Jordan Journal of Social Sciences, Volume **17**, No. **2**, **2024**

Development of a GIS Tool to Find Fiber Cable Fault Using Python Scripting for ArcGIS in Amman City

Madeleine Taiseer Shehadeh[1], Mohammed Jamil Al-Qaralleh [2]

ABSTRACT

Optical fiber cables are characterized by a larger bandwidth than other transmission media, which increases the amount of data transmitted per unit of time. Managing a fiber optic network using Geographic Information Systems (GIS) has benefits for the operation and maintenance of the communication network. The study aims to develop a GIS tool to detect fiber optic cable faults using Python Scripting for ArcGIS in the city of Amman. To achieve this goal, the Esri data model for telecommunications was used. The study concluded that it is possible to locate faults using the Python programming language, and the program provides accurate coordinates based on the precision of the OTDR device.

Keywords: Fiber, Python, Scripting, Accuracy, ArcGIS, Esri telecommunication template.

INTRODUCTION

We are experiencing an incredible evolution in digital information. Fiber networks are the infrastructure that transports all types of data, including voice, video, and text, from point to point. Fiber is becoming increasingly popular due to its high bandwidth capabilities. Therefore, using GIS for managing a fiber network can consolidate all this data into a relational location database.

GIS capabilities are becoming increasingly important in the construction of fiber networks. Not only are visual representations of fiber optic networks required, but also tools that aid in defining network-related questions, such as where the cable is located and which fibers inside those cables have services. Is there any spare inner duct in the conduit? How many free fiber strands are there? The only way to determine the fiber cable fault location is to use optical time-domain reflectometer (OTDR) test equipment to measure the distance of the fault location from one side of the cable, usually from the optical distribution frame (ODF) inside the data center, and then use GIS drawings to manually calculate the distance by adding fiber lengths until it equals the OTDR reading. However, this method often takes more time and is inaccurate.

Therefore, using a Geographic Information System (GIS) can provide a solution by developing a Python tool, which is the focus of this study, to determine the cable fault distance from an optical time-domain reflectometer (OTDR). This approach will give a precise location to assist maintenance engineers in determining cable faults quickly and effectively. The process is accomplished by following the steps below:

- Using the trace capabilities provided by the "Esri Telecommunication Template."
- Using a Python Add-in to create a toolbar, which is a container for the buttons.
- Using Python scripting for ArcGIS to dissolve the selected fiber cable features after tracing the defined strands into one feature, split it into two lines at the desired distance, and get the coordinates of the fault

[1] Ministry of Digital Economy and Entrepreneurship, Jordan. madeleine.shehadeh@modee.gov.jo

[2] Department of Geography, Faculty of Arts, Mu'tah University, mqaralleh@mutah.edu.jo

Received on 12/6/2023 and Accepted for Publication on 22/2/2024.

position in both the JTM and WGS84 coordinate systems.

Anyone who plans, constructs, or operates a fiber-optic network system will benefit from this study. The time factor is essential in managing the operation of a fiber network. Determining the location of a fiber fault quickly and conveniently will help maintenance engineers address fiber cuts and restore service to customers promptly.

-Study Area

Amman, the capital and largest city of Jordan, is also the country's economic, political, and cultural heart. It is located between longitudes E 35.6660433 and E 37.2479238 and latitudes N 31.2663185 and N 32.1132768, as shown in Figure 1. In 2019, Amman had a population of 4,430,700 people and covered an area of 7,579 km² (Department of Statistics, 2019).

The National Broadband Network (NBN), owned by the Ministry of Digital Economy and Entrepreneurship (MoDEE), is an optical fiber network that this study focuses on. Its goal is to connect all public schools, government agencies, healthcare organizations, community colleges, and knowledge stations. To reach the connection points, the network is designed using point-to-point Gigabit Ethernet (GE) through fiber optic technology via aggregation nodes. The network is divided into three phases: the middle phase, the north phase, and the south phase. These phases are linked by an Overlay Network, which provides physical communication between the Points of Presence (PoPs) for each module and the National Data Centre via 10GE connections on the Ethernet Switch using single-mode optical fiber cables.

The Am Katheer Aggregate in Amman, part of the middle phase, serves nineteen schools, three governmental entities, and two health entities. It will be used as a pilot area, as shown in Figure 1.

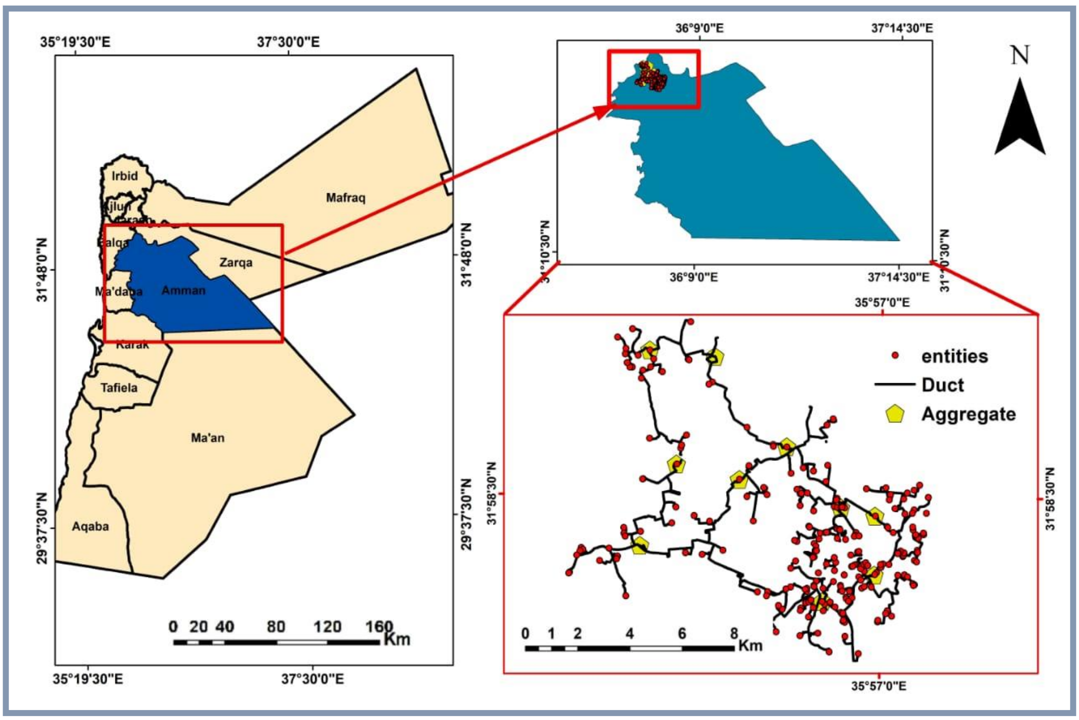


Figure (1) Location of Study Area. Source: Researchers Work & (MODEE)

- Previous studies

- Halaseh, Jumana Jaleel (2016). “The investigation of optical soliton in optical fiber.” This thesis demonstrates how optimal handling of both GVD (Group Velocity Dispersion) and SPM (Self Phase Modulation) results in optical solitons, which are very short pulses that can transmit without alteration over long distances while maintaining their form. The results show how linear (GVD) and nonlinear (SPM) factors can alter the shape of a pulse. It is also demonstrated that the best scenario for soliton transmission may be achieved by meeting specific soliton requirements, such as soliton peak power, soliton order, and attenuation coefficient.
- Skendzic, Sandra (2015). “ASIC-enabled High Resolution Optical Time Domain Reflectometer.” This thesis reports on a high-resolution OTDR that utilizes a compact and programmable ASIC. The ASIC functions as a tunable clock, pattern generator, precise timer, electrical receiver, and signal sampling circuit. Its features can be optimized for OTDR performance. In this thesis, the ASIC-based OTDR is designed for single-mode fiber testing. The theory of operation of the ASIC is explained, and its programmable settings are explored. The implemented OTDR successfully detects backscattered light pulses in a fiber with a PIN photodiode and a 1 cm sampling resolution. The bandwidth of the receiver is shown to be a limiting factor in the OTDR's performance, which may be attributed to stray parasitics, such as those associated with the photodiode packaging.
- Zahraa M. Matrood and Others (2014). “A Simple GIS-Based Method for Designing Fiber Networks.” This paper presents a new, straightforward strategy for optimizing the deployment of GIS-based network design tools. In this study, a prototype system was created for automatically planning a Fiber to the Home (FTTH) network based on the geographic data of the area. First, the region's geographic urban information is acquired, and common inaccuracies are resolved. The necessary GIS tables are then created, which include street maps, building maps, and housing maps. The network configuration is designed using either a star or a bus topology. The network design process involves determining the distribution of network nodes and the optimal routes for cables connecting the nodes to achieve comprehensive coverage of the entire region at the lowest cost.
- Nixon Lemlem (2012). “GIS in the Management of Fiber Optic Cable Network in Urban Centers: Case Study of Frontier Optical Networks Limited.” This study examines the application of Geographic Information Systems in the management of Frontier Optical Network (FON) Limited’s fiber optic cable distribution network in Nairobi, Kenya. A spatial database was constructed to store information about the fiber optic cable network. Fiber tracing was performed using network analysis to show the pathways of data communication flow between a transmitting and receiving station.
- Hu Chaoju and Sheng Rui (2009). “The Fault Location of Optical Cable for Communications Based on GIS.” This research designed an algorithm to pinpoint the location of fiber cable faults using GIS based on the OTDR (Optical Time Domain Reflectometer) working principle. The algorithm starts by obtaining the distance from the OTDR, then finds the record from the geodatabase that corresponds to this distance. After that, the attribute information of the fault spots is identified.
- Sheng Rui and Jianhua Tan (2008). “Communication Cable Fault Location in Power System.” This paper describes the design of a GIS system that combines the characteristics of fiber optic communications. Based on this design, an effective algorithm was developed for locating faults in optical cables. Once the fault location on the fiber optic cable is identified, the attribute information in the GIS system is researched to find the surrounding environment of the fault spot, assign the specific location of the fault, and carry out emergency repairs.

• Nashed, Ahmad Ibrahim AbdAlfattah (2006). "Optical Bragg Fiber Design." This thesis uses the FDFD method to study Bragg fibers produced by a core coated with multiple sheaths of high and low refractive index layers. The three and four cladding sheaths of Bragg fibers were mainly studied. A neural network approach was then used to design low-dispersion fiber. A fiber fault investigation is also conducted.

• GuoRay Cai (2002). "A GIS Approach to the Spatial Assessments of Telecommunication Infrastructure." The telecommunications infrastructure has grown to be the region's most comprehensive spatial structure. An examination of the telecommunications infrastructure in the United States exemplifies this process. This study proposes a GIS approach for analyzing the adequacy of regional broadband infrastructure by spatially linking infrastructure facilities with demand. In the telecom infrastructure and demand overlay analysis, the paper highlights the inherent difficulties of dealing with conceptual and geometric incompatibilities. It proposes a new method for achieving meaningful data integration based on an extension of Chrisman's transformational approach.

Theoretical framework

-Telecommunication

People used fire, smoke, and light to communicate in the early days. Telecommunication refers to the process of exchanging speech, data, and video transmissions via electronic technologies such as telephones (wired and wireless), microwave communications, fiber optics, satellites, radio and television broadcasts, and the internet. Optical fiber technology is one of the fastest types; it sends and receives data-encoded messages at speeds of up to 300,000 km/sec using light generated from an electrical signal. However, in practice, light travels slower through optical fiber cables than it does in a vacuum.

The three essential components of optical fiber communication are the light signal transmitter, the optical fiber cable, and the photo-detecting receiver. Additional parts, such as cable splicers, connectors, regenerators, beam splitters, and optical amplifiers, are employed to improve the communication system's performance (see Figure 2).

