# Prediction and Classification of the Optical Transport Network Faults Using Hybrid Deep Learning Model

Girma Negese Hordofa

A Thesis Submitted to The Department of Computer Science and Engineering,

School of Electrical Engineering and Computing

Presented in Partial Fulfillment of the Requirement for the Degree of Master of Science in Computer Science and Engineering

Office of Graduate Studies

Adama Science and Technology University

September, 2024

Adama, Ethiopia

# Prediction and Classification of the Optical Transport Network Faults Using Hybrid Deep Learning Model

Girma Negese Hordofa

Advisor: Frezewd Lemma Tena (Dr.-Ing.)

A Thesis Submitted to The Department of Computer Science and Engineering, School of Electrical Engineering and Computing

Presented in Partial Fulfillment of the Requirement for the Degree of Master of Science in Computer Science and Engineering

Office of Graduate Studies
Adama Science and Technology University

September, 2024

Adama, Ethiopia

# Declaration

I declare that this master thesis "**Prediction and Classification of Optical Transport Network Faults Using Hybrid Deep Learning Model**" is my original work and has not been submitted to any universities for a similar purpose. The references used in this thesis are duly recognized by proper citations.

| Girma Negese Hordofa | _____ | _____ |
|---|---|---|
| Name of student | Signature | Date |

# Advisor Recommendation

I, the advisor of this thesis, hereby certify that I have closely advised the student while developing this thesis and read the draft thesis entitled **"Prediction and Classification of Optical Transport Network Faults Using Hybrid Deep Learning Model"** prepared under my guidance by **Girma Negese Hordofa.** Therefore, I recommend the submission of the thesis to the department for further review and evaluation.

| | | |
|---|---|---|
| Frezewd Lemma Tena (Dr.-Ing.) | _____ | _____ |
| Main Advisor | Signature | Date |

# Approval Page

I/we hereby certify that the recommendation and suggestion given by the thesis review committee are appropriately incorporated into the final thesis entitled “Prediction and Classification of Optical Transport Network Faults Using Hybrid Deep Learning Model” by Girma Negese Hordofa.

| | | |
|---|---|---|
| Frezewd Lemma Tena (Dr.-Ing.) | _____ | _____ |
| Major Advisor | Signature | Date |

We, the undersigned, members of the Board of the thesis open defense by Girma Negese Hordofa have read and evaluated the thesis entitled “Prediction and Classification of Optical Transport Network Faults Using Hybrid Deep Learning Model” and assessed the understanding of the thesis. Therefore, to certify that the thesis is accepted for the partial fulfillment of the requirement of the degree Master of Science in Computer Science and Engineering.

| | | |
|---|---|---|
| _____ | _____ | _____ |
| Chairperson | Signature | Date |
| _____ | _____ | _____ |
| Internal Examiner | Signature | Date |
| _____ | _____ | _____ |
| External Examiner | Signature | Date |

Final approval and acceptance of the thesis is contingent upon submission of its final copy to the Office of Postgraduate Studies (OPGS) through the Department Graduate Council (DGC) and School Graduate Committee.

| | | |
|---|---|---|
| _____ | _____ | _____ |
| Department Head | Signature | Date |
| _____ | _____ | _____ |
| School Dean | Signature | Date |
| _____ | _____ | _____ |
| Office of Postgraduate Studies | Signature | Date |

# ACKNOWLEDGEMENT

Before anything else, I want to thank God from the bottom of my heart for giving me the courage, discernment, and persistence to accomplish this task.

I would like to convey my profound appreciation to Dr. Eng Frezewd Lemma, my advisor, whose knowledge, insightful criticism, and constant encouragement have been indispensable to this study. His support and insightful criticism have greatly influenced the direction and quality of my work. I appreciate his trust in my abilities and his unwavering support and inspiration. His supervision has been greatly valued, and his efforts have been crucial to finishing this study.

Finally, I owe particular gratitude to my relatives for, their unconditional love and sacrifices, which have laid the foundation for all my achievements. Their belief in me has always been an inspiration. To my siblings, for their patience, understanding, and support. They have all been my rock, providing me with the emotional and moral support needed to overcome obstacles and stay focused on my goals. I also give special thanks to my friends, due to their companionship and encouragement have made this journey enjoyable. Their continuous support, whether through late-night discussions, or study sessions has been a source of motivation and comfort.

Without these remarkable individual contributions and support, this research would not have been possible. Thank you all for being a part of this work.

Table of contents

# LIST OF TABLES

# LIST OF FIGURES

# LIST OF EQUATIONS

# LIST OF ACRONYMS

| | |
|---|---|
| 5G | Five Generation |
| AI | Artificial Intelligent |
| ANN | Artificial Neural Network |
| BP | Back Propagation |
| CNN | Convolutional Neural Network |
| DOCSIS | Data Over Cable Service Interface Specification |
| DSL | Digital Subscriber Line |
| EPON | Ethernet Passive Optical Network |
| FEC | Forward Error Correction |
| FTTH | Fiber-to-the-Home |
| GFP | Generic Framing Procedure |
| GPON | Gigabit Passive Optical Network |
| GRU | Gated Recurrent Unit |
| IoT | Internet of Things |
| IP | Internet Protocol |
| ITU-T | International Telecommunication Union Telecommunication |
| LCAS | Link Capacity Adjustment Scheme |
| LSTM | Long Short-Term Memory |
| MAE | Mean Absolute Error |
| MSE | Mean Squared Error |
| NE | Network Equipment |
| NMS | Network Management System |

| | |
|---|---|
| OADM | Optical Add-Drop Multiplexer |
| ODU | Optical Channel Data Unit |
| OLTs | Optical Line Terminals |
| ONTs | Optical Network Terminals |
| OTDR | Optical Time-Domain Reflectometer |
| OTN | Optical Transport Network |
| OTU | Optical Channel Transport Unit |
| OXC | Optical Cross-Connect |
| PON | Passive Optical Network |
| PTCL | Pakistan Telecommunications |
| QoS | Quality of Service |
| RMSE | Root Mean Squared Error |
| SDH | Synchronous Digital Hierarchy |
| SLA | Service Level Agreement |
| SLP NN | Single-Layer Perceptron Neural Network |
| SONET | Synchronous Optical Network |
| SOON | Self-Optimized Optical Network |
| VCAT | Virtual Concatenation |
| WDM | Wavelength-Division Multiplexing |

# ABSTRACT

*The Optical Transport Network is a critical constituent of modern telecommunication infrastructure, enabling high-capacity data transmission channels over the same fiber. However, the occurrence of faults in OTNs can lead to service disruptions and network downtime, causing significant financial losses for telecommunication operators. Many types of research have dealt with alarm analysis, fault detection, identification, localization, and diagnosis of OTN, while fault management is important, they are a reactive method that finds faults after they have occurred. Therefore, there is an upward demand to develop efficient and accurate fault prediction techniques to proactively manage faults and ensure network reliability. This research attempts to enhance fault management capabilities, minimize downtime, optimize operational efficiency, and deliver enhanced services by utilizing a hybrid deep learning algorithm. The research methodology involves a comprehensive analysis of historical fault data to identify fault patterns and trends. This analysis serves as the foundation for developing a fault prediction mode. By leveraging advanced monitoring techniques and data in the real world, potential faults can be predicted promptly, and proactive measures to prevent service disruptions. This empowers network administrators to proactively address potential issues, allocate resources efficiently, and improve overall network performance. Hybrid Deep learning models CNN-LSTM, CNN-BiLSTM, and CNN-GRU were evaluated for Optical Transport Network fault prediction using MSE, MAE, RMSE, and R2 metrics. The CNN-LSTM model outperformed others with the lowest MSE (1.5203), MAE (2.7686), RMSE (1.2330), and highest R2 (0.9392), making it the most effective. Both CNN-BiLSTM and CNN-GRU models also showed improvements over their standalone versions, with CNN-GRU slightly better in MAE and R2 but not as robust as CNN-LSTM. Generally, the outcomes of this research contribute to the field of fault management in OTNs, providing valuable insights for telecommunication operators to enhance their fault prediction capabilities.*



***Keywords: Optical Transport Network, Fault Prediction, Fault Classification, Hybrid Deep Learning, Telecommunication Infrastructure.***

# CHAPTER ONE

# 1. INTRODUCTION

## 1.1. Background of the Study

(Jiao et al., 2023) OTN is standardized by ITU-T and made up of several optical network components linked together by optical fibers that can manage, multiplex, route, encapsulate, transport, and ensure the survivability of signals. OTN is a critical element of modern telecommunication infrastructure, enabling high-capacity data transmission channels over the same fiber (Antonio et al., 2000). In OTN, wavelength-division multiplexed fiber-optic links connect switching nodes (Optical Cross-Connect, OXC) to provide many, parallel, extremely bandwidth-intensive communication channels over the same fiber (Marín-Tordera et al., 2006). The network facilitates the development of new technologies and services as well as a wide spectrum of companies and government organizations. As a result, fiber optics are the main incentives for new emerging technologies and networks such as smart cities, the IoT, data centers, and 5G (Saif et al., 2020). Optical Transport Network technology is used to provide high-capacity, long-distance transmission of data, voice, and video traffic over optical fibers, and works by transforming optical signals from electrical signals and transmitting them over optical fibers.

