

**DEVELOPMENT OF COST-EFFECTIVE FAULT
LOCALIZATION PLATFORM FOR UNDERGROUND CABLE
SYSTEM**

A Thesis Submitted to the College of
Graduate and Postdoctoral Studies
In Partial Fulfillment of the Requirements
For the Degree of Master of Science
In the Department of Electrical and Computer Engineering
University of Saskatchewan
Saskatoon, SK, Canada

By

MD SALAUDDIN

© Copyright Md Salauddin, February 2023. All rights reserved.

Unless otherwise noted, copyright of the material in this thesis belongs to the author.

Permission to Use

In presenting this thesis in partial fulfillment of the requirements for a Postgraduate degree from the University of Saskatchewan, I agree that the Libraries of this University may make it freely available for inspection. I further agree that permission for copying of this thesis in any manner, in whole or in part, for scholarly purposes may be granted by the professor or professors who supervised my thesis work or, in their absence, by the Head of the Department or the Dean of the College in which my thesis work was done. It is understood that any copying or publication or use of this thesis or parts thereof for financial gain shall not be allowed without my written permission. It is also understood that due recognition shall be given to me and to the University of Saskatchewan in any scholarly use which may be made of any material in my thesis.

Requests for permission to copy or to make other uses of materials in this thesis/dissertation in whole or part should be addressed to:

Head of the Department of Electrical and Computer Engineering

57 Campus Drive

University of Saskatchewan

Saskatoon, Saskatchewan S7N 5A9Canada

OR

Dean

College of Graduate and Postdoctoral Studies

University of Saskatchewan

116 Thorvaldson Building, 110 Science Place

Saskatoon, Saskatchewan S7N 5C9

Canada

Disclaimer

Reference in this thesis to any specific commercial products, process, or service by trade name, trademark, manufacturer, or otherwise, does not constitute or imply its endorsement, recommendation, or favoring by the University of Saskatchewan. The views and opinions of the author expressed herein do not state or reflect those of the University of Saskatchewan and shall not be used for advertising or product endorsement purposes.

Abstract

The mining industry is highly dependent on electricity or other forms of energy converted into electricity. Here, one of the common forms of electricity distribution is through underground cables, but the electricity systems regularly confront contingencies due to cable breakdowns or defects. Depending on fault types, detective tools, and cable position, it may take several hours to days to detect the exact fault location with existing methods and tools, which affects the entire mining operation with financial losses and safety issues. Besides, industrial fault localization platforms are very expensive and seldom tailored for the mining industry. To locate such faults, the online fault localization platform has drawn wide attention but is limited in functionalities and applications.

This research aims to develop a cost-effective double-ended online fault localization platform while proposing solutions for data synchronization and accurate traveling wave arrival time detection problems. At first, extensive market research has been done on available platforms to access those platforms' functionalities and costings. Then, a hardware setup has been developed by combining appropriate sensors, a data acquisition unit, a computational platform, and other necessary components. Furthermore, to establish communication between remote platforms, a Python-based software program has been developed. Besides, query-based server management has been introduced to handle and manage huge amounts of data. Combining the hardware and software, the overall cost of a single platform is CAD \$ 2,850.00, which is at least ten times less than the least expensive market option.

The chosen double-ended traveling wave-based online fault localization method requires accurate synchronized time to properly compute the traveling wave arrival time difference. Thus, coordinated universal time alignment of acquired time series data is needed. Global Positioning System (GPS) based universal time synchronization is one of the popular ways. There are several other techniques, but all of these have some practical difficulties and dependencies. To solve this problem, a cost-effective novel zero-crossing point-based data synchronization approach has been proposed. This approach doesn't rely on GPS receivers or any other existing methods; rather, the measurement is synchronized by calibrating the zero-crossing points of the sinusoid measurements before the fault. In this way, appropriate synchronization has been realized.

The accuracy of traveling wave-based fault localization highly depends on how accurately its arrival times are detected from ends of cable. Accurate detections are achieved when signals sampling rates are high. Usually, a data acquisition system with a higher sampling rate may address the issue, but it increases cost. On the other hand, available interpolation-based up-sampling techniques have several constraints. Machine learning model can solve this problem if trained by real inputs and outputs with desired sampling rates for different types of faults. In this research, a machine learning-based up-sampling model has been presented, which improves the accuracy of traveling wave arrival time detection. Besides, it is cost-effective and outperforms interpolation techniques.

Therefore, the proposed fault localization platform combines all the above solutions, which makes the platform very advance, reliable, and cost-effective.

Acknowledgments

At first, all praise goes to the Almighty ALLAH for HIS blessings upon me.

I would like to extend my sincere gratitude to my supervisors Prof. Dr. C. Y. Chung, Prof. Dr. Seok-Bum Ko, Dr. S.M. Mazhari and Dr. Tongkun Lan. Without their continuous support, guidance, and supervision my program, and this thesis work at the University of Saskatchewan would not have been possible.

Besides my supervisors, I would like to thank the rest of my graduate committee for insightful comments and suggestions.

I would also like to thank the Natural Sciences and Engineering Research Council of Canada (NSERC), International Minerals Innovation Institute (IMII), Cameco, Nutrien, The Mosaic Company and the University of Saskatchewan, for providing financial support during my master's study.

Last but not least, I would like to thank my parents, my wife and sisters for their support and unconditional love. I would not have accomplished this achievement without them.

Thank you all for your support.

Dedication

Dedicated to my parents and my wife for their incredible love and support.

Table of Contents

	Page Number
Permission to Use	i
Abstract.....	iii
Acknowledgments	v
Dedication	vi
Table of Contents	vii
List of Tables	x
List of Figures.....	xi
List of Abbreviations	xiv
Chapter 1: Introduction	1
1.1. Underground Cable Fault Localization in Mining Industries	1
1.2. Fault Localization Techniques	2
1.2.1. Offline Fault Localization Methods	2
1.2.2. Online Fault Localization Methods	5
1.3. Problem Statements and Research Objectives	7
1.4. Structure of the Thesis.....	9
Chapter 2: Underground Cable Online Fault Localization Platform Development	10
2.1. Market Research on Available Online Fault Localization Platforms	10
2.2. Fault Localization Platform Design Requirements	14
2.3. Full Hardware Design of the Online Fault Localization Platform	15
2.3.1. Hardware Connection Overview of the Proposed Setup	16
2.3.2. Cost of the Platform.....	18

2.3.3. Sensor	19
2.3.4. Data Acquisition Unit.....	20
2.3.5. Computational and Server Platform	21
2.3.6. Display.....	23
2.3.7. Power Supply.....	24
2.3.8. Additional storage.....	25
2.4. Software Design	26
2.4.1. Python Based Software Program for Bi-Directional Communication Establishment.	27
2.4.2. Sqlite3 Based Database Management.....	29
2.5. Tests in Experimental Setup.....	31
2.6. Summary	33
Chapter 3: Low-cost Data Synchronization	34
3.1. Introduction and Available Synchronization Techniques	34
3.2. Methodology of Zero Crossing Point-Based Algorithm.....	36
3.3. Data Analysis and Synchronization Algorithm Development	39
3.4. Simulation Model.....	40
3.5. Validation Results for Different Case Studies	41
3.5.1. Influence of Fault resistance	42
3.5.2. Influence of Phase shifts.....	44
3.5.3. Influence of Noise	46
3.6. Cost-Effectiveness of the Proposed Technique.....	49
3.7. Test in Experimental Setup	50
3.8. Summary	50

Chapter 4: Data Up-Sampling by Machine Learning	52
4.1. Introduction and Available Up-Sampling Techniques.....	52
4.2. The Proposed Machine Learning Based Up-Sampling Model.....	54
4.2.1. Data Pre-processing.....	54
4.2.2. Machine Learning Model Development.....	56
4.3. Validation Study.....	59
4.3.1. Data Generation Model	60
4.3.2. Data Generation Automation Algorithm Development.....	61
4.3.3. Results and Comparisons Between Actual and Predicted Data	63
4.3.4. Comparisons Between Interpolation and Proposed Method	66
4.4. Test in Experimental Setup	70
4.5. Summary	70
Chapter 5: Conclusion and Future Work.....	71
5.1. Conclusion.....	71
5.2. Case Specific Accomplishments	72
5.3. Future Work	73
References	75
Appendix A	84
Appendix B	84

List of Tables

	Page Number
Table 1-1: A list of most needed equipment for offline fault localization.....	5
Table 2-1: Available fault localization platforms with costing.....	11
Table 2-2: Available data acquisition units with prices	13
Table 2-3: Cost of one computational platform setup	19
Table 2-4: Specifications of Jetson TX2 Developer Kit.....	22
Table 2-5: Pinouts of Jetson TX2 Developer Kit.....	23
Table 3-1: Results for faults at different phases with different fault resistance	42
Table 3-2: Results for faults at different phases with different fault resistance & phase shifts (φ)	44
Table 3-3: Comparison of results for different noise levels	47
Table 3-4: Industrial GPS time synchronization modules with prices	49
Table 4-1: Results and comparisons between actual and predicted data	66
Table 4-2: Results and comparisons between interpolation and machine learning based prediction	69

List of Figures

	Page Number
Figure 1-1: Process flow diagram of the methodology of offline fault localization method.....	2
Figure 1-2: Typical set-up of a double ended online fault localization platform.	6
Figure 2-1: Total setup of an online fault localization platform.....	15
Figure 2-2: Connection overview of the proposed computational platform. Sensors are connected to the CH 1 and CH 2 of the DAQ unit. Via USB connection, DAQ unit and Jetson TX2 is connected. These devices are powered by a dual-output switching mode power supply. Display is connected to Jetson TX2 by HDMI and USB connection, while SSD storage is connected by SATA cable. Ethernet connection provides internet to the Jetson device.....	17
Figure 2-3: Connection overview of the server. Powered by a 12 V DC power supply while additional SSD storage is connected via SATA.	18
Figure 2-4: HFCT Sensor, Model-HVDP (HFCT 100 HCP)	20
Figure 2-5: Data Acquisition Unit, Model-Multicomp (MP720646 US)	21
Figure 2-6: 10.1inch capacitive touch screen LCD display	24
Figure 2-7: Power Supply. A dual output switching power supply is used as the power source. 24	24
Figure 2-8: Solid state drive. Additional storage for computational platform and server.	25
Figure 2-9: The process flow chart of python-based software program allowing bi-directional communication between computational platforms and server.....	27
Figure 2-10: Server listening for client after running the program.....	28
Figure 2-11: Client connecting with sever.....	28
Figure 2-12: Connection established with the client with IP and Port number.	28
Figure 2-13: The data sharing process between computational platforms and server.....	30
Figure 2-14: Pictures show local databases in computational and server platform with stored information in structured format.....	31

Figure 2-15: Computational platform generating data and sending to server.	32
Figure 2-16: Server platform getting data from computational platform for analysis.....	32
Figure 3-1: Time synchronization by GPS time receivers.....	35
Figure 3-2: Propagation of traveling waves at ends of the cable.....	37
Figure 3-3: Lag observed in the same phase between ends.....	38
Figure 3-4: Waveforms indicating zero crossings to align.....	38
Figure 3-5: Flow diagram of the developed Python program.....	39
Figure 3-6: 35 kV underground cable transmission system.....	40
Figure 3-7: Observed signal waveforms from sending end.....	41
Figure 3-8: Observed signal waveforms from receiving end.....	41
Figure 3-9: Signal waveform with added Gaussian noise of SNR 30 dB.....	48
Figure 3-10: Signal waveform with added Gaussian noise of SNR 40 dB.....	48
Figure 4-1: Sample signal with lower sampling rate.....	53
Figure 4-2: Sample signal with higher sampling rate.....	53
Figure 4-3: Single input sample has five corresponding output samples to predict.	55
Figure 4-4: Desired time stamps are inserted into input signals.....	55
Figure 4-5: Original 1MHz signal pre-processed to 5 MHz by interpolation.....	56
Figure 4-6: Step by step development of machine learning model.	57
Figure 4-7: PSCAD model of a 35 kV underground cable transmission system.....	60
Figure 4-8: (a) 1 MHz faulty signal, (b) Zoom in image of 1 MHz faulty signal, (c) 5 MHz faulty signal, (d) Zoom in image of 5 MHz faulty signal.	61
Figure 4-9: Flow diagram of the developed Python program for automated data generation.....	62

Figure 4-10: (a) Actual 1 MHz signal (b) Zoom-in samples of actual 1 MHz signal (c) Predicted 5 MHz signal (d) Zoom-in samples of predicted 5 MHz signal (e) Actual 5 MHz signal (f) Zoom-in samples of actual 5 MHz signal, for fault at 0.4 km with fault resistance of 10 Ohm. . 64

Figure 4-11: (a) Actual 1 MHz signal (b) Zoom-in samples of actual 1 MHz signal (c) Predicted 5 MHz signal (d) Zoom-in samples of predicted 5 MHz signal (e) Actual 5 MHz signal (f) Zoom-in samples of actual 5 MHz signal for fault at 0.6 km with fault resistance of 0.01 Ohm. 65

Figure 4-12: (a) Actual 1 MHz signal (b) Zoom-in samples of actual 1 MHz signal (c) Predicted 5 MHz signal by interpolation (d) Zoom-in samples of predicted 5 MHz signal by interpolation (e) Predicted 5 MHz signal by machine learning (f) Zoom-in samples of predicted 5 MHz signal by machine learning (g) Actual 5 MHz signal (h) Zoom-in samples of actual 5 MHz signal for fault at 0.4 km with fault resistance of 10 Ohm. 68

List of Abbreviations

GPS	Global Positioning System
TDR	Time Domain Reflectometry
USB	Universal Serial Bus
AI	Artificial Intelligence
BNC	Barrel Nut Connector
DAQ	Data Acquisition
HDMI	High-Definition Multimedia Interface
SATA	Serial Advanced Technology Attachment
PCI	Peripheral Component Interconnect
SSD	Solid State Drives
CT	Current Transformer
HFCT	High Frequency Current Transformer
WIFI	Wireless Fidelity
PD	Partial Discharge
ADC	Analog to Digital Converter
GS/S	Giga-Samples Per Second
LCD	Liquid Crystal Display
LAN	Local Area Network
HMI	Human Machine Interface
SDK	Software Development Kit
GPU	Graphics Processing Unit
SD	Secure Digital

GB	Gigabyte
MB	Megabyte
eMMC	Embedded Multi Media Card
DSI	Display Serial Interface
MIPI	Mobile Industry Processor Interface
ISP	Image Signal Processor
LPDDR	Low Power Double Data Rate
CSI	Camera Serial Interface
Mbps	Megabits per second
UART	Universal Asynchronous Receiver/Transmitter
SPI	Serial Peripheral Interface
I2C	Inter-Integrated Circuit
I2S	Inter-IC Sound
GPIO	General Purpose Input/Output
TTP	Thermal Transfer Plate
GSM	Global System for Mobile Communications
JTAG	Joint Test Action Group
PCIE	Peripheral Component Interconnect Express
SQL	Structured Query Language
VAC	Volts Alternating Current
DC	Direct Current
AC	Alternative Current
MV	Medium Voltage
HV	High Voltage

dB	Decibel
MHz	Megahertz
ReLU	Rectified Linear Unit
MSE	Mean Squared Error
s	Second
μ s	Micro Second
W	Watt
km	Kilometer
mm	Millimeter
$^{\circ}$ C	Degree Celsius
TEV	Transient Earth Voltage
HVCC	High Voltage Coupling Capacitor
AA	Airborne Acoustic
BTS	Bushing Tap Sensor

Chapter 1: Introduction

1.1. Underground Cable Fault Localization in Mining Industries

Electricity is the main power source of mining industry. In 2017, the total energy used in the Canadian mineral sector was 9.7% of the total energy used in Canada [1]. Just in 2015, the Canadian mining industry spent \$ 2.7 billion on power sources [2]. This is almost half of the total wages paid for production and related workers. Other than general-purpose applications, mining industries are using electricity to power the equipment and systems for mining operations. There are several crucial equipment using electricity in mine sites, such as continuous electric miner machines, electric shuttle cars, electric roof bolters, longwall systems, etc. Besides, several operations in mines are also run by electric power, for example, conveyors, draglines, shovels, etc.

The power distribution systems in mining sites are similar to those in conventional distribution systems. Here, one of the common forms of electricity distribution is through underground cables, which are buried under the soil or placed inside the ducts under the ground. These cables can be several kilometers long. Electricity distribution through underground cables is the most feasible way for mining sites, because underground cables are reliable and safe due to not being very much affected by calamities and outer environments.

Though underground cable is one of the reliable ways of power supply, it is not entirely safe. Usually, cable breakdowns or defects are caused because of poor installation, maintenance issues, and insulation degradation, which is led by many years of operations, overstresses from electrical, thermal, mechanical, and environmental reasons. Once a fault occurs in cable, modern power protection systems are used to isolate the faulty area in mining site. Generally, after the fault occurs, the primary task for the maintenance crew is to use fault localization kits to trace the entire faulty area to locate the fault position. But, due to the cables being several kilometers long and installed underground, it is difficult to trace the fault location. In these circumstances, it is very common to face delays for fault localization. The delay may reach hours to days. It eventually leads to the whole mining operations being suspended. Due to the shutdown operations, the mining

sites would suffer substantial economic losses. Moreover, power failure is directly related to safety issues for mine workers, which is one of the most critical factors in mining sites to consider. Therefore, fast and accurate fault localization is essential for mining sites, and this is the major focus of this thesis.

1.2. Fault Localization Techniques

Fault localization is a challenging task though it is among the most common issues. Researchers are continuously working to develop better and easier ways to solve the above problems [3]-[11]. Generally, fault localization techniques are divided into offline and online-based methods. The following sections illustrate details on both methods.

1.2.1. Offline Fault Localization Methods

The most common fault localization platforms are based on offline method and are widely used in commercial packages [3]. The offline method is a post-fault localization method that requires the line to be tripped and de-energized. It follows a multi-step process as shown in Fig. 1-1.

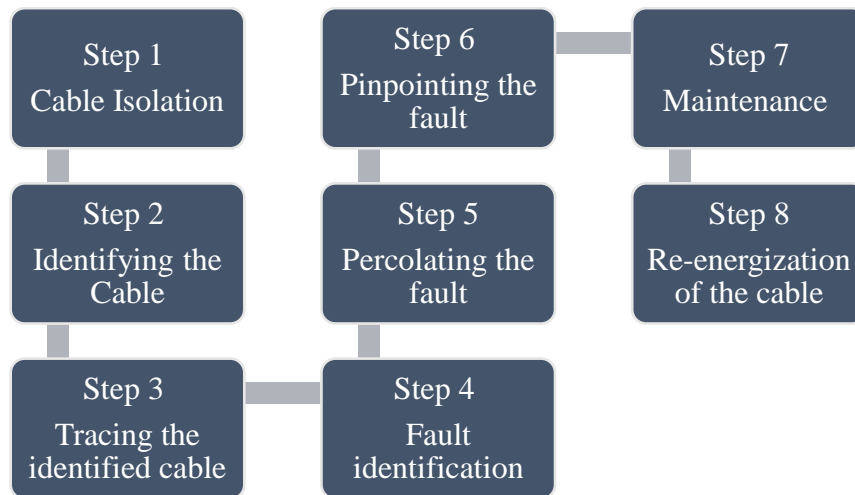


Figure 1-1: Process flow diagram of the methodology of offline fault localization method.

The offline fault localization method is a combination of terminal and tracer techniques. In the terminal technique, the approximate location of the fault is assumed based on measurements at either one or both ends of the underground cable [4]. Since the faulted cable differs in electrical

characteristics compared to the non-faulted cable, the fault localization is made based on the change in electrical characteristics. The terminal technique is divided into several localization methods, such as the bridge method, radar method, resonance method, etc.

The bridge method is subdivided into the Murry loop bridge and the capacitance bridge methods. In Murrey loop bridge, faulted cable and non-faulted cable are combined to form a loop to make a proportional measurement to approximate the fault location. But open circuit faults can't be measured in Murrey loop. On the other hand, comparisons of capacitance from one end between faulted and non-faulted cable are the main criteria of fault localization in capacitance bridge.

The Radar method is quite different from bridge because it calculates the time difference of the injected radar pulse to reach a fault point and then reflect back, but faces difficulties for nonlinear faults. In resonance method, a frequency generator is connected at the end of the faulted cable. The resonance frequency is observed by changing the frequency because there is a relation between frequency and fault distance. Resonance method also faces same ineffectiveness in detecting nonlinear faults.

However, these above terminal-based methods do not pinpoint faults, only approximates the fault location. So, in the second step of offline fault localization, for more accurate pinpointing of the fault location, tracer technique takes measurements along the approximate fault area found in terminal technique. This approach is divided into different methods, such as tracing current, audio frequency, impulse/thumper, time domain reflectometry, earth gradient, etc.

The tracing current method is divided into DC and AC-based current injections to one ends of the formed circuit between faulted conductor and the ground. Then electromagnetically coupled circuits or current transformers are used to take the measurement on the other side manhole location to calculate the fault location. Cables that are buried underground with shorted to the ground is the main focus of this method, but is not very effective for other types of faults.



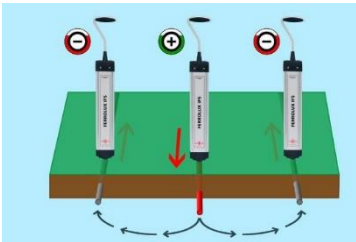





In the audio frequency method, audio frequency is injected into the faulty cable, which creates a magnetic field at the fault point. Then, magnetic loop antennas are used to detect the fault location. This method is particularly designed for the low-voltage system. On the other hand, there is impulse/thumper method which uses charger capacitors to transmit high resistance pulses in the faulty cable, and causes arcs at the fault point, which eventually heats up the surrounding air and

transmits audible thumps. Afterward, this thump is traced by listening or using magnetic field detectors but faces accuracy problem.

One of the most famous methods is time domain reflectometry (TDR), which is a low-voltage method used for locating low-resistive faults, open-circuit faults, short-circuit faults, etc. It is also useful to measure the cable length. Similar to radar, high frequency pulse is injected into the faulty cable. Depending on the discontinuity or differences of TDR impedance due to the fault, a portion of the pulse or full pulse reflects back. Comparing the sending and receiving pulse in the time axis, the fault location can be determined. Additionally, depending on the shape of the pulse, the fault type can be determined. However, this method is not used for insulation and sheath-to-ground fault localization. Similar to this, the arc-reflection method is a combination of TDR method with an impulse generator. High voltage impulse breaks down the high resistance fault to act like a short circuit fault, then the TDR method is used. The velocity propagation constant of the cable fault distance is measured in this method. This method is mostly applicable to in the secondary distribution system with insulated cable for cable-to-ground faults.

Despite the offline methods can find the fault location, the main issues of offline fault localization methods are the cost, complexities, multi-step process using several test equipment, and the long localization time [6]. Depending on the cable structure and types of faults, the localization can take several hours to days. A set of multiple equipment work together as a whole setup. According to the type of faults, combinations of separate equipment are used, as listed in table 1-1. Moreover, considering several sets of multiple equipment are used, the costs can exceed a couple of hundred thousand dollars for a single unit, which is a substantial amount for mining companies who have many sites. So, it can be easily understood that offline fault localization a time consuming, complex, and expensive solution. Last but not least, it is not feasible and practical to be able to use the mentioned methods without adequate experience and training provided to a dedicated team. In total, it is a big burden for any industry to manage all these tasks together.

Table 1-1: A list of most needed equipment for offline fault localization

<p style="text-align: center;">Cable Identifier</p> 	<p style="text-align: center;">Cable and Phase Identifier</p> 	<p style="text-align: center;">Cable Localization Device</p> 
<p style="text-align: center;">Cable Fault Localization Device</p> 	<p style="text-align: center;">Cable Fault Pin Pointer</p> 	<p style="text-align: center;">Sheath Fault Locator</p> 
<p style="text-align: center;">Cable Ground Fault Locator</p> 	<p style="text-align: center;">Portable Cable Fault Location System</p> 	<p style="text-align: center;">Time and Frequency Domain Reflectometers</p> 

1.2.2. Online Fault Localization Methods

Online methods are based on acquiring voltage, current or other electrical parameters in real-time so that the data can be utilized to calculate the location once the fault occurs [3][6]. These data go through processes to pinpoint the fault [7]. This method can be divided into two different categories, which are the impedance-based method and the traveling wave-based method [12].