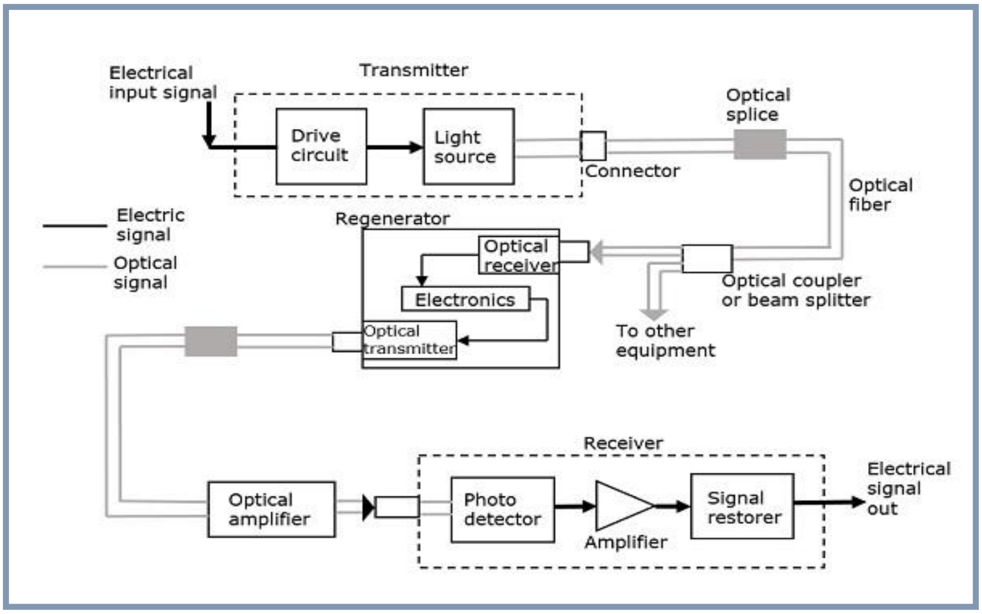


Figure (2) Optical fiber communication components. Source: (www.tutorialspoint.com)

-Laser

Lasers, or "Light Amplification by Stimulated Emission of Radiation," are the best light source for optical fiber communication. A process known as stimulated emission amplifies light energy to exceptionally high intensity. The laser generates energy in or near the visible light range of the electromagnetic spectrum (Sean P. O'Duill and Others, 2020).

- Electromagnetic spectrum

Scientists use the phrase *electromagnetic spectrum* to define the complete range of light that exists, from radio waves to gamma rays. Light is a wave that consists of alternating electric and magnetic fields that are perpendicular to one another. Like every other wave, light has a few essential qualities that define it. One is its frequency, measured in hertz (Hz), which counts how many waves pass by a point in a second. Another related property is the wavelength, which is the distance between the peaks of two waves. These two properties are inversely related: as the frequency increases, the wavelength decreases, and vice versa (*EarthSky*).

-Python

Python is an open-source scripting language. It was introduced to ArcGIS in version 9. The objective of this study is to write Python scripts to carry out tasks in ArcGIS. Python is used as an interpreted language to work directly with existing functions available in ArcGIS.

Comparing scripting vs programming

A scripting language refers to automating certain functionality within another program, while a programming language involves developing more sophisticated, multifunctional applications. Scripting is a programming task that allows the connection of diverse existing components to accomplish a new, related task. Scripting acts as the "glue" that helps put various existing elements together. Programming, on the other hand, allows the building of components from scratch (Zandbergen, Paul A. *Python Scripting for ArcGIS*. Esri Press, New York, 2013, p. 4).

-Python Add-in

For customization, we can use an Add-in that builds a toolbar containing a collection of tools that plug into an ArcGIS Desktop application (such as ArcMap, ArcCatalog, ArcGlobe, and ArcScene) to provide supplemental functionality for accomplishing custom tasks (see Figure 3; ArcGIS Help).

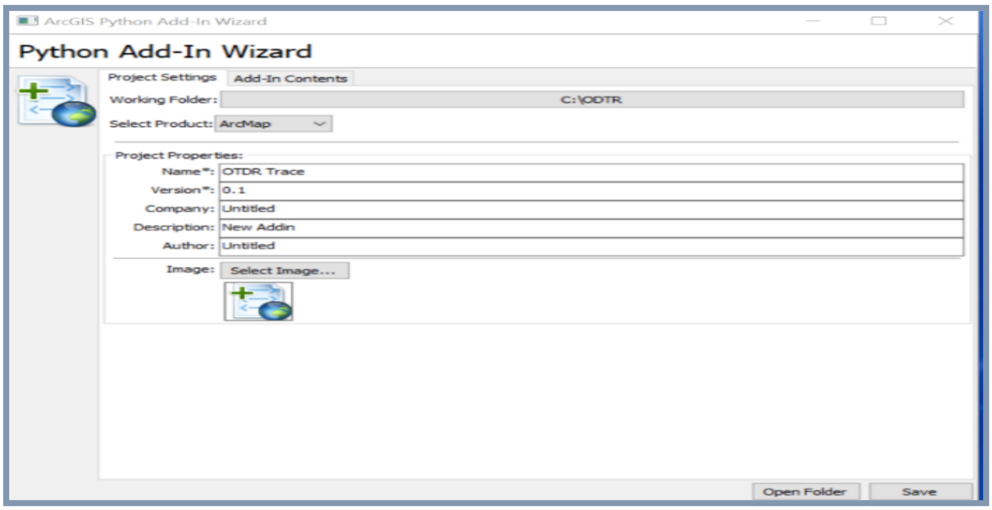


Figure (3) Python Add-In Wizard. Source: Researchers Work & Python

The Add-in toolbar contains buttons, combo boxes, tools, menus, and tool palettes. In desktop applications, toolbars can be docked or floating and can be set to activate automatically when the application starts. To create a new toolbar, right-click on "Toolbar" in the Add-in Contents tab and select "New Toolbar." The properties of the toolbar will be displayed in the right panel of the wizard. The toolbar has an ID and a caption; the label of the toolbar is entered in the caption field. The unique name of the ID is used to distinguish between different toolbars and must not contain spaces. To ensure the toolbar appears when the desktop application starts, the option "Show Initially" must be checked. If this box is unchecked, the toolbar will be hidden when the application starts (see Figure 4; desktop.arcgis.com).

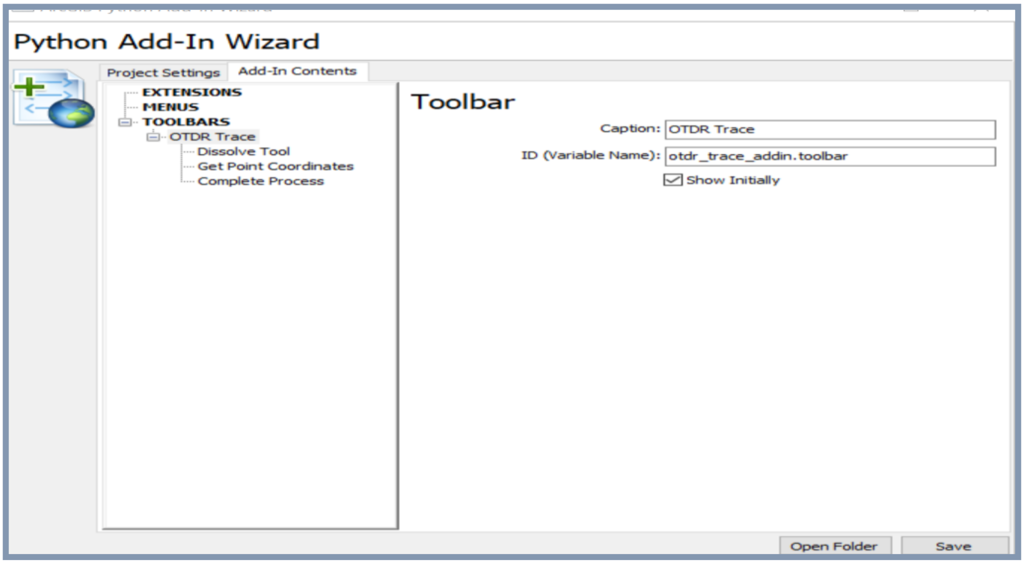


Figure (4) Python Add-In Toolbar. Source: The Add-in contents tab

-Esri data model for telecommunication

The Esri data model for telecommunications is an example of an ArcGIS Desktop configuration for the telecom industry that allows users to edit and update fiber network information stored in a geodatabase (www.esri.com). The geodatabase consists of a feature dataset called Telecom Dataset, which includes two geometric networks: ConduitNetwork (which contains Conduit, Structures, and Poles) and TelecomNetwork (which includes Fiber Cable, PatchPanel (devices), and Splice Closure). Additionally, the model includes relationship classes that preserve connectivity between elements.

-Fiber Network components and rules

1. The Duct and Inner Duct configurations of Conduit features can be specified once the Conduit has been created. Ducts and Inner Ducts can later be related to Fiber Cables to model how the cable is contained within the Conduit network.
2. Once all of the Ducts have been created for the selected Conduit, any associated Inner Ducts can be created.
3. After the Conduit network elements have been created, the related fiber telecommunications network features (such as Fiber Cables, devices,

and Splice Closures) can be added. In addition to creating spatial Fiber Cable, Splice Closure, and device features on the map, spatial records for individual fiber strands, device ports, and fiber splices will generally be created at this time. Their interconnectivity and relationships will also be specified and established.

4. Fiber Cables are digitized in a downstream direction from the source (capacity-supplying) equipment towards the elements being fed. Additionally, the mainline Fiber Cables should be created before branching elements.
5. Once all the Fiber Cables have been created, they should be associated with the existing Conduits, Ducts, or Inner Ducts that they pass through. Each Fiber Cable should be associated only with the lowest level element it passes through. In other words, if it passes through an Inner Duct, it should only be associated with that Inner Duct and not with the Duct or Conduit containing the Inner Duct.
6. Create a Splice Closure and place it at the end of the Fiber Cable.
7. Place a device at the appropriate locations at the ends of the Fiber Cables.

-Adding new Ducts and Inner Ducts to a Duct Bank (Conduit)

Below are the steps to add new ducts and inner ducts to the Conduit feature. First, select the desired duct bank (Conduit).

1. Open the attributes using the "Show Attributes" tool.
2. Expand the duct bank tree and find the duct child within the expanded tree.
3. Right-click and choose "Add New Duct" to this duct bank. Repeat this process until you have added all the ducts needed for this duct bank.
4. To add an inner duct to one of the previously added ducts, select the desired duct from the duct tree view. Find the inner duct child within the expanded tree, right-click, and choose "Add New" to add a new inner duct to the selected duct. Repeat this process until you have added all the inner ducts required.

-Association between Conduit Network and Fiber Network

1. To associate Conduit network components with fiber network components, an association between their components must be created. The types of associations are as follows:
2. **Conduit and Fiber Cable**: ConduitHasFiberCable
3. **Duct and Fiber Cable**: DuctHasFiberCable
4. **Inner Duct and Fiber Cable**: InnerDuctHasFiberCable

If ducts have inner ducts, then the association should be made between the Fiber Cable and the inner ducts only. If not, the association should be made between the Fiber Cable and the duct.

To create an association, use the attributes tree view by following these steps:

1. Select the Fiber Cable and the Duct Bank (Conduit) to create an association between them.
2. Expand the tree of the selected Duct Bank (Conduit).
3. Find the Inner Duct child from the expanded tree. Then, expand the tree of the desired Inner Duct, select the Fiber Cable, right-click, and choose "Add Selected."

-Fiber Network Editing Template Tools

1. IMPLEMENTATION STEPS

The following implementation steps were executed to configure the Fiber Network Editing Template:

1. **Install the Fiber Network Editing Toolbar Add-in** on the ArcMap application. To do this, double-click on "Esri Telco Tools." This will create a toolbar called "Esri Fiber Editing"

within the ArcGIS Desktop environment.

2. **Open the map document** "Fiber Network Data Management.mxd."
3. **Turn on the Esri Fiber Editing Toolbar** by navigating to **View** > **Toolbars** > **Esri Fiber Editing**.

2. TOOLS OVERVIEW

The tools include network devices, Fiber Cables, Splice Closures, Conduit, and Structures, as well as specialized functions for updating and maintaining fiber telecommunications and Conduit network map features. They also facilitate the configuration, maintenance, and tracing of lower-level interconnections and relationships between individual ports, fiber splices, fiber strands, buffer tubes, and ductwork.

3. COMMANDS AVAILABLE IN ESRI FIBER EEDITING TOOLS TOOLBAR

Once the tools are installed, add the Telecom Editing Tools toolbar to ArcMap. The tools available on the toolbar are shown in Figure 5.