The Network of Optical Transport has been implemented as the spine transmission infrastructure to accommodate the increasing need for high-capacity transmission lines brought on by the growth in data traffic, broadband services, and bandwidth-intensive applications. However, the optical network experiences regular failures and unavailability despite the greatest efforts to preserve and protect it from various defects (Bekele, 2022). Large-scale loss of data and disruption of services are the results of optical network faults. For telecom operators to transport the necessary quality of service to their clients and handle the enlarged capacity of traffic on the infrastructure, the optical backbone network's dependability is essential. Fault prediction performs an essential role in improving the operational dependability and availability of optical networks. The foundation of modern telecom networks comprises the components of OTNs. However, OTNs are complex systems that are susceptible to a variety of faults. These faults can be categorized into hard and soft faults. A single failure can cost revenue to service

providers. So, fault management is necessary to guarantee that users receive uninterrupted services (Paul et al., n.d.). While the methods used for optical network protection are unable to prevent problems beforehand, they do take passive precautions. The operator is having trouble reliably and high-quality transmitting various kinds of services due to these flaws. Network service interruptions can also be planned outages, which are frequently arranged ahead of time to carry out specified operations at particular times, for a specific length, and with particular services to be impacted. This makes it possible to carry out specific operations like traffic relocation and system upgrades. Unplanned outages, on the other hand, are caused by malfunctions in the optical transceiver, transceiver module, power supply, hardware, software, fiber bend or cut, and loss of fiber of NE (Nyarko-boateng, 2020).

The availability of directly connected network elements, such as IP backbone routers that use the optical network as a transport medium, and other indirectly connected access network devices is impacted by the availability of backbone optical transmission. Because it can impact their availability and performance, a problem in the optical backbone network can therefore be regarded as a defect in other linked problems. As a result, the QoS is impacted by frequent optical backbone network breakdowns, which also cause the loss of a significant amount of revenue and expose large maintenance costs, causing the network to run less than optimally. Both preventive and reactive management are typically carried out in OTN faults when a network malfunction or a service failure signal sounds on the Network Management System (NMS) (Jaudet et al., 2004).

To overcome the challenges of low proficiency and difficult performance improvement a hybrid deep learning algorithm fault prediction method is proposed. This fault prediction method, which is based on a neural network learning model, is used to extract features from the dataset as much as possible to mine its hidden information. It also predicts the network's future working state to determine whether the network will fail at a specific point in the future. In addition, the operator can schedule labor, time, and resources to guarantee the optical transport network's dependability and services' availability and make the frequency of reported fault alarms and the operator's operating costs, preventive measures can be applied to links and nodes identified as the most frequent sources of network faults through the performance of fault occurrence time prediction in optical networks.

## 1.2. Motivation of the Study

(Budka et al., 2010) OTN network is the essential backbone of modern telecommunications networks to the country's economy and infrastructure. The technology for low latency, high transmission speeds, and massive data capacities are optical networks. As the requirement for data transmission capacity grows, optical transport networks must be able to handle it. The high bandwidth and low latency required to facilitate a variety of applications, including data, video, and audio services. But OTNs are also intricate and prone to a range of errors, which renders them frequently imprecise and ineffective (G. K. Silva et al., 2023). The early prediction and classification of OTN faults are used to minimize service disruptions and maintain network performance. For predicting and classifying optical transport network faults to improve the performance and reliability of networks that are difficult to identify using traditional methods. Traditional methods are often reactive, addressing faults after they occur, and rely on static predefined processes. To solve this problem, we propose the hybrid deep learning technique as a solution to this issue since it can overcome the fault's dynamic character by using the data it is provided to forecast faults in complicated scenarios.

## 1.3. Statement of the Problem

In the present big data era, optical networks are the pillar for high-capacity and long-distance data communication, and failure will cause tremendously serious consequences, such as significant data loss, large-scale computing disruption, and core information transfer blocking (D. Wang et al., 2022). There will be a massive loss of data if the OTN fails due to the increasing volume of traffic passing through it. (Sheetal, 2024) the amount and variety of information and services that optical network technology can transport are always growing, as is the network manner and access method of those services. With the OTN serving as the main infrastructure for upcoming generations, it is becoming more and more responsible for transmitting larger volumes of data. OTN uses several service signals including Ethernet, IP, and SDH all at once to provide transparent, dependable, fast, and efficient transmission. (M. Zhang & Wang, 2019) as the OTN grows in size, over a million alerts in the optical layer occur within only one week, causing significant challenges for network administration, maintenance, and operation. For instance, the network administrator must quickly identify the issue if alerts arise as a result of an exceptionally high bit-error rate (BER) or an extended service delay. But in real networks, a

failure is typically accompanied by a lot of alarms, making it challenging to quickly extract meaningful information from these warnings(M. Zhang & Wang, 2019).

The primary cause of outages and service interruptions from these networks is the optical transport network, which can be attributed to loss of fiber, software problems, amplifier, transponder, transceiver, power, BER, and OSNR are the main. So, the quality of the optical transmission network is being impacted by these vital telecom facilities. In addition to directly costing the telecom operator more money, the faults also raise maintenance costs and lower customer service quality. Due to these, the network has enormous challenges in meeting SLA requirements and delivering high-quality QoS. The optical network's architecture, hardware and software types, optical channels, number of nodes, and linkages are all susceptible to failure.

OTN fault problems are a challenge in the largest telecommunications service provider which operates a nationwide network that provides to millions of customers (Skubic et al., 2015). Generally, the existing studies still have unsolved problems, because the existing literature mainly focuses on reactive fault management, high error prediction, and limited dataset obtained from a single telecommunication (Cui et al., 2019) (B. Zhang et al., 2020) (Bekele, 2022). As a result, telecommunication is facing the challenge of increasing fault rates and the need to improve the efficiency of its fault prediction and classification process. Therefore, to decrease the number of faults, a proactive fault occurrence time prediction mechanism should be in place. This method would assist operational and maintenance staff in promptly preventing and eliminating potential fault causes(Tan & Pan, 2019).

Faults in OTNs can arise from a variety of sources, including physical layer impairments, signal degradation, equipment failures, and environmental factors. These faults may not follow clear patterns, making traditional rule-based or statistical models insufficient. Hybrid DL models, however, are designed to handle the complexities of multivariate time series data and the interactions between various network parameters. OTNs generate vast amounts of data in real-time, such as BER, OSNR, and power levels. These metrics are critical for assessing the health of the network. Hybrid models can effectively process this data through feature engineering and preprocessing steps that identify the most relevant attributes for fault prediction which, are particularly well-suited for the complex, multivariate nature of OTN data, and their application is paving the way for more efficient, automated fault managing in modern telecom networks.

By correctly predicting faults, these models enhance network reliability, reduce downtime, and ensure the continuous availability of telecom services. (Zabin et al., 2023) a hybrid DL model for the prediction and classification of faults in OTN networks can have several benefits, including; improving network performance, reliability, and reducing network maintenance costs. Because this model can learn hidden patterns in the dataset that help to work proactively.

## 1.4. Research Questions

**RQ1**. How can a hybrid deep learning model be developed to accurately predict and classify faults in OTN?

**RQ2**. Which features are the most relevant for predicting and classifying OTN faults?

**RQ3.** How can a hybrid deep learning model be used to improve the efficiency and accuracy of OTN fault prediction?

## 1.5. Objectives of the Study

### 1.5.1. General Objective

Developing a hybrid deep-learning model for OTN fault prediction and classification based on an optical fault dataset is the main goal of this thesis.

### 1.5.2. Specific Objectives

- To collect and clean datasets.
- To prepare the dataset for the hybrid deep learning model (feature selection and engineering).
- To identify the important features for predicting and classifying OTN faults.
- To develop a hybrid DL model for predicting and classifying OTN faults.
- To assess the performance of the hybrid deep learning model on a held-out dataset of the OTN dataset.

## 1.6. Significance of the Study

The significance of prediction and classification of optical transport network faults employing a hybrid deep learning model can help enhance the performance and reliability of the network. Additionally, the prediction and classification of faults in OTN network can have many benefits, including:

- Reduce network maintenance costs: By predicting faults before they occur and recommending preventive maintenance actions.
- To enhance the quality of its services and user satisfaction.
- Improve network performance and reliability.
- Can help to avoid service disruptions.
- Used to identify the root cause of faults. This information can be used to take corrective action to prevent the faults from happening again.
- Can be used to recommend preventive maintenance actions to avoid faults. This can help to minimize, and decrease the number of faults and the cost of maintaining the network.