As the name suggests, the impedance-based method calculates the fault location by observing the change in impedance during fault with pre-fault condition impedance. The

calculation of the impedance can be done by single-ended or double-ended techniques. Estimating fault location by the measurement from one end is known as the single-ended technique. The same can be done by both end measurement through double-ended technique. However, the impedance-based method accuracy depends not only on values of all phase-to-ground voltages and currents but also on how accurately the system impedance is measured. This impedance measurement can be affected by inaccurate relay measurements, system homogeneities, and mutual coupling of zero sequence, etc. [13].

The other online method is the traveling wave-based method, which depends on the calculation based on the arrival times of traveling waves at ends of the cable. These traveling waves are the voltage and current surges that travel toward the ends of the line. The traveling waves arrival times at ends are directly related to the fault distance, and it is the main base of traveling-waved-based fault localization method. These measurements can also be done from a single or double ends. But the single ended method suffers accuracy problems. To improve the accuracy, traveling-wave-based online fault localization mostly uses double-ended measurements. So, data from both ends of the cable is necessary in such a case. So, in this double-ended set-up, current sensors are installed at the ends of each cable. Readings from these sensors are monitored and analyzed to localize the fault. Figure 1-2 shows the typical set-up of a double ended online fault localization platform.

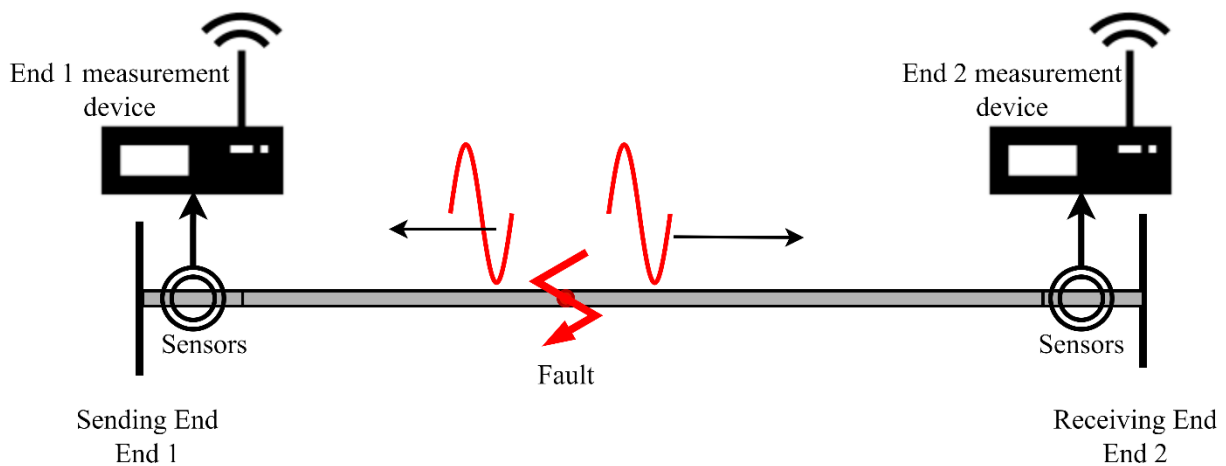


Figure 1-2: Typical set-up of a double ended online fault localization platform.

Online methods are far more promising for underground cable fault localization than offline methods. Online methods are quick to localize faults and less complex by reducing multi-step processes. Taking into account the quick restoration; the highest priority is to repair the feeder cables as soon as possible. Besides, based on the observed signals from the cable, the prediction of fault can also be made before it turns into a permanent fault. In this regard, a suitable online fault localization platform is very desirable for quick, easy, and accurate fault localization.

1.3. Problem Statements and Research Objectives

Industrial fault localization platforms are being used for decades but seldom of them are tailored for the mining industry. At potash mining sites, substations are connected with HV feeder cables which may exceed several kilometers from a supply point. The mining industry feeder protection system is mostly focused on worker safety which is tailored to operate in low fault currents. Also, a medium-high resistance grounding is usually used due to the safety of workers in hazardous work conditions. But, this medium-high resistance grounding limits fault currents in a range of ~25-50 Amperes. Because of fault current is so low, it becomes a challenge to detect it by conventional analysis. Besides, because of following the usual electrical standards, multiple cables are incorporated with a single relay for power protections in mining sites. This makes exact fault localization a very difficult task in mines.

The double-ended traveling wave-based online method is the most feasible option due to being able to realize fault location with a weak fault current, and also it is independent of system parameters. However, one major challenge is that it requires high frequency data acquisition with accurate time reference to detect the traveling wave arrival times from the weak signals. So, the set-up becomes quite costly. Also, these are not very available in the market. Only a few manufacturers offer similar platforms with limited functionality, non-customizable analysis options, and very high amount of costs.

The primary goal of this thesis is to develop a modernized low-cost double ended online fault localization platform. The platform needs to handle high frequency data for traveling wave-based fault localization. Moreover, it should be able to send and receive data from remote ends of mining sites to perform fault localization algorithms. In addition to that, big data analytics combined with advanced machine-learning tools, such as deep learning, ensemble trees, tools can

be useful for accurate fault location of the low fault current conditions in mines. Considering the importance of using advanced machine-learning algorithms, the platform should provide the option of performing advance machine learning algorithms.

Apart from that, the double-ended measurement-based online fault localization platforms generally use costly Global Positioning System (GPS) time receivers to synchronize the data acquired at both ends, while GPS would face problems due to satellite invisibility, atmospheric condition, and installation constraint in deep underground mines. This thesis also focuses on solving data synchronization problem with a novel low-cost data synchronization technique.

Lastly, the accuracy of traveling wave-based fault localization is dependent on how accurately its arrival time is detected. So, highly sampled information is needed for the accurate detection of traveling waves arrival times. Focusing on cost effective solution, this thesis aims to solve this problem by developing a machine-learning model for the predictive up-sampling of fault data.

To address all the mentioned issues, the following research objectives were set to meet the goal of developing an affordable system:

- Develop an online fault localization platform in a more advance and affordable way.
- Develop necessary software algorithms for bi-directional communication establishment and sever management in fault localization platform.
- Develop a novel data synchronization technique without reliance on GPS.
- Develop necessary algorithm to perform data synchronization in fault localization platform.
- Develop a machine learning model for predictive data up-sampling to accurately detect traveling wave arrival time.
- Develop necessary algorithm to perform data up-sampling in fault localization platform.

1.4. Structure of the Thesis

The chapters are organized as follows:

Chapter 2 presents the development of an online fault localization platform. To begin, extensive market research on available online fault localization platforms with cost is presented. Then, a low-cost hardware set-up is presented with cost and necessary details. Afterward, the development of software to establish bidirectional communication and server management are explained elaborately. Finally, in the experimental setup the developed software algorithms are tested.

Chapter 3 presents a noble low-cost synchronization technique. First, the available synchronization techniques and drawbacks are discussed. Afterward, a zero-crossing point-based algorithm is presented with the necessary simulation model and Python-based algorithm. Cost-effectiveness of the proposed technique is also justified comparing with the available GPS modules. Finally, the simulation results are discussed elaborately. Then, the developed algorithm is tested in the experimental setup.

Chapter 4 proposes a machine learning-based approach for predictive data up-sampling. The necessity of data up-sampling and different up-sampling approaches are described at first, and then, a machine learning model is presented where pre-processed datasets are used to train the model. Subsequently, in the validation study, the simulation model for data generation process is presented. Then, the feasibility of the machine learning-based up-sampling approach is examined through comparative analysis. Finally, the developed machine learning model is loaded and tested in the experimental setup.

Chapter 5 concludes the findings of this research work and provides direction for future works to upgrade the system.

Chapter 2: Underground Cable Online Fault Localization Platform Development

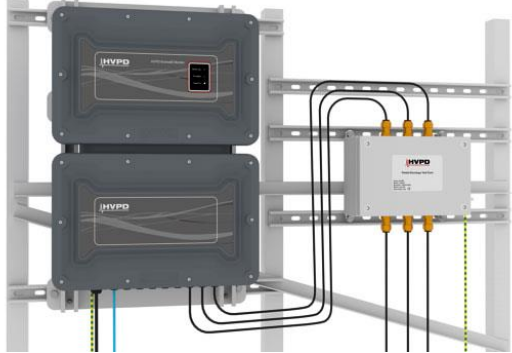


An underground cable fault localization platform is developed in this chapter. First, section 2.1 presents market research on the available online fault localization platforms with costs. Based on the observation on market available options and project goals to achieve, the overall design requirements are highlighted in section 2.2. Then following the design requirements, in section 2.3, the developed hardware setup with connection overviews and cost are given in details. This section also describes each setup component with required details. As an important part of fault localization platform, in section 2.4, developed python software programs related to bi-directional communication establishment and database management are presented. These algorithms are tested in experimental hardware setup in the section 2.5. Finally, this chapter is summarized in section 4.6.

2.1. Market Research on Available Online Fault Localization Platforms

In this section, market research on available online fault localization platforms is presented. The purpose of the market research is to understand and know the functions, specifications, costs and drawbacks of available options. Besides, it may provide a guidance for developing an online localization platform. In the market research, the focus has been made on the platform's capability of acquiring high frequency data online. It has also been checked that whether the available platforms permit to implement advance machine learning algorithms for more accurate fault localization.

A list of the most suitable market available online fault localization platforms with costs is given in the following table 2-1.

Table 2-1: Available fault localization platforms with costing




Ser.	Manufacturer and Model	Bandwidth	Voltage Level	Cost/set
1.	<p>HVPD Kronos® KPM Condition Monitoring [14]</p> 	50 MHz	HV and MV	\$48,843.00 CAD
2.	<p>Omicron MONTESTO 200 [15]</p> 	40 MHz	HV and MV	\$94,720.00 CAD
3.	<p>Rugged Monitoring HPM601-P [16]</p> 	100 MHz	HV and MV	\$27,490.00 CAD

From the market research, it has been found that very few companies have solutions specific to online fault localization and these are very expensive. Even a single unit can cost hundreds of thousand dollars. These solutions generally come with limited functionalities. Most of these don't comply with installing customized algorithms by restricting any manipulation of their products. Moreover, due to acquiring high frequency data, the available platforms use expensive data acquisition units with higher sampling rates, which leads to high costs. Besides, to synchronize the data from both ends of the cable, GPS-based synchronization is used with added cost. Additionally, these platforms are not very easy to use, and almost all of the companies suggest training with additional cost.

Manufacturers are offering the whole platform by combining all separate parts but making it more expensive than it should be. In these platforms, the most important part is the data acquisition units and the analysis software. The main function of the data acquisition unit is to convert the signals to digital values. So, further market research has been done to find suitable data acquisition units to incorporate it in the development of customized fault localization platform at a lower cost.

Table 2-2 shows the costs of most suitable data acquisition units in the market. Prices vary depending on the maximum frequency range along with different functions such as the number of available input channels, input bandwidth, output resolution and sampling rate, different analysis options, capability to incorporate various types of sensors, etc.

Table 2-2: Available data acquisition units with prices

Ser.	Manufacturer and Model	Bandwidth	Cost/set
1.	AlazarTech ATS9352 PCI Express Digitizer [17] 	250 MHz	\$ 7,000.00 CAD
2.	Keysight DAQ970A Data Acquisition System [18] 	2 GHz	\$ 3364.00 CAD
3.	Dataman 522 USB Oscilloscope [19] 	60 MHz	\$ 810.00 CAD
4.	MP720646 US Multicomp Pro PC Oscilloscopes [20] 	100 MHz	\$ 650.00 CAD

It is clear from the market research that a DAQ unit is far less expensive than a whole setup. It will be cost-effective to develop a localization platform combining cheaper DAQ unit, computational platform, and software programs. So, with this understanding, design requirements are set in the following sections to develop an online fault localization platform.

2.2. Fault Localization Platform Design Requirements

According to the discussion in Section 2.1 and the goals to achieve, the following design requirements are set for the development of a fault localization platform:

- The platform should be able to acquire data from both ends of the cable.
- High frequency data acquisition compatibility is required to acquire high frequency fault information.
- The computational platform should be capable of performing advanced machine learning algorithm for more accurate fault localization.
- Remote fault localization platforms should be capable of communicating with server bi-directionally to transfer and receive data while each platform should be running standalone.
- The platforms should have proper data management facilities for data analysis from stored data.
- The whole system should be low cost compared to the market available options.

The platform combines both hardware and software to achieve the design requirements, which are discussed in the following sections.

2.3. Full Hardware Design of the Online Fault Localization Platform

The proposed prototype is depicted in Fig. 2-1 and designed according to the set requirements. In this platform, current transformers are placed at the ends of the cable to acquire current signals. Data acquisition (DAQ) units are connected to corresponding current transformers and convert data from analog to digital. Standalone embedded AI computing devices are used as computational platforms and server. Each computational platform is connected with a DAQ unit to acquire data and then analyzes data for fault localization. Computational platform installed at remote ends sends information to the server through internet-based communication.

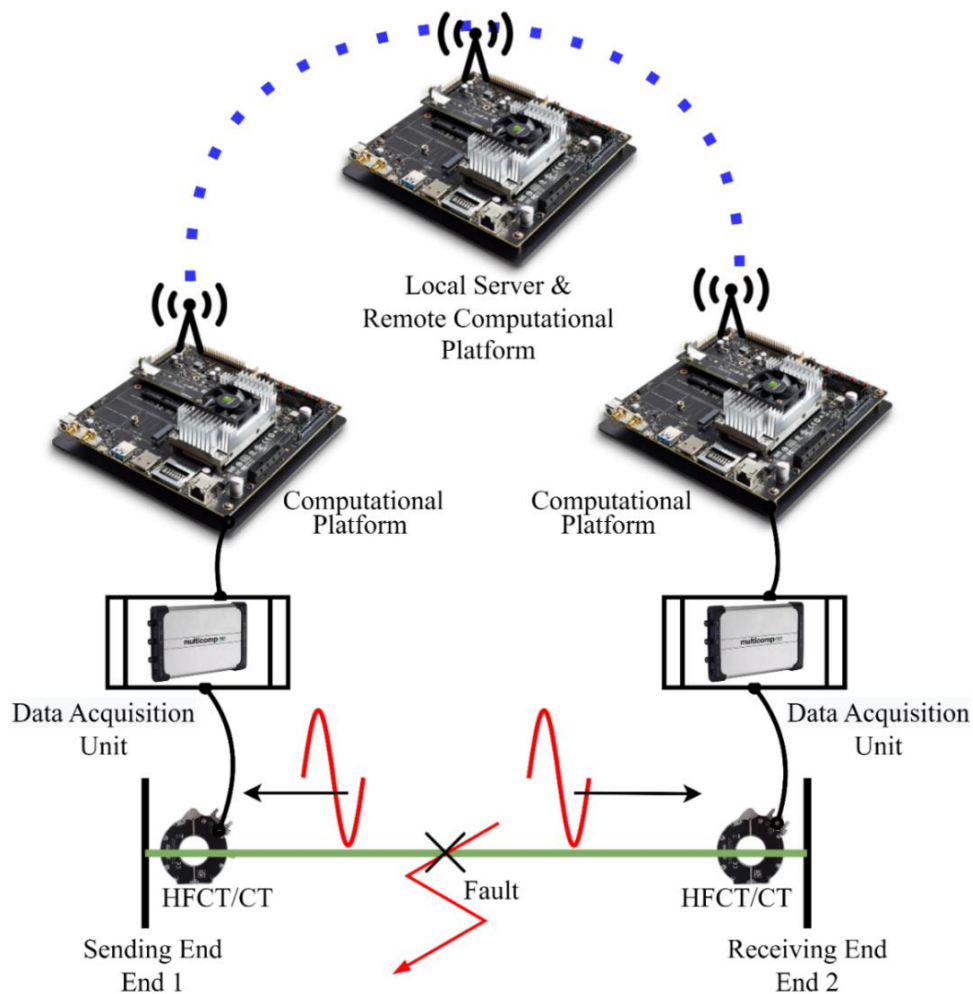


Figure 2-1: Total setup of an online fault localization platform

2.3.1. Hardware Connection Overview of the Proposed Setup

The connection diagram for the standalone computational platform is shown in Fig. 2-2. The selected DAQ unit has two input channels to incorporate two different sensors. Here, the sensors are connected via BNC cables to the inputs of the DAQ unit, where 5V DC power is provided to the DAQ unit from the power supply. Dedicated USB port is used to connect the DAQ with Jetson TX2 computational platform with a male-to-male USB cable. Computational platform is also powered by the same power supply, where the power supply itself is powered by 60 Hz 120 VAC power source on the site. The touchscreen display is connected by a male-to-male HDMI cable using the provided HDMI ports in the display and the Jetson TX2. 5V DC power is shared among Display and DAQ unit from channel 1 of the power supply. To add an additional SSD storage device with Jetson TX2, a male-to-male SATA connector is used. Jetson TX2 can be connected to internet via ethernet port from a router or it can be connected through WIFI. To acquire information from DAQ unit, the SDK software of the DAQ unit is used in Jetson's Linux platform. Data can be stored in sequential orientation for further query-based data fetching by SQLite3 based database management system. Developed python programs establish the connection with server by socket based bi-directional communication management while also manage the query-based SQLite3 database. The programs also display acquired information. This platform has enabled the options to further install advance machine learning-based algorithms to pinpoint the fault location more accurately.

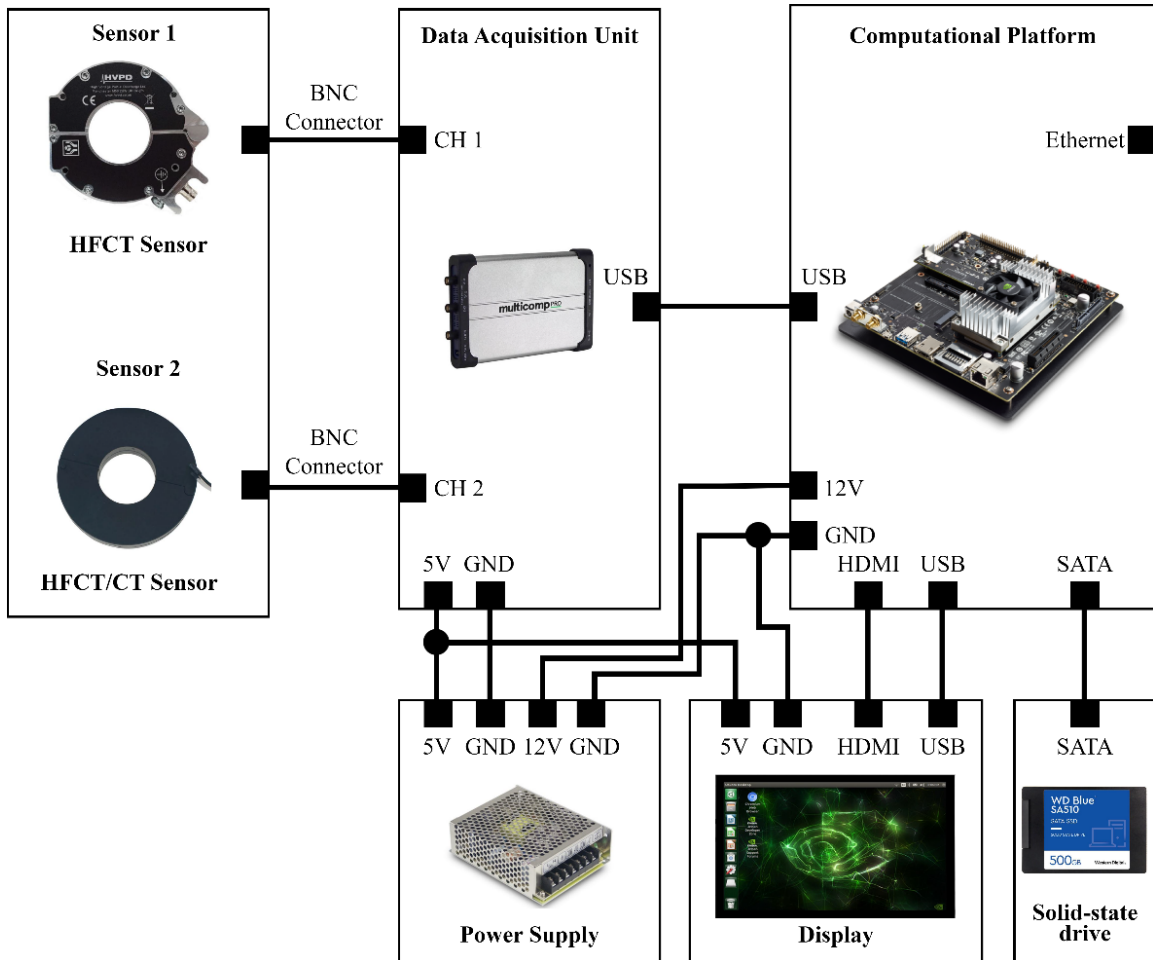


Figure 2-2: Connection overview of the proposed computational platform. Sensors are connected to the CH 1 and CH 2 of the DAQ unit. Via USB connection, DAQ unit and Jetson TX2 is connected. These devices are powered by a dual-output switching mode power supply. Display is connected to Jetson TX2 by HDMI and USB connection, while SSD storage is connected by SATA cable. Ethernet connection provides internet to the Jetson device.

Server communicates with the computational platforms to fetch data by using the bi-directional communication program. The server hardware connections are depicted in Fig. 2-3. It also incorporates SQLite3 based database to store information. Due to continuously fetching data from both computational platforms, it requires a large storage space. So, as like as computational platforms, additional data storage facility is added via a SATA connection to the solid-state drive. A power supply of 12V is used as the power source of the server. Server is also capable of running machine learning algorithms based on acquired data to pinpoint faults. Besides, users will be

notified about the fault and the location of the fault from the server. Also, access can be granted from remote locations to the server to see related information.

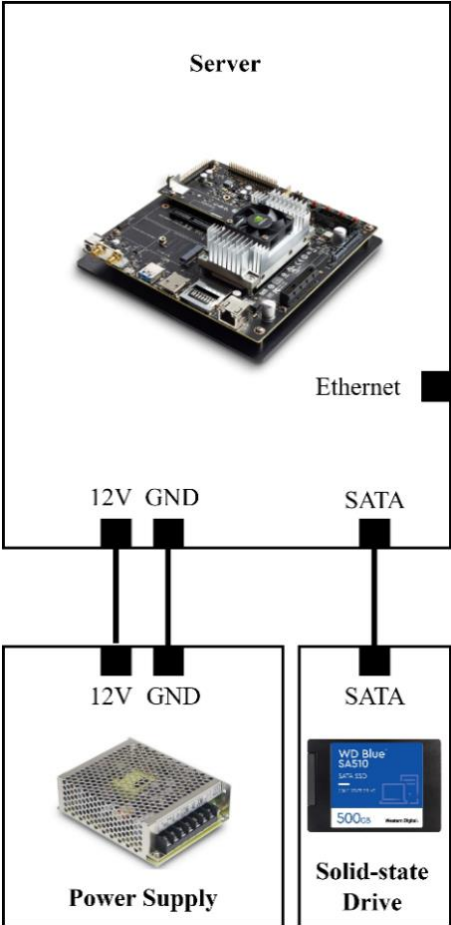


Figure 2-3: Connection overview of the server. Powered by a 12 V DC power supply while additional SSD storage is connected via SATA.

2.3.2. Cost of the Platform

Based on the set-up in the previous sections, the basic prototype will cost around CAD \$ 2,850.00, as calculated in December 2022. The breakdown is presented in the following Table 2-3.

Table 2-3: Cost of one computational platform setup

Component Name	Approximate Cost (CAD)
HFCT Sensor [21]	\$ 644.00
BNC Extension Cable [22]	\$ 12.00
Power Supply [23]	\$ 32.00
Data Acquisition Unit [20]	\$ 650.00
Jetson TX2 Development Kit [24]	\$ 1,170.42
10.1inch Touch Screen LCD Display [25]	\$ 129.99
500 GB SSD Drive [26]	\$ 54.99
Male to male USB cable X 2 [27]	\$ 8.99 X 2 = \$ 17.98
Male to male HDMI cable [28]	\$ 12.00
SATA Connection Set [29]	\$ 26.62
Accessories and Manufacturing	\$ 100.00
Total	\$ 2,850.00

In the following sections, the components used in the design are described in details.