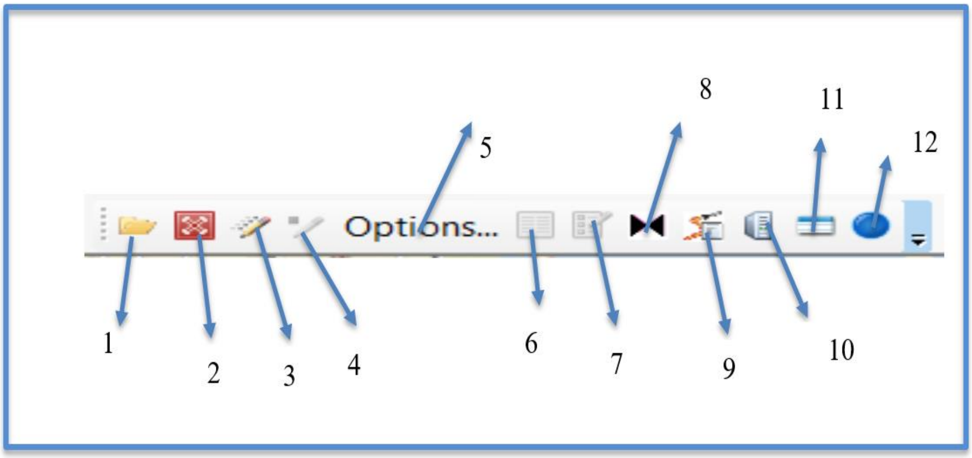


Figure (5) Esri fiber Editing Tools toolbar. Source: Esri Template Bar

1. **Open Workspace**: Select a telecom workspace to edit.
2. **Close Workspace**: Closes the current telecom workspace and ends any editing sessions.
3. **Start Editing**: Start an edit session so you can modify features and attributes.
4. **Stop Editing**: Ends the edit session. If there are any unsaved edits, you will be prompted to save them.
5. **Editing Options**: Opens the Editing Options dialog box, where you can specify various options related to editing.
6. **Attributes**: Opens the Attributes window, where you can modify the attribute values of selected features in the layers being edited. You can also edit relationships among features in this window.
7. **Create Features**: Opens the Create Features window, allowing you to add new features.
8. **View or Edit Fiber Splice**: Define the fiber strand ranges for Fiber Cables connected to a selected Splice Closure and specify the type of Splice Closure and its loss value.
9. **Run a Fiber Network Trace**: Provides low-level tracing functionality at the individual device ports or fiber strands level, based on the interconnectivity configured with the other

Telecom Editing tools.

10. **View or Edit Fiber Device Connection**: Specify the from and to ranges for Fiber Cables connected to equipment (e.g., Cross Connects).
11. **Telecom Tools Log**: Access the log for telecom tools.
12. **Run DB Integrity Check**: Execute a database integrity check.

-Editing and Tracing Network Data

Starting to Use the Fiber Network Editing Toolbar

1. Open the "Fiber Network Data Management.mxd" file from the startup dialog or by selecting File > Open from the menu.
2. In ArcMap, click the Customize menu, point to Toolbars, and select the Esri Fiber Editing option. To start editing and placing features, open a workspace using the Open Workspace command. Choose the desired workspace and click OK (see Figure 6).
3. Start editing by click on start editing command
4. Click on create features window, to start features editing and features placement, and to create features, Figure (7).
5. Set splicing relations between Fiber Cables and splice closures network.

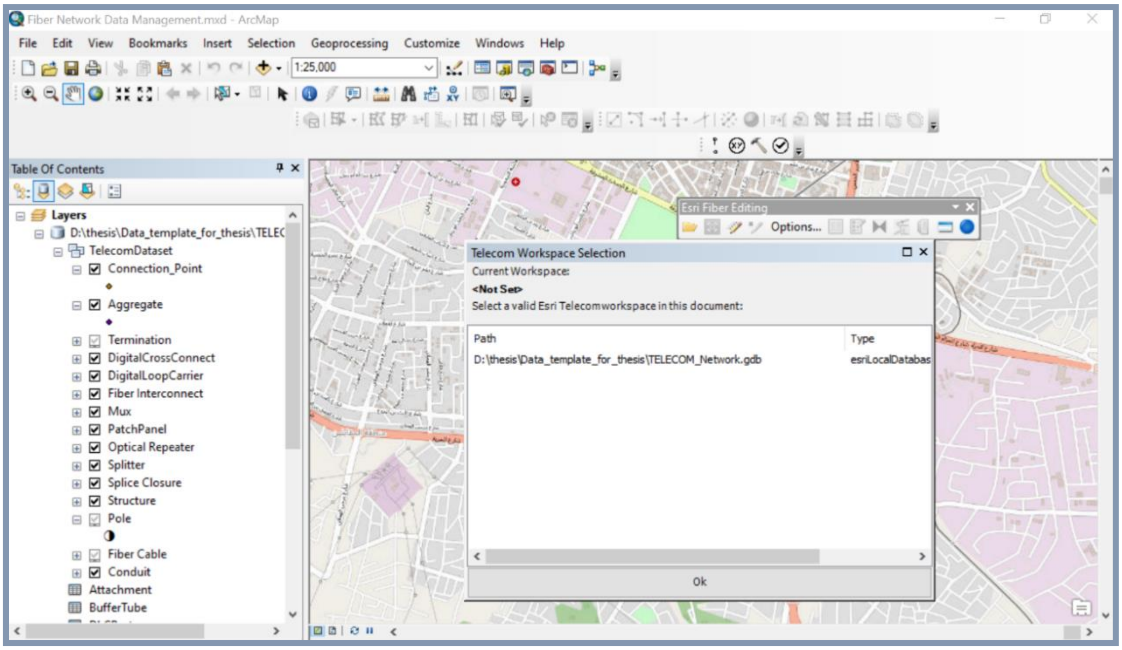


Figure (6) Workspace Open Window. Source: ArcMap

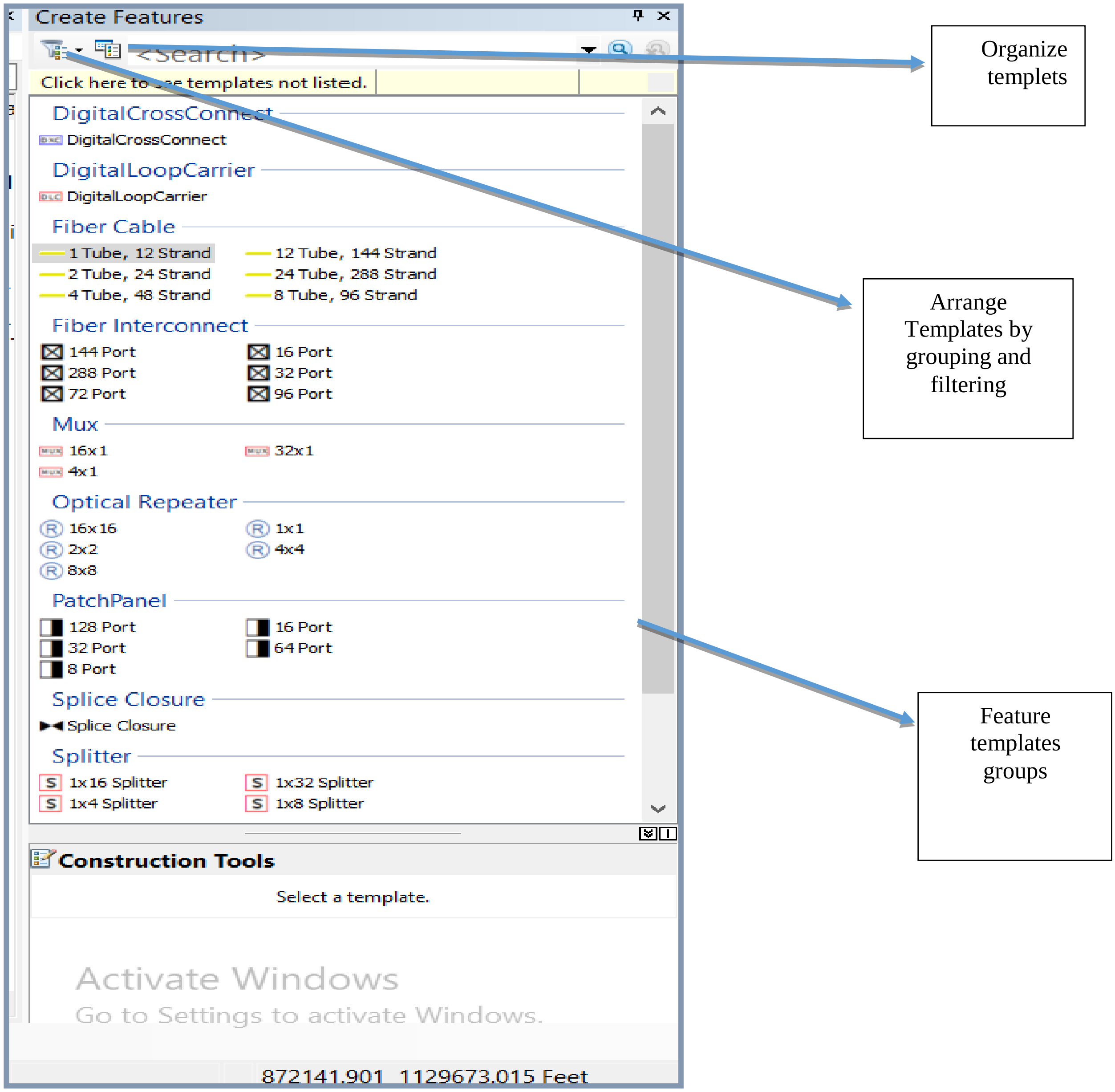


Figure (7) Create Features Window. Source: Researchers Work & Esri fiber editing toolbar

Click on Edit or view fiber splice command  to open Edit or view fiber splice window from fiber management toolbar, Figure (8).

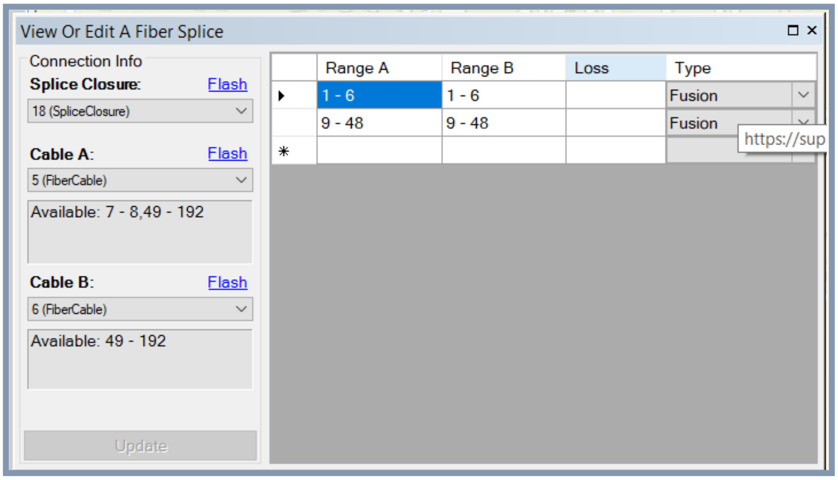


Figure (8) View or Edit a Fiber Splice. Source: View or Edit a Fiber Splice in Esri Fiber Editing toolbar

I. The dialog displays the selected Splice Closure in the Splice Closure combo box drop-down control. The Cable A and Cable B controls list the cables connected (snapped) to the selected Splice Closure. Select the correct features for the Splice Closure, Cable A, and Cable B controls from the available options. Cable A should be the cable upstream from the splice. If necessary, use the "Flash" links to highlight the selection in the map view for confirmation. If there are no available pair ranges, the dialog will state, "No available ranges!" An error message will appear if an update is attempted with unavailable ranges.

II. Define the Range A, Range B, Loss, and Type values for each Fiber Splice within the Splice Closure, and then set these values by clicking the **Update** button.

III. When finished creating Splice Closures and editing their fiber splice connectivity, save the edits by selecting **Editor > Save Edits** from the Editor Toolbar.

6. Set device connections between Fiber Cables and devices (ODFs)

 Clicking View or edit Fiber device connection to open View or edit Fiber device connection window from fiber management toolbar, Figure (9).