## 1.7. Scope and Limitations

### 1.7.1. Scope of the Study

The scope of this thesis focuses on developing advanced models that combine deep learning architectures of CNN with LSTM, BiLSTM, and GRU for fault prediction. The study leverages multivariate time series data, recorded at frequent intervals. Through data preprocessing and feature selection, the study aims to increase optical transport networks' fault prediction accuracy. The study also evaluates the models using standard metrics of MAE, MSE, RMSE, and R2, focusing on the enhancement of OTN fault management systems by reducing network downtime through accurate fault prediction. However, this study does not take into account the types of faults that follow fault prediction.

### 1.7.2. Limitations of the Study

Relying on a single vendor like Ericsson for OTN infrastructure presents limitations that could impact fault prediction, network flexibility, and overall performance. This study will only use the Ericsson vendor historical data to predict the upcoming values of the OTN. We evaluate our model only on this network infrastructure dataset generated from two Ericsson SPO 1400 devices in the InRete Lab for only ten hours within five-second time intervals.

## 1.8.Thesis Organization

This thesis contains seven chapters that are organized in the following ways.

**Chapter One**: Deals with the introduction of the OTN. It mentions the problem of the statement, the objective of the study, research questions, motivation of the study, significance, scope, and limitation of this paper.

**Chapter Two:** Provides an introduction to the OTN and covers its generation, characteristics, architecture, layers, and faults in OTN. It also discusses time-series analysis, the OTN Network Management System, the OTN's fault causes and severity levels, and a summary of related work.

**Chapter Three:** In this chapter, we will address the methods and techniques used in this study to design the desired outcome. The methodology including the data source, data analysis, and processing steps, is thoroughly described. We also go over the evaluation metrics that are applied in this research.

**Chapter Four:** This chapter will present the overall model design, discuss the architecture of the suggested model for the study, and deal with hyperparameter optimization.

**Chapter Five:** The experimentation and implementation of the suggested solution, including developing a working implementation environment, model design, and performance evaluation implementation, will be addressed in this chapter.

**Chapter Six:** In chapter six, we compare evaluation metrics and go over the outcomes and results of the suggested architecture as well as an overview of the overall research findings.

**Chapter Seven:** This chapter deals with the discussion about research contribution and makes recommendations for future work of this study.

# CHAPTER TWO

# 2. LITERATURE REVIEW

## 2.1. Overview of Optical Transport Network

OTN is a digital signal transport platform that provides flexible and powerful signal communication capabilities desired to create telecommunication services networks and provide social media, online gaming, online video streaming, and cloud storage of data, which require high-capacity optical transport network to carry the traffic (Mcdermott, n.d.) (J. Chen et al., 2022). OTN is a standardized hierarchical network architecture for transmitting massive amounts of digital information via optical fiber networks. It is designed to provide efficient, reliable, and flexible transport of diverse traffic types, including voice, video, and data (Rahman et al., 2022).

(Antonio et al., 2000) The key components and features of OTN are Optical Channels OTN uses wavelength-division multiplexing (WDM) to transmit multiple optical channels over a single optical fiber. Each channel operates at a specific wavelength and can carry a separate data stream. Digital Wrapper OTN employs a digital wrapper around the client data, which adds overhead information and enables error detection, monitoring, and performance management. The digital wrapper ensures the integrity and reliability of the transmitted data. OTN uses FEC techniques to correct errors that may occur during transmission. FEC codes are added to the digital wrapper to enhance the error resilience of the optical signal. Optical Multiplexing Hierarchy OTN defines a hierarchical structure that allows the aggregation of multiple optical channels into higher-capacity units. It provides different levels of granularity, such as OTU and ODU, to accommodate various traffic demands. Network Management OTN incorporates network management capabilities for monitoring and controlling the optical network. This includes performance monitoring, fault management, provisioning, and restoration mechanisms to ensure efficient operation and troubleshooting. Compatibility OTN is designed to be compatible with various client interfaces, including SONET/SDH, Ethernet, and Fiber Channel. It enables seamless integration of different network technologies and facilitates the transport of different types of traffic over the optical infrastructure. OTN plays a crucial role in modern telecommunications networks by providing a scalable and robust optical transport solution that

can handle high-capacity data transmission and support diverse traffic types. OTN consists of a collection of nodes, or optical network elements, connected by optical fibers. These node elements perform multiplexing, switching, amplification, power management, and service forwarding on optical channels.

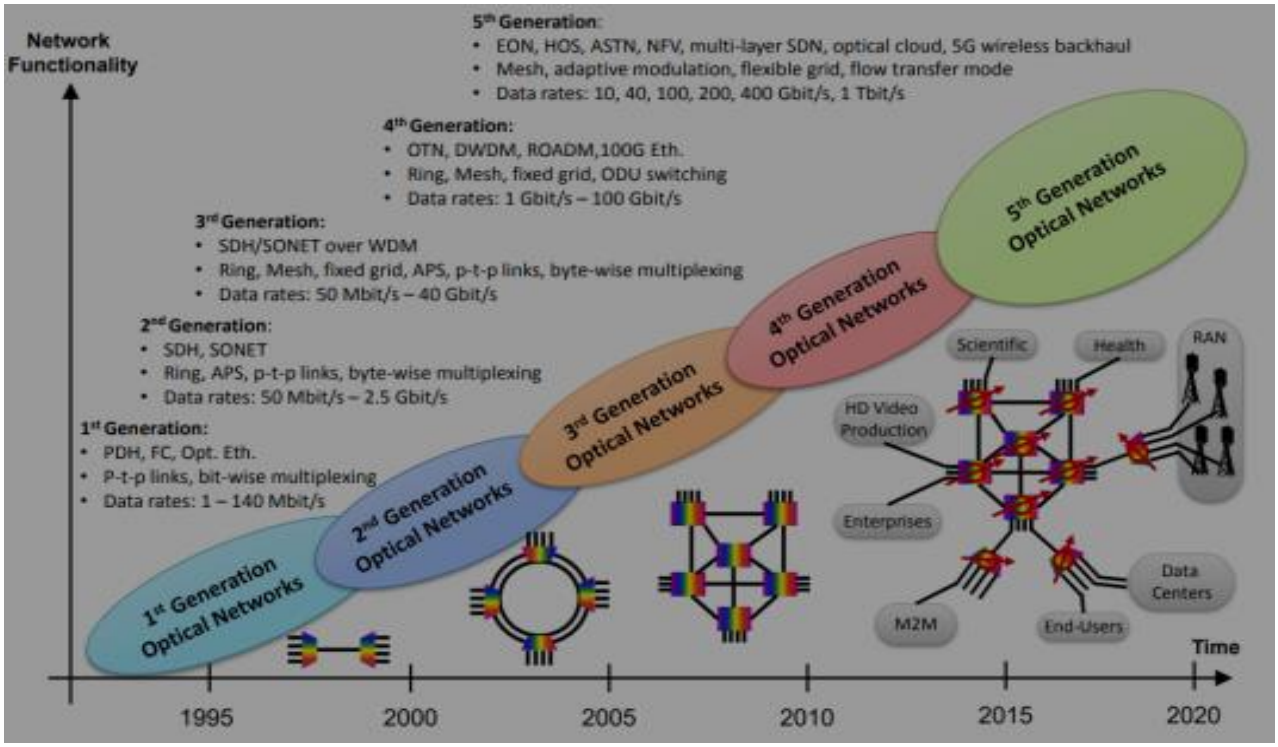


*Figure 2. 1. The generation of an OTN (Aleksic & Member, 2015)*

## 2.2. Architecture of OTN

The OTN architecture is designed to facilitate the efficient and reliable transport of large volumes of data over optical fiber networks. The architecture is based on standardized protocols and layers, ensuring interoperability and compatibility across different vendors' equipment. The architecture of OTN is defined by international standards, with the ITU-T (International Telecommunication Union Telecommunication Standardization Sector) playing an important function in the growth and maintenance of these standards. This ensures that, in a multi-vendor OTN environment, equipment from several vendors can function together without problems. The standardized architecture contributes to the scalability, flexibility, and reliability of optical transport[1].

[1] https://www.ciena.com/insights/what-is/What-is-Optical-Transport-Networking-OTN.html

(Grobe et al., 2018) physical Layer consists of Optical Fiber and Optical Transceivers. Optical Fiber is the physical medium for data transmission, typically using single-mode optical fibers to carry modulated optical signals. Optical transceivers convert electrical signals into optical signals for transmission, and vice versa. These are used at the endpoints of the optical link. This layer has WDM and OADMs. WDM multiple wavelengths (channels) of light are used on a single optical fiber, each carrying independent data streams. This increases the overall capacity of the network. Optical Amplifiers boost the optical signal strength periodically along the fiber to compensate for signal attenuation. OADMs allow the addition or removal of specific wavelengths at intermediate points in the network without affecting others. ODU and FEC are components of the Data Link Layer. ODU is the protocol used at the data link layer in OTN. ODU frames encapsulate client signals, providing error detection and correction. Forward FEC is implemented at the data link layer to enhance the reliability of data transmission by detecting and correcting errors. Network Layer has three sub-layers namely OTN, GFP, VCAT, and LCAS. OTN is the network layer protocol that manages the multiplexing, switching, and routing of optical channels. It provides a hierarchical structure for transporting client signals. GFP encapsulates client signals into OTN frames. VCAT and LCAS techniques are used for efficient bandwidth utilization by dynamically adjusting the capacity of optical channels. Client Layer at this layer there are Client Signal and Client Interface. Client Signals: There are various types of client signals, such as SONET/SDH, Ethernet, and Fiber Channel, which are transported over the OTN. Client Interfaces adapt client signals to the OTN network by providing the necessary mapping and encapsulation.