2.3.3. Sensor

A split core high frequency current transformer from HVPD (HFCT 100 HCP) is used to measure high frequency information [21], which is shown in Fig. 2-4. The sensor is suitable for placing around high voltage (HV) and medium voltage (MV) underground power cables. The Aluminum enclosure improves its sensitivity to online Partial discharge (PD) measurements in noisy environments, which makes it a better choice. Its frequency bandwidth is within 0.30 to 60 MHz, which is adequate for PD signals or high frequency measurements. The standard unit has an inner dimension of 46 mm, where width x depth x height are 115 mm x 24 mm x 117 mm. Different

sensor models can be adopted depending on the cable size and requirements. There is a female BNC port to transfer the sensor output via BNC cable, while the other end of the BNC cable is connected to the DAQ unit. Due to being a split core, sensor can be split into two parts to wrap around the cable without cutting the cable.



Figure 2-4: HFCT Sensor, Model-HVPD (HFCT 100 HCP)

2.3.4. Data Acquisition Unit

Here, the basic requirement of the data acquisition unit is to acquire high frequency data from sensors within Hz to MHz range and then to convert analog data from sensors to digital format. So, the main use of the DAQ unit is to function as an analog to digital converter. The selected DAQ unit is a 2-channel PC-based oscilloscope from Multicomp (MP720646 US), with maximum bandwidth of 100MHz and sampling rate of 1 GS/s, which can perfectly incorporate the HFCT/CT sensors [20]. The operating voltage of the DAQ is 5-15 VDC. 5 V DC power is provided to the DAQ from the power supply. The selected computational platform has Linux operating system, and this DAQ unit is supported on Linux platform. It also provides secondary development support via a software development kit. This DAQ unit transfers acquired data to the computational platform via USB Device (type-C) or USB host, or LAN.



Figure 2-5: Data Acquisition Unit, Model-Multicomp (MP720646 US)

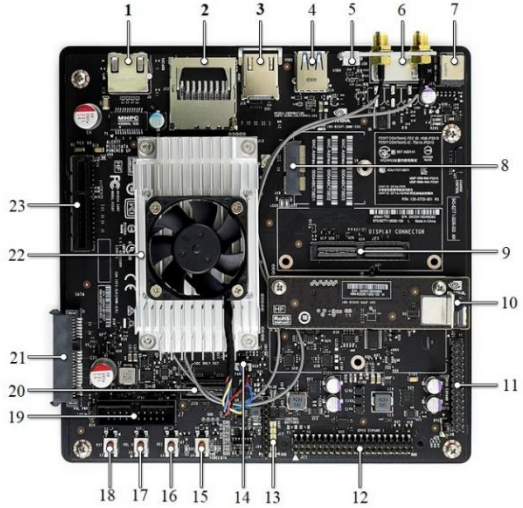
2.3.5. Computational and Server Platform

The Jetson TX2 developer kit is selected as computational and server platforms. It helps developing a hardware and software a system in fast and easy way. It has quad-core ARM A57 processor and dual-core Denver 2 CPU, 256-core NVIDIA GPU [24][30]. It has an opensource SDK (Linux Based) that includes software libraries for machine and deep learning, GPU computing, multi-processing, etc. Installed units at the both ends of the cable work as computational platforms. Apart from that, a separate unit is used as a server at a remote location. Besides working as computational and server platforms, it is capable of saving high volume data in SD card and SATA storage. Ethernet/wireless peripherals make it capable of transferring information over internet and performing remote operations. As a standalone platform, HMI interface can also be added to display information and to enable users to navigate through manus. Following Tables 2-4 and 2-5 describe its capability to work as a standalone computational or server platform.

Table 2-4: Specifications of Jetson TX2 Developer Kit

Processing Components	Ports & Peripherals	Form-Factor
<ul style="list-style-type: none"> • Dual-core NVIDIA Denver2 + quad-core ARM Cortex-A57 CPU • 256-core Pascal GPU • 8GB LPDDR4, 128-bit interface • 32GB eMMC • 4kp60 H.264/H.265 encoder & decoder • Dual ISPs (Image Signal Processors) • 1.4 gigapixel/sec MIPI CSI camera 	<ul style="list-style-type: none"> • HDMI 2.0 • 802.11a/b/g/n/ac 2x2 867Mbps WiFi • Bluetooth 4.1 • USB3, USB2 • 10/100/1000 BASE-T Ethernet • PCIe gen 2.0, 1x4 + 1x1 or 2x1 +1x2 • Sdcard & SATA • UART, SPI, I2C, I2S, GPIOs 	<ul style="list-style-type: none"> • Dimensions: 50x87mm (1.96" x 3.42") • Thermal Transfer Plate (TTP), -25°C to 80°C operating temperature • Mass: 85 grams, including TTP • 5.5-19.6VDC input power (consuming 7.5W under typical load)

Table 2-5: Pinouts of Jetson TX2 Developer Kit

Pinouts		NVIDIA Jetson TX2
1. Gigabit Ethernet port	12. 40PIN GPIO expansion header	
2. SD card slot	13. UART interface	
3. HDMI (Type-A)	14. Fan connector	
4. USB3.0 port	15. Power button	
5. Micro USB port	16. Recovery button	
6. Antenna connector	17. Audio button	
7. Power jack	18. Reset button	
8. M.2 key A interface	19. JTAG connector	
9. Display interface	20. Debug interface	
10. Camera	21. SATA connector	
11. 30PIN GPIO expansion header	22. JetsonTX2	
	23. PCIE connector	

2.3.6. Display

Jetson TX2 has MIPI DSI/HDMI based display interface to directly connect a MIPI or HDMI display. Shown in Fig. 2-6 is a 10.1-inch capacitive touchscreen LCD display with resolution of 1024×600 from Waveshare (SKU: 18096) [25]. This display requires 5V/3A power supply to run with the Jetson device. The purpose of this display is to show real-time data or saved data, outputs from performed algorithms and to navigate to different menus of the computational platform.

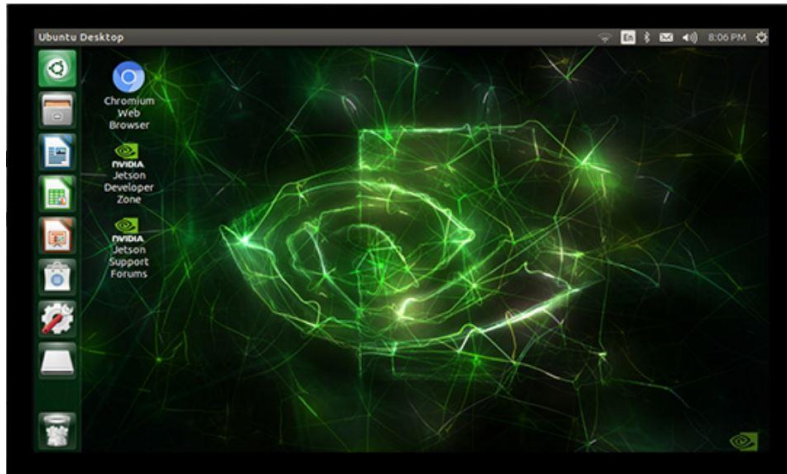


Figure 2-6: 10.1inch capacitive touch screen LCD display

2.3.7. Power Supply

As a power supply, dual output switching power supply (Model- MEAN WELL RD-50A) is used, which is shown in Fig. 2-7 [23]. It takes universal AC as input. Due to the requirement of two separate voltage levels, it is chosen, as it has two separate output channels of CH 1-5V 6A and CH 2-12V 2A with a combined maximum output power of 54W. 5V from CH1 is used as a common power source for data acquisition unit and display, on the other hand, 12V from CH2 is dedicated to Jetson TX2 board. This power supply has built-in short circuit, overload, and over voltage protection facility, with withstanding capacity for 5s for 300 VAC surge.



Figure 2-7: Power Supply. A dual output switching power supply is used as the power source.

2.3.8. Additional storage

Jetson TX2 can avail additional storage capability by using SD Card or Serial advanced technology attachment (SATA) based disks. Here, a 500 GB (Western Digital-Internal Solid-State Drive) is selected primarily, as additional storage which is shown in Fig. 2-8 [26]. If needed, a higher capacity storage can be used. The selected solid-state drive has read speed of up to 560MB/s and has very low power draw. This high speed will help faster handling of acquired data. The dimension of the SSD is 1.08 (W) x 0.27 (H) x 0.03 (D) inch, which makes it a slim storage disk to use with the platform.



Figure 2-8: Solid state drive. Additional storage for computational platform and server.

2.4. Software Design

To perform the double ended fault localization algorithm, it is required to have data from both corresponding ends. In underground mining, cables are several kilometers long (>5 km), so no physical connection is present between the installed computational platforms at remote ends. Here, only Ethernet/Wi-Fi/GSM based internet communication is the most feasible way. Besides, continuous high-volume data need to be stored at each computational platform. So, sophisticated software programs are required to establish reliable communication and database management. So, to build a fault localization platform, application-specific software programs are equally important as hardware.

Here, the data volume is large and continues real-time transfer to the server can create data transmission problem and will require constant communication facility with higher bandwidth. Additionally, there is a high risk of losing data in case of network interruption, as systems installed underground have more network redundancy issues. Also, there is no need to pass continuous data to the fault localization algorithm or to the server unless there is a fault or doing fault prediction. But there should always be a communication established between the server and computational platform whether or not data needs to be transmitted and fetched. To address these issues, a python-based software program is developed to establish bi-directional communication between server and platforms.

Besides, managing high-volume data is problematic and requires large data storage. In addition to that, the management of data is very crucial as those need to be saved in a structured manner with proper time stamps. So, a query-based data management is required, where data can be pulled out later from the saved database with specific timeframe and can be passed to the fault localization algorithm. All these reasons create the requirement to develop python-based software programs for reliable query-based database management.

The following sections describe the developed python-based programs to establish socket based bi-directional communication and SQLite3 based database management.

2.4.1. Python Based Software Program for Bi-Directional Communication Establishment

The developed python-based program establishes communication between the server and computational platforms; the procedures are depicted in Fig. 2-9 [31]. The developed program has two separate parts, one for the server and another for computational platforms. To begin, both server and computational platforms initiate the separate parts of the program. Communication between separate machines or platforms can be established by using Socket based two-way communication process [32]. So, each program incorporates socket-based approach. In the second step, separate IP addresses and port numbers are initiated for both server and computational platforms, and then server starts listening for incoming connections [34]. Subsequently, computational platforms as TCP clients initiate the connection with server by dialing IP address and port number of the server. As soon as the communications are established, either way, data requests can be sent; thus, it establishes the way for simultaneous reading and writing of data. Multiple clients can be added from remote locations to the server for concurrent operation.

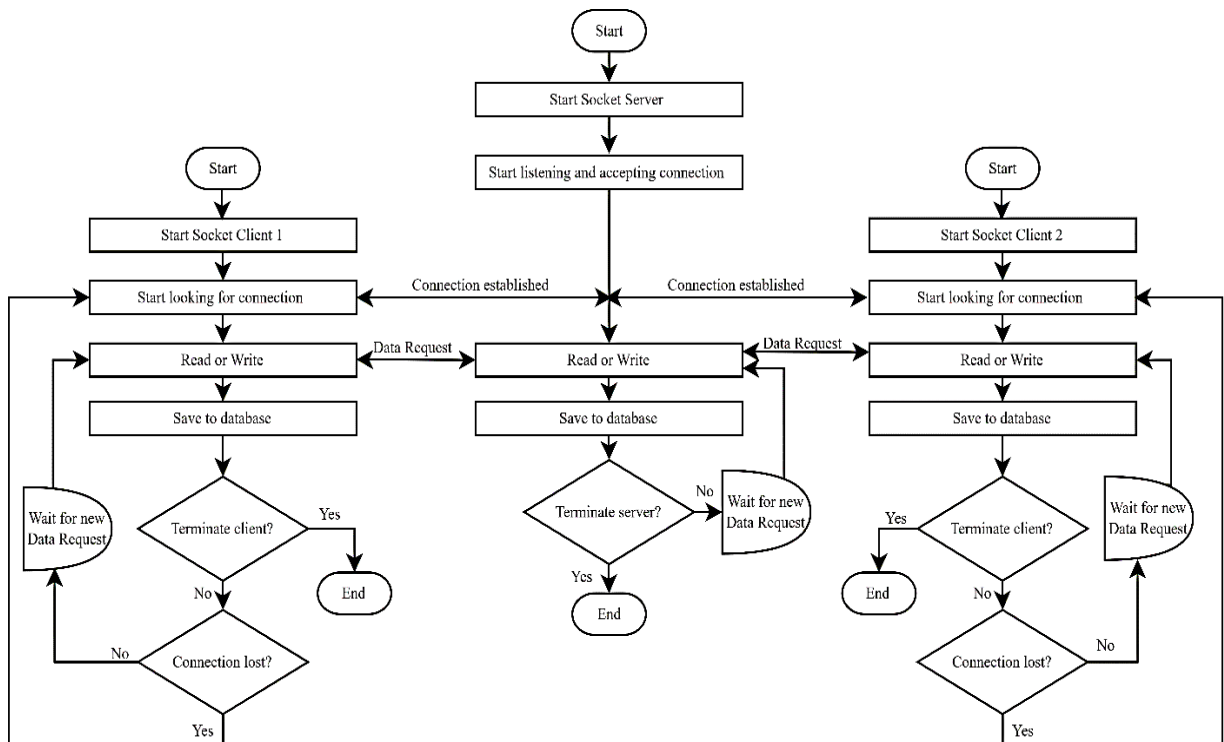


Figure 2-9: The process flow chart of python-based software program allowing bi-directional communication between computational platforms and server.

Python program is designed in such way that if the client losses it's connection with the server, it keeps trying to re-connect until the connection re-establishes. In the meanwhile, due to the interruption of the network, if the requested data is not received, it can be retrieved again from the client database. The following Figures 2-10 to 2-12, present how the server looks for connections and further client establishes the connection.

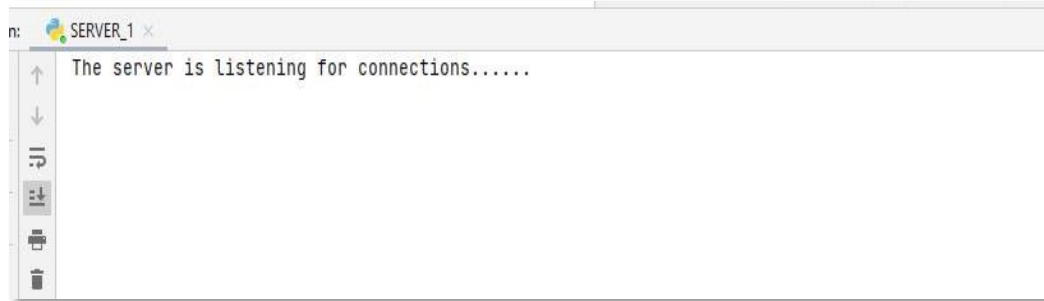


Figure 2-10: Server listening for client after running the program

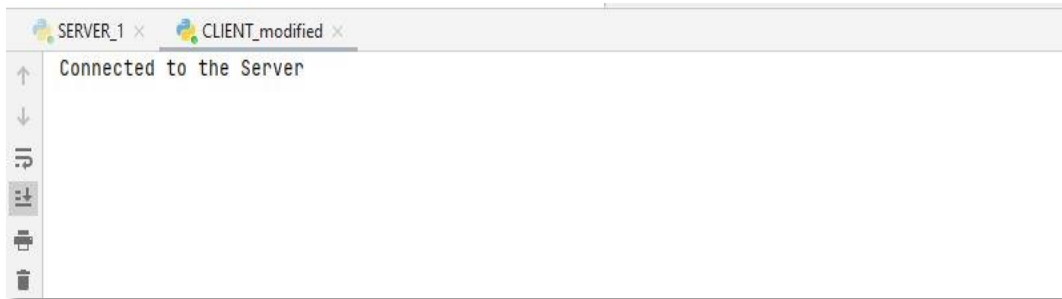


Figure 2-11: Client connecting with sever

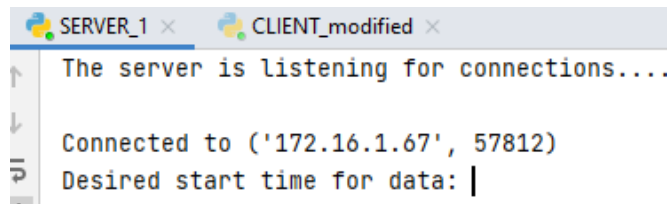


Figure 2-12: Connection established with the client with IP and Port number.

2.4.2. Sqlite3 Based Database Management

Data are fetched continuously from sensors, but usually, data will be analyzed after the fault happens. The fault localization algorithm may only analyze the data which are around the fault time. But there is a huge amount of data continuously stored in the database. So, to retrieve the data around the fault time from the database, there needs to be a time-specific query-based data read and write process. So, it is necessary to incorporate a database management system to address the query-based process. Additionally, the contents should be updated continuously and automatically in a structured way so that little or no data is lost in any failure. Additionally, this process is desired to be simple, lightweight, and faster. Also, it should be compatible with Python programming language, as the developed platform is based on Python.

Based on the above requirements SQLite3 based database management system has been found to be the most suitable option in this case. SQLite3 is not only an open-source database management system but also a simple, lightweight file-based SQL Database [35]. Here, data with values and types are stored sequentially with time in a structured manner. Usually, database management requires a separate server process, but it is not required for SQLite3 based database management. Besides, nonstandard formats of commands from SQL query language are enough to work with this database management system. The most important factor is that without any additional procedure or software installation, it works seamlessly with Python programming language [36]. The database files are portable across all available operating systems, so the data can be read in any platform for analysis with no need of changing or converting to other formats.

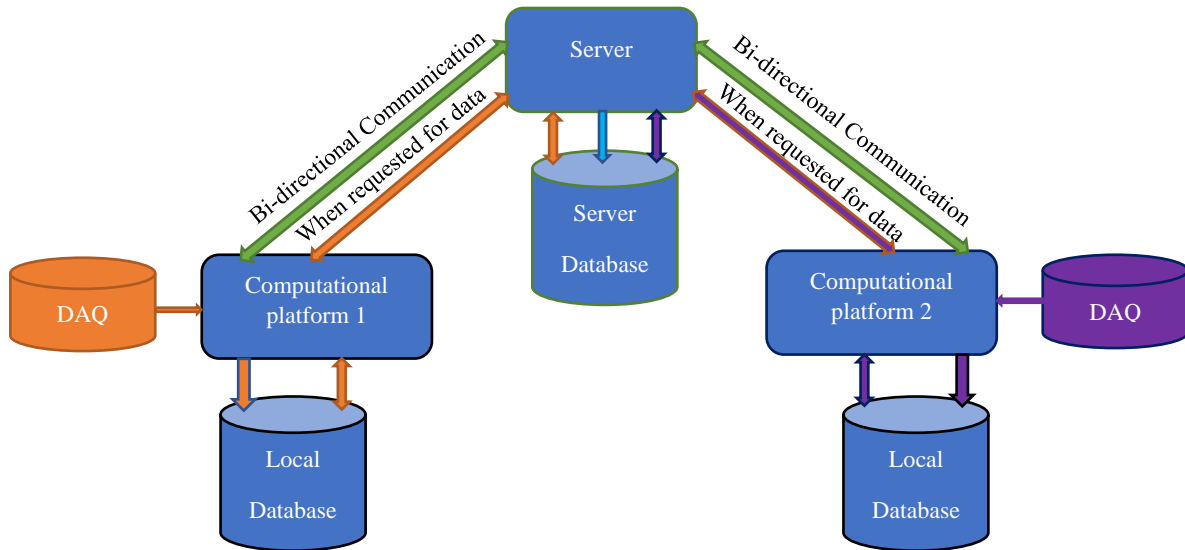


Figure 2-13: The data sharing process between computational platforms and server.

The SQLite3 based data storage procedure is depicted in Fig. 2-13, which consists of both local and server storage. In order to transfer data, the server will request data for a certain time frame from the computational platform. The computational platform has a local storage unit that saves the data which are being received from DAQ unit in real-time. When requested by the server, the module on the computational platform extracts the data from the local database and sends to the server. In the same way, computational platforms can request data from the server if needed.

Similar to the computational platforms, the server also has a local storage unit to store data which are being received from the remote computational platforms. The database stores data with timestamps. In the following Fig 2-14 shows how, data are saved on computational and server platforms with proper time stamps.

Computational platform local database auto updates with acquired data

	Timestamps	Data
	Filter	Filter
255	2022-05-06 22:06:30.603499	218
256	2022-05-06 22:06:31.633761	116
257	2022-05-06 22:06:32.686630	139
258	2022-05-06 22:06:33.717475	196
259	2022-05-06 22:06:34.737312	147
260	2022-05-06 22:06:35.776131	176
261	2022-05-06 22:06:36.802598	173
262	2022-05-07 23:30:14.156530	167
263	2022-05-07 23:30:15.201112	110
264	2022-05-07 23:30:16.247850	170
265	2022-05-07 23:30:17.290839	149
266	2022-05-07 23:30:18.334997	185
267	2022-05-07 23:30:19.394384	159
268	2022-05-07 23:30:20.436266	175
269	2022-05-07 23:30:21.476264	172
270	2022-05-07 23:30:22.504842	192
271	2022-05-07 23:30:23.549257	179

Server local database auto updates with acquired data

	Timestamps	Data
	Filter	Filter
185	2022-05-02 02:17:53.196018	126
186	2022-05-02 02:17:54.229764	192
187	2021-09-05 18:46:20.903161	116
188	2021-09-05 18:46:20.907468	118
189	2021-09-05 18:46:20.912404	120
190	2021-09-05 18:46:20.916471	122
191	2021-09-05 18:46:20.920471	124
192	2021-09-05 18:46:20.925456	126
193	2021-09-05 18:46:20.930443	128
194	2021-09-05 18:46:20.934432	130
195	2022-05-02 02:17:48.010440	210
196	2022-05-02 02:17:49.040297	216
197	2022-05-02 02:17:50.068865	163
198	2022-05-02 02:17:51.117126	213
199	2022-05-02 02:17:52.153606	190
200	2022-05-02 02:17:53.196018	126
201	2022-05-02 02:17:54.229764	192

Figure 2-14: Pictures show local databases in computational and server platform with stored information in structured format.

2.5. Tests in Experimental Setup

In the experimental setup, one computational platform and one server platform were set in the lab to test the developed software programs in real-time, which are shown in the following figures 2-15 and 2-16. Each setup was combined with a JetsonTX2 device along with a display monitor, keyboard, mouse, internet, and power connection.

First, the Linux operating system was installed in both Jetson TX2 devices. Then those were configured to connect to the internet. Here, separate internet connections were used for each platform, so that remote communication algorithms could be verified.

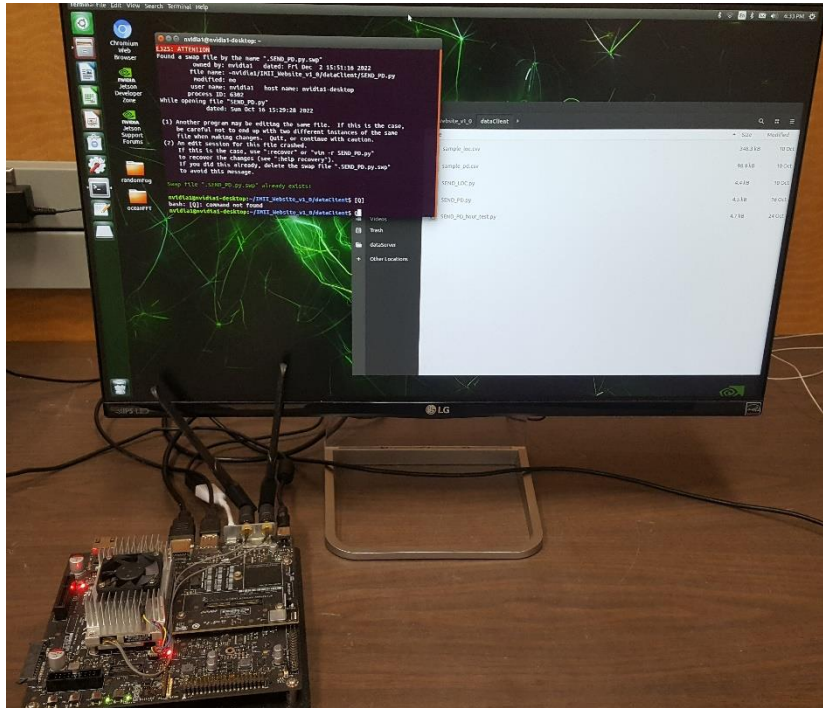


Figure 2-15: Computational platform generating data and sending to server.