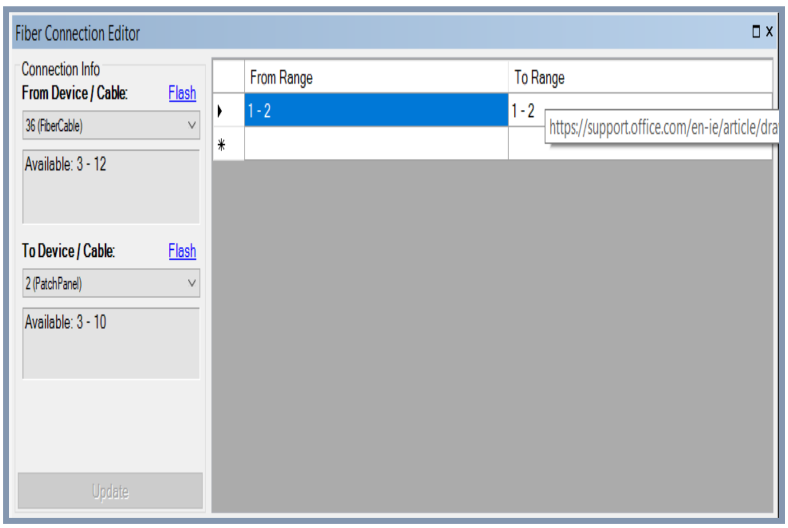


Figure (9) View or Edit Fiber Device Connections. Source: Fiber Connection Editor in Esri Fiber Editing toolbar

I. The Connections Editor dialog will be displayed. This dialog lists the new device and any cables connected to it in the From Device/Cable and To Device/Cable combo box drop-down controls. Select the appropriate device and cable from the available options. If necessary, click the Flash links to highlight the selection in the map view for confirmation.

II. Define the From Range and To Range values for the ports of the device that the Fiber Cable strands connect to, and then set these values by clicking the Update button. If there are no available pair ranges, the dialog will state, "No available ranges," and an error message will appear if an update is attempted with unavailable ranges.

III. Save the edits by selecting Editor > Save Edits from the Editor Toolbar.

8. Tracing the Telecom Network

I. The Esri Fiber Editing toolbar provides the ability to trace a specified port or fiber strand from a selected device or Fiber Cable. This analysis relies on a combination of the Telecom Network geometric network connectivity for the spatial map features and the connectivity specified with the Splice Editor and Connection Editor tools (see Figure 10). The process is as follows:

II. Zoom in to an appropriate map scale.

III. The Telecom Trace tool operates on a selected device or Fiber Cable feature. Therefore, the Fiber Cable and/or the device(s) containing the fiber or port to be traced must be selectable. If necessary, add these layers to the list of selectable layers.

IV. Use the Select Features tool from the Standard toolbar to select the Fiber Cable or device to be traced in the map.

V. Open the Telecom Trace tool by clicking the Telecom Trace button on the Telecom Editing Tools toolbar.

VI. The dialog lists the selected devices or Fiber Cables in the Existing Cables/Devices combo box drop-down.

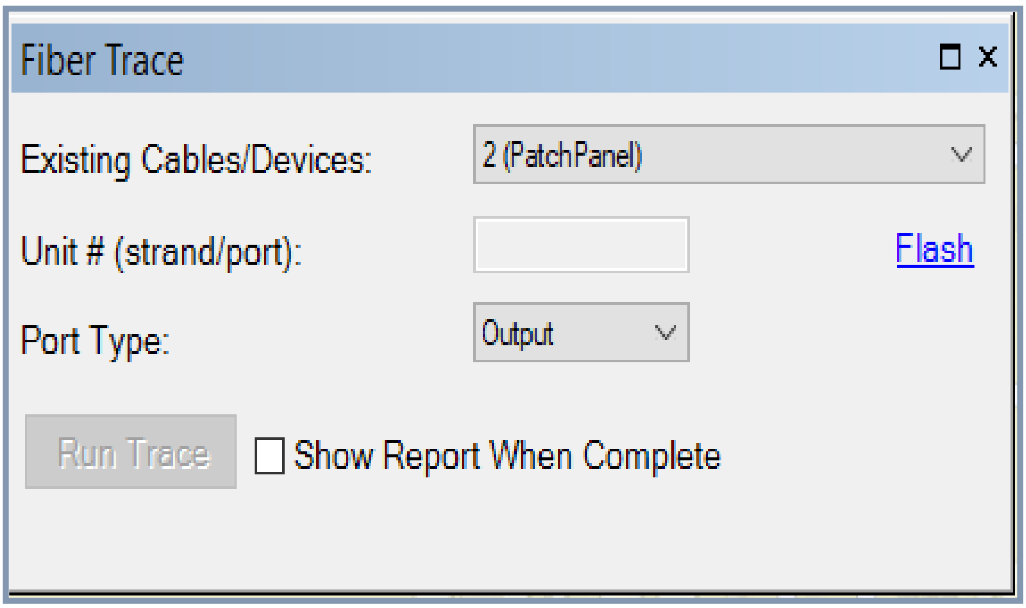


Figure (10) Fiber Trace Window. Source: Fiber Trace in Esri fiber Editing toolbar

VII. Choose the appropriate specifications for tracing the network.

VIII. Click the Trace Fiber Path button on the Telecom Trace dialog to run the analysis. The results of the trace will be displayed in the Report Results dialog and highlighted on the map. The Report Results dialog lists details of the trace, including an ordered list of all the devices, Splice Closures, and Fiber Cables that the fiber strand or port is connected to.

-TelecomNetwork – GeometricNetwork

The TelecomNetwork in the Esri template is a geometric network consisting of connected edges and junctions, along with connectivity rules. The edges in this network are Fiber Cables, which are represented as polylines, and the junctions are Splice Closures and Patch Panels, which are represented as points. Connectivity rules come in two forms: edge-junction rules and edge-edge rules.

An edge-junction rule specifies that an edge of type A may link to a junction of type B. An edge-edge rule establishes that an edge of type A may link to an edge of type B through a set of junctions. In edge-edge rules, a set of junctions is always present (desktop.arcgis.com).

The Utility Network Analyst tool (see Figure 11) helps in identifying nine built-in trace operations that reveal the characteristics of the geometric network. These operations include:

I. Find common Ancestors.
II. Find connected.
III. Find loop.
IV. Find path upstream.
V. Find path.
VI. Trace downstream.
VII. Fund upstream accumulation.
VIII. Trace upstream.

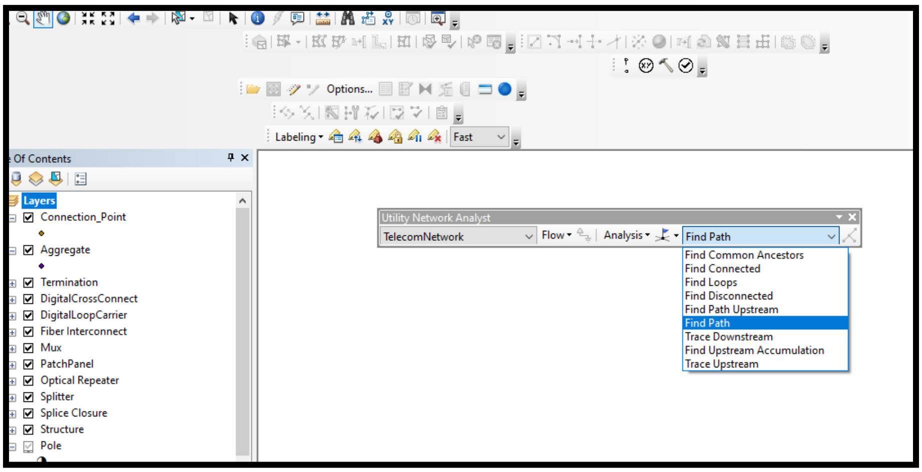


Figure (11) Utility Network Analyst Toolbar. Source: Arc Map 10.7.1 Application

Design and Methodology

-Study Methodology

To accomplish the study objectives, the following methodologies were used:

1. Programming Methodology: The methodology employed in Python is object-oriented, which relies on objects and classes. The concept of objects is central to object-oriented programming. Data structures, or objects, are defined in object-oriented programming, each with its own set of traits or attributes. Each object can have its own set of procedures and methods. Software is created by using objects that interact with one another.

2. Analytical Methodology: This methodology was used to determine the approach for solving the problem faced by maintenance engineers in the field. The program was developed using Python scripting along with the Telecom Fiber Esri template and add-in tool in Python. The program's main function is to determine the location of the cable fault by inputting the OTDR readings provided by the maintenance engineer. The fault location is represented as a point with coordinates. The coordinates are determined in both WGS84, for use in Google Maps applications, and in JTM, for calibration with GPS systems using JTM coordinates.

- Study procedures

According to the methodology followed, the study was conducted through the following stages:

Data Collection and Preparation Stage:

1. The sample data were obtained from the Jordan Ministry of Digital Economy and Entrepreneurship GIS department. This data is contained in a Feature Dataset within a File Geodatabase, as shown in Table 1 and Figure 12.

Table (1): NBN fiber network elements

Layer Name	Layer Type	Layer Description
Duct Route Civil	Line	Underground route
Arial Cable Fiber	Line	Cable hanging on electrical poles
Duct cable Fiber	Line	Cable that passing throughunderground route
Aggregate	Point	the location from which cables are distributed
Aggregate Boundary	Polygon	Aggregate boundary
Manhole	Point	Duct cables pass through.
Handhole	Point	Duct cables pass through
Connection Point	Point	School, government and Health entities
Coil	Point	The extra cable length
Poles	Point	Electrical poles where the Aerial cable connect together
Pull Box	Points	This the place where the cable enter the connection point
Splice	Point	Connect two cable together

Source: Arc Map 10.7.1 Application

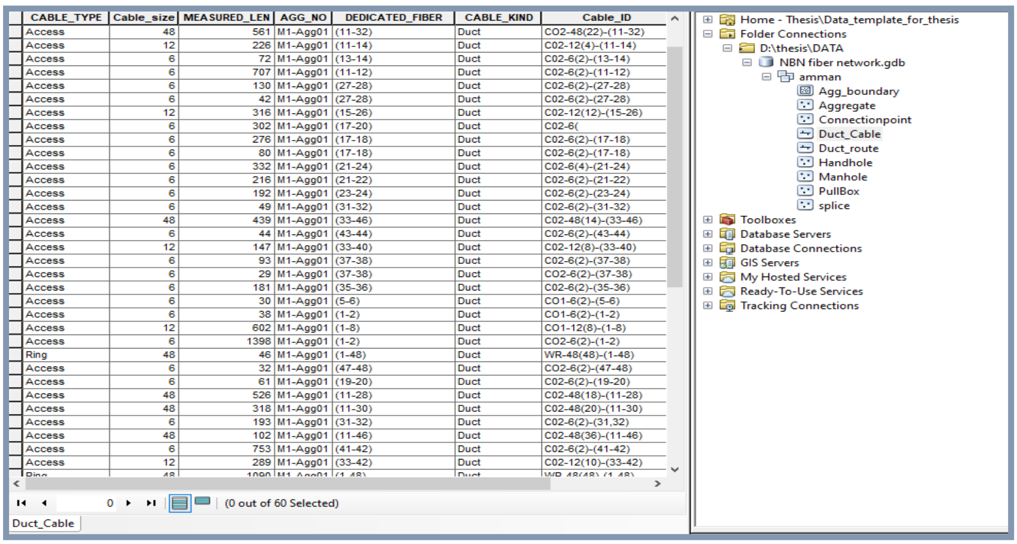


Figure (12) Feature Dataset of data in study area. Source: Arc Map 10.7.1 Application & (MODEE)

Preparation Stage: A topology was created, as shown in Figure 13, to check the connectivity between NBN network elements before loading them into the Esri Telecom Template. The rules are as follows:

I. **Duct Cable**: "Endpoint Must Be Covered by Splice."
II. **Splice**: "Must Be Covered by Endpoint of Duct Cable."
III. **Aggregate (Patch Panel)**: "Must Be Covered by Endpoint of Duct Cable."
IV. **Manhole**: "Must Be Covered by Endpoint of Duct Route."
V. **Handhole**: "Must Be Covered by Endpoint of Duct Route."
VI. **Duct Cable**: "Must Be Covered by Feature Class of Duct Route."
VII. **Connection Point**: "Must Be Covered by Endpoint of Duct Cable."