Management Layer; Management Plane and Control Plane. The Management Plane includes network management functions for configuration, monitoring, and fault detection. The Control Plane is responsible for setting up, maintaining, and tearing down connections within the network (Long et al., 2006).

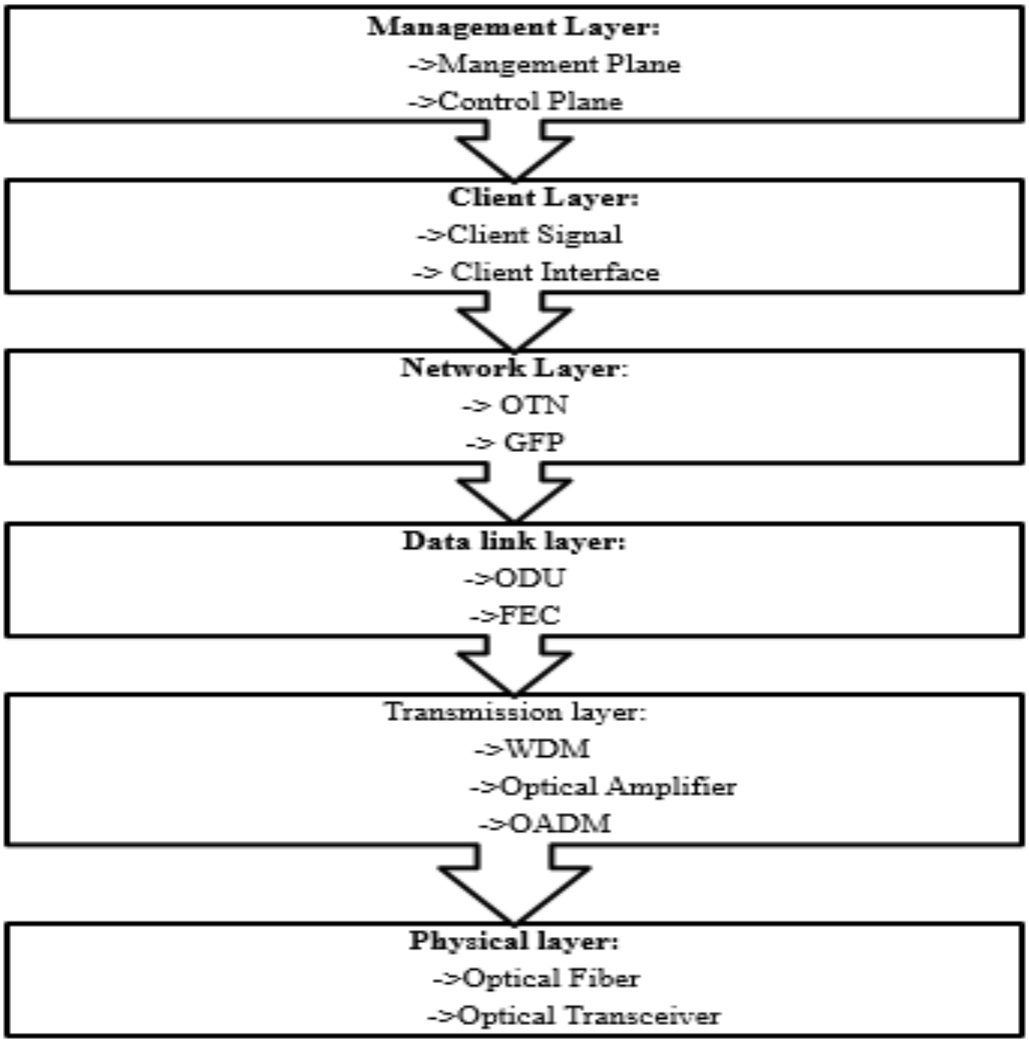


*Figure 2. 2. OTN Architecture*

## 2.3. The Layers of Optical Transport Network

(Paraschis et al., 2020) (P. Wang, 2020) The layers of the optical network, which include the access, aggregation, and core layers, provide a comprehensive structure for reliable and effective data transfer as depicted below in Figure 2.3.

**Access Layer:** The access layer serves as a point of entry for end users and devices, managing connectivity, and initial data transmission. This layer functions as a vital interface that links end users to the optical fiber communication network and is essential in delivering last-mile connectivity. The access layer ensures high-bandwidth and high-speed connectivity for a range of user needs by utilizing a variety of technologies, including cable networks, DOCSIS, DSL,

FTTH, PON, and Wi-Fi. In the access layer, optical line terminals and optical network terminals play crucial roles in controlling connections and guaranteeing smooth data transfer between end-user devices and the optical network. This change is a result of innovations like Wavelength Division Multiplexing, which increases data capacity and provides better dependability in the last-mile connection market. Broadband, phone, and video services are just a few of the many business access options offered by the dependable and efficient Access Layer. With the help of technologies like EPON and GPON, several users can share a single optical cable.

**Aggregation Layer:** The center for traffic management and optimization in optical communication systems is the optical network aggregation layer, which is located between the access and core layers. This layer uses cutting-edge technologies, such as OTN electrical cross-connect technology, which is used to map, multiplex, and cross with ODUk as the granule. It is in charge of aggregating heterogeneous traffic streams from various access points, such as subnetworks and service providers. To increase network deployment flexibility and economy, it is utilized to achieve arbitrary cross-connections between branches of a certain level of N input signals. The aggregation layer serves as a sophisticated middleman, ensuring effective traffic consolidation before sending it to the core layer for additional processing and distribution. It does this by emphasizing fault tolerance mechanisms, scalability to meet growing network demands and seamless service integration. The backbone of the optical infrastructure, the core network, is established with the help of the aggregation layer. This layer is essentially a necessary part that is carefully engineered to satisfy the requirements of contemporary optical networks by coordinating the effective management, aggregation, and transmission of various traffic kinds.

**Core Layer: -** The backbone of a network infrastructure is its core layer, which processes massive volumes of data quickly and efficiently. It uses the OTN to transport data at high speeds and capacities while being engineered for maximum efficiency. Ensuring high levels of network availability and reliability, the Core Layer notable for its resilience incorporates sophisticated optical protection and restoration techniques. Minimizing signal propagation delays for real-time services is the goal of this crucial layer's low-latency architecture. The uniform structure of its framework enables smooth communication between different services, including Ethernet and SONET/SDH, and allows for integrated transport. The Core Layer's scalability allows it to

easily handle the growing demand for network capacity by including more wavelengths or cutting-edge transmission technologies[2].

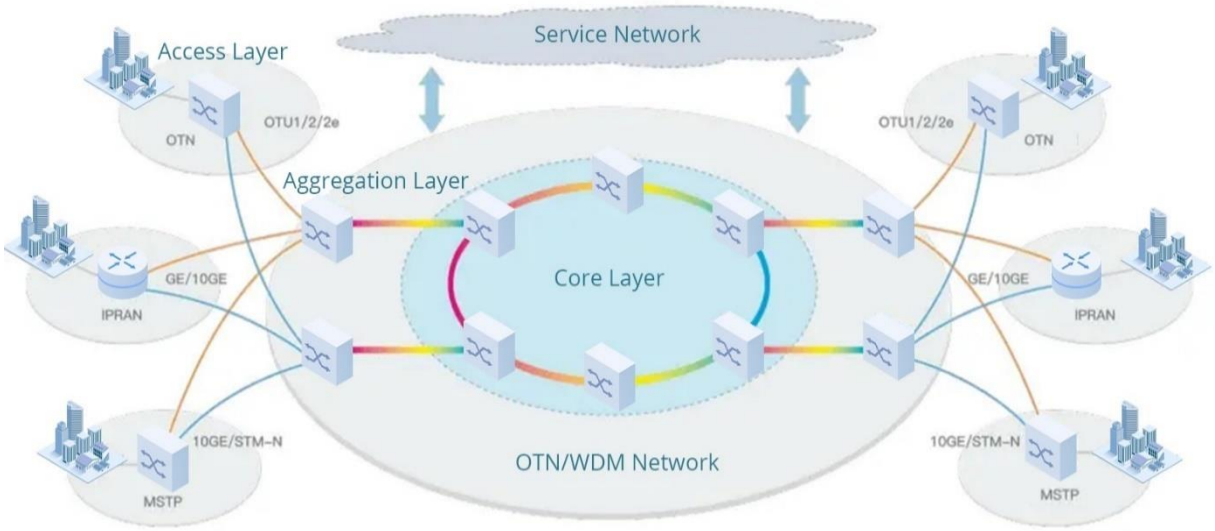


*Figure 2. 3. The Layers of OTN*

Sourced from (https://community.fs.com/article/the-layers-of-optical-transport-network-core-aggregation-and-access-layer.html)

## 2.4. Characteristics of Optical Transport Network

OTN represents a vital infrastructure in modern telecommunications, facilitating high-capacity and efficient data transmission over fiber-optic networks. One main characteristic of OTN is its capability to support numerous types of traffic, including voice, data, and video, by employing wavelength-division multiplexing. With WDM, different light wavelengths can be transmitted simultaneously over a single optical cable, greatly boosting the capacity of the network. Another notable feature is the robustness and reliability of OTN. The architecture incorporates fault tolerance mechanisms, such as automatic protection switching, which ensures seamless communication in the case of a network outage. This high level of reliability is essential for meeting the stringent demands of mission-critical applications and services. Scalability is a defining characteristic of OTN, allowing network operators to easily expand the network's capacity by adding more wavelengths or upgrading equipment.