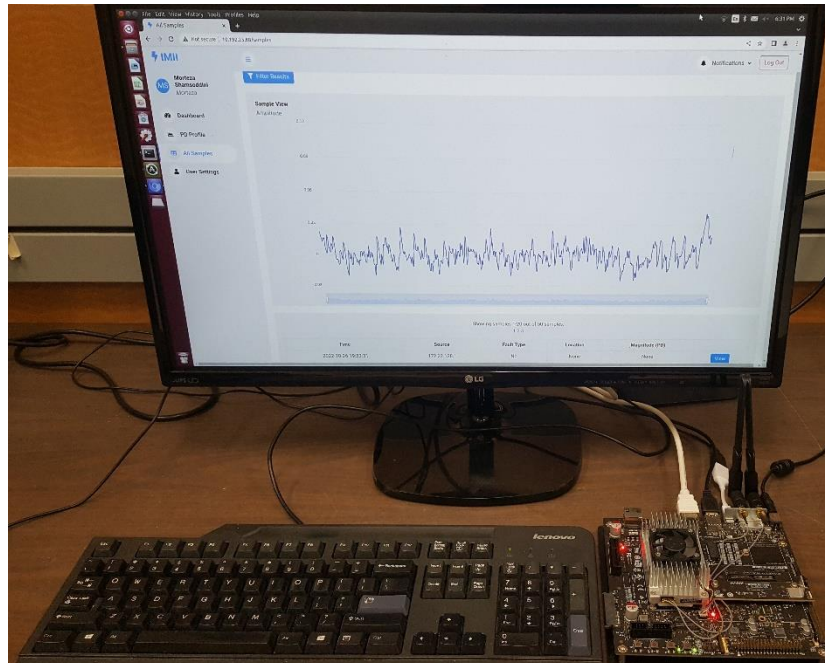


Figure 2-16: Server platform getting data from computational platform for analysis.

After both platforms were connected to separate internet connections, platform-specific remote communication algorithms were installed and initiated. Each platform had separate IP address and port number. By dialing the server IP address and port number, the computational platform initiated the connection process with the server. After the sever accepted the request from the computational platform, bi-directional communication was established.

In the second step, the computational platform was generating data by using a random data generation algorithm to mimic that the data are coming from current sensors on the cable. At the same time, these data were being saved sequentially with timestamps in the local storage by the sqlite3 based database management algorithm. In the meanwhile, the server was requesting data from the computational platform within a specific time frame. Data were fetched and being showed in the server platform, as in Fig. 2-16. All these data were saved in the local database of the server as well.

In the same way, the computational platform requested data from the server, and those were successfully fetched too. Communication redundancy was also tested by disconnecting one platform from the internet for a while. The computational platform kept trying to reconnect as long as the connection with the server was not re-established. In the meanwhile, the data generation process was going on. After the connection re-established, with time-specific requests, the server fetched the data from the computational platform, which were generated during the network disconnection.

2.6. Summary

In this chapter, the total fault localization platform has been developed along with the Python based software programs. The developed hardware platform is cost-effective and has the capability to acquire, process, and analyze data. Furthermore, it is an open platform that gives the facility to perform advance machine learning algorithms for better fault localization. Additionally, the developed software programs are capable of establishing reliable bi-directional communication and data management. As a whole, the setup addresses all the design requirements that were set to accomplish. The following chapter focuses on developing a low-cost data synchronization technique which will be incorporated into the developed online fault localization platform.

Chapter 3: Low-cost Data Synchronization

This chapter describes a novel idea of low-cost data synchronization. Section 3.1 introduces the problem background, available synchronization techniques, and drawbacks. Section 3.2 presents the methodology of the zero-crossing point-based algorithm. A data analysis and synchronization algorithm are developed in section 3.3. The simulation model is described in section 3.4. In the subsequent section 3.5, results are discussed, which validates the idea with different fault conditions. The cost-effectiveness of the proposed technique is compared with the available GPS synchronization modules in section 3.6. The developed algorithm and proposed technique are tested in experimental hardware setup in the section 3.7. Finally, section 3.8 summarizes the chapter. This chapter was presented in 2022 IEEE Electrical Power and Energy Conference (EPEC 2022) and published in IEEE Xplore.

3.1. Introduction and Available Synchronization Techniques

When a fault occurs, voltage and current surges will travel toward the ends of the line, known as the traveling wave; its arrival time at both ends is directly related to the fault distance, which is an effective index to locate the fault [37]. This approach is widely recognized in the industry as the double-ended traveling wave-based fault localization method [38]. However, this method requires accurate synchronized time to properly compute the traveling wave arrival time difference [39]. So, coordinated universal time alignment of acquired time series data is needed when aggregating data from different measurement or logging units [37]. Global Positioning System (GPS) based universal time synchronization is one of the most popular ways to do that [40]. GPS receiver chip consists of clocks, and it synchronizes the time in data acquisition platforms on both ends [41]-[42].

Despite the time synchronization provided by GPS, there are a number of practical difficulties in achieving the required data synchronization. One of the critical issues is the reliability of GPS signal due to affected by various unpredictable and uncontrolled factors, such as loss of signal, satellite invisibility, electromagnetic interference, atmospheric disturbances,

jamming, failure of the GPS antenna, weather change, etc. [43]. Even though the antennas are located in places with an unobstructed view of satellites, GPS receivers face occasional loss of signals. Due to these reasons, the quality of synchronized measurement data is influenced. For underground cables, the effects are severe due to cables being laid deep under the surface [41]. All these circumstances make GPS time synchronization-based fault localization a costly and complicated technique. GPS time synchronization devices are integrated with each measurement unit placed at ends, which syncs time with the satellite, as shown in Fig. 3-1.

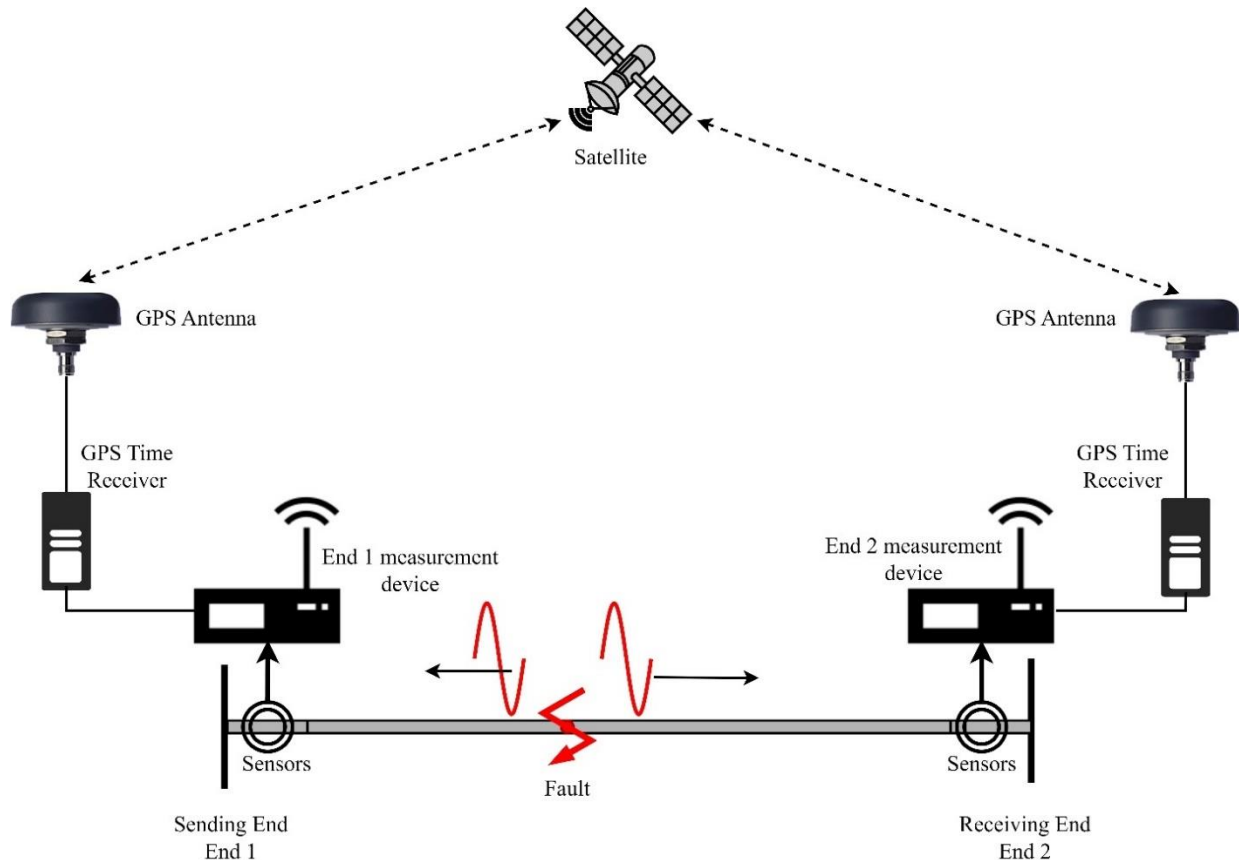


Figure 3-1: Time synchronization by GPS time receivers

Other than synchronizing by GPS, using phasor-based synchronizing technique can estimate the synchronization error aiming at subsequent compensation [44], it relies on the extraction of fundamental-frequency components, but the distortion of fundamental components reduces the accuracy. Besides, some research proposed compensation of time synchronization errors by analyzing line energization or fault records, where the synchronization factor is computed to align the remote times for synchronization. However, this factor solely depends on fault records

with high sampling rates [39]. On the other hand, studies on locating the fault without data synchronization have been carried out in recent years, but certain constraints are brought in, such as only being applicable for metallic faults [45], or requiring both voltage and current measurements [46]-[47].

To deal with the above issues, a zero-crossing point-based data synchronization approach is proposed. This approach doesn't rely on GPS receivers or other methods; rather the measurements are synchronized by calibrating the zero-crossing points of the sinusoid measurements before the fault. In this way, appropriate synchronization is realized, and the performance of the proposed method is evaluated by observing the results for different fault conditions. The maximum synchronization error found without noise is 0.02 ms, and with proper filtering or denoising, the same can be achieved for noisy signals.

3.2. Methodology of Zero Crossing Point-Based Algorithm

A cable with a certain length of l is considered and installed underground. In the ideal condition, the acquired signal is considered to be sinusoidal.

Under fault conditions, traveling waves will be generated at the fault position and propagate along the line from the fault point to both ends [39], [48]-[49]. Propagations of traveling waves are shown in Fig. 3-2. With acquired traveling wave arrival time, the fault location problem can be formulated as follows [37]-[38],[42]:

$$L = 1/2 (l + (t_a - t_b)v) \dots \dots \dots (3.1)$$

where L is the location of the fault, t_a and t_b are the arrival times of the traveling waves at each end, l is the length of the cable, and v is the propagation velocity.

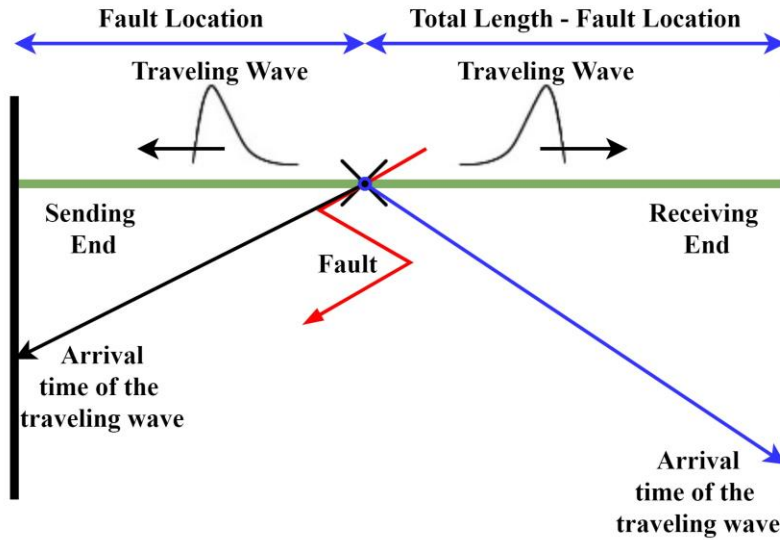


Figure 3-2: Propagation of traveling waves at ends of the cable

The calculated location depends on the accurate arrival time of the detected traveling wavefronts, so accurate time synchronization is essential. To synchronize the measurements at both ends, the idea is to calibrate the zero-crossing points of the sinusoidal measurements before the fault occurs. The pre-fault component is a sinusoidal signal, which will not be interfered by the traveling waves. As a result, this pre-fault component should have a fixed phase lag between ends as shown in Fig. 3-3, and the data synchronization at both ends can be realized by calibrating the pre-fault zero crossing points.

To locate the pre-fault zero-crossing points, an overcurrent criterion is utilized for activation as expressed in the following:

$$| I_{observed} | > | I_{set} | \dots \dots \dots (3.2)$$

where $I_{observed}$ is the observed current amplitude when fault happened. I_{set} is the threshold. It is mentionable that the activation is not limited to the overcurrent criterion, other criterion can be applied.

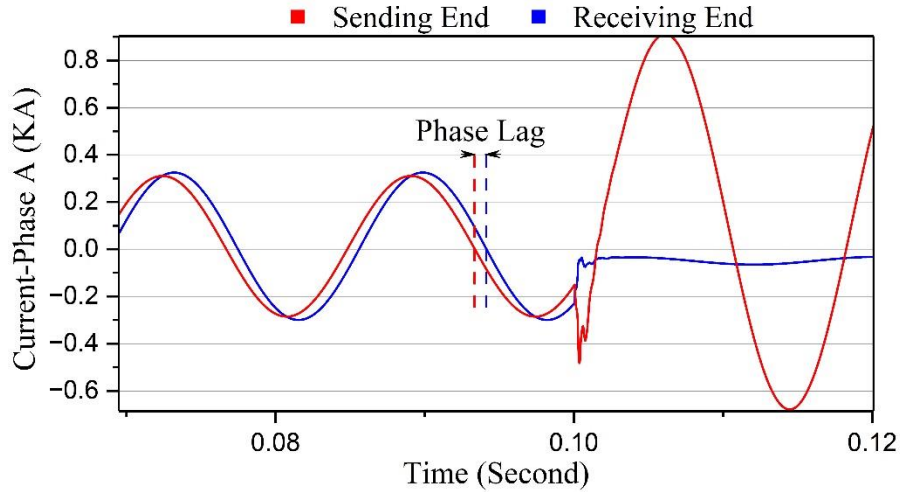


Figure 3-3: Lag observed in the same phase between ends

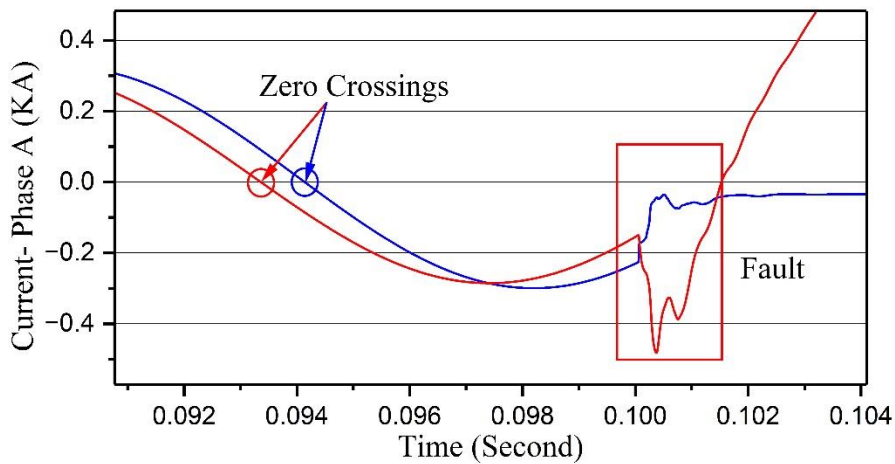


Figure 3-4: Waveforms indicating zero crossings to align

After the activation, the synchronization will be based on aligning the pre-fault zero crossing points as indicated in Fig. 3-4. The offset between the unsynchronized measurement at both ends is calculated as follows:

$$t_0 = t_2 - t_1 - t_\alpha \dots \dots \dots (3.3)$$

where t_0 is the time offset that leads to synchronization error; t_1 is the zero-crossing time from sending end; t_2 is the zero-crossing time from receiving end; t_α is the constant phase lag due to the non-resistive cable transmission.

3.3. Data Analysis and Synchronization Algorithm Development

A program has been developed based on Python programming language, which analyzes data from both ends of the cable. The procedures are depicted in Fig. 3-5. First, to find the approximate fault time, an overcurrent criterion is utilized for activation as mentioned in equation (3.3). And then, it identifies all the zero crossing points before the fault. Subsequently, it pinpoints the exact pre-fault zero crossing point. The same program is applied to the data from both ends to find the corresponding pre-fault zero crossing point. Finally, the measurements are synchronized by offsetting the calculated t_0 . The same procedures are applied to synchronize data for all phases.

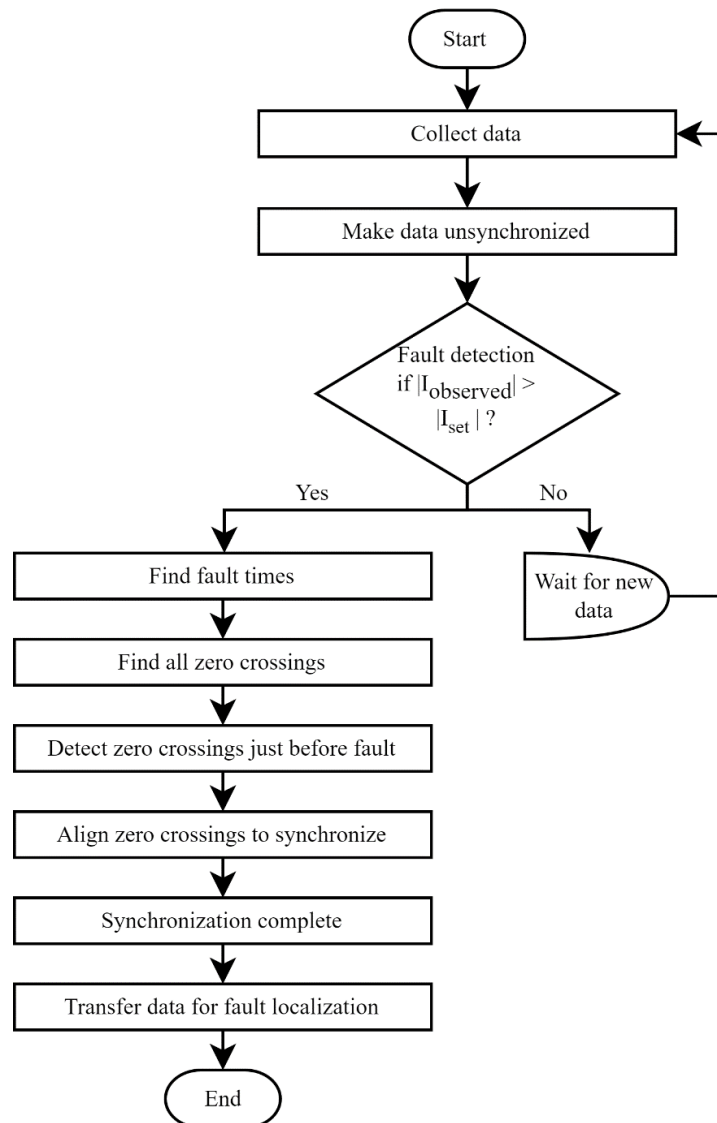


Figure 3-5: Flow diagram of the developed Python program.

3.4. Simulation Model

In this section, PSCAD simulations are carried out. In the practical field, CT sensors are placed on cable ends to acquire current information. Instead of field measurements, the PSCAD simulation data are produced and then fed into the Jetson TX2 for analysis. Assuming the length of the 3 Phase 35 kV underground cable is 20 km, the fault can occur at any phase. In the model setup, different scenarios based on variations of fault at different phases, different fault resistances, different phase shifts, and the addition of different levels of white gaussian noises have been observed to validate the approach. The underground cable transmission system modeled in PSCAD is depicted in fig 3-6.

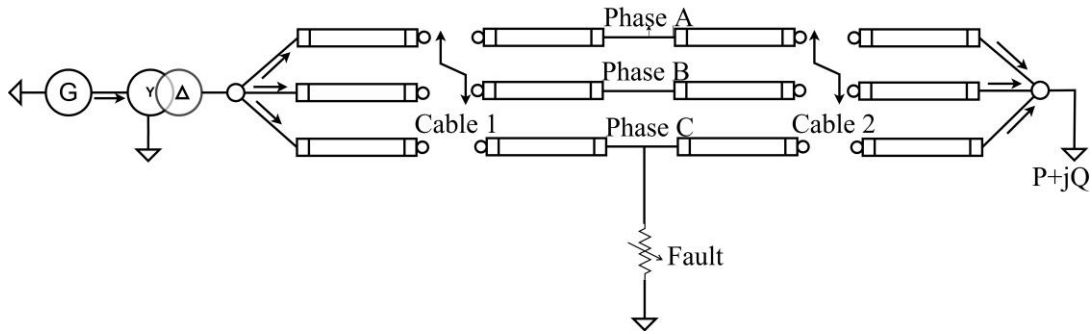


Figure 3-6: 35 kV underground cable transmission system

Phase to ground faults are set at 0.10 s to 0.15 s, where the simulation duration is 0.20 sec, the solution time and channel plot stamp are 10 us. 20000 samples in 0.20 sec are analyzed in the simulations. From simulations, the following effects in three-phase currents during fault have been observed from both ends, as shown in Fig. 3-7 and Fig. 3-8.