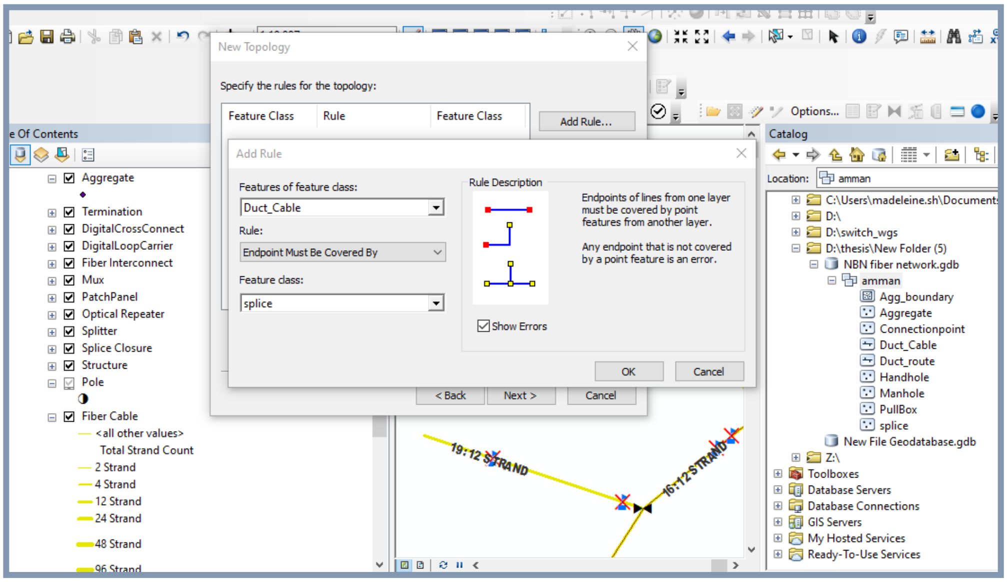


Figure (13) Topology. Source: Arc Catalog in ArcGIS 10.7.1

- Loading data Into Esri Telecom Template

Before loading the data, it was necessary to transform the coordinate system because the Esri template uses WGS84, while the NBN fiber network is in Jordan Transverse Mercator (JTM). The transformation was performed by projecting the template to JTM using the following steps: ArcMap > ArcToolbox > Data Management Tools > Projections and Transformations > Project, as shown in Figure 14.

To load data into the template, it must be done in ArcMap using the Esri Fiber Editing toolbar's "Create Feature" window. This is because any feature created within the template in any layer of the Telecom Dataset will have an "Item of Plant ID" field, which is a unique identifier that establishes relationships between features.

Conduit Network (Duct Route, Manhole, Handhole, Pull Box):

First, copy the Conduit Network to the template by clicking the Copy command on the Standard toolbar, selecting the correct feature template from the feature template groups, and then clicking the Paste command on the ArcMap Standard toolbar.

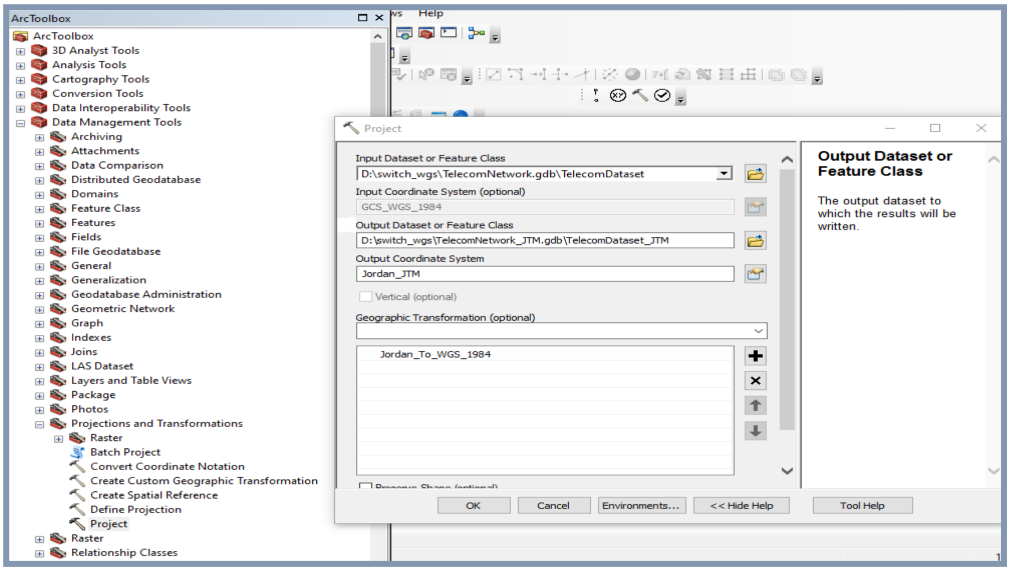


Figure (14) Project Window. Source: Arc Toolbox in Arc Map 10.7.1

Telecom Network (PatchPanel, Splice Closure, and Duct Cable):

Next, copy the Telecom Network features to the appropriate feature template, following the same procedure as in step 1.

Association Between Fiber Cable and Conduit:

The GIS tool of this study will rely on Fiber Device Connection and Splice Closure Connection. The process was carried out as follows: I. The aggregate was represented as a PatchPanel with 48 ports for access cable C02-48 (48). II. An access cable of size 48 was snapped to the PatchPanel point feature. III. The connection between the access cable and the PatchPanel was established using the Fiber Connection Editor (Figure 15).

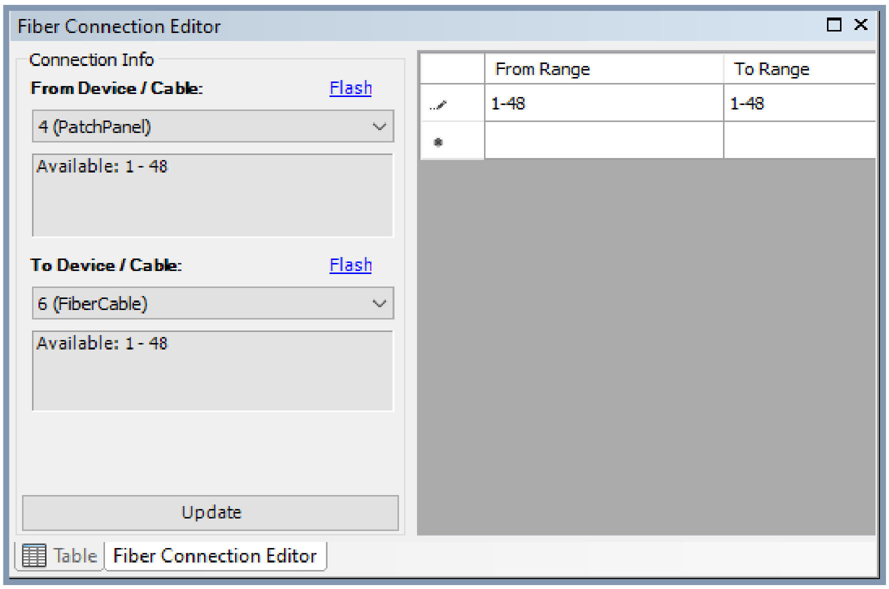


Figure (15) Fiber Connection Editor. Source: Esri Fiber Editing Toolbar in Arc Map 10.7.1

4. The splices were configured in fiber splice tool, Figure (16).

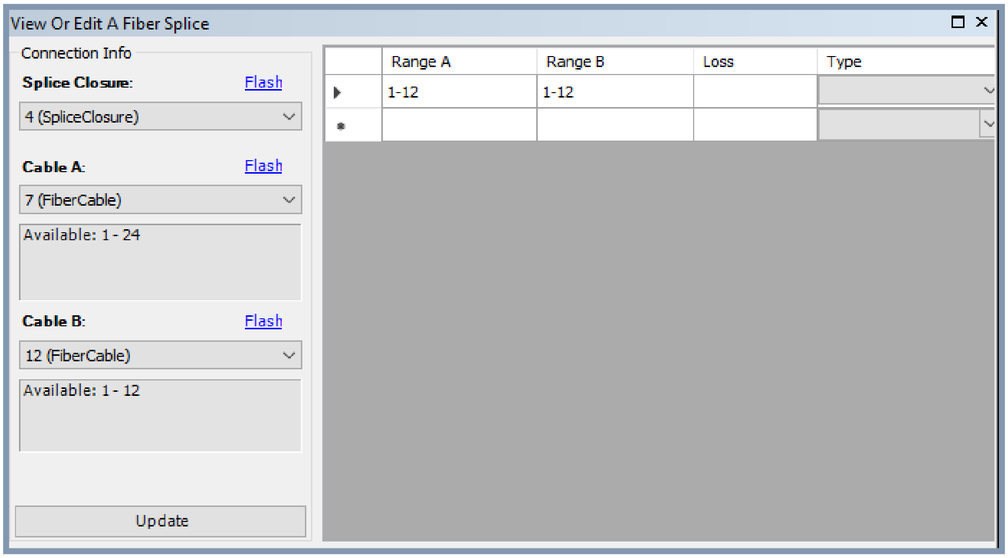


Figure (16) Edit Fiber Splice window. Source: Esri Fiber Editing Toolbar in Arc Map 10.7.1

-Network Tracing

To trace the strands provided by the maintenance engineer, configure the Fiber Trace tool in Esri Fiber Editing (Figure 17). The result will include the selected features of the Fiber Cable feature class.

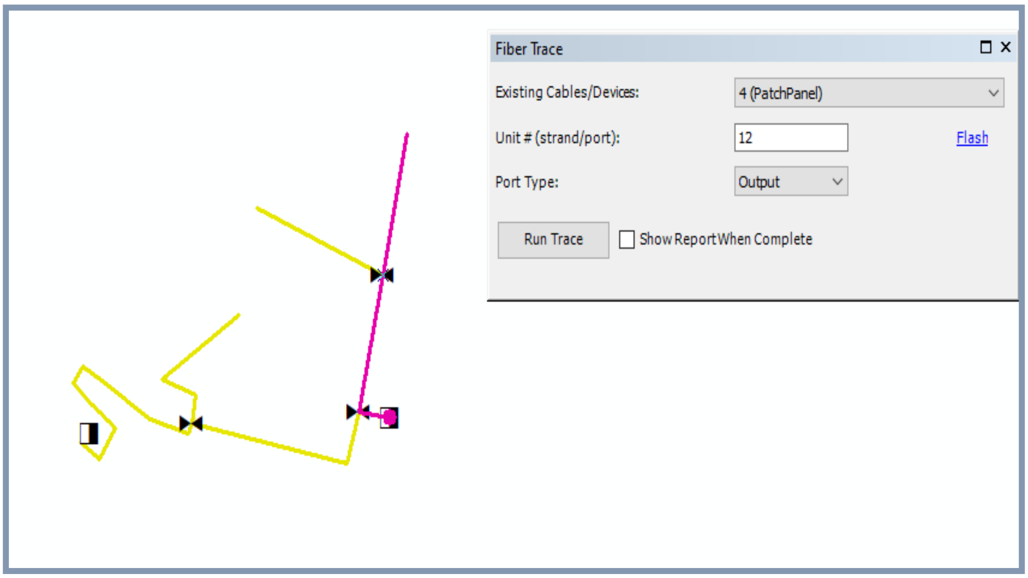


Figure (17) Fiber Trace. Source: Esri Fiber Editing Toolbar in Arc Map 10.7.1

-Python scripting program of GIS Tool

The OTDR Trace tool (Figure 18) was created in this study and was accomplished through the following steps.

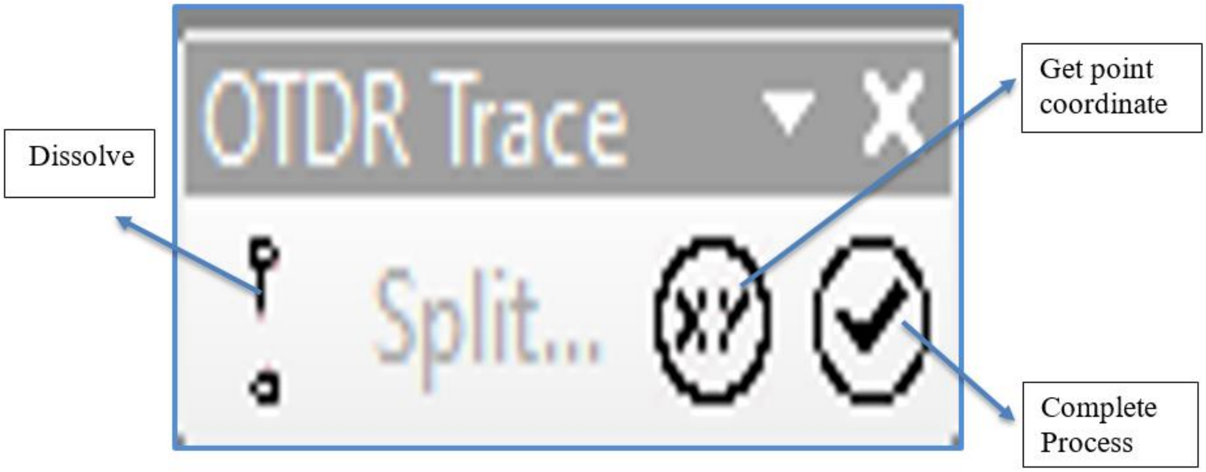


Figure (18) OTDR Trace Toolbar. Source: Researchers Work

1. Double-click the Python Add-in executable file. The wizard will open and prompt you to choose a location for creating the tool. Select C:\ODTR as shown in Figure (19).