[2] https://community.fs.com/article/the-layers-of-optical-transport-network-core-aggregation-and-access-layer.html

This scalability is vital as the need for bandwidth continues to grow exponentially with the increasing prevalence of data-intensive applications and services. Moreover, OTN is designed with flexibility, accommodating different data rates and protocols. It provides a standardized framework for encapsulating and transporting diverse types of data, offering a versatile solution that can adapt to the evolving needs of telecommunications networks.

In terms of management and control, OTN incorporates advanced network management protocols and tools, facilitating efficient monitoring, provisioning, and maintenance. This centralized control ensures optimal performance and allows for quick response to changing network conditions(Networks, 2008) (B. Chen et al., 2021) (Destras & Nicolescu, 2023).

Generally, the characteristics of Optical Transport Networks, including their high capacity, reliability, scalability, flexibility, and efficient management, make them a cornerstone in modern telecommunications infrastructure, supporting the ever-increasing demand for fast and reliable data communication.

## 2.5. Faults in Optical Transport Network

Optical transport networks are facing a variety of faults that can disrupt or even disable network services. (D. Wang et al., 2022) faults in an Optical Transport Network can manifest in various forms, posing challenges to the network's resilience and performance. Physical damage, such as fiber cuts due to construction activities or natural disasters, can lead to partial or complete loss of communication on affected fibers, necessitating prompt repair or rerouting. Equipment failures, ranging from malfunctioning transceivers to issues in amplifiers or add-drop multiplexers, can disrupt services, resulting in potential network downtime. Power outages, whether due to grid failures or other reasons, impact critical components. Wavelength interference on the same fiber can degrade signal quality, causing disruptions. Bit errors and signal degradation may result from factors like dispersion or attenuation, compromising data integrity and network performance. Multiplexing or demultiplexing errors and network congestion can lead to incorrect signal routing and reduced performance, respectively.

(Patell et al., 2001) Security breaches pose risks of data compromise, network disruptions, and integrity loss. Environmental factors like extreme temperatures or humidity may cause equipment damage or malfunctions. There are several reasons why optical transceiver fault

errors in the optical transport network occur, including fiber bends or cuts, power outages, board malfunctions, excessive loss, attenuation, incorrect parameter settings, and amplifier malfunctions. Furthermore, the amount of network elements, optical channels, transponders, optical transceiver modules, and all other hardware and software components of the optical transport network strongly correlates with the fault occurrence rate. Overheating, excessive optical power input, and other factors can result in component-level faults, including those in the optical transceiver module. Circuit breakers, rectifiers, and other power system malfunctions can give rise to problems at the network element level. Interconnected backbone layer devices, such as Internet Protocol routers, that depend on Optical Transport Networks to transfer services, are also susceptible to faults. Faults in the network could be caused by other access layer devices connected to the OTN. Additionally, configuration errors can result in incorrect routing, service disruptions, or degraded performance. Addressing these faults requires a comprehensive fault management system, including proactive monitoring, rapid fault detection, and efficient resolution strategies to maintain the OTN's reliability and optimal functionality.

Telecom operators employ a reactive method to maintain and recover the services back to their normal circumstances. This approach does not stop the error from occurring. The reactive method uses identification, localization, and fault detection to recover from faults. Therefore, a proactive method to predict that a fault is going to happen in a certain amount of time and help to take actions to prevent the fault occurrence is crucial. The Network Monitoring System stores optical performance and fault occurrence time information. This information is kept as a time series of events on the NMS. Therefore, employing fault prediction time series forecasting models could be a useful way to prevent malfunction through planned preventive maintenance.

### 2.5.1. Fault Severity Level in OTN

An essential component of network management is fault severity levels, which offer a methodical manner to categorize and rank problems that have been found inside a network. Fault severity levels are useful in the context of different network infrastructures, such as OTN, as they enable businesses to assess the relevance and urgency of individual issues. Administrators are guided in resource allocation and timely incident response by these severity levels, which are often classified as Critical, Major, Minor, and Informational (warning alarms).

Critical errors are serious problems that cause large service interruptions and need to be fixed right away. Major errors signal serious problems that need to be fixed right away. Even while minor errors are less serious, they still need to be fixed to keep the network operating at its best. Informational errors don't affect services directly; instead, they give information for observation or inquiry. The categories of fault severity are determined by the alarm notification of a particular fault or anomalous state that arises within the network. A quick notice indicating the occurrence of a malfunction appears on the monitoring system's screen (Systems, 2019) (Djomadji et al., 2023)

## 2.6. Fault Management in OTN

(D. Wang et al., 2022) Fault management in an OTN is a critical aspect of network operations, ensuring the rapid detection, isolation, and resolution of issues to maintain optimal performance and reliability. A robust fault management system involves continuous monitoring of the network infrastructure, utilizing tools and protocols to detect deviations from normal behavior. Alarms and alerts are configured to notify administrators of critical faults, allowing for swift responses to minimize downtime. The fault management system distinguishes between various severity levels, such as critical, major, minor, and informational, allowing for prioritized responses based on the impact and urgency of identified issues. Proactive measures help anticipate and address potential faults before they lead to network disruptions. Generally, the fault management system contributes to ongoing improvement and the overall resilience of the OTN.

## 2.7. Application of OTN Fault Forecasting

Optical Transport Network fault prediction is a critical aspect of network management in modern telecommunications systems. By utilizing advanced algorithms and predictive analytics, OTN fault prediction aims to anticipate potential network failures before they occur, allowing for proactive maintenance and minimizing service disruptions. One key application of OTN fault prediction is in network reliability enhancement. By examining historical data and real-time network performance metrics, predictive models can recognize trends or anomalies that may show an impending fault. By taking a proactive approach, network administrators can avoid downtime and guarantee that end users receive uninterrupted service by rerouting traffic or carrying out maintenance (Manias et al., 2023).

Another important application of OTN fault prediction is resource optimization. By accurately predicting potential faults, network resources can be allocated more efficiently to address the identified issues before they escalate into major problems. This optimization not only improves network performance but also decreases operational costs by diminishing the need for reactive maintenance and emergency repairs. Additionally, OTN fault forecasting has a vital role in service quality assurance by helping operators maintain high levels of service availability and reliability(Li, 2022). By leveraging predictive analytics and machine learning techniques, network operators can stay ahead of potential issues, optimize resource utilization, maintain service quality, and bolster network security.

### 2.7.1. Machine Learning for OTN Fault Forecasting

[3] Artificial intelligence includes machine learning as a subfield that allows computers to automatically learn from their knowledge and improve it without the need for explicit programming. [4] The creation of computer programs that can access data and use it to further their education is the main focus. To seek patterns in data and make better results in the future based on the examples we offer, learning starts with observations or data, such as examples, immediate experience, or instruction. The main goal is to empower computers to learn on their own, with independent assistance or supervision from humans, and adjust their behavior accordingly.

(Szostak et al., 2021) In the field of failure prediction for Optical Transport Networks, machine learning has become a potent tool. OTNs are essential to the current telecom infrastructure because they enable dependable and fast data transfer across optical fibers. OTN defects may cause service interruptions that affect users and companies. Telecom businesses can increase network performance and reliability by using ML techniques to anticipate and avoid OTN problems.

[3] https://www.javatpoint.com/machine-learning-algorithms

[4] https://www.javatpoint.com/machine-learning

(Z. Wang et al., 2017) (Du et al., 2021) Large-scale historical data can be analyzed by machine learning models to find patterns and trends that might point to an upcoming OTN problem. Telecom providers can avert network breakdowns by being proactive with this predictive capability. Telecom firms can plan maintenance efforts during off-peak hours or before they impact vital services by anticipating failures in advance. This proactive strategy reduces downtime and ensures that customers will always receive service. Reducing operating costs related to reactive maintenance methods can be achieved through machine learning-enabled predictive maintenance. (Du et al., 2021) telecom operators can maximize resource allocation and prevent expensive emergency repairs by addressing possible problems before they become more serious. Ensuring dependable OTN performance is imperative in providing consumers with superior services. By leveraging machine learning for fault forecasting, telecom providers can enhance network stability and customer satisfaction by minimizing service disruptions. Several machine learning methods are available for application with optical transport networks. These include supervised, unsupervised, and reinforcement.