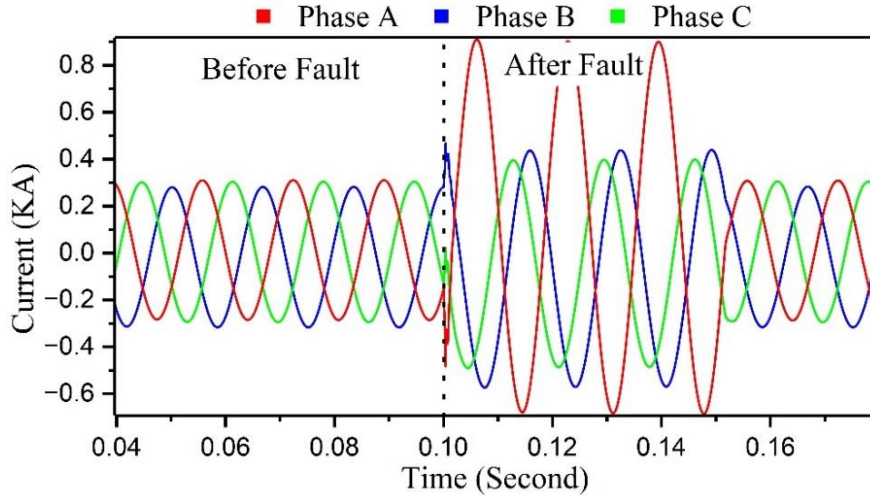


Figure 3-7: Observed signal waveforms from sending end

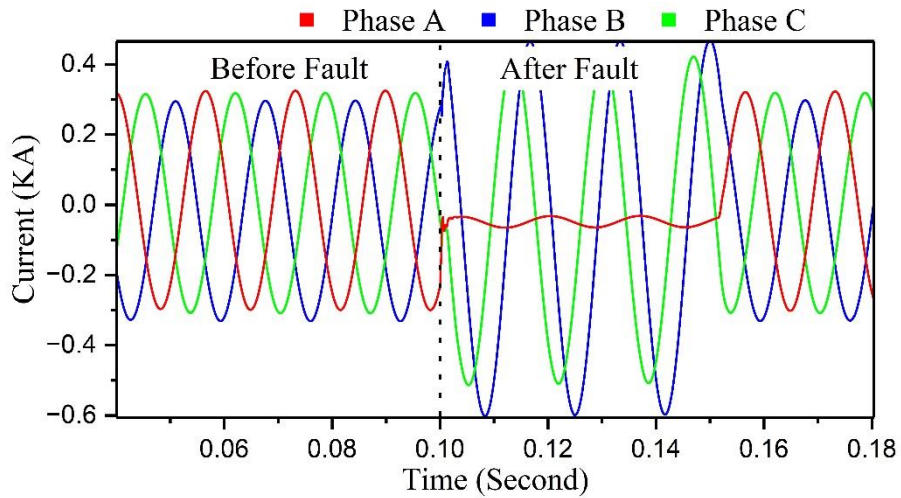


Figure 3-8: Observed signal waveforms from receiving end

3.5. Validation Results for Different Case Studies

In the validations, at first, different fault resistances from 0.01 Ohm to 50 Ohm are considered for faults at different phases. Subsequently, different phase shifts from 0° to 40° are considered for similar fault conditions. Finally, the noise levels of 30 dB and 40 dB are superimposed to the measurement to observe the influence of noise. These parameter variations are considered to look for any possible effects on synchronization error in the proposed method. The online data analysis for synchronization have been done by the developed Python-based program.

3.5.1. Influence of Fault resistance

The results due to faults at three separate phases with different fault resistances are given in Table 3-1, where t_1 , t_2 are the zero-crossing points of the unsynchronized measurements at both ends. t_0 is offset to synchronize the measurement, t_{error} and S are the synchronization errors in the format of time and sampling point, respectively.

Table 3-1: Results for faults at different phases with different fault resistance

Fault Condition	Phase	Unsynchronized Zero Crossing points		t_0 (s)	Synchronization Error	
		t_1 (s)	t_2 (s)		t_{error} (s)	S
Phase A R= 0.01 Ω	A	0.10038	0.09755	0.00300	0.00000	0
	B	0.10580	0.10299	0.00298	0.00002	2
	C	0.10324	0.10041	0.00300	0.00000	0
Phase A R= 5 Ω	A	0.10063	0.09477	0.00603	0.00000	0
	B	0.10605	0.10021	0.00601	0.00002	2
	C	0.10349	0.09763	0.00603	0.00000	0
Phase A R= 10 Ω	A	0.09610	0.09801	0.00174	0.00000	0
	B	0.10152	0.10345	0.00176	0.00002	2
	C	0.09896	0.10087	0.00174	0.00000	0
Phase A R= 20 Ω	A	0.09795	0.09509	0.00303	0.00000	0
	B	0.10337	0.10053	0.00301	0.00002	2
	C	0.10081	0.09795	0.00303	0.00000	0
Phase A R= 50 Ω	A	0.09576	0.10058	0.00465	0.00000	0
	B	0.10118	0.10602	0.00467	0.00002	2
	C	0.09862	0.10344	0.00465	0.00000	0
Phase B R= 0.10 Ω	A	0.10202	0.09869	0.00350	0.00000	0
	B	0.10744	0.10413	0.00348	0.00002	2
	C	0.10488	0.10155	0.00350	0.00000	0
Phase B R= 5 Ω	A	0.09727	0.09961	0.00217	0.00000	0
	B	0.10269	0.10505	0.00219	0.00002	2

Fault Condition	Phase	Unsynchronized Zero Crossing points		t_0 (s)	Synchronization Error	
		$t_1(s)$	$t_2(s)$		t_{error} (s)	S
	C	0.10013	0.10247	0.00217	0.00000	0
Phase B R= 10 Ω	A	0.09588	0.09646	0.00041	0.00000	0
	B	0.10130	0.10190	0.00043	0.00002	2
	C	0.09874	0.09932	0.00041	0.00000	0
Phase B R= 30 Ω	A	0.09823	0.09485	0.00355	0.00000	0
	B	0.10365	0.10029	0.00353	0.00002	2
	C	0.10109	0.09771	0.00355	0.00000	0
Phase B R= 50 Ω	A	0.09937	0.09635	0.00319	0.00000	0
	B	0.10479	0.10179	0.00317	0.00002	2
	C	0.10223	0.09921	0.00319	0.00000	0
Phase C R= 0.01 Ω	A	0.09838	0.09754	0.00101	0.00000	0
	B	0.10380	0.10297	0.00100	0.00001	1
	C	0.10124	0.10040	0.00101	0.00000	0
Phase C R= 5 Ω	A	0.10263	0.09973	0.00307	0.00000	0
	B	0.10805	0.10516	0.00306	0.00001	1
	C	0.10549	0.10259	0.00307	0.00000	0
Phase C R= 10 Ω	A	0.10046	0.10088	0.00025	0.00000	0
	B	0.10588	0.10631	0.00026	0.00001	1
	C	0.10332	0.10374	0.00025	0.00000	0
Phase C R= 20 Ω	A	0.09887	0.10211	0.00307	0.00000	0
	B	0.10429	0.10754	0.00308	0.00001	1
	C	0.10173	0.10497	0.00307	0.00000	0
Phase C R= 50 Ω	A	0.09450	0.10134	0.00667	0.00000	0
	B	0.09992	0.10677	0.00668	0.00001	1
	C	0.09736	0.10420	0.00667	0.00000	0

From Table 3-1, it is observed that the offset t_0 can be obtained successfully, and the maximum error for synchronized measurements is within 0.02 ms or 2 sampling points, which demonstrates that the proposed approach has great synchronization capability. Additionally, it is found that variations of fault resistance at different phases have no influence on synchronization error.

Here, only phase B shows some errors, which are believed to be associated with non-constant phase lag in PSCAD software. The phase lag t_α is due to the non-resistive cable transmission which is found as 0.00017 seconds from the PSCAD simulations for both A and C phases. On the other hand, it has varied between 0.00017 seconds to 0.00019 seconds for phase B. Thus, it is kept as constant value of 0.00017 seconds to calculate the offset t_0 . So, the resulted offsets t_0 showed deviations for phase B, thus associate synchronization errors.

3.5.2. Influence of Phase shifts

In different systems, the phase shifts of the feeding sources can be different. The proposed approach should be applicable to all systems. Therefore, phase shifts from 5° to 40° are applied on top of the fault conditions similar to the previous analysis, and the results are given in Table 3-2.

Table 3-2: Results for faults at different phases with different fault resistance & phase shifts (φ)

Fault Condition	Phase	Unsynchronized Zero Crossing points		$t_0(s)$	Synchronization Error	
		$t_1(s)$	$t_2(s)$		$t_{error}(s)$	S
Phase A R= 0.30 Ω $\Phi= 10^\circ$	A	0.09711	0.09437	0.00291	0.00000	0
	B	0.10262	0.09989	0.00290	0.00001	1
	C	0.10002	0.09728	0.00291	0.00000	0
Phase A R= 10 Ω $\Phi= 10^\circ$	A	0.10131	0.10093	0.00055	0.00000	0
	B	0.10682	0.10645	0.00054	0.00001	1
	C	0.10422	0.10384	0.00055	0.00000	0
Phase A R= 20 Ω $\Phi= 20^\circ$	A	0.09686	0.09467	0.00236	0.00000	0
	B	0.10247	0.10028	0.00236	0.00000	0
	C	0.09981	0.09762	0.00236	0.00000	0

Fault Condition	Phase	Unsynchronized Zero Crossing points		$t_0(s)$	Synchronization Error	
		$t_1(s)$	$t_2(s)$		$t_{error}(s)$	S
Phase A R= 30 Ω $\Phi= 30^\circ$	A	0.09389	0.09637	0.00231	0.00000	0
	B	0.09958	0.10207	0.00232	0.00001	1
	C	0.09688	0.09936	0.00231	0.00000	0
Phase A R= 50 Ω $\Phi= 5^\circ$	A	0.10047	0.10082	0.00018	0.00000	0
	B	0.10594	0.10630	0.00019	0.00001	1
	C	0.10336	0.10371	0.00018	0.00000	0
Phase B R= 0.20 Ω $\Phi= 5^\circ$	A	0.09458	0.09934	0.00459	0.00000	0
	B	0.10005	0.10482	0.00460	0.00001	1
	C	0.09747	0.10223	0.00459	0.00000	0
Phase B R= 5 Ω $\Phi= 10^\circ$	A	0.09574	0.09978	0.00387	0.00000	0
	B	0.10134	0.10539	0.00388	0.00001	1
	C	0.09869	0.10273	0.00387	0.00000	0
Phase B R= 10 Ω $\Phi= 10^\circ$	A	0.09485	0.10174	0.00672	0.00000	0
	B	0.10036	0.10726	0.00673	0.00001	1
	C	0.09776	0.10465	0.00672	0.00000	0
Phase B R= 20 Ω $\Phi= 30^\circ$	A	0.09978	0.10043	0.00048	0.00000	0
	B	0.10547	0.10613	0.00049	0.00001	1
	C	0.10277	0.10342	0.00048	0.00000	0
Phase B R= 50 Ω $\Phi= 5^\circ$	A	0.09527	0.09963	0.00419	0.00000	0
	B	0.10074	0.10511	0.00420	0.00001	1
	C	0.09816	0.10252	0.00419	0.00000	0
Phase C R= 0.40 Ω $\Phi= 20^\circ$	A	0.09604	0.10084	0.00463	0.00000	0
	B	0.10164	0.10645	0.00464	0.00001	1
	C	0.09899	0.10379	0.00463	0.00000	0
Phase C R= 5 Ω $\Phi= 10^\circ$	A	0.09702	0.09713	0.00006	0.00000	0
	B	0.10253	0.10265	0.00005	0.00001	1
	C	0.09993	0.10004	0.00006	0.00000	0

Fault Condition	Phase	Unsynchronized Zero Crossing points		$t_0(s)$	Synchronization Error	
		$t_1(s)$	$t_2(s)$		$t_{error}(s)$	S
Phase C R= 10 Ω $\Phi= 10^\circ$	A	0.10087	0.10180	0.00076	0.00000	0
	B	0.10638	0.10732	0.00077	0.00001	1
	C	0.10378	0.10471	0.00076	0.00000	0
Phase C R= 20 Ω $\Phi= 20^\circ$	A	0.09636	0.09863	0.00210	0.00000	0
	B	0.10196	0.10424	0.00211	0.00001	1
	C	0.09931	0.10158	0.00210	0.00000	0
Phase C R= 50 Ω $\Phi= 5^\circ$	A	0.09991	0.09627	0.00381	0.00000	0
	B	0.10538	0.10175	0.00380	0.00001	1
	C	0.10280	0.09916	0.00381	0.00000	0

From the results in Table 3-2, it is observed that the errors are all within 0.01 ms or 1 sampling point, which is almost same as in the previous analysis. Therefore, it validates that the proposed approach is also feasible for different phase shifts and is not influenced by it.

3.5.3. Influence of Noise

The measurements in practical projects often come with noise; therefore, it is also important to consider the noise in the analysis. Considering more realistic scenarios, white Gaussian noises with SNR 30 dB and 40 dB are added to the acquired data before analysis. The results are given in Table 3-3.

Table 3-3: Comparison of results for different noise levels

Fault Condition	Phase	Error for 30 dB		Error for 40 dB	
		$t_{error} (s)$	S	$t_{error} (s)$	S
Phase A R= 5 Ω	A	0.00088	88	0.00007	7
	B	0.00015	15	0.00002	2
	C	0.00010	10	0.00005	5
Phase A R= 50 Ω	A	0.00013	13	0.00005	5
	B	0.00001	1	0.00009	9
	C	0.00001	1	0.00000	0
Phase B R= 0.01 Ω	A	0.00013	13	0.00005	5
	B	0.00000	0	0.00010	10
	C	0.00081	81	0.00002	2
Phase B R= 30 Ω	A	0.00149	149	0.00004	4
	B	0.00007	7	0.00004	4
	C	0.00017	17	0.00006	6
Phase C R= 20 Ω	A	0.00027	27	0.00006	6
	B	0.00018	18	0.00005	5
	C	0.00008	8	0.00000	0
Phase C R= 50 Ω	A	0.00006	6	0.00010	10
	B	0.00018	18	0.00005	5
	C	0.00013	13	0.00002	2

From Table 3-3, it is observed that the synchronization error increases with stronger noise. For the noise level of 40 dB, synchronization errors are within .10 ms or 10 sample points. Considering the wave velocity is 180 m/us, the localization error for 10 sampling points is 1.8 km, which is relatively high. But, when 30 dB noise is considered, the synchronization error can reach 149 sampling points. It is because the noise results in multiple zero-crossing points around amplitude zero, which are shown in Fig. 3-9 and Fig. 3-10 within the blue rectangle, and the influence is more significant when the noise becomes stronger. Thus, finding the exact zero-

crossing point is more difficult in noisy signals than that without noise, which makes synchronization a challenging task for noisy scenarios.

However, the synchronization error can be as low as zero if the zero crossings are detected accurately. The noise influence on the accurate detection of zero-crossing points can easily be solved by using filters or denoising techniques, which should be applied at the time of acquiring data before applying it to the proposed approach. With proper filtering or denoising techniques, the proposed approach will work perfectly as it has been without noise.

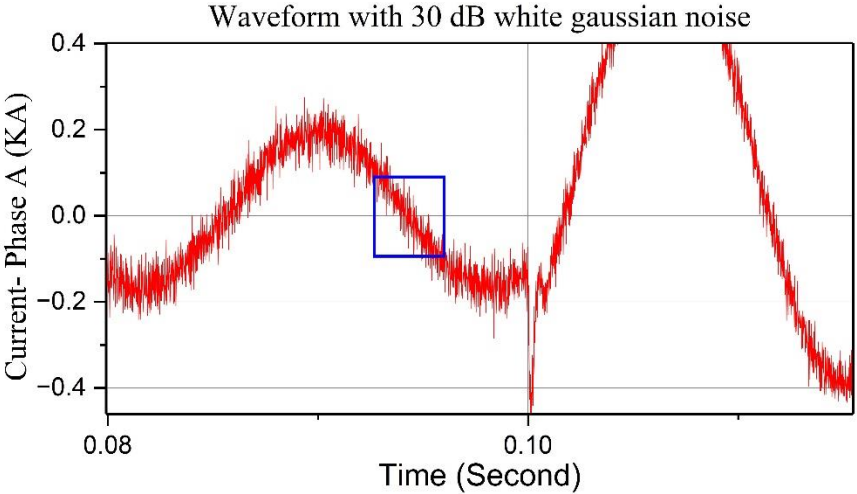


Figure 3-9: Signal waveform with added Gaussian noise of SNR 30 dB

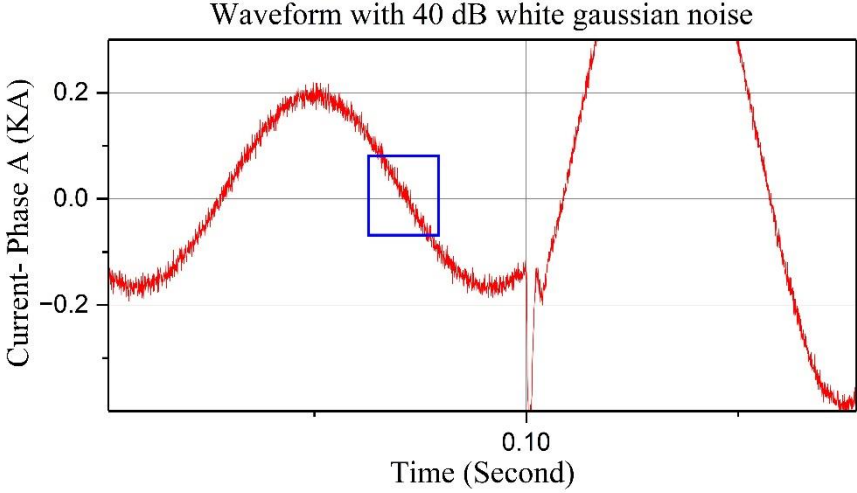


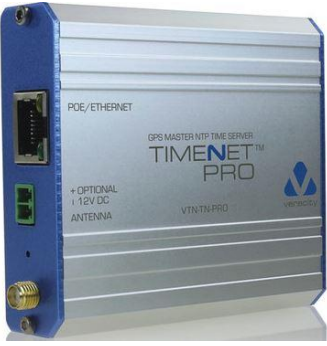


Figure 3-10: Signal waveform with added Gaussian noise of SNR 40 dB

3.6. Cost-Effectiveness of the Proposed Technique

Generally, each GPS synchronization module consists of a GPS receiver and an antenna. An industrial GPS synchronization model requires additional processes and software integration in the measurement platform. Besides, these modules are quite expensive depending on different functions and capabilities. In the following Table 3-4, few available industrial GPS time synchronization modules with prices are given.

Table 3-4: Industrial GPS time synchronization modules with prices

Ser.	Manufacturer and Model	Cost/set
1.	GPS NTP Network Time Server (TM1000A) by Time Machines [50]. 	\$ 470.00 CAD
2.	GPS Receiver Module I-87211W by ICP DAS [51]. 	\$ 617.00 CAD
3.	TIMENET Pro Master NTP Time Server with Antenna by Veracity [52]. 	\$ 659.00 CAD

It is clear from the table that it is costly to add a GPS time synchronization module with a measurement unit. This cost can reach several thousands of dollars with added complexity to the hardware and software setup. However, the proposed approach completely eliminates the use of additional synchronization modules. So, it reduces the additional cost. Besides, in the proposed approach, very small calculations can do the synchronization.

3.7. Test in Experimental Setup

Testing of the developed algorithm was done on the similar experimental platform described in the previous chapter, section 2.5. Data from both ends were needed for synchronization. Since there were only two Jetson devices available, so, computational platform 1 was pre-loaded with PSCAD simulated data of end 1 and the server platform was pre-loaded with PSCAD simulated data of end 2.

Generally, PSCAD simulation data for both ends are synchronized. But to test the algorithm, data should be unsynchronized. So, using a software algorithm, random time offsets were added to the loaded data before analysis. Afterward, bi-directional communication was established between platforms, and data were fetched by the server as like as before. The, the server had both end data to perform synchronization. After initiating the developed algorithm in the server platform, it was seen that, the experimental setup perfectly synchronized data using the developed algorithm without any additional GPS synchronization module.

3.8. Summary

This chapter has presented a synchronization technique in a double-ended data acquisition approach without using external synchronization techniques, only by aligning zero crossing points for each phase before the fault time. The proposed approach is validated through PSCAD/EMTDC simulation and by a developed synchronization program. It is found that regardless of fault at different phases, different fault resistances, and different phase shifts, the proposed approach has accurately pinpointed zero crossing points before the fault with zero synchronization error for most of the cases, which proves that the approach is effective. Proper synchronization can be achieved by filtering or using denoising techniques in noisy scenarios, to help accurate identification of pre-fault zero crossing points. With the characteristics of high accuracy and small calculation, the

proposed approach eliminates the necessity of external synchronization techniques to accurately and quickly detect faults in underground cables.

The proposed method solely depends on finding exact zero-crossing points before fault, which is a pre-fault condition and is not affected by types of faults. So, the validations are done based on data from single line to ground faults with varying scenarios.

The following chapter focuses on developing a machine learning-based approach for predictive data up-sampling for accurate detection of traveling wave arrival time.

Chapter 4: Data Up-Sampling by Machine Learning

A machine learning-based data up-sampling model is developed in this chapter. First, section 4.1 introduces the necessity of up-sampled data for accurate traveling wave arrival time detection. Subsequently, available techniques for up-sampling and drawbacks are discussed. To solve the problems, the step-by-step process of the state of art machine learning model development is described in detail in section 4.2 along with data preprocessing steps. The validation study is carried out in subsequent section 4.3, where simulation and automated models for data generation are explained. Then comparative studies between interpolation and machine learning approaches are presented for different fault conditions. The developed machine learning model and data analysis algorithm are tested in experimental hardware setup in the section 4.4. Finally, section 4.5. concludes the chapter with an overall summary.

4.1. Introduction and Available Up-Sampling Techniques

The accuracy of traveling wave-based fault localization is dependent on how accurately its arrival time is detected. Recalling equation (3.1) from chapter 3, the fault location L is solely dependent on t_a and t_b , which are the arrival times of the traveling waves at each end. The length of the cable l and propagation velocity v are the known parameters.

$$L = 1/2 (l + (t_a - t_b)v) \dots \dots \dots (3.1)$$

If the signals are acquired with low sampling rate, the signals will have fewer data points over time, which leads to inaccurate traveling wave arrival time detection. The following figures 4-1 and 4-2 provide an overview of how signal with lower and higher sampling rates differ in finding accurate traveling wave arrival times. Here, sudden change in amplitude is the considered activation function for traveling wave arrival time detection. From the high-sampled data, the detected arrival time is at 1.2 seconds, whereas the detection time from lower-sampled data is at 1.5 seconds. So, it is clear that with the higher number of samples in the signal, the detection of traveling wave arrival time is more accurate.

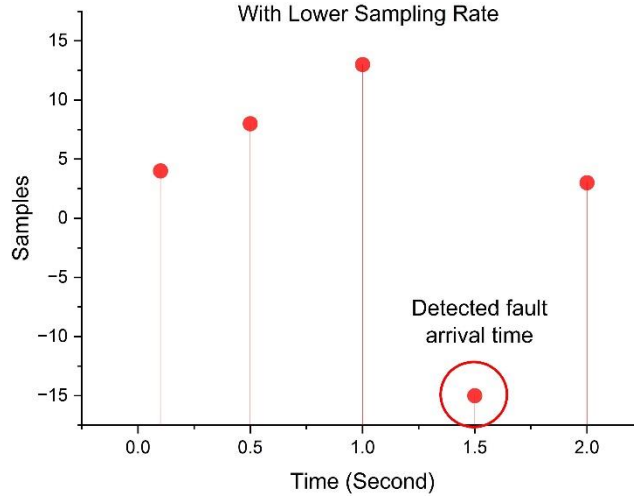


Figure 4-1: Sample signal with lower sampling rate

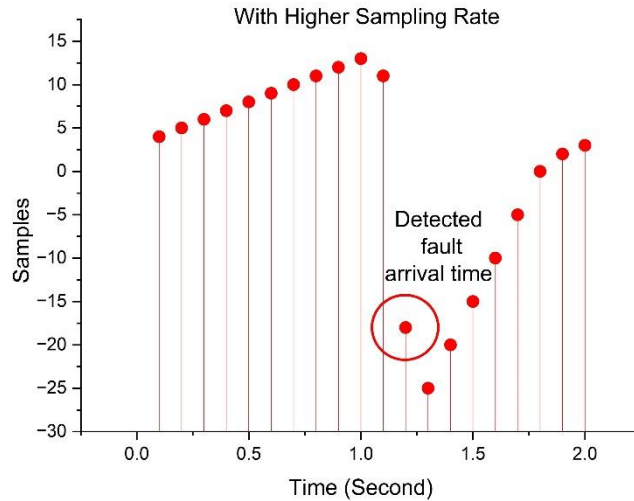


Figure 4-2: Sample signal with higher sampling rate

However, acquiring signals with higher sampling rate is not cost-effective; it requires expensive data acquisition units. Besides, acquiring high-volume data over time creates data processing and storage issues. There are cost-effective approaches for up-sampling based on interpolation techniques [53]-[54].