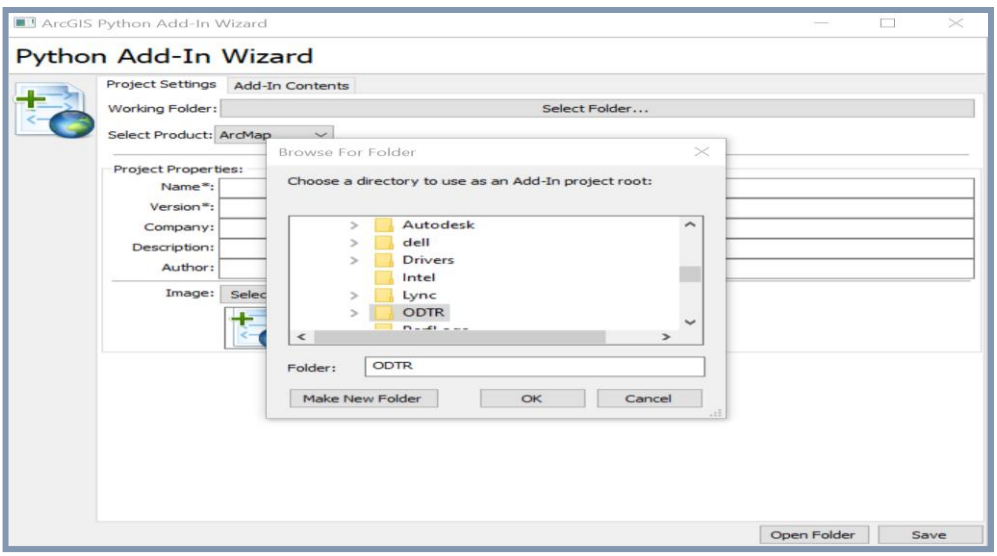


Figure (19) Python Add-In wizard. Source: Python Add-in executable file

2. On the Add-in Contents tab, right-click on Toolbars and choose "Add New Toolbar." In the right panel, enter a label and ID for the toolbar to distinguish it from others, as shown in Figure (20).

.

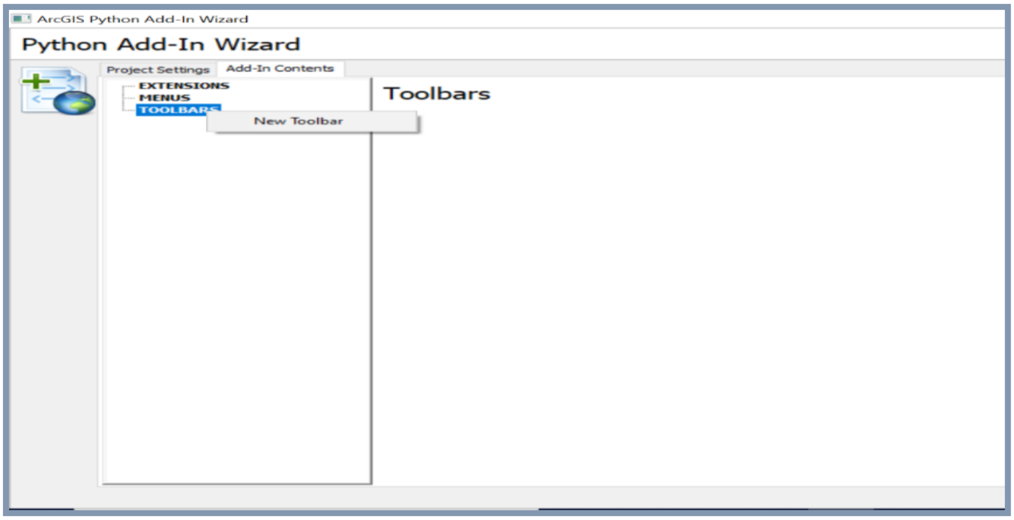


Figure (20) Python Add-In Contents Window. Source: Add-in Contents Toolbars

2. Right click on toolbar then choose new button. See Figure (21).

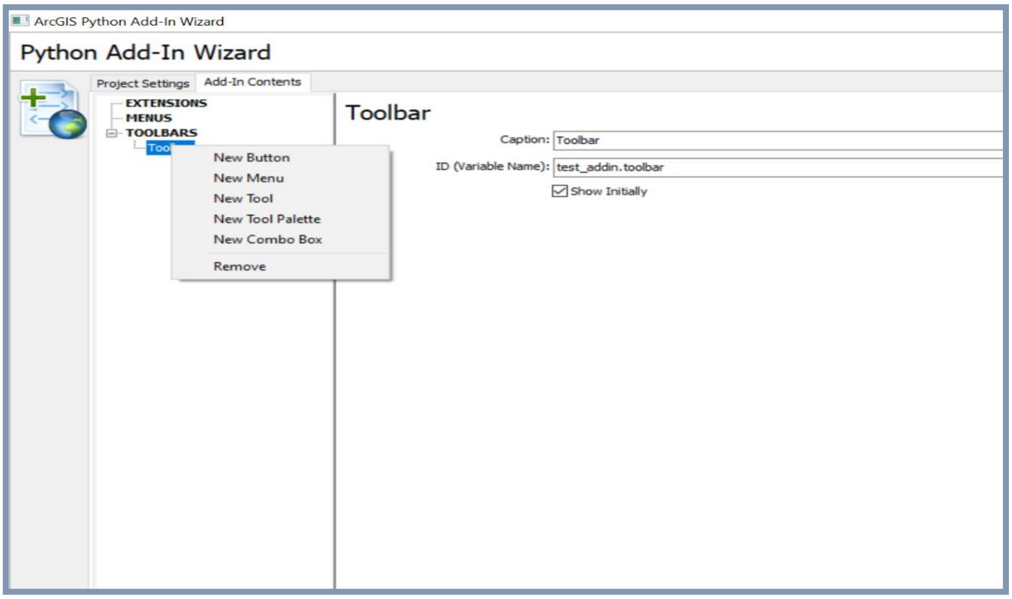


Figure (21) Python Add-In Toolbar Window. Source: Add-in Contents Toolbar

4. Fill in the right panel with the appropriate parameters. Once you have completed entering the properties, click the Save button at the bottom of the wizard. This will create all the necessary files and folders within the working directory, as shown in Table (2).

Table (2): The parameters description of button creation

Property	Description
Caption	The caption is defined by this property and used in the Add-In Manager (Figure 22) to identify the various types of customizations available. This is

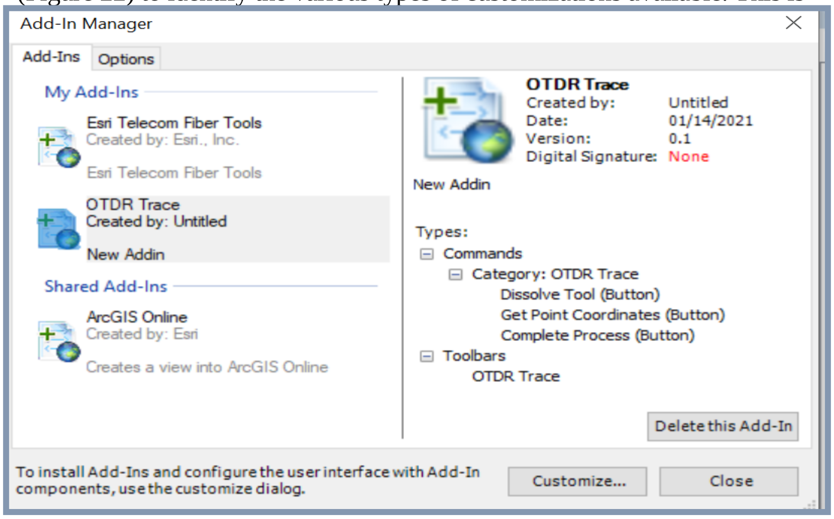


Figure (22) Add-In Manger. Source: Arc Map10.7.1 Customize Tap

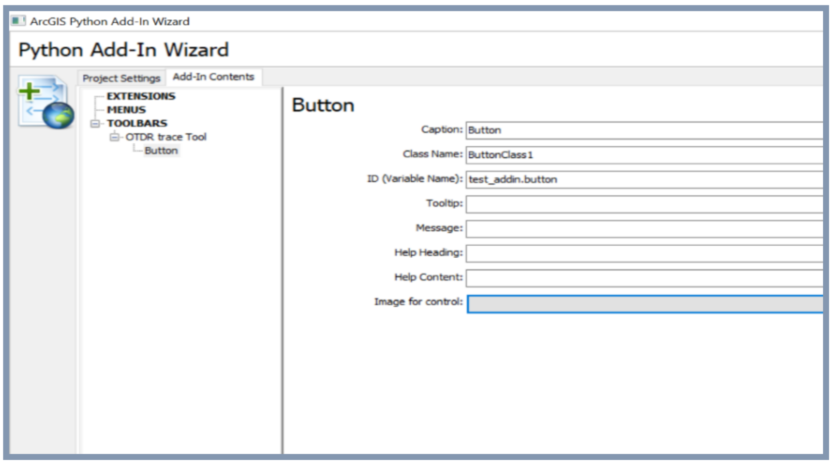


Figure (23) Python Add-In Button Window. Source: Add-in Contents Button

Property	Description
Class	When a button in a desktop program is clicked, the associated Python class is executed. The business logic for the button is written in this Python class.
ID	The button's unique name is used to identify it. It is possible to construct multiple buttons for a single project, and this ID is used to differentiate between them.
ToolTip	When the pointer pauses over a button in a desktop application, a short description appears.
Message	A full description of the function of the button. When the mouse pointer pauses over the button, the message shows beneath the ToolTip.
Image	The button should be represented by a 16X16-pixel picture. One of the most widely used picture formats should be utilized (that is, .bmp, .jpg, and so on). The image will be saved in the Add-in project's Images folder.
Help Heading	This indicates what the help content is about.
Help Content	The help content for the button. These characteristics provide data that will be used when a user requests context-sensitive assistance. These are pop-up subjects that stay on the screen until the user clicks away.

Source: Arc Map 10.7.1 Help

6. Three buttons are created, the dissolve Tool button, and Get Point Coordinates button, Complete Process button, Figure (24).

7. The Python script (otdr_trace_addin) is located in the c:\ODTR\install folder, where the code for the three buttons has been written. To access the file, right-click on it and choose "Edit with IDLE," as shown in Figure (25).