### 2.7.2. Deep Learning for OTN Fault Forecasting

Deep Learning is a subclass of machine learning that utilizes neural networks with multiple layers to automatically learn and model intricate patterns in large datasets, commonly used for tasks including predictive analytics, natural language processing, and image recognition. These algorithms are designed to learn from and make predictions or decisions based on data[5].

Artificial neural networks (ANNs) are the building blocks of deep learning. These networks are composed of layers of connected nodes (neurons) that analyze data. Each node receives input, processes it using an activation function, and passes the output to the next layer of nodes[6].

(Qiu et al., 2023) DL models are "deep" because they have many layers, allowing them to learn intricate patterns in data. A model can be trained to identify patterns, categorize items, make predictions, or produce new material by varying the weights (connections) between nodes. One of the key benefits of deep learning is its ability to automatically extract features from raw data eliminating the need for manual feature extraction.

[5] https://www.ibm.com/topics/deep-learning

[6] https://aws.amazon.com/what-is/deep-learning/

This makes deep learning mainly effective for tasks such as image and speech recognition, natural language processing, and many other complex problems.

(P. Wang, 2020) Deep learning has developed as a hopeful approach for fault forecasting in Optical Transport Networks. Its capability to learn complex patterns from large datasets makes it well-suited for analyzing the enormous amount of performance monitoring data produced by these networks. Ensuring the reliability and availability of OTN networks is essential for providing uninterrupted services to end-users. One approach to enhancing network reliability is through fault forecasting, which involves predicting potential faults before they occur. Deep learning, a subclass of machine learning, has arisen as a promising technique for OTN fault forecasting. Models of deep learning are capable of learning complex patterns and dependencies in large datasets, making them suitable for fault forecasting in OTN networks. These models can analyze various parameters, such as optical power, signal-to-noise ratio, and bit error rate, to predict potential faults (Huang et al., 2021). The most commonly used deep learning techniques for OTN fault forecasting include:

**Convolutional Neural Networks:** (Rafidison et al., 2023) are a type of artificial neural network that is mostly utilized for auto-correlated data, image processing, segmentation, and classification. It is intended to automatically and adaptively learn the important hierarchies of features from tasks with grid-like structures, such as an image. The biological visual cortex serves as its source of inspiration. Include fully connected layers, pooling layers, and convolutional layers, three distinct types of layers. An arrangement of these levels creates a network architecture.

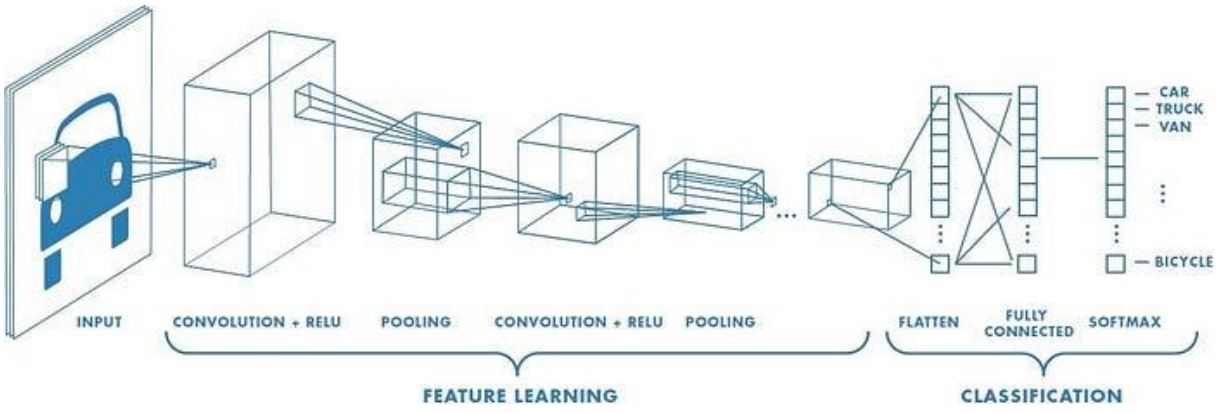


*Figure 2. 4. CNN Architecture*

Sourced from (https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53)

**Recurrent Neural Network:** (Hewamalage et al., 2021) With the ability to process sequential data, RNNs have become a potent family of artificial neural networks. RNNs have seen tremendous advancements over time, resulting in a variety of designs and applications that have revolutionized many industries, including banking, healthcare, and natural language processing. This offers a thorough introduction to recurrent neural networks, exploring their structures, training methods, difficulties, and a wide range of applications. When processing sequential or time-series data, a specific neural network called a recurrent neural network is used. It has a feedback connection, and at each time step, the updated input and the output are sent back into the network. The neural network may recall previous data while processing an output thanks to the feedback connection. Recurrent neural networks are another name for this architecture since the processing involved can be described as a recurrent process.

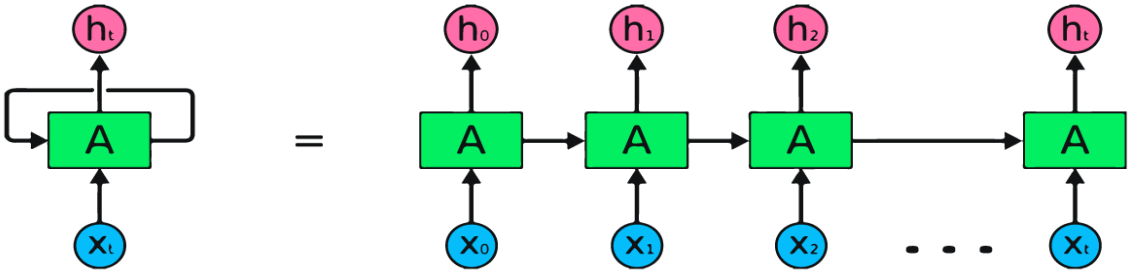


*Figure 2. 5. RNN Architecture (Le et al., 2019)*

**Long Short-Term Memory:** (Tian et al., 2024) is a type of RNN architecture that is designed to overcome the vanishing gradient problem encountered in traditional RNNs. LSTM is particularly effective in tasks involving time series data, natural language processing, and speech recognition. LSTM networks consist of memory cells, input gates, output gates, and forget gates. These components work together to process and remember relevant information over long periods. (Le et al., 2019) during the forward pass, the LSTM network processes input data sequentially. The current input, the previous hidden state, and the memory cell state are used to calculate the input gate, forget gate, and output gate at each time step. The memory cell state is updated using the input gate, forget gate, and the new input. Finally, the output gate controls the information to be passed to the next layer or the final output.

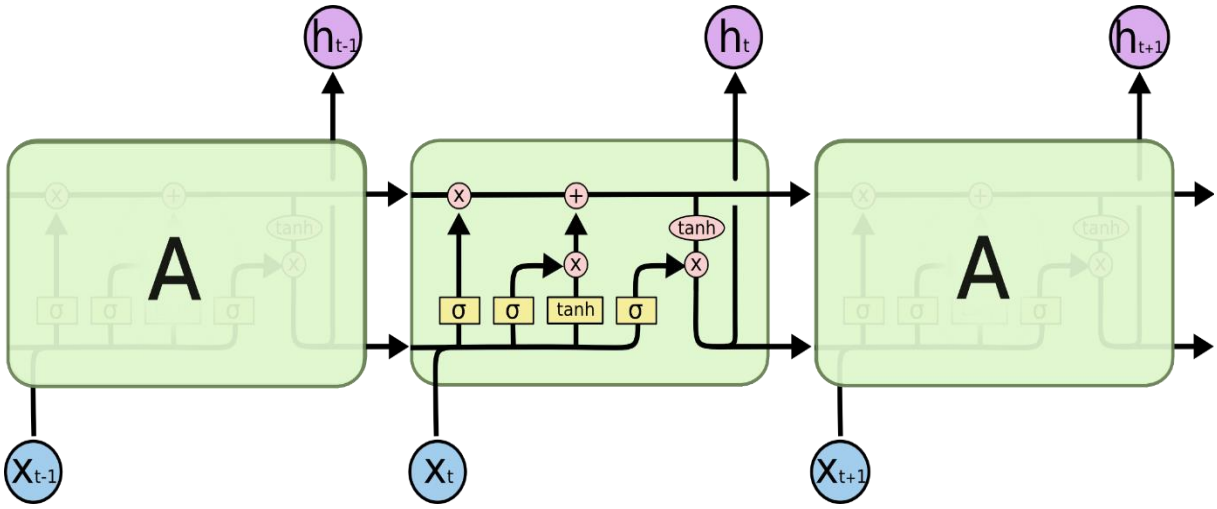


*Figure 2. 6. LSTM Architecture*[8]

**Auto Encodes:** (Molefe & Tapamo, 2023) A particular kind of feedforward neural network called an autoencoder uses the same input as its output. After compressing the input into a lower-dimensional code, they use this representation to recreate the output. The code is a condensed "compression" or "summary" of the input; it is also known as the latent-space representation. An autoencoder consists of three components: the encoder, the code, and the decoder. The encoder creates a code in addition to compressing the input. Then, the decoder uses this code to reconstruct the input.