Interpolation is the process where a simple function is created based on the samples from given data to estimate samples or unknown points in between. There are a group of different interpolation techniques, such as nearest-neighbor interpolation, polynomial interpolation, cubic interpolations, quadratic interpolation, linear interpolation, etc. [55]-[59]. However, several non-

negligible drawbacks exist in interpolation, such as predicted samples by interpolations may suffer smoothing effects on cliff or fault lines, difference in predicted minimum and maximum values with the input data set, estimation problems for above maximum or below minimum values, prediction problems in data with abrupt high/low values, poor performance due to extreme difference in values, etc. Especially, when the data patterns are unknown, such as in the case of fault, interpolation techniques may suffer to predict realistic samples.

Machine learning model can solve the above-mentioned problems if trained by real inputs and outputs for different fault types [60]. The machine learning model can learn the fault patterns, which cannot be done by the conventional interpolation techniques. By learning from different fault patterns, predicted up-sampled data can be as realistic as actual, which leads to accurate detection of traveling wave arrival time.

4.2. The Proposed Machine Learning Based Up-Sampling Model

A machine learning-based up-sampling model is developed to transform 1 MHz data into 5 MHz data by predictive up-sampling. The model is trained by real input and output samples. But, before training, the datasets need to be pre-processed to help the machine learning model to make reasonable correlations between features for better results. So, data pre-processing steps are first explained in detail in the following section. Then the step-by-step process of machine learning model development is illustrated elaborately.

4.2.1. Data Pre-processing

Generally, a machine learning model makes prediction based on the training by real input and output data. It learns correlation within the input parameters by defining the weights and biases to relate to the actual outputs. The datasets should contain at least multiple input features to learn the correlation to predict the outputs. Besides, input datasets should not have missing samples. But it is quite the opposite case here because, based on one input feature, the corresponding outputs need to be predicted. Also, the output size is five times more than the input samples, which is shown in Fig. 4-3; besides, the input datasets don't have enough features and samples to make reasonable correlations in between. So, it is realized that in such conditions, it will be difficult to develop a good predictive model.

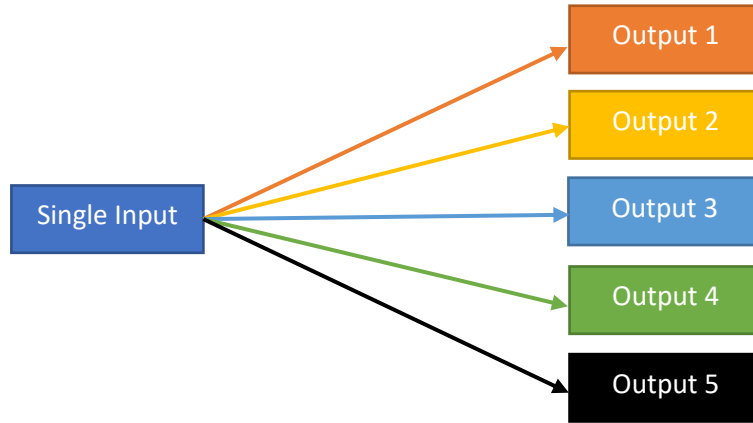


Figure 4-3: Single input sample has five corresponding output samples to predict.

To help building better machine learning model, a few initial steps of data pre-processing are followed. The rate of up-sampling is a known parameter. For example, in 1 MHz data, between 0.200000 seconds to 0.200001 seconds, there will be no more time stamps. But to make it 5 MHz data, in between, there will be four more-time stamps. With a defined rate of up-sampling, it is known how many extra time stamps is there. So, in the input dataset, desired time stamps are inserted as an initial step of data pre-processing, which is shown in Fig. 4-4. In this way, the machine learning model can have a better understanding to make reasonable correlations within available input samples and time stamps.

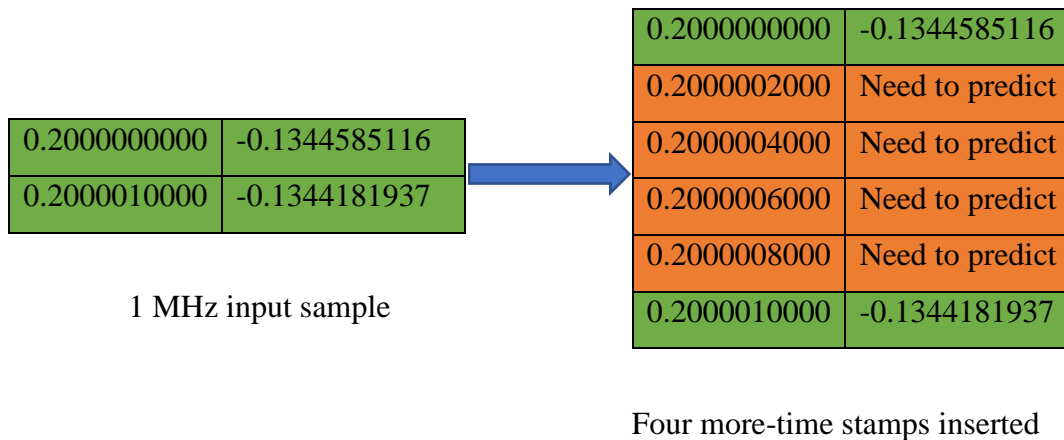


Figure 4-4: Desired time stamps are inserted into input signals.

After inserting the time stamps into the input datasets, there is only one sample value is available within five-time stamps. The other four samples remain as Nan values within the dataset. With 80% of missing values, still, it is not a good input dataset for prediction. To help preparing the dataset for better machine learning model creation, missing samples can be filled using interpolation techniques. Interpolation techniques have drawbacks, as discussed at the beginning of the chapter, but it is still a great idea to fill the Nan values with interpolation, as shown in Fig. 4-5. So, different types of interpolation techniques are applied, such as backfill, forward fill, nearest interpolation, cubic, quadratic, linear interpolation, etc. Finally, these pre-processed data are fed into the machine learning model for prediction.

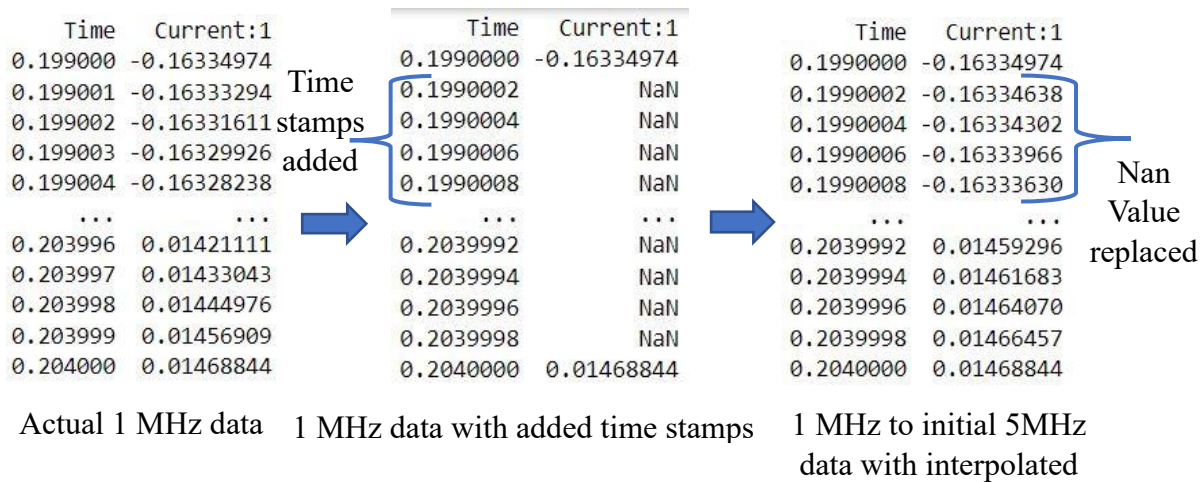


Figure 4-5: Original 1MHz signal pre-processed to 5 MHz by interpolation.

4.2.2. Machine Learning Model Development

Supervised and unsupervised learnings are the two general categories in machine learning [61]. In supervised learning, there are labeled datasets. To predict the outputs accurately, the machine learning model is trained by these labeled datasets in supervised learning. In the training process, the model learns from real input-output datasets by keep measuring model accuracy over time. On the other hand, when datasets are not labeled, machine learning algorithms analyze patterns from datasets and try making co-relations to produce outputs. This is widely known as unsupervised learning. Here, supervised learning-based machine learning is the approach, as input data are to be increased into the higher sampling data by learning from labeled input-output datasets with actual fault patterns.

In the process of developing a machine learning model, it follows a sequential step-by-step process. These steps are dependent on the type of data and the goal to achieve. Here, the goal is to increase the number of samples. Here, this model is to transform the 1 MHz input dataset to 5 MHz output datasets containing fault data. So, the actual 1 MHz input dataset and corresponding 5 MHz datasets are organized for training the model. But, before that, the raw datasets are pre-processed in the way, as mentioned in the previous section. Then, starting with the training, these pre-processed datasets go through data preparation steps such as multi-category encoding and normalizations to extract valuable information and features from datasets. Following that, neural network layers are created to make reasonable correlations and set trainable permeates. In between, there are non-linear layers that work as activation functions to speed up the process and increase computational efficiency. Finally, regression layers predict continuous output data from input variables. After the model is trained, whenever a 1 MHz signal goes through the model, depending on the type of fault in it, a realistic prediction is made from the knowledge of training, and the predicted data is an up-sampled 5 MHz signal. The developed predictive model is depicted in Fig. 4-6, and subsequently, details of each step are illustrated.

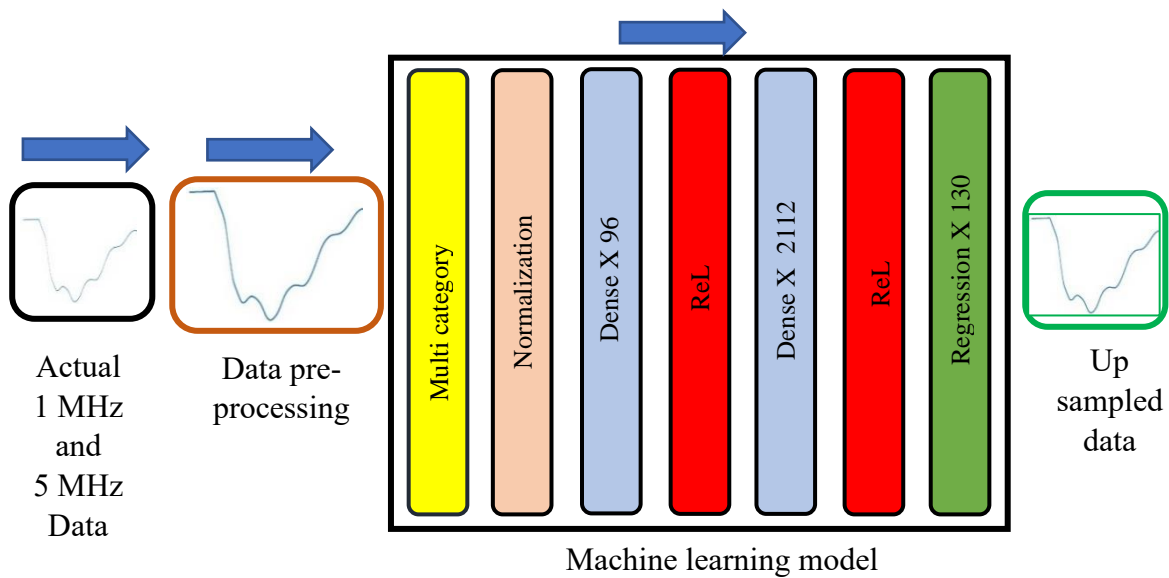


Figure 4-6: Step by step development of machine learning model.

a) Multi-Category Encoding

As the very first step, the datasets go through a multi-category encoding process, which can also be derived as a data pre-processing step. The model needs to extract information from data and understand the pattern, but unless categorical variables are converted into numerical variables, the machine learning model can't process the information [62]. So, before giving it for training the model, the primary step is to pre-process the data by converting into numerical variables.

b) Normalization

In the subsequent step, the normalization technique is applied for data preparation. When datasets have differences in the range of values, it becomes difficult for the machine learning model to extract features. Using a common scale can solve the problem while dealing with data having large differences in values, such as fault data [63]. In the common scale, data remains in shape without losing information, and it is done by normalizing the data within a common scale.

c) Dense Layers

One of the main parts of a neural network-based machine learning model is the dense layers. After the previous steps of data preparation, dense layers are created where every single layers of neurons are connected to all the previous input neurons to make a dense neural network [64]. This is important to manipulate or change the matrix vectors. These matrix vectors are trainable permeates. These get trained and updated by forward and backward propagations to produce accurate output data layers.

d) Rectified Linear Unit (ReLU)

The training input and output datasets contain a large volume of data with different fault patterns. To process this large volume of data requires a huge number of neurons which eventually leads to a long processing time to produce outputs. To solve this problem, an activation function is used to decide which neurons to activate by the weighted sums and biases. This can speed up the process. It also introduces non-linearity, so that the complex patterns of faults can be learned by the model. The chosen activation function is rectified linear unit (ReLU), which is a non-linear

function that activates the neurons to produce outputs only when input is positive, otherwise produces zero [65]. In this way, all the neurons are not activated at the same time, which is the main benefit of using ReLU. Eventually, the process of training becomes faster. Besides, due to only a certain number of neurons being activated at a time, it is more computationally efficient.

e) Regression

Predictive modeling in machine learning is divided into regression and classification-based approaches. When the requirement is to predict continuous data from input variables, regression algorithms are mostly used [66]. This algorithm creates an estimated mapping function combining inputs and outputs for continuous predictive output generation. On the other hand, in classification, the mapping function predicts discrete output variables with labels or categories. Here, the goal is to predict continuous discrete values in the form of integer numbers, so a regression-based model is the best fit. Co-relations are established by learning from variables in training datasets and estimating the effects of one variable on others.

The developed machine learning model combines all the discussed steps to finally produce predicted up-sampled data. Finalizing all these steps required many trials and error analysis by keep correcting or adding different techniques in between.

4.3. Validation Study

To train the machine learning model, input and the corresponding output datasets are required. In the validations, PSCAD/EMTDC simulation model is used to generate data where different fault conditions are considered. Here, fault can occur anywhere within a 5 km long underground cable with different fault resistances and different phase-to-ground faults. A Python-based program is developed to automate the simulation process to generate data for different fault conditions. In the subsequent stages of the validation study, to see whether the machine learning approach improves the accuracy of finding the traveling wave arrival time, comparisons are made between actual and predicted data. Then, comparisons are observed between interpolation and machine learning approach to understand whether machine learning approach works better than the conventional interpolation techniques or not.

4.3.1. Data Generation Model

A PSCAD model has been created to generate 1 MHz fault data and corresponding higher frequency 5 MHz fault data for the same fault conditions. Here, 3 Phase 35 kV underground cables of 5 km are considered. The fault can occur at any phase, at any location within the 5 km, with varying fault resistances of 0.01 Ohm to 50 Ohm. The PSCAD model of the underground cable transmission system is depicted in Fig. 4-7.

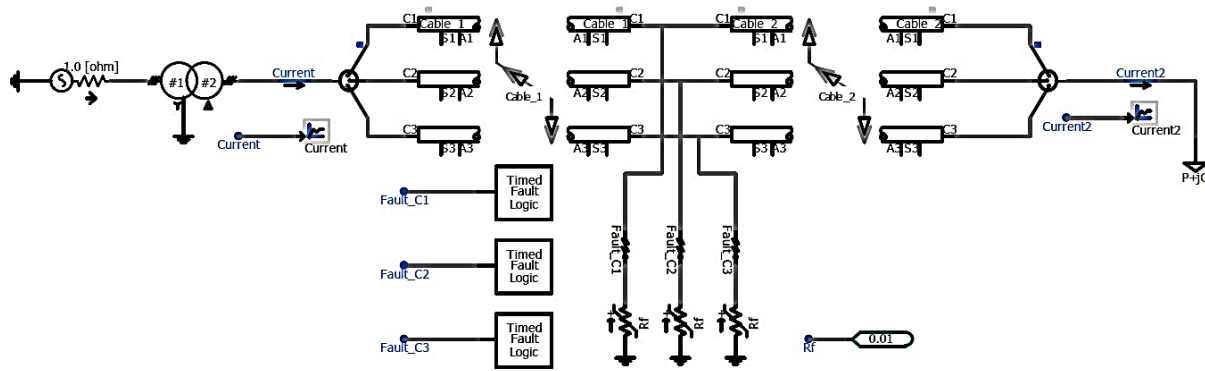
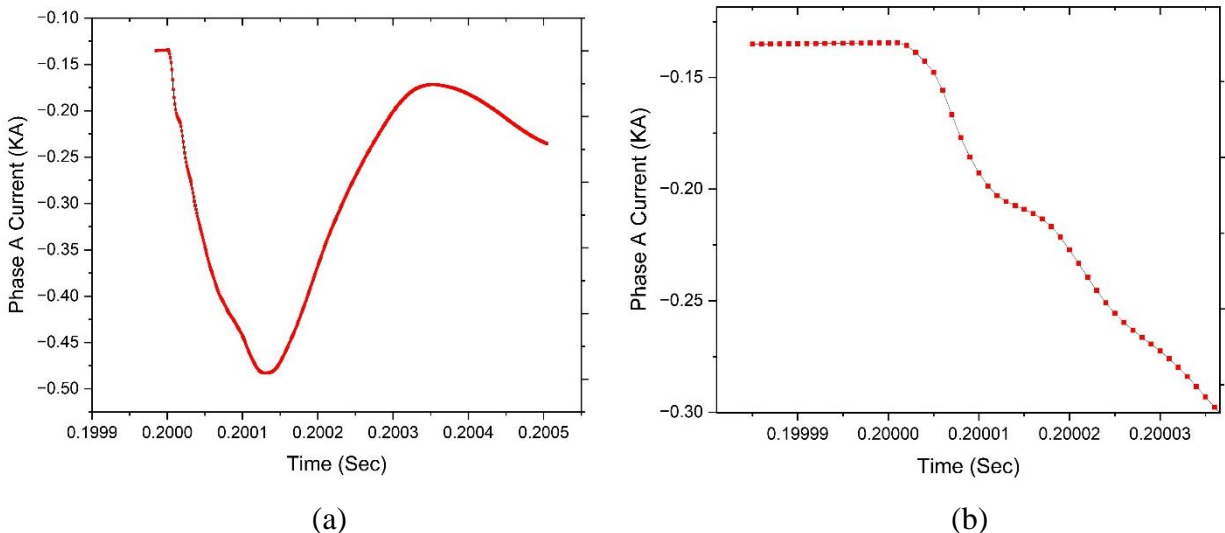


Figure 4-7: PSCAD model of a 35 kV underground cable transmission system.

From 1 MHz and 5 MHz simulation data, observed signals due to single phase to ground faults are shown in Fig. 4-8. The 5 MHz signals are constructed with five times higher number of samples as shown in Fig. 4-8 (d). So, each sample in 1 MHz signal has corresponding five samples in 5 MHz signal. So, the machine learning model needs to predict 5 samples from one input sample.



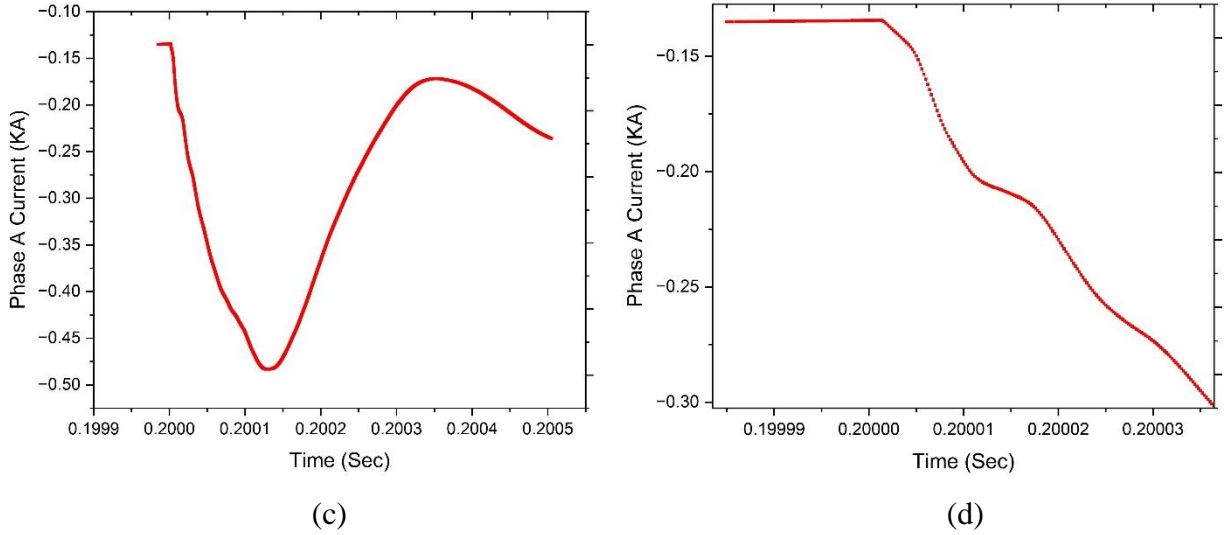


Figure 4-8: (a) 1 MHz faulty signal, (b) Zoom in image of 1 MHz faulty signal, (c) 5 MHz faulty signal, (d) Zoom in image of 5 MHz faulty signal.

4.3.2. Data Generation Automation Algorithm Development

For each fault condition, the considered PSCAD simulation time is .25 seconds. In 1 MHz simulation, each simulation consists of 250000 samples with an average CSV file size of 29 MB. Each 5MHz file has five times more samples with an average CSV file size of 144 MB. So, it is easily understood that generating such amount of data by PSCAD for each case takes a long time. Besides, manually changing the fault conditions where fault can occur at different locations with varying fault resistances and phase-to-ground faults are not practically feasible and would be very time-consuming. To solve this problem, a Python-based program is developed. In this program, the automated algorithm can initiate the PSCAD software with predefined commands [67], where the fault conditions can be changed by setting different phase-to-ground faults, fault resistances, and locations of the fault. Subsequently, PSCAD model can be simulated automatically for different fault conditions and then the simulation results can be saved individually. The whole process is depicted in Fig. 4-9 as in the flow diagram below.