8. The file opened and there is a default code comes with the creation of the buttons, Figure (26).

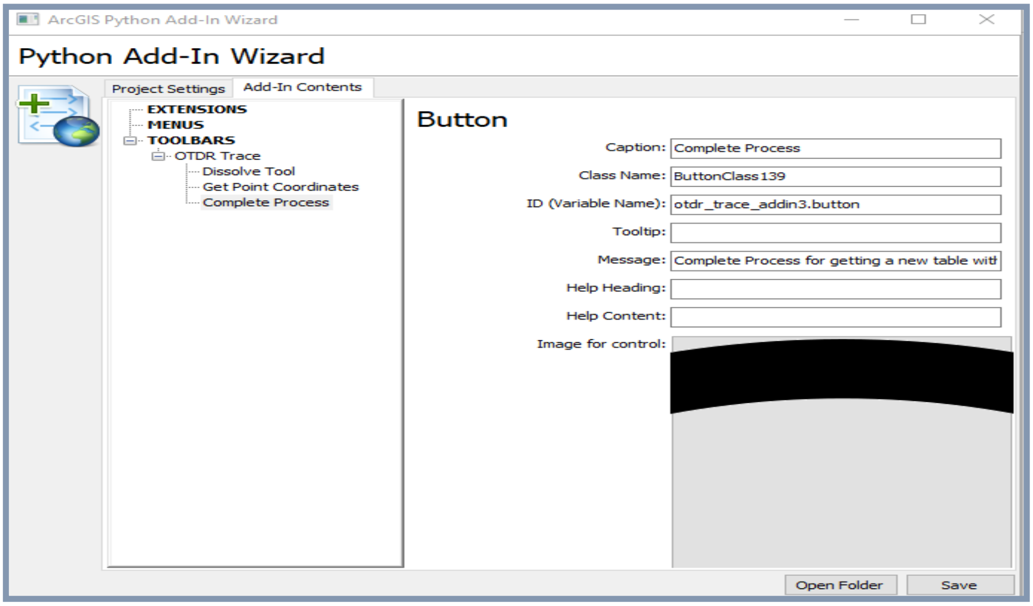


Figure (24) Button Description. Source: Add-in Contents Button

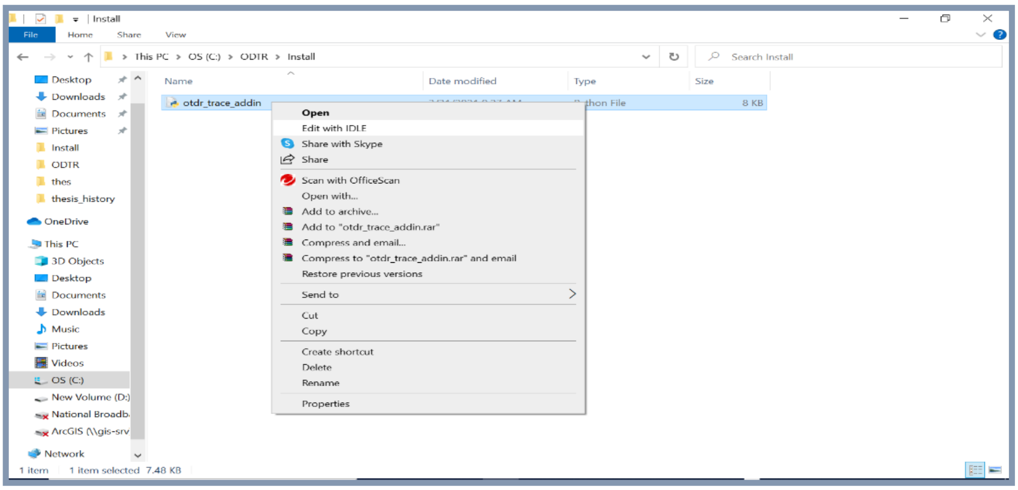


Figure (25) Install Folder. Source: Windows Explorer

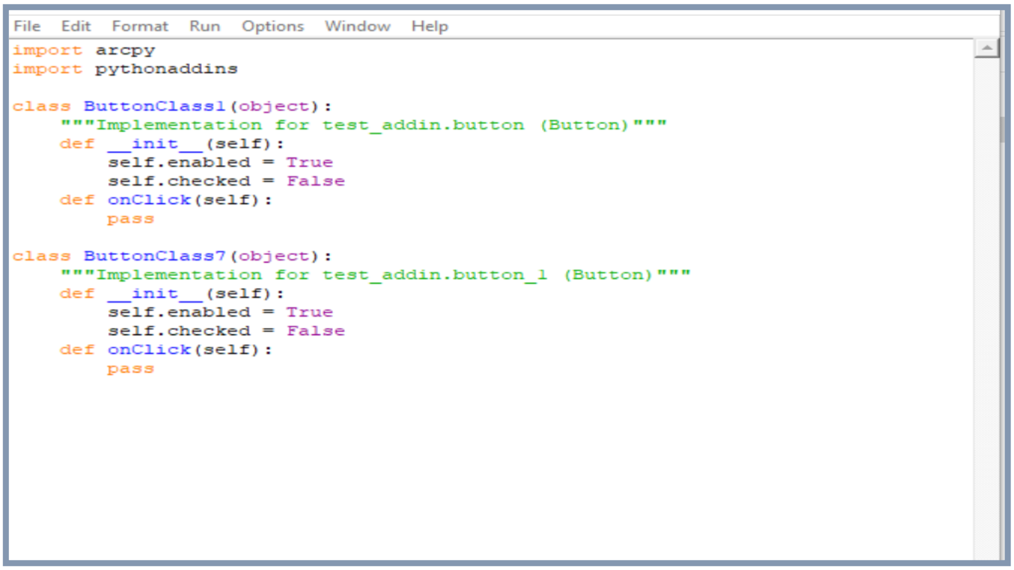
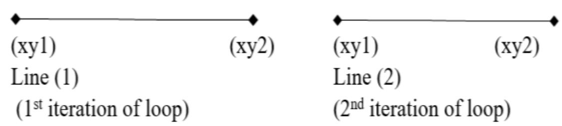


Figure (26) IDLE file of OTDR Tool. Source: Researchers Work by using Python Scripting

8. The source code of the tool (Appendix I) will be explained as follows:

1. The **dissolve** button Union the selected lines in the Esri template into a single line.
2. Get the count of the selected features in the Fiber Cable feature class.
3. If more than one feature is selected, the code will check if the FiberCable_Dissolve feature class exists. If it does, it will be removed from the current MXD and then deleted. This step ensures that FiberCable_Dissolve contains only one object at a time.
4. Dissolve management is used to union the selected features of “Fiber Cable”.
5. The Split tool splits the selected feature of "FiberCable_Dissolve" at a specified distance provided by the maintenance engineer. After splitting the union line, the process checks for intersection points between the two lines to display the midpoint and its coordinates in a new feature class called FinalDetails_SplitLines. The process follows these steps.

- The loop starts with Line (1) and retrieves its xy1 and xy2 coordinates, creating a new feature class called IntersectMidPointBetween_SplitLines to add points at (xy1) or (xy2) coordinates.
- These coordinates are checked for intersections with the split lines of the FiberCable_Dissolve feature class.
- If neither of these coordinates intersects with the two lines, the process repeats with Line (2) until the specified point intersects with both lines.
- Finally, ArcMap will refresh the map to display the midpoint.



6. **Get point Coordinate Button** get the coordinates of the middle Point after splitting the line

Discussions, Results and Recommendations

- Discussion & Results

This study demonstrated that it is possible to determine the locations of fiber cable faults using this newly developed GIS tool, provided that the tool is given accurate distance measurements. The precise OTDR-measured distance is achieved by entering both the Index of Refraction (IOR) and the helix percentage (a ratio between the fiber and cable length), which are supplied with the cable documentation from the factory. These parameters compensate for the distance difference between the cable sheath and the fiber (Figure 27). Because the OTDR measures the length of the fiber, not the physical cable, some OTDRs do not have a parameter to enter the helix percentage during calibration. In such cases, simple math is used to adjust the IOR as follows:

1. Cable markings on a bulk spool indicate the cable length.
2. The fiber length will be reported after the OTDR test using an initial IOR (factory value) in the OTDR setup screen.
3. The helix percentage is determined by dividing the fiber distance by the cable distance.
4. The adjusted IOR is calculated by multiplying the helix factor by the fiber's initial IOR.
5. The adjusted IOR is then entered into the OTDR settings, which will provide a more accurate, though not perfect, location (https://learn.lightbrigade.com/).

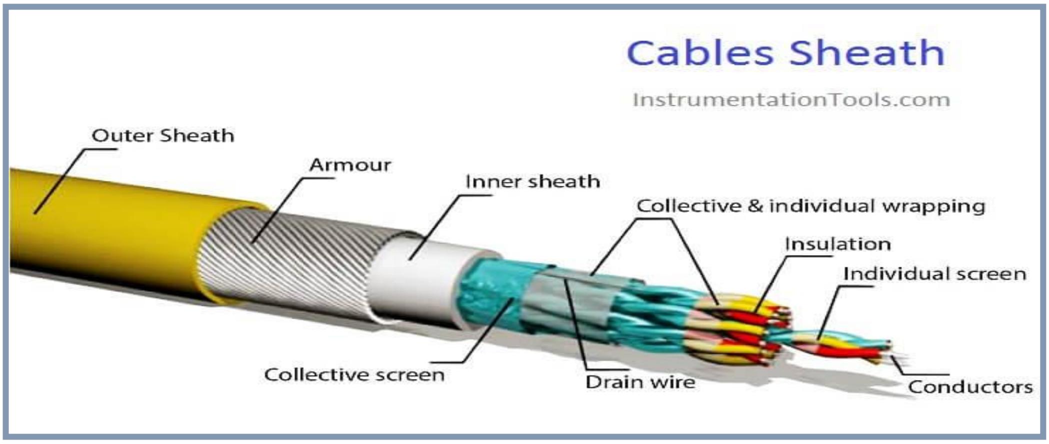


Figure (27): Cable sheath material. Source: https://instrumentationtools.com

To determine the locations of fiber-optic cable faults with minimal time and effort—whether the fault is a complete fiber cut or a high-attenuation issue affecting communication continuity—it is essential to use innovative methods, such as the developed GIS tool. This tool enhances the efficiency of maintenance operations in the field, reduces the downtime of optical fiber networks, and increases customer satisfaction.

In practice, the concept of this tool can be summarized as follows: When the NOC (Network Operation Center) at the Jordan Ministry of Digital Economy and Entrepreneurship (MODEE) informs the maintenance team of a fiber-optic cable failure with preliminary information, the team uses this initial data, along with cable distribution drawings and a geographic information system (GIS), to identify the affected entities. The maintenance team then begins testing operations, typically starting from the aggregate rooms with the Optical Distribution Frame (ODF). The OTDR test equipment is used to assess the cable, providing results such as the type of fault and its distance. This distance is then used as input for the developed GIS tool, which immediately provides the (X, Y) coordinates in JTM for use with the Global Positioning System (GPS) calibrated to the JTM coordinate system. Additionally, the GIS tool also provides coordinates in the WGS84 system, which can be entered into Google Maps on smartphones. The Esri data model for telecommunication helps organize GIS data and trace specified strands. The Python Add-in customization was used to create a toolbar with three buttons for pinpointing the cable fault.

Case Study 1: The maintenance engineer was informed that “Ali Reda Al Rakabi” Primary School was out of service. The engineer used the GIS drawing to determine which aggregate it belonged to and the strands associated with it. The aggregate was Am Katheer, and the strands were (23, 24) of C02 Fiber Cable. The next step involved using the OTDR at the C01 cable’s ODF in the aggregation room to specify the fault distance. The distance was calculated to be 1500 meters. This distance was entered into the Developed GIS Tool, which provided coordinates in JTM as (x = 390529.136269, y = 544565.91255). These coordinates were then input into a GPS device, and the WGS84 coordinates (E 35.840724, N 32.024046) were provided for entry into Google Maps on a smartphone.

Case Study 2: For Jubeiha Comprehensive Secondary School for Boys, the fiber cut was located 2500 meters from the aggregate. The detected fibers were (9, 10) of C02 Fiber Cable. The coordinates of the fiber cut location were (x = 392764.197744, y = 545804.359877) in JTM, and (E 35.864245, N 32.035429) in WGS84.

After applying the optical distance to the Developed GIS Tool, the software provided accurate coordinates based on the OTDR equipment's accuracy.

RECOMMENDATIONS

Many elements of the fiber network contain important spatial information. Without a map, this spatial information loses context and becomes essentially meaningless. Therefore, creating tools to address specific questions is crucial.

GIS applications can support operations and maintenance in telecom companies, and one of their key applications is locating fiber cuts in the network by integrating with OTDR devices. Such applications can be developed using licensed and specialized programs designed for managing communication networks, which operate within the geographic information systems (GIS) environment.

Esri, the company behind ArcGIS, the world’s most powerful mapping and spatial analytics software, partners with developers to create applications built on the Esri ArcGIS platform. In an optical fiber network, there are three primary solutions, which are:

I. The Esri data model for telecommunications, used in this study, is available for free.

II. 3GIS, a web GIS solution for fiber networks, offers a framework for gathering, organizing, and analyzing telecommunication information with a spatial basis.

III. The Ericsson Network Engineer application provides network operators with a centralized graphical information system (GIS) for planning, designing, maintaining, and documenting end-to-end communications networks. Its graphical user interface resembles ArcMap. This application is expensive.

I recommend that the Jordan Ministry of Digital Economy and Entrepreneurship study the characteristics of each of these solutions to enhance the performance of its fiber network. Doing so will save time and effort in managing the network.

ACKNOWLEDGMENT

In the name of God, the Most Gracious, the Most Merciful, and prayers and peace be upon our Prophet Muhammad. I would like to thank my supervisor, Prof. Dr. Mohammed Jamil Al-Qaralleh, for his support and trust, which gave me the strength to complete this thesis. I also want to express my deep gratitude to my family, especially my husband, Mohammad Al-Hadidi.