[8] https://colah.github.io/posts/2015-08-Understanding-LSTMs/

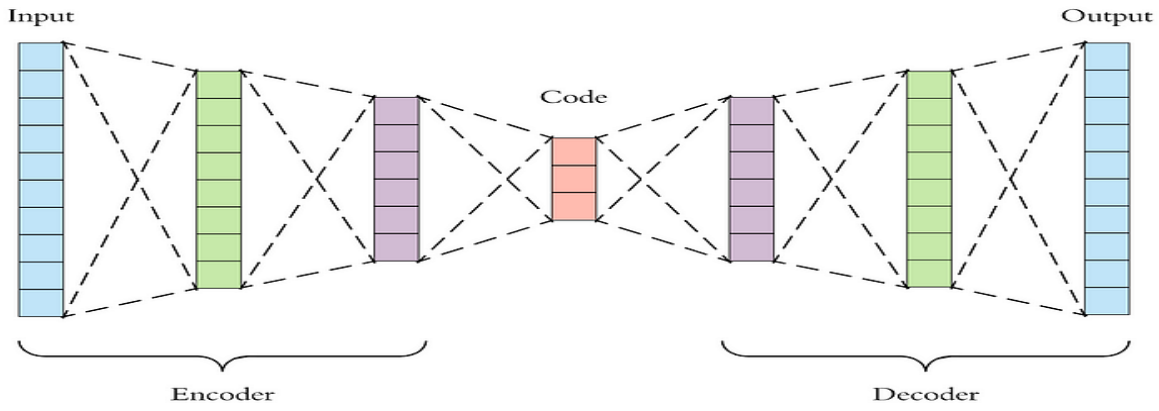


*Figure 2. 7. AutoEncoder*[9]

**Multi-Layer Perception:** This is a kind of artificial neural network made up of several node layers coupled to one another. There are no cycles in the connections between the nodes since it is a feedforward neural network. MLPs are frequently utilized in many different applications, including regression, pattern recognition, and classification[10]. Encompasses an input layer, one or more hidden layers, and an output layer, among other three sorts of layers. The original data is received by the input layer, processed by the hidden layers using activation functions and weighted connections, and finally, output by the output layer yields the desired outcome. An MLP node generates an output by performing a weighted sum of its inputs, adding a bias term, and then applying an activation function[11]. MLPs frequently employ the activation functions sigmoid, tanh, softmax, and ReLU.

### 2.7.3. Hybrid Deep Learning Model for OTN

(Akpatsa et al., 2021) A hybrid approach refers to the combination of two or more different models or techniques to improve the accuracy and robustness of the fault prediction system. The increasing complexity and scale of OTNs pose challenges in identifying and addressing potential faults before they lead to service disruptions.

[9] https://towardsdatascience.com/applied-deep-learning-part-3-autoencoders-1c083af4d798

[10] https://medium.com/codex/introduction-to-how-an-multilayer-perceptron-works-but-without-complicated-math-a423979897ac

[11] https://www.shiksha.com/online-courses/articles/understanding-multilayer-perceptron-mlp-neural-networks/

Here is an overview of a hybrid approach for the prediction and classification of OTN faults. Leveraging deep learning models, such as CNNs, BiLSTMs, GRUs, and LSTMs enhances the hybrid approach's capacity to handle complex and dynamic fault patterns. Deep learning excels in learning hierarchical features and capturing intricate relationships within datasets, which is beneficial for OTN fault prediction.

## 2.8. Time Series Prediction and Analysis

Time series prediction and analysis involve examining data points collected or recorded at specific time intervals to identify trends, patterns, and underlying structures. These are typically sequential, making it essential to account for temporal dependencies in the data. One particular method of examining a set of data points gathered over a while is time series analysis. Time series analyzers record data points at regular intervals over a specified period, as opposed to only sometimes or arbitrarily capturing them. However, this form of analysis is more than just collecting data over time[12]. The capability to show how variables change over time through analysis differentiates time series data from other types of data. Stated differently, time is a significant variable since it reveals how the data changes throughout the data points and the end outcomes. It offers a different information source as well as a predetermined hierarchy of data relationships[13]. For time series analysis to be reliable and consistent, a lot of data points are usually needed.

Having a large dataset ensures that the analysis can effectively filter out erratic data, making the sample size more representative. It also helps ensure that any patterns or trends identified are not simply anomalies but are robust enough to account for seasonal variations. Additionally, time series data is often used for forecasting, which involves predicting future values based on historical data. Time series analysis is applied across numerous sectors, such as finance, economics, weather prediction, and signal processing. Accurate forecasting enables organizations to make more informed decisions and better prepare for upcoming changes, ultimately leading to enhanced outcomes and greater success (Ahmed et al., 2022; Zhu et al., 2020; Du et al., 2021).

[12] https://blog.quantinsti.com/time-series-analysis/

[13] https://www.tableau.com/learn/articles/time-series-forecasting

## 2.9. Related Work

(Z. Wang et al., 2017)The paper presents a novel approach to predicting equipment failures in optical networks using machine learning techniques, specifically SVM and DES. The paper begins by addressing the challenges faced by optical networks, particularly the significant data loss that occurs during failures. Traditional protection algorithms, such as shared-path protection and shared risk link group failure protection, are reactive and only mitigate damage after a failure has occurred. The authors argue for the necessity of proactive measures that can predict and prevent failures before they happen, thereby minimizing data loss and service disruption.

The authors identify key indicators that correlate with board failures. Through analysis, they determined that the laser bias current is the most significant predictor, followed by environmental temperature. This step involves sorting indicators based on their relationship with failure and selecting them for further analysis. Using the selected indicators, the authors build an SVM model to diagnose potential failures. They employ ten-fold cross-validation to assess the model's accuracy, ensuring that the model generalizes well to unseen data. The authors utilize DES to predict future values of the selected indicators based on historical data. This is crucial for maintaining an updated prediction curve that reflects the latest monitoring data. Finally, the SVM model is used to predict whether equipment will fail based on the predicted values of the indicators. The authors conducted experiments using real data collected from a wavelength-division multiplexing network operated by a telecommunications provider. They focused on balancing the dataset by screening out redundant normal boards and concentrating on those that were more prone to failure. This resulted in a dataset of 320 boards over a 44-day observation period, totaling 14,080 data items.

The results demonstrated that the proposed method achieved an impressive average prediction accuracy of 95%. The authors conclude that their proactive approach to failure prediction using SVM and DES significantly enhances the ability to manage risks in optical networks. By predicting failures before they occur, network operators can

(Panayiotou et al., 2018) This document presents an innovative approach to fault localization in optical networks utilizing machine learning techniques. The proposed approach is based on a path correlation procedure and a Gaussian process classifier that generates a failure possibility

for each suspect link. The classifier is trained on historical data to model and predict failure probability. The approach attains a high localization accuracy and significantly reduces network costs compared to conventional fault localization methods. It also discusses related work, the dataset generation procedure, a fault localization approach established for comparison purposes, and the performance evaluation results. The proposed approach could significantly improve network reliability and decrease maintenance expenses. However, further research is needed to validate the approach on a larger scale and under different network conditions.

(Tan & Pan, 2019) this author discusses the CNN-LSTM model for fault prediction in network systems. It explains the elements of the prediction model as well as the steps involved in data preprocessing. Sequence prediction is performed with the LSTM model, and feature extraction is done with the CNN model. The document also mentions the use of time windows and compares the performance of diverse models based on the length of the time window. Time windows are used to divide the input sequence into input and label windows. Further, the performance of the model is compared based on the length of the time window.

(M. Zhang & Wang, 2019) this document proposes a scheme for combined alarm analysis and failure prediction in optical transport networks using machine learning and deep learning algorithms. The approach uses a combination of time series segmentation, time sliding window, K-means, back propagation neural network, and modified apriori with weights for alarm analysis. A support vector machine or long short-term memory is used for device failure prediction. The proposed scheme is evaluated using actual data from provincial spines of telecommunications workers. The document also discusses the challenges faced by network administrators when dealing with alarms in optical transport networks and how the proposed scheme addresses these challenges. Overall, the scheme aims to prevent data loss and make network operation, administration, and maintenance more efficient.