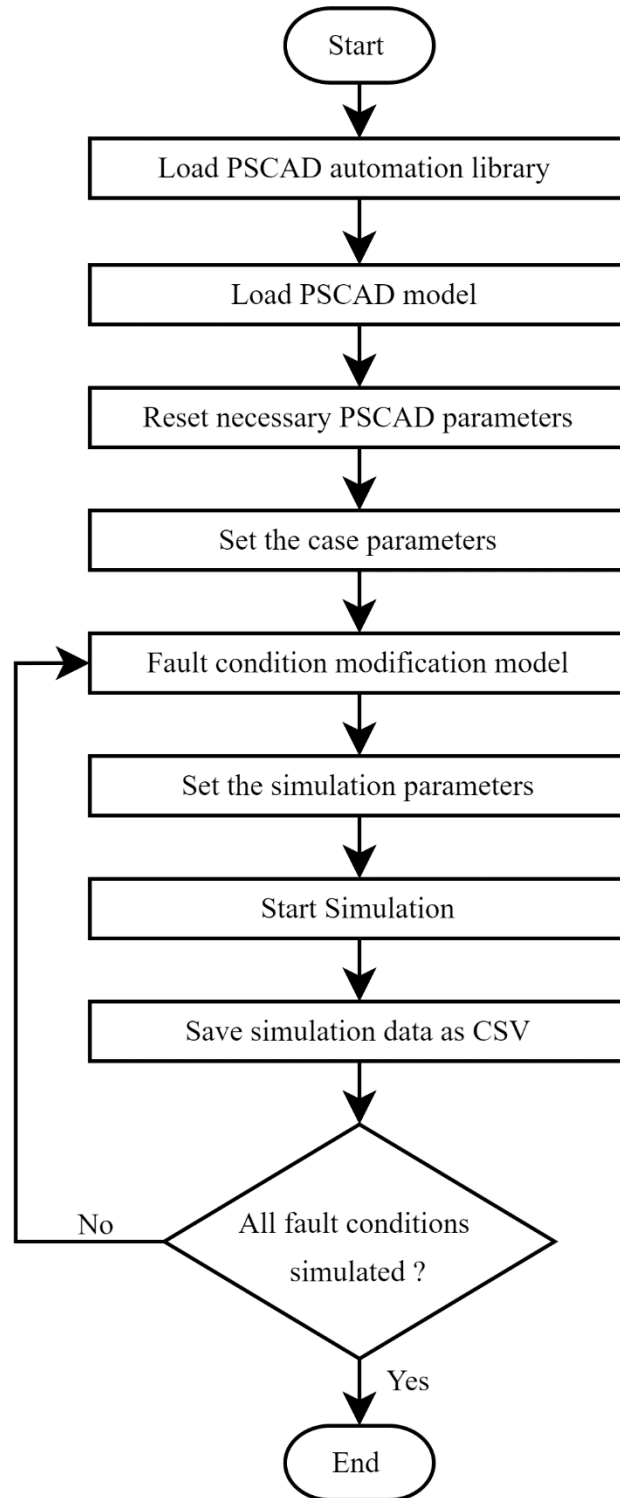
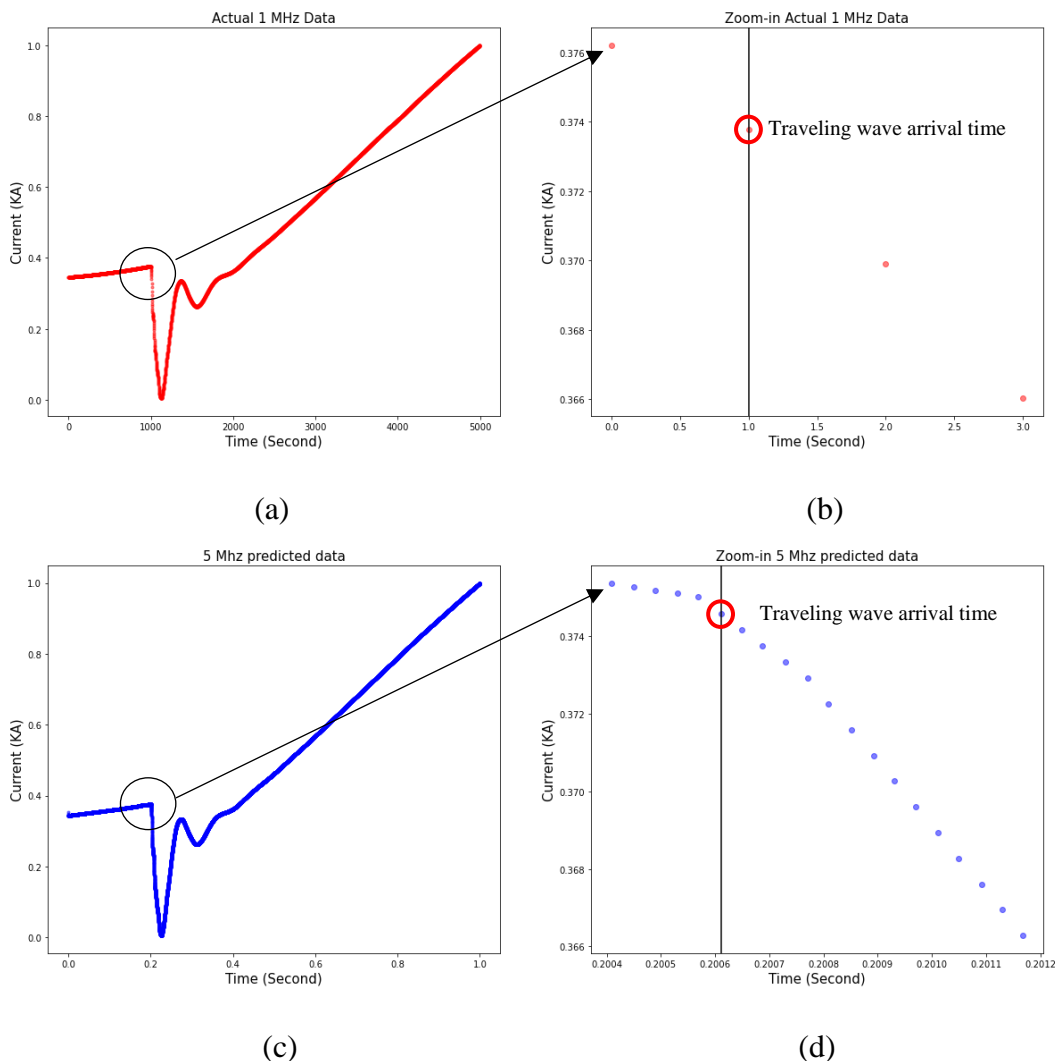


Figure 4-9: Flow diagram of the developed Python program for automated data generation.

4.3.3. Results and Comparisons Between Actual and Predicted Data

The machine learning model increases the number of samples and mimics the pattern of the actual high-sampled signal. The model is trained by different fault conditions. Upon checking the comparative results between actual and predicted data, the arrival times of the traveling waves are found to be more accurate in predicted signals than in actual 1 MHz signals.

Considering a single line to ground fault at 0.4 km with fault resistance of 10 Ohm, the detected traveling wave arrival time is 0.20067 s in an actual 1 MHz signal. But, when the predicted data by the machine learning model is observed, the detected traveling wave arrival time is 0.200612 s. In the actual 5 MHz signal, the observed traveling wave arrival time is 0.20057 s. Here, the improved detection time by the machine learning model is $= (0.20067 - 0.200612) = 58 \mu\text{s}$. Following figures 4-10 (a-f) show traveling wave arrival times for this fault condition.



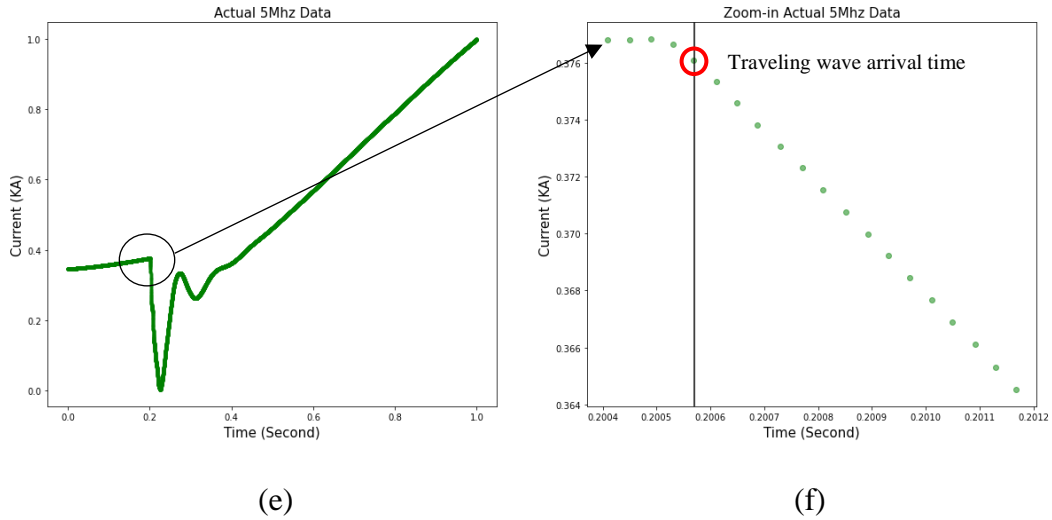
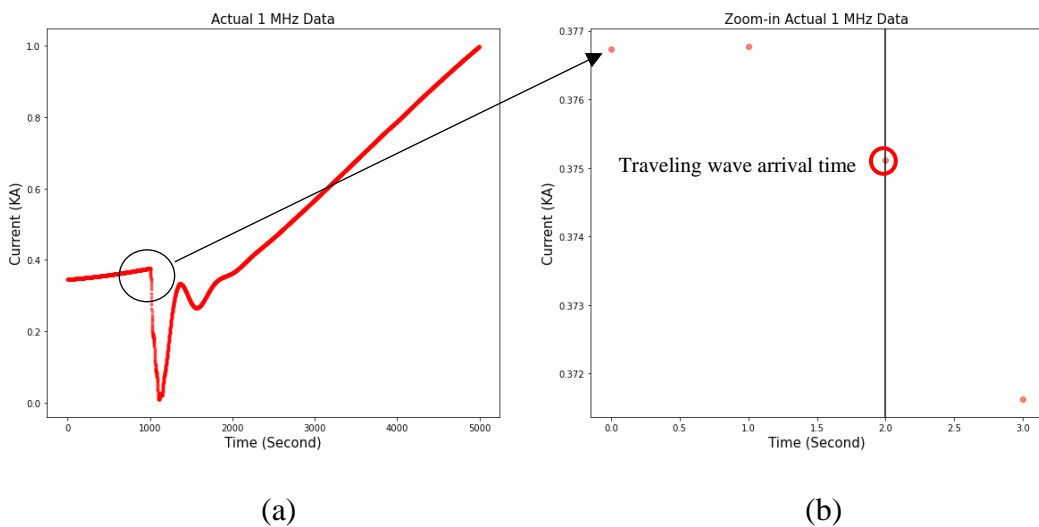
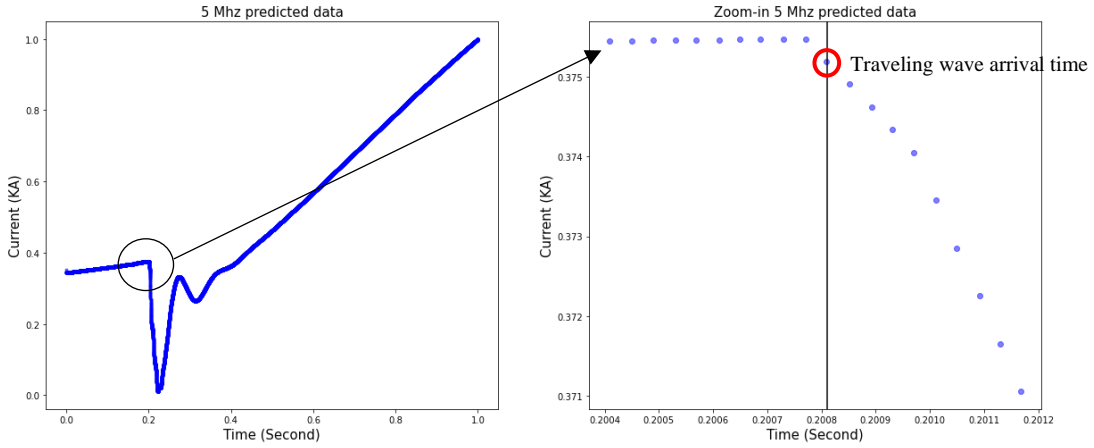


Figure 4-10: (a) Actual 1 MHz signal (b) Zoom-in samples of actual 1 MHz signal (c) Predicted 5 MHz signal (d) Zoom-in samples of predicted 5 MHz signal (e) Actual 5 MHz signal (f) Zoom-in samples of actual 5 MHz signal, for fault at 0.4 km with fault resistance of 10 Ohm.

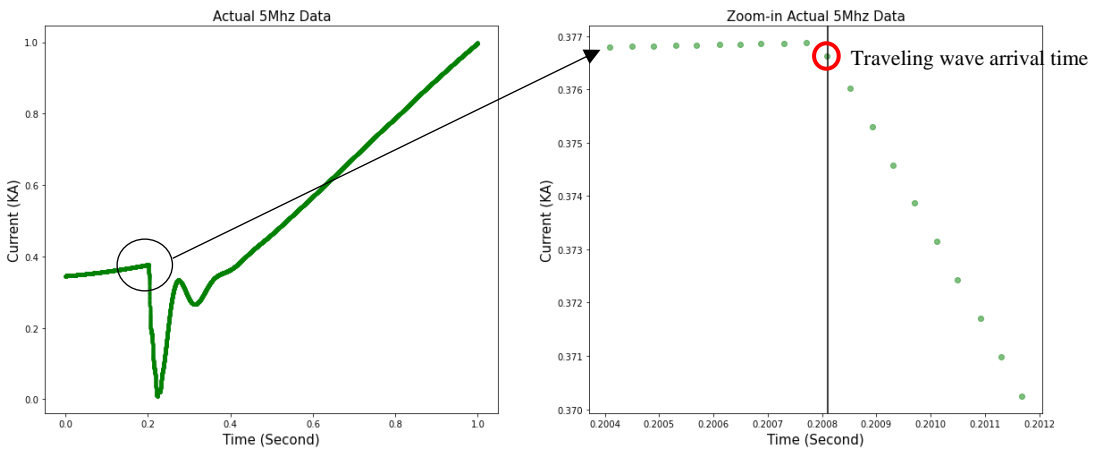
For different fault conditions, similar improvements are achieved. The maximum accuracy has been observed for several cases, where the predicted signal and actual 5MHz signal has the same traveling wave arrival times. As an example, for a fault at 0.6 km with fault resistance of 0.01 Ohm, the traveling wave arrival times in the predicted signal and actual 5 MHz signals are the same, which is 0. 20081 s, where the arrival time in actual 1 MHz signal is 0.20091 s. The following figures 4-11 (a-f) show traveling wave arrival times for this fault condition.





(c)

(d)



(e)

(f)

Figure 4-11: (a) Actual 1 MHz signal (b) Zoom-in samples of actual 1 MHz signal (c) Predicted 5 MHz signal (d) Zoom-in samples of predicted 5 MHz signal (e) Actual 5 MHz signal (f) Zoom-in samples of actual 5 MHz signal for fault at 0.6 km with fault resistance of 0.01 Ohm.

More results due to different fault conditions are given in Table 4-1, where t_1 , t_{p5} , and t_5 are the detected traveling wave arrival times in actual 1 MHz, predicted 5 MHz and actual 5 MHz data respectively. t_{imp} is the improved traveling wave arrival time due to machine learning-based prediction.

Table 4-1: Results and comparisons between actual and predicted data

Fault Condition	Actual 1 MHz	Predicted 5 MHz	Actual 5 MHz	t_{imp} (μ s)
	t_1 (s)	t_{p5} (s)	t_5 (s)	
0.4 KM 30 Ohm	0.200670	0.200612	0.200570	58
1 KM 20 Ohm	0.200910	0.200810	0.200810	100
1.2 KM 1 Ohm	0.201470	0.201390	0.201370	80
2.2 KM 1 Ohm	0.200910	0.200810	0.200810	100
2.4 KM 30 Ohm	0.201711	0.201662	0.201611	49
4.2 KM 30 Ohm	0.202110	0.202040	0.202010	70

From the above observations, it is clear that the accurate finding of traveling wave arrival time improves when the machine learning-based predicted up-sampled signal is used. Detection time improvement can reach the highest accuracy of 100 μ s when predicted and the actual 5 MHz signal has the same arrival times. Any influences due to different fault conditions are not found.

4.3.4. Comparisons Between Interpolation and Proposed Method

The developed machine learning model helps finding traveling wave arrival times more accurately by predictive up-sampling. The model is developed by a rigorous training process with actual input-output fault datasets to make the predictions more accurate. There are interpolation techniques to perform the up-sampling as well. So, a comparative study has been done to understand how much improvements can be achieved by machine learning. Here, evaluation matrices are based on time error comparisons and mean squared error comparisons between interpolation and machine learning approaches. First, for both approaches, differences between detected traveling wave arrival times in predicted and actual 5MHz signals are observed. Then, average squared differences are calculated.

Looking at the predicted signals in graphs as shown in Fig. 4-12 for fault at 0.4 km with fault resistance of 10 Ohm, if samples are very closely observed, differences between the two approaches are found. The followings are the observations in this fault condition.

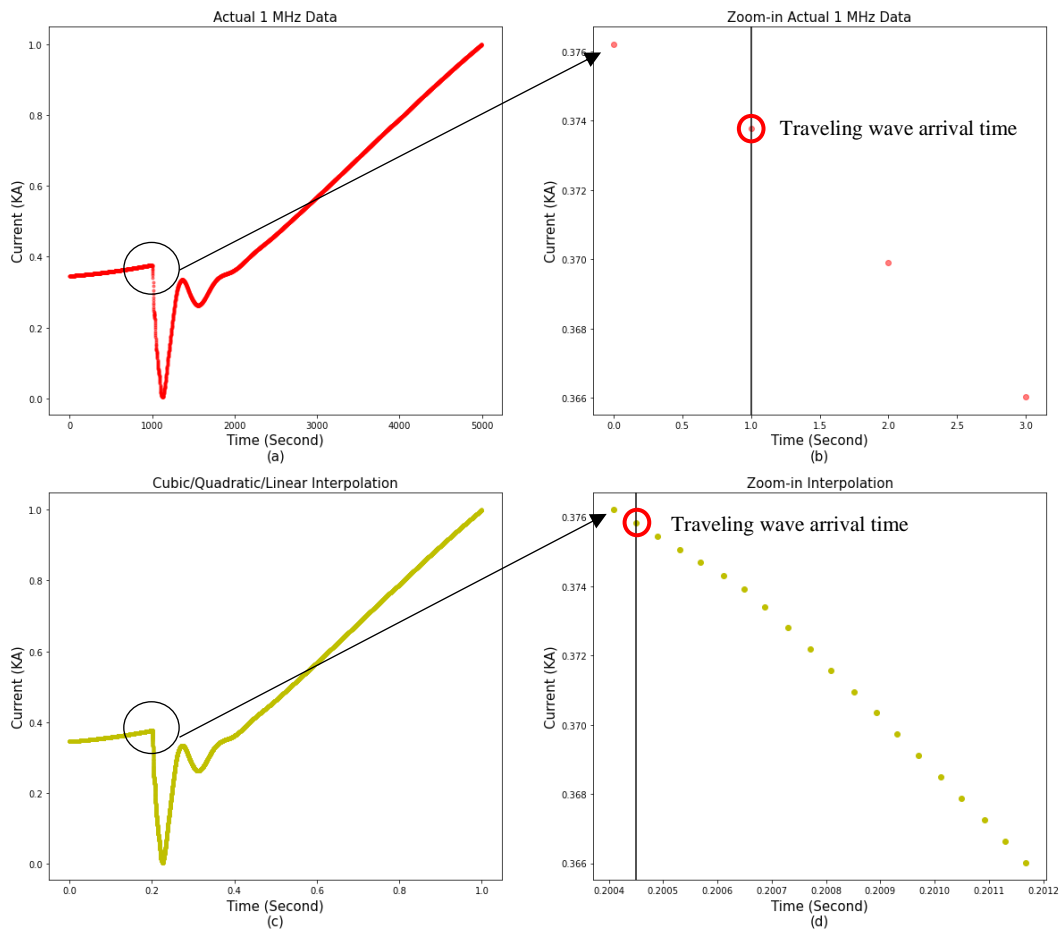
Detected traveling wave arrival time in predicted 5MHz signal by interpolation = 0.20045 s

Detected traveling wave arrival time in predicted 5MHz signal by machine learning = 0.200612 s

Detected traveling wave arrival time in actual 5 MHz signal = 0.20057 s

Time error in interpolation = (0.20057 - 0.20045) s = 120 μ s

Time error in machine learning = (0.200612 - 0.20057) s = 42 μ s



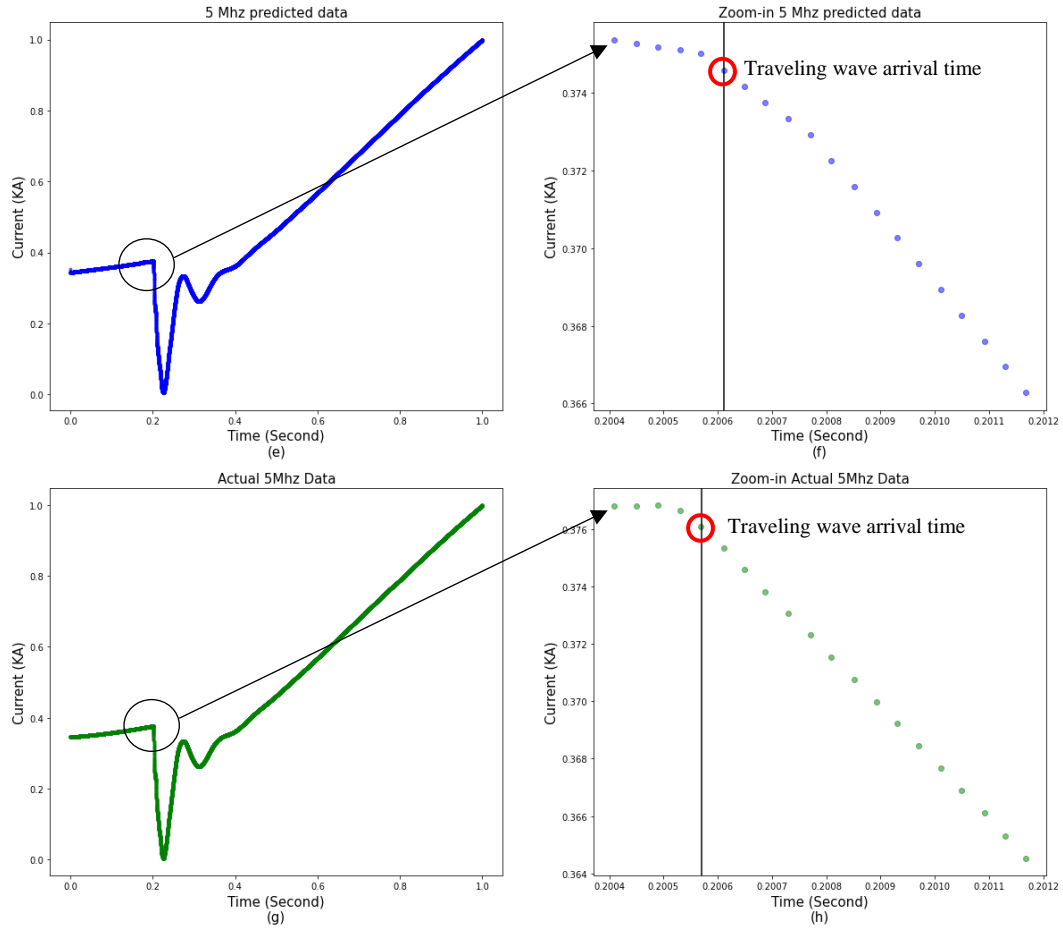


Figure 4-12: (a) Actual 1 MHz signal (b) Zoom-in samples of actual 1 MHz signal (c) Predicted 5 MHz signal by interpolation (d) Zoom-in samples of predicted 5 MHz signal by interpolation (e) Predicted 5 MHz signal by machine learning (f) Zoom-in samples of predicted 5 MHz signal by machine learning (g) Actual 5 MHz signal (h) Zoom-in samples of actual 5 MHz signal for fault at 0.4 km with fault resistance of 10 Ohm.

More comparisons are presented in the following table 4-2. It is clearly seen that the machine learning approach performs better in all observed fault conditions. The maximum time error found in the machine learning approach is 85 μs . On the other hand, it reaches up to 283 μs for interpolation. Besides, the machine learning approach can achieve the highest accuracy with a null time error. But the lowest time error observed in interpolation is 120 μs , which is higher than the highest time error found in the machine learning approach. This undoubtedly proves that the machine learning approach performs better while finding accurate traveling wave arrival time.

Table 4-2: Results and comparisons between interpolation and machine learning based prediction

Fault Conditions	Interpolation	Machine Learning	Interpolation	Machine Learning
	Time Error (S)	Time Error (S)	MSE	MSE
0.2 KM 50 Ohm	0.000245	0.000085	2.0645834E-05	8.4430360E-07
0.4 KM 10 Ohm	0.000120	0.000042	2.0575682E-05	6.3892130E-07
0.4 KM 30 Ohm	0.000120	0.000042	2.0575682E-05	6.3892130E-07
0.6 KM 0.01 Ohm	0.000120	0.000000	2.0439455E-05	8.6522120E-07
0.8 KM 1 Ohm	0.000240	0.000080	2.0444513E-05	1.6121558E-06
1 KM 20 Ohm	0.000121	0.000000	2.0508724E-05	8.6525980E-07
1.4 KM 40 Ohm	0.000241	0.000081	2.0512562E-05	8.4422476E-07
1.8 KM 1 Ohm	0.000120	0.000042	2.0378651E-05	2.7949795E-06
2.2 KM 1 Ohm	0.000120	0.000000	2.0508724E-05	8.6525980E-07
2.8 KM 30 Ohm	0.000120	0.000000	2.0301228E-05	4.4875733E-06
3.2 KM 25 Ohm	0.000283	0.000163	2.0178124E-05	8.9751320E-06
3.8 KM 0.01 Ohm	0.000239	0.000080	2.0247096E-05	6.5568847E-06
4 KM 15 Ohm	0.000241	0.000081	2.0242860E-05	8.4408470E-07
4.4 KM 1 Ohm	0.000235	0.000080	2.0650004E-05	6.5571040E-06

By comparing predicted signals for both approaches in terms of average squared differences, the machine learning approach reaches the lowest MSE of 8.6525980E-07, where the lowest MSE in interpolation is 2.0650004E-05. Even the lowest MSE in interpolation is higher than the highest MSE 1.6121558E-06 in the machine learning approach, which proves that the machine learning approach makes better overall predictions.