Appendix (I)
The Developed GIS Tool Source Code

```
import arcpy
import Pythonaddins
import glob
import os, shutil

mxd = arcpy.mapping.MapDocument ("CURRENT")
sr = mxd.activeDataFrame.spatialReference

arcpy.env.workspace = r"C:\ODTR\Default2.gdb"
arcpy.env.overwriteOutput = True

class ButtonClass139(object):
    """Implementation for otdr_trace_addin3.button (Button)"""
    def __init__(self):
        self.enabled = True
        self.checked = False
    def onClick(self):

        # Spatial reference: JTM Jordan
        in_coordinate_system = arcpy.SpatialReference(102158)
        # Spatial reference: GCS_WGS_1984
        out_coordinate_system = arcpy.SpatialReference(4326)
        arcpy.env.overwriteOutput = True
        arcpy.Project_management(r"C:\ODTR\Default2.gdb\FinalDetials_SplitLines",
r"C:\ODTR\Default2.gdb\FinalDetials_Projected",
out_coordinate_system,"Jordan_To_WGS_1984", in_coordinate_system)

        wgs = arcpy.SpatialReference(4326)
        with arcpy.da.UpdateCursor("FinalDetials_Projected", ['SHAPE@',
'Longitude_X', 'Latitude_Y']) as rows:
             for row in rows:
                    pnt_wgs = row[0].projectAs(wgs,"Jordan_To_WGS_1984")
                    row[1:] = [pnt_wgs.centroid.X, pnt_wgs.centroid.Y] #will be
in decimal degrees
                    rows.updateRow(row)


        print "Progrss Point Complete"
        arcpy.RefreshActiveView()

class ButtonClass284(object):
    """Implementation for otdr_trace_addin.button (Button)"""
    def __init__(self):
```

```
        self.enabled = True
        self.checked = False
    def onClick(self):
        if arcpy.Exists(r"C:\ODTR\Default2.gdb\FiberCable_Dissolve"):

arcpy.Delete_management(r"C:\ODTR\Default2.gdb\FiberCable_Dissolve")
        else:
             arcpy.CreateFileGDB_management("C:\ODTR", "Default2.gdb")
        selCount = len([int(fid) for fid in arcpy.Describe("Fiber
Cable").fidset.split(";") if fid != ''])
        if(selCount > 0):
            df=arcpy.mapping.ListDataFrames(mxd)[0]
            for lyr in arcpy.mapping.ListLayers(mxd, "*",df):
                if lyr.name == "FiberCable_Dissolve":
                      arcpy.mapping.RemoveLayer(df, lyr)
                      print(lyr.dataSource)
                      arcpy.Delete_management(lyr.dataSource)
                      print("Layer Deleted")
            arcpy.Dissolve_management("Fiber Cable",
"C:\ODTR\Default2.gdb\FiberCable_Dissolve")
            workspace = r'C:\ODTR\Default2.gdb'
            edit = arcpy.da.Editor(workspace)
            edit.startEditing(False, True)
            edit.startOperation()
            arcpy.SelectLayerByAttribute_management("Fiber Cable",
"CLEAR_SELECTION")
            arcpy.SelectLayerByAttribute_management("FiberCable_Dissolve",
"NEW_SELECTION", "OBJECTID = 1")
            # Stop the edit operation.
            edit.stopOperation()
            # Stop the edit session and save the changes
            arcpy.RefreshActiveView()
        else:
            Pythonaddins.MessageBox("Please Select at least one Line for
Split","Note",0)


class ButtonClass285(object):
    """Implementation for otdr_trace_addin1.button (Button)"""
    def __init__(self):
        self.enabled = True
        self.checked = False
    def onClick(self):
         inFeatures = "FiberCable_Dissolve"
         shapeName = arcpy.Describe (inFeatures).shapeFieldName
         rows = arcpy.SearchCursor(inFeatures)
         whichxypoint = 0
         for row in rows:
```

```
            feat = row.getValue(shapeName)
            xy1 = feat.firstPoint
            xy2 = feat.lastPoint
            fc = r"C:\ODTR\Default2.gdb\IntersectMidPointBetween_SplitLines"
            df=arcpy.mapping.ListDataFrames(mxd)[0]
            for lyr in arcpy.mapping.ListLayers(mxd, "*",df):
                if lyr.name == "IntersectMidPointBetween_SplitLines":
                      arcpy.mapping.RemoveLayer(df, lyr)
                      print(lyr.dataSource)
                      arcpy.Delete_management(lyr.dataSource)
                      print("Layer Deleted")
            arcpy.CreateFeatureclass_management("C:\ODTR\Default2.gdb",
"IntersectMidPointBetween_SplitLines", "POINT")
            cursor = arcpy.da.InsertCursor(fc, ["SHAPE@XY"])
            cursor.insertRow([xy1])
            whichxypoint = 1
            del cursor

            arcpy.SelectLayerByLocation_management(inFeatures, "intersect", fc,
0, "new_selection")
            selCount = len([int(fid) for fid in
arcpy.Describe(inFeatures).fidset.split(";") if fid != ''])

            #process with xy1
            if(selCount > 1 and whichxypoint == 1):
                midpoint_xy = xy1
                break
            elif(selCount == 1 and whichxypoint == 1):
                df=arcpy.mapping.ListDataFrames(mxd)[0]
                for lyr in arcpy.mapping.ListLayers(mxd, "*",df):
                   if lyr.name == "IntersectMidPointBetween_SplitLines":
                      arcpy.mapping.RemoveLayer(df, lyr)
                      print(lyr.dataSource)
                      arcpy.Delete_management(lyr.dataSource)
                      print("Layer Deleted")
                arcpy.CreateFeatureclass_management("C:\ODTR\Default2.gdb",
"IntersectMidPointBetween_SplitLines", "POINT")
                cursor = arcpy.da.InsertCursor(fc, ["SHAPE@XY"])
                cursor.insertRow([xy2])
                whichxypoint = 2
                del cursor

            arcpy.SelectLayerByLocation_management(inFeatures, "intersect", fc,
0, "new_selection")
            selCount = len([int(fid) for fid in
arcpy.Describe(inFeatures).fidset.split(";") if fid != ''])
            #process with xy2
            if(selCount == 1 and whichxypoint == 2):
```

```
                continue
            else:
                midpoint_xy = xy2
                break

        df=arcpy.mapping.ListDataFrames(mxd)[0]
        checkiflayeravaliableornot = 0
        for lyr in arcpy.mapping.ListLayers(mxd, "*",df):
                if lyr.name == "FinalDetials_SplitLines":
                     fc = r"C:\ODTR\Default2.gdb\FinalDetials_SplitLines"
                     cursor = arcpy.da.InsertCursor(fc,
["SHAPE@XY","XCOORD","YCOORD"])
                     cursor.insertRow([midpoint_xy, midpoint_xy.X,
midpoint_xy.Y])
                     del cursor
                     checkiflayeravaliableornot = 1
                     break
        if checkiflayeravaliableornot == 0:
                     arcpy.CreateFeatureclass_management("C:\ODTR\Default2.gdb",
"FinalDetials_SplitLines", "POINT")
                     fc = r"C:\ODTR\Default2.gdb\FinalDetials_SplitLines"
                     arcpy.AddField_management(fc, "XCOORD", "DOUBLE")
                     arcpy.AddField_management(fc, "YCOORD", "DOUBLE")
                     arcpy.AddField_management(fc, "Longitude_X", "DOUBLE")
                     arcpy.AddField_management(fc, "Latitude_Y", "DOUBLE")
                     cursor = arcpy.da.InsertCursor(fc,
["SHAPE@XY","XCOORD","YCOORD"])
                     cursor.insertRow([midpoint_xy, midpoint_xy.X,
midpoint_xy.Y])
                     del cursor
        df=arcpy.mapping.ListDataFrames(mxd)[0]
        for lyr in arcpy.mapping.ListLayers(mxd, "*",df):
              if lyr.name == "IntersectMidPointBetween_SplitLines":
                      arcpy.mapping.RemoveLayer(df, lyr)
                      print(lyr.dataSource)
                      arcpy.Delete_management(lyr.dataSource)
                      print("Layer Deleted")
        arcpy.RefreshActiveView()
```

REFERENCES

Fu, M., *et al*. (2020), Study on the generation and influence factors of the new wavelength components in backscattering light of OTDR, ***Optics Communications***, *474*, November 2020.

Guo, R., & Cai, J. (2002), AGIS approach to the spatial assessments of telecommunication infrastructure, ***Networks and Spatial Economics***, *2*, 35–63.

Halaseh, J. J. (2016), ***The investigation of optical soliton in optical fiber,*** Master Thesis, Mutah University, Karak-Jordan.

Hu, C., & Sheng, R. (2009), The fault location of optical cable for communications based on GIS, In ***International Forum on Information Technology and Applications***, School of Computer Science and Technology, North China Electric Power University.

Hogari, K., & Others. (2010), Design and performance of ultra-high-density optical fiber cable with rollable optical fiber ribbons, ***Optical Fiber Technology***, ***16***(4), June 2010.

Shlyagin, M. G., & Others. (2013), Optical-fiber self-referred refractometer based on Fresnel reflection at the fiber tip, ***Sensors and Actuators B: Chemical***, *178*, 263–269.

Nixon, L. (2012), ***GIS in the management of fiber optic cable network in urban centers: Case study of Frontier Optical Networks Limited,*** Master Thesis, College of Architecture and Engineering, University of Nairobi, Republic of Kenya.

Nashed, A. I. A. (2006), ***Optical Bragg fiber design,*** Master Thesis, Mansoura University, Egypt.

Sheng, R., & Tan, J. (2008), Communication cable fault location in power systems, In ***International Conference on Artificial Intelligence Commerce (AIMSEC)***.

O'Duill, S. P., & Others. (2020), Applicability of small-signal laser and fiber models for passive optical networks operating at the 1550 nm window, ***Optical Fiber Technology***, *56*, May 2020.

Skendzic, S. (2015), ***ASIC-enabled high resolution optical time domain reflectometer,*** Master Thesis, University of California, Santa Barbara-USA.

Matrood, Z. M., *et al.* (2014). A simple GIS-based method for designing fiber networks, ***International Journal of Engineering and Innovative Technology (IJEIT)***, ***4***(2), August 2014.

Zandbergen, P. A. (2013), ***Python scripting for ArcGIS***, California: Esri Press.

Arabic References

Ahmad, M. A. H. (2011), ***Monitoring oil pipelines using optical fibers,*** Master Thesis, Umm Al-Durman Islamic University, Sudan.

Almutari, N. F. (2007), ***Evaluation of the optical fiber network to the home in the State of Kuwait,*** Master Thesis, Arab Gulf University, Bahrain.

Hamadi, M. K. (2002), ***The birefringent effect on coherent light propagation in single-mode fiber optics,*** Master Thesis, Al-Mustansiriya University, Iraq.

Mohammed, N. M. A. (2018), ***Optical fiber performance evaluation,*** Master Thesis, Umm Al-Durman Islamic University, Sudan.

Web Site References

https://www.esri.com

https://www.dos.gov.jo

https://www.tutorialspoint.com/principles_of_communication/principles_of_optical_fiber_communications.htm.

https://www.instrumentationtools.com

https://www.alibaba.com/product-detail/12-24-48-Core-Vertical-Type_60800998747.html,

تطوير أداة نظم المعلومات الجغرافية (GIS) لإيجاد أعطال كوابل الألياف الضوئية باستخدام لغة بايثون لتطبيقات "ArcGIS" في مدينة عمان

مادلين تيسير شحادة [1]، محمد جميل القراله [2]

الملخّص

تتميز كوابل الألياف الضوئية بنطاق ترددي أكبر من وسائط النقل الأخرى، مما يزيد من كمية البيانات المنقولة بوحدة الوقت. إن إدارة شبكة الألياف باستخدام نظم المعلومات الجغرافية له فوائد لتشغيل شبكة الاتصالات وصيانتها. تهدف الدراسة إلى تطوير أداة GIS لتحديد أعطال كوابل الألياف باستخدام Python Scripting For ArcGIS في مدينة عمان. لتحقيق هذا الهدف، تم استخدام نموذج بيانات Esri للاتصالات. توصلت الدراسة إلى أنه من الممكن تحديد موقع الخطأ باستخدام لغة البرمجة بايثون ويعطي البرنامج إحداثيات دقيقة اعتمادًا على دقة جهاز الـ OTDR.

الكلمات الدالة: الألياف، بايثون، البرمجة النصية، الدقة، ArcGIS، قالب اتصالات Esri.

---

[1] وزارة الاقتصاد الرقمي وريادة الأعمال، الأردن

madeleine.shehadeh@modee.gov.jo

[2] قسم الجغرافيا، كلية العلوم الاجتماعية، جامعة مؤتة، الأردن

mqaralleh@mutah.edu.jo

تاريخ استلام البحث 2023/6/12، وتاريخ قبوله 2024/2/22.