(T. Liu et al., 2019) This author explores the use of artificial neural networks in fault localization for optical transport networks. The paper introduces the thought of F-measure and measures the positioning effect using location time, mean square error (MSE), and F-measure. The experiment demonstrates that while both LSTM and BP neural networks are capable of accurately locating problems in optical transport networks, LSTM performs better overall. Compared to the BP neural network, the LSTM localization time is shorter, and after 45

iterations, the F1-score of the LSTM can achieve 0.96156689296, meeting the fault location requirements for accuracy and real-time. The paper also discusses the practical applications and prospects of introducing neural networks in fault localization for optical transport networks. The evaluation indexes used in the experiment include MSE, location time, and F1 score.

(B. Zhang et al., 2020) Proposes a novel alarm prediction method based on cross-layer artificial intelligence to reduce the burden on the controller during alarm prediction in self-optimized optical networks. The proposed method extends the self-optimized optical network system functions by deploying device AI in the data plane. The cross-layer AI can perform task decomposition and multi-party cooperation to improve processing and system management efficiency. The proposed method was evaluated using equipment data collected from a commercial SDH network, and the experimental results demonstrate that the alarm prediction method has high precision. The paper concludes by discussing future work and open issues, such as algorithm selection in various scenarios and the determination of superparameters in model training.

(Nyarko-boateng, 2020) The paper proposes a machine-learning approach to predict the genuine location of a fiber cable fault in an underground optical transmission link. The authors use linear regression in the Python sci-kit learn library to predict the actual location of a fault in an underground optical network. The employed MAE evaluation matrix and mean square error yielded good accuracy results of 8.0143 and 6.1291, respectively. The result obtained in this paper shows that faults in underground optical networks can be found quickly to avoid delays in the fault tracing process, which leads to an excessive revenue loss.

(Bekele, 2022) The study focuses on the use of machine learning algorithms for predicting faults in optical transport networks, with a specific focus on the case of Ethio Telecom. The study presents an overview of optical transport networks, fault reasons, and fault severity levels along with presenting the research's problem formulation and system model, it also covers a variety of machine learning algorithms appropriate for failure prediction, data collection methods, data preparation strategies, time series ideas, and metrics for measuring model performance. The author concludes with the presentation of the experiment's details, results, and analysis. The performance evaluation metrics described in the study are used to assess the created models, and a comparison of the models is conducted to determine which model best

captures the fault interarrival time. The study finds that the developed models can predict faults in optical transport networks with high accuracy, and the availability of the general optical transport network increases. In summary, the study provides valuable insights into predicting faults in optical transport networks using machine learning algorithms. The study's findings can be useful for telecommunication companies and network engineers in predicting faults and improving the availability of optical transport networks. According to the model prediction performances, the ANN model has predicted the fault with 16.08% MAPE while GRU shows 16.23% MAPE, and the LSTM model has 17.93% MAPE on average.

(P. Liu et al., 2023) This research article presents a mechanism that can quickly locate and respond to faulty links for optical networks in high-density interconnection scenarios of data centers. The technology is designed to promptly identify and address malfunctioning links in high-density data center interconnection environments. The mechanism combines a customized AI module with the optical time-domain reflectometer device and the optical power monitoring module to make full use of the data from optical links. The author shows that the average fault detection efficiency of the link is improved. The document includes a detailed description of the system architecture, the AI model used in the platform, and the practical application and performance analysis of the platform. The article concludes that the mechanism can greatly improve the efficiency of failure finding and location in data centers. In general, the summary of related work is concise as shown in Table 2.1.

*Table 2. 1. Summary of Related Work*

| Author, Publication year, and title | Methods and algorithms | Contribution | Limitation |
|---|---|---|---|
| (Z. Wang et al., 2017) Failure prediction using machine learning and time series in optical network | Involves DES and SVM for predicting equipment failures in optical networks. | With an impressive average prediction accuracy of 95%, the paper approach significantly enhances the stability of optical networks and safeguards services from potential disruptions. | The focus on SVM and DES may limit the exploration of other potentially more effective machine-learning algorithms that could enhance prediction accuracy and robustness. |

| | | | |
|---|---|---|---|
| (Panayiotou et al., 2018) Leveraging statistical machine learning to address failure localization in optical networks | An innovative approach to fault localization in optical networks using ML techniques. | The proposed approach attains a high localization accuracy and significantly reduces network costs compared to conventional fault localization methods. | Tested on a limited dataset, created for 10,000 failure incidents injected into the network. Attempt to localize single-link failure and attain localization accuracy of 91%. |
| (M. Zhang & Wang, 2019)Machine Learning-Based Alarm Analysis and Failure Forecast in Optical Network | Alarm prediction method based on cross-layer artificial intelligence (AI) to reduce the burden on the controller during alarm prediction in self-optimized optical networks. | The document makes several contributions to the field of optical transport networks like device failure prediction, and methods for alarm analysis, and addresses the challenges faced by network administrators in dealing with alarms in optical transport networks, providing a potential solution to the difficulties of prompt failure location and alarm compression and analysis. | It does not provide a comparative analysis of the proposed scheme against existing methods or approaches for alarm analysis and failure prediction in optical transport networks, which could limit the understanding of its novelty and effectiveness. The document lacks detailed information on the specific. Performance metrics, validation methodologies, and potential limitations of the evaluation process could impact the robustness of the findings. |

| | | | |
|---|---|---|---|
| (T. Liu et al., 2019) Application of Neural Network in Fault Location of Optical Transport Network | LSTM and BP neural networks | The document discusses the practical applications and prospects of introducing neural networks in fault localization for optical transport networks, highlighting the potential for real-time fault localization with high accuracy. | The fault localization models do not cover a wide range of diverse network configurations and fault scenarios, potentially limiting the robustness of the findings. |
| (B. Zhang et al., 2020) Optical Fiber Technology Alarm classification prediction based on cross-layer artificial intelligence interaction in SOON | The novel method is proposed in self-optimized optical networks | Provides valuable insights into the potential of cross-layer AI architectures for improving alarm prediction accuracy and system efficiency in optical networks. | The document does not provide a detailed comparison of the proposed method with existing alarm prediction methods in optical networks. |
| (Nyarko-boateng, 2020) Predicting the actual location of faults in underground optical networks using linear regression | The machine learning (SLP NN) technique was applied in a simple linear regression predictive model to predict the precise distance of fault in an | The approach uses linear regression to predict the actual location of faults in underground optical networks. Providing evidence that the proposed approach can help overcome the problems involved in tracing the actual spot of fault in underground optical networks, leading to reduced delays and loss of revenue. | Limited dataset obtained from a single telecommunication. Not be representative of all underground optical networks. The achieved accuracy results are 6.1291 for MSE and 8.0143 for MAE. The specific details of the dataset and its size are not mentioned. |

| | | | |
|---|---|---|---|
| | underground fiber cable cut. | | |
| (Bekele, 2022) Fault Prediction in OTN using Machine Learning Algorithms - Case of Ethio Telecom | ANN, LSTM, and GRU algorithms. | Improve the understanding of optical transport networks and machine learning algorithms' application in predicting faults and improving the availability of optical transport networks. | The study uses a limited dataset, which does not represent the entire faults in optical transport networks with only one vendor optical network and high error prediction is another limitation. |
| (M. F. Silva et al., 2022) Learning Long- and Short-Term Temporal Patterns for ML-Driven Fault Management in Optical Communication Networks | RNN, LSTM, and LSTNet algorithms | Experimental results in the lab demonstrate that the LSTNet diminishes mean squared errors for unseen data by 95%, suggesting robustness and applicability in real-world scenarios. | The model misclassifies some failure samples as normal conditions, leading to prediction inaccuracies. No feature engineering and selection methods are applied. |
| (Djomadji et al., 2023)AI-Assisted Failure Location Platform for Optical Network | AI module with OTDR. | Locate potential faults in optical networks within data centers. Improve fault detection and response times. Leveraging data analytics for network maintenance and troubleshooting. | Predict the normal state as the failure state or high error prediction. |

| | | | |
|---|---|---|---|
| (Sheetal, 2024) Fault Identification in Optical Transport Network Using CNN | utilizing the CNN model for fault detection in OTN. | By effectively identifying and classifying faults, the paper contributes to improved fault localization strategies in OTN, which is critical for maintaining network reliability and performance. | The researcher uses only one evaluation metric (F1-score) and only one algorithm (CNN) for fault identification which makes it hard to generalize the model. |

Although identifying a fault is useful, it typically addresses the issue only after the fault has already occurred, which can lead to negative consequences as the system has already been affected. This reactive approach often results in operational downtime, increased costs, and potential loss of data or service quality. According to many journals, actions are taken post-failure, which may be too late to prevent significant damage or service degradation.

Moreover, a significant limitation of current research in OTN fault detection is the narrow scope of most studies. These works often focus on identifying specific types of faults in isolation, without accounting for the complexity of real-world scenarios where multiple fault types can occur simultaneously or evolve. This slight focus presents a challenge for applying these methods in operational environments where dynamic and multifaceted fault conditions are