4.4. Test in Experimental Setup

Once the machine learning model is developed, the model can be saved to be loaded later. Similarly, after a rigorous training process in personal computer, the developed model was saved as HDF5 format with a .h5 extension. Then this saved model was loaded into the experimental setup.

First, the computational platform was loaded with 1 MHz fault data of different fault conditions. Then, these fault data went through the loaded machine learning model in the computational platform to increase the sample five times. It was found that the model works perfectly in the computational platform to increase samples. Afterward, the data analysis algorithm traced out the improved arrival times of traveling waves.

4.5. Summary

This chapter has presented a machine learning-based data up-sampling approach to detect the traveling wave arrival times more accurately. To validate the approach, PSCAD/EMTDC simulation data for different fault conditions have been used for training and testing. With pre-processed training data, the machine learning approach has followed a step-by-step development process to finalize the data up-sampling model. It has been found that machine learning-based data up-sampling improves the accuracy of finding traveling wave arrival times. In the comparative analysis between conventional interpolation and machine learning, it has been observed that the machine learning approach performs better. The machine learning approach has less time errors than interpolation while finding accurate traveling wave arrival times. Similarly, mean squared errors are less in the machine learning approach, which prove that overall signal predictions are better by the machine learning. In a nutshell, the proposed machine learning model is suitable and feasible for finding the traveling wave arrival times more accurately. However, the machine learning model will produce realistic outputs for different fault conditions if initially trained by similar fault conditions.

Chapter 5: Conclusion and Future Work

5.1. Conclusion

Fault localization in underground cables in mining industries calls for improvement of existing localization platforms due to these platforms are not solely tailored for the mining industry. These are not fast enough and cost-effective also. Traveling wave-based double ended online fault localization platforms are able to solve these problems with advanced hardware and software setup. With development of fault localization platform, this thesis has majorly contributed to the following developments:

- 1) Finding available fault localization platforms and corresponding equipment with costs.
- 2) A Low-cost double ended online fault localization platform development.
- 3) Development of five separate software algorithms.
 - Bi-directional communication establishment algorithm.
 - Sever management algorithm.
 - Data analysis algorithm for data synchronization.
 - PSCAD data generation automation algorithm.
 - Machine learning model training and data analysis algorithm.
- 4) A novel low-cost data synchronization technique development.
- 5) Development of a machine learning model to improve the accuracy of finding traveling wave arrival times.

5.2. Case Specific Accomplishments

In this thesis, a low-cost double ended online fault localization platform has been developed. It is capable of running advance machine learning algorithms for more accurate fault localization. The features of the developed localization platform are given below.

- It can acquire data from both ends of the cable.
- It has high frequency data acquisition capability.
- It can perform advanced machine learning algorithms for faster and more accurate fault localization.
- The developed software establishes bi-directional communication to transfer and receive data from remote platforms for analysis.
- The software also provides a query-based data management facility for data analysis from stored data.
- The overall cost of a single platform is CAD \$ 2,850.00 which is less than ten times of the available market options.

Secondly, a novel low-cost data synchronization technique has been proposed in this thesis. In a nutshell, the proposed synchronization technique has the following features:

- It doesn't rely on external synchronization devices, which reduces cost.
- Regardless of fault at different phases, different fault resistances, and different phase shifts, the proposed approach is effective.
- In noisy scenarios, with filtering or denoising, it can achieve proper synchronization.
- It contains very small calculations for accurate and quick synchronization.
- It depends only on a pre-fault condition and is not affected by types of faults.

Finally, a machine learning model is developed in this thesis to do predictive up-sampling of fault data. This model has the following features:

- The developed model up-samples data, which improves the accuracy of finding traveling wave arrival times.
- The model eliminates the necessity of costly data acquisition units with high sampling rates.
- The model performs better than conventional interpolation techniques in terms of increased accuracy in finding traveling wave arrival times and better overall predictions.

The combination of all the above approaches makes the fault localization platform a well-rounded cost-effective solution for underground cable fault localization. However, there are scopes of future work, and more developments can be made.

5.3. Future Work

Although the proposed fault localization platform has solved the main problems, there are possible scopes to improve the platform, such as:

- Implementation of the platform in real sites may give more practical aspects to improve the hardware-software platform.
- The server platform can be integrated into an online-based server application to remotely accessed and manage via web application.
- The data acquisition unit is not limited to the use of one type of sensor rather it is an open platform. Other sensors can be used and tested for different scenarios. The following sensors can also be incorporated with the platform for PD based fault localization; Transient Earth Voltage (TEV) for on-line PD measurements in decibels [68], High Voltage Coupling Capacitor (HVCC) sensors to measure PD pulses occurring in the machine's stator winding [69], Airborne Acoustic (AA) sensors to detect airborne acoustic PD signals [70], Bushing Tap Sensor (BTS) to detect, measure and monitor PD in HV transformers and bushings[71], etc.

- More input channels can be incorporated to measure from many cables at a time. It will reduce setup costs, as well as can improve the setup functionality.
- Here, the computational platform and the data acquisition unit are separate units. There are scopes to combine these into a single circuit design.
- The synchronization approach depends only on a pre-fault condition. Here, single line to ground faults with different conditions have been examined. But further studies can be done on line-to-line and multiple-phases-to-ground faults to observe whether there are any effects due to other types of faults.
- Single line to ground fault is the most likely fault that occurs in underground cables. So, the machine learning model has been trained with these fault conditions. It can be trained by more different fault conditions to get desired results for diverse fault conditions.
- Both the synchronization approach and machine learning approach have used simulation data. Using practical field data will definitely improve the platform for practical aspects.

References

- [1] “Mining Sector Performance Report 2011-2022”, Published by Energy and Mines Ministers’ Conference, Newfoundland and Labrador, July 2022.
- [2] The Mining Association of Canada, “Facts and Figures of the Canadian Mining Industry”, 2017.
- [3] K. Chen, C. Huang and J. He, “Fault detection, classification and location for transmission lines and distribution systems: a review on the methods,” *High Voltage*, vol. 1, no. 1, pp. 25-33, 2016.
- [4] C.M. Wiggins, D.E. Thomas, T.M. Salas, F.S. Nickel, and H.W. Ng, “A novel concept for URD cable fault location,” *IEEE Trans. Power Del.*, vol. 9, no. 1, pp. 591–597, Jan. 1995.
- [5] S. Potivejkul, P. Kerdonfag, S. Jamnian, and V. KinnaresN, “Design of a low voltage cable fault detector,” in *Proc. IEEE Power Eng. Soc. Winter Meeting, 2000*, vol. 1, pp. 724–729.
- [6] H. Diaz and M. López, “Fault location techniques for electrical distribution networks: A literature survey,” in *Proc. 5th IASTED Int. Conf.*, Jun. 15–17, 2005, pp. 311–318.
- [7] J. Sadeh and H. Afradi, “A new and accurate fault location algorithm for combined transmission lines using adaptive network-based fuzzy inference system,” *Elect. Power Syst. Res.*, vol. 79, pp. 1538–1545, 2009.
- [8] P. Ray and D. Mishra, “Application of extreme learning machine for underground cable fault location,” *International Transactions on Electrical Energy Systems*, vol. 25, no. 3, pp. 227–3247, 2015.
- [9] J. Moshtagh and R. K. Aggarwal, “A new approach to ungrounded fault location in a three-phase underground distribution system using combined neural networks & wavelet

- analysis,” in Proc. Canadian Conf. Electrical and Computer Engineering, May 2006, pp. 376–381.
- [10] H. Livani, “Supervised Learning-Based Fault Location in Power Grids,” *Big Data Application in Power Systems*, 2018, PP. 303-319.
- [11] Y. Xiang and J.F.G. Cobben, “A Bayesian Approach for Fault Location in Medium Voltage Grids With,” *IEEE Power and Energy Technology Systems Journal*, vol. 2, no. 4, pp. 116-124, 2015.
- [12] G. Xu, Y. Zhou and Y. Chen, "Model-Based Fault Location with Frequency Domain for Power Traction System," *Energies*, vol. 6, no. 7, pp. 3097-3114, 2013.
- [13] Power System Relaying Committee of the IEEE Power Engineering Society, C37.114/D9-IEEE Guide for Determining Fault Location on AC Transmission and Distribution Lines, IEEE, 2005.
- [14] “HVPD Kronos® KPM Condition Monitoring- HVPD.” [Online]. Available: <https://www.hvpd.co.uk/kronos/>. [Accessed: 13-Oct-2021].
- [15] “MONTESTO 200- Omicron.” [Online]. Available: <https://www.omicronenergy.com/en/products/montesto-200/>. [Accessed: 14-Oct-2021].
- [16] “HPM601-P- Rugged Monitoring.” [Online]. Available: <https://www.ruggedmonitoring.com/partial-discharge-products/partial-discharge-monitors/hpm601-p/5df1e5f93f52a90001c14ee3>. [Accessed: 19-Jul-2021].
- [17] “ATS9352 PCI Express Digitizer- AlazarTech.” [Online]. Available: <https://www.alazartech.com/en/product/ats9352/6/>. [Accessed: 20-Dec-2022].

- [18] “DAQ970A Data Acquisition System- Keysight.” [Online]. Available: <https://www.keysight.com/us/en/products/modular/data-acquisition-daq/digital-acquisition-benchtop-system/daq970a-data-acquisition-system.html>. [Accessed: 20-Dec-2022].
- [19] “Dataman 522 60 MHz USB Oscilloscope- Dataman.” [Online]. Available: <https://www.dataman.com/oscilloscopes/dataman-522-60-mhz-usb-oscilloscope.html>. [Accessed: 20-Dec-2022].
- [20] “MP720646 US Multicomp Pro PC Oscilloscopes - Multicomp Pro.” [Online]. Available: <https://canada.newark.com/multicomp-pro/mp720646-us/pc-oscilloscope-2ch-100mhz-500msps/dp/83AH9507>. [Accessed: 20-Dec-2022].
- [21] “HFCT 100 HCP - HVPD.” [Online]. Available: <https://www.hvpd.co.uk/hfct/>. [Accessed: 13-Oct-2021].
- [22] “BNC Extension Cable- YOTENKO.” [Online]. Available: <https://www.amazon.ca/s?k=YOTENKO>. [Accessed: 20-Dec-2022].
- [23] “MEAN WELL RD-50A - MEAN WELL.” [Online]. Available: <https://www.mouser.ca/ProductDetail/MEAN-WELL/RD-50A?qs=V9a8iPeg90x33CByXejgiw%3D%3D>. [Accessed: 20-Dec-2022].
- [24] “Jetson TX2 Development Kit- NVIDIA.” [Online]. Available: <https://www.amazon.ca/NVIDIA-945-82771-0000-000-Jetson-TX2-Development/dp/B06XPFH939?th=1>. [Accessed: 20-Dec-2022].
- [25] “10.1inch Capacitive Touch Screen LCD (SKU: 18096)- Waveshare.” [Online]. Available: <https://www.waveshare.com/10.1inch-hdmi-lcd-e.htm>. [Accessed: 20-Dec-2022].

- [26] “WD Blue 500GB Internal Solid State Drive (WDBB8H5000ANC-WRSN)- Western Digital.” [Online]. Available: <https://www.bestbuy.ca/en-ca/product/wd-blue-500gb-internal-solid-state-drive-wdbb8h5000anc-wrsn/16366472>. [Accessed: 20-Dec-2022].
- [27] “Male to Male USB Cable (80790)- UGREEN.” [Online]. Available: https://www.amazon.ca/UGREEN-Transfer-Braided-Enclosure-Printer/dp/B089FB2G8J?ref_=ast_sto_dp&th=1&psc=1. [Accessed: 20-Dec-2022].
- [28] “Male to male HDMI Cable (B0B6RDSQGS)- Twozoh.” [Online]. Available: <https://www.amazon.com/s?k=Twozoh>. [Accessed: 20-Dec-2022].
- [29] “22-pin (7+15) Sata Male to Female Data and Power Combo Extension Cable (FBA_18501) - SMAKN.” [Online]. Available: https://www.amazon.ca/dp/B00L9R3AKA/ref=as_li_ss_tl?psc=1&ie=UTF8&linkCode=gg2&linkId=795d712db669e1ef9909e93c86f08093&tag=jetsonhacks-20. [Accessed: 20-Dec-2022].
- [30] “Jetson TX2- Nvidia.” [Online]. Available: <https://www.nvidia.com/en-us/autonomous-machines/embedded-systems/jetson-tx2/>.
- [31] “Socket Programming,” www.cs.dartmouth.edu. <https://www.cs.dartmouth.edu/~campbell/cs60/socketprogramming.html> [Accessed: 20-Dec-2022].
- [32] “Socket-based Communication,” sites.ualberta.ca. https://sites.ualberta.ca/~smartynk/Resources/CMPUT%20379/lab%20notes/Network_Programming_and_the_Sockets_API.pdf [Accessed: 20-Dec-2022].
- [33] “Socket programming,” www.ibm.com. <https://www.ibm.com/docs/en/i/7.1?topic=communications-socket-programming> [Accessed: 20-Dec-2022].
- [34] “What Is a Socket?,” docs.oracle.com. <https://docs.oracle.com/javase/tutorial/networking/sockets/definition.html> [Accessed: 20-Dec-2022].

- [35] “What Is SQLite?,” www.sqlite.org. <https://www.sqlite.org/index.html> [Accessed: 20-Dec-2022].
- [36] “sqlite3 — DB-API 2.0 interface for SQLite databases,” [docs.python.org](https://docs.python.org/3/library/sqlite3.html). <https://docs.python.org/3/library/sqlite3.html> [Accessed: 20-Dec-2022].
- [37] J. C. C. Vieira, D. F. Júnior, W. A. Neves and F. V. Lopes, "Challenges in Application of the Traveling Wave-Based Fault Location Methods Applied to HVDC Systems: Evaluation of Classical One- and Two-Terminal Methods," 2022 20th International Conference on Harmonics & Quality of Power (ICHQP), 2022, pp. 1-6, doi: 10.1109/ICHQP53011.2022.9808569.
- [38] H. Jia, “An Improved Traveling-Wave-Based Fault Location Method with Compensating the Dispersion Effect of Traveling Wave in Wavelet Domain,” *Mathematical Problems in Engineering*, vol. 2017, Feb. 2017, pp. 1-11, doi: 10.1155/2017/1019591.
- [39] V. Lopes and E. J. S. Leite, “Traveling wave-based solutions for transmission line two-terminal data time synchronization,” *IEEE Transactions on Power Delivery*, vol. 33, no. 6, pp. 3240–3241, Dec. 2018, doi: 10.1109/TPWRD.2018.2831458.
- [40] S. Wang and G. Xu, “Research on traveling wave fault technology based on ground potential,” *AIP Advances*, vol. 11, no. 11, Nov. 2021, doi: 10.1063/5.0069963.
- [41] B. Li, H. Li, X. Zeng, W. He and X. Zeng, "Novel time synchronization method based on duo-satellite triggering for double-ended PD location in MV cables," 2017 1st International Conference on Electrical Materials and Power Equipment (ICEMPE), 2017, pp. 436-439, doi: 10.1109/ICEMPE.2017.7982121.
- [42] G. Ma, L. Jiang, K. Zhou, and G. Xu, “A Method of Line Fault Location Based on Traveling Wave Theory,” *International Journal of Control and Automation*, vol. 9, no. 2, pp. 261–270, Feb. 2016, doi: 10.14257/ijca.2016.9.2.25.

- [43] W. Yao et al., "Impact of GPS signal loss and its mitigation in power system synchronized measurement devices," 2017 IEEE Power & Energy Society General Meeting, 2017, pp. 1-1, doi: 10.1109/PESGM.2017.8274578.
- [44] M. M. Saha, J. Izykowski, and E. Rosolowski, *Fault Location on Power Networks*. Berlin, Germany: Springer, ISBN: 978-1-84882-885-8, 2010.
- [45] F. V. Lopes, "Settings-free traveling-wave-based earth fault location using unsynchronized two-terminal data," *IEEE Transactions on Power Delivery*, vol. 31, no. 5, pp. 2296-2298, 2016.
- [46] F. Lopes, K. Silva, F. Costa, W. Neves, and D. Fernandes, "Realtime traveling-wave-based fault location using two-terminal unsynchronized data," *IEEE Transactions on Power Delivery*, vol. 30, no. 3, pp. 1067-1076, 2014.
- [47] S. M. Hashemian, S. N. Hashemian, and M. Gholipour, "Unsynchronized parameter free fault location scheme for hybrid transmission line," *Electric Power Systems Research*, vol. 192, p. 106982, 2021.
- [48] T. Hensler, C. Pritchard, N. Fischer, and B. Kasztenny, "Testing Superimposed Quantities and Traveling-Wave based Protection," in *Locating Faults and Protecting Lines at the Speed of Light: Time-Domain Principles Applied, Vol.1*, E. Schweitzer, B. Kasztenny, A. Guzman, V. Mynam, H. Altuve, Ed. Washington: Schweitzer Engineering Laboratories, Inc., 2018, pp. 211-222.
- [49] M. Gilany, D. k. Ibrahim and E. S. Tag Eldin, "Traveling-Wave-Based Fault-Location Scheme for Multiend-Aged Underground Cable System," in *IEEE Transactions on Power Delivery*, vol. 22, no. 1, pp. 82-89, Jan. 2007, doi: 10.1109/TPWRD.2006.881439.

- [50] "GPS NTP Network Time Server (TM1000A)- Time Machines." [Online]. Available: <https://timemachinescorp.com/product/gps-time-server-tm1000a/>. [Accessed: 20-Dec-2022].
- [51] "GPS Receiver Module (I-87211W-G)- ICP DAS." [Online]. Available: https://www.logicbus.com/I-87211W-G_p_2122.html. [Accessed: 20-Dec-2022].
- [52] "TIMENET Pro Master NTP Time Server with Antenna- Veracity." [Online]. Available: <https://www.veracityglobal.com/products/networked-video-integration-devices/timenet-pro.aspx>. [Accessed: 20-Dec-2022].
- [53] M. Lepot, Aubin, J.B. Aubin and F.H.L.R. Clemens, "Interpolation in Time Series: An Introductory Overview of Existing Methods, " Their Performance Criteria and Uncertainty Assessment, *Water*, vol. 9, no. 10:796, 2017, doi: 10.3390/w9100796.
- [54] S.R. Brubacher and G. T. Wilson, "Interpolating time series with application to the estimation of holiday effects on electricity demand, " *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, vol. 25, no. 2, pp. 107-116, 1976, doi: 10.2307/2346678.
- [55] E. Carrizosa, N.V. Olivares-Nadal and P. Ramirez-Cobo, "Times series interpolation via global optimization of moments fitting," *European Journal of Operational Research*, Elsevier, vol. 230(1), pp. 97-112, 2013, doi: 10.1016/j.ejor.2013.04.008.
- [56] R. Sibson, "A brief description of natural neighbor interpolation," *Interpreting Multivariate Data*, John Wiley & Sons, New York, pp. 21-36, 1981.
- [57] T. Athawale and A. Entezari, "Uncertainty Quantification in Linear Interpolation for Isosurface Extraction," in *IEEE Transactions on Visualization and Computer Graphics*, vol. 19, no. 12, pp. 2723-2732, Dec. 2013, doi: 10.1109/TVCG.2013.208.

- [58] A. Gnauck, "Interpolation and approximation of water quality time series and process identification," *Analytical and Bioanalytical Chemistry*, vol.380, pp.484–492, Oct 2004, doi: 10.1007/s00216-004-2799-3.
- [59] J.P. Musial, M.M. Verstraete and N. Gobron, " Technical Note: Comparing the effectiveness of recent algorithms to fill and smooth incomplete and noisy time series, " *Atmospheric Chemistry and Physics*, vol. 11, pp. 7905–7923, 2011, doi: 10.5194/acp-11-7905-2011.
- [60] Y. Chen, G.A. Kopp and D. Surry, " Interpolation of wind-induced pressure time series with an artificial network," *Journal of Wind Engineering and Industrial Aerodynamics*, vol. 90, pp. 589-615, 2002, doi:10.1016/S0167-6105(02)00155-1.
- [61] J. Delua, "Supervised vs. Unsupervised Learning: What’s the Difference?" [www.ibm.com. https://www.ibm.com/cloud/blog/supervised-vs-unsupervised-learning](https://www.ibm.com/cloud/blog/supervised-vs-unsupervised-learning) [Accessed: 21-Nov-2022].
- [62] “Category Encoding layer,” [keras.io. https://keras.io/api/layers/preprocessing_layers/categorical/category_encoding/](https://keras.io/api/layers/preprocessing_layers/categorical/category_encoding/) [Accessed: 21-Nov-2022].
- [63] “Normalization,” [developers.google.com. https://developers.google.com/machine-learning/data-prep/transform/normalization](https://developers.google.com/machine-learning/data-prep/transform/normalization) [Accessed: 21-Nov-2022].
- [64] “dense layer,” [developers.google.com. https://developers.google.com/machine-learning/glossary#dense-layer](https://developers.google.com/machine-learning/glossary#dense-layer) [Accessed: 21-Nov-2022].
- [65] “Rectified Linear Unit (ReLU),” [developers.google.com. https://developers.google.com/machine-learning/glossary#rectified-linear-unit-relu](https://developers.google.com/machine-learning/glossary#rectified-linear-unit-relu) [Accessed: 21-Nov-2022].

- [66] “Identify regression machine learning scenarios,” learn.microsoft.com. <https://learn.microsoft.com/en-us/training/modules/create-regression-model-azure-machine-learning-designer/2-regression-scenarios> [Accessed: 21-Nov-2022].
- [67] “Automation Library,” /www.pscad.com. <https://www.pscad.com/software/pscad/automation-library> [Accessed: 21-Nov-2022].
- [68] “Transient Earth Voltage (TEV) Sensor- HVPD.” [Online]. Available: <https://www.hvpcd.co.uk/tev/> [Accessed: 20-Dec-2021].
- [69] “High Voltage Coupling Capacitor (HVCC) - HVPD.” [Online]. Available: <https://www.hvpcd.co.uk/hvcc/> [Accessed: 20-Dec-2021].
- [70] “Airborne Acoustic (AA)- HVPD.” [Online]. Available: <https://www.hvpcd.co.uk/aa/> [Accessed: 20-Dec-2021].
- [71] “Bushing Tap Sensor (BTS)- HVPD.” [Online]. Available: <https://www.hvpcd.co.uk/bts/> [Accessed: 20-Dec-2021].

Appendix A

Publication:

- [1] M. Salauddin, T. Lan, C. Y. Chung, S. -B. Ko and S. M. Mazhari, "A Novel Zero-Crossing Point Calibration-Based Data Synchronization Approach for an Underground Cable Fault Localization Platform," *2022 IEEE Electrical Power and Energy Conference (EPEC)*, Victoria, BC, Canada, 2022, pp. 137-142, doi: 10.1109/EPEC56903.2022.10000214.

Appendix B

Copyright Reference:

- [1] © [2022] IEEE. Reprinted, with permission from [M. Salauddin, T. Lan, C. Y. Chung, S. -B. Ko and S. M. Mazhari, "A Novel Zero-Crossing Point Calibration-Based Data Synchronization Approach for an Underground Cable Fault Localization Platform," *2022 IEEE Electrical Power and Energy Conference (EPEC)*, Victoria, BC, Canada, 2022, pp. 137-142, doi: 10.1109/EPEC56903.2022.10000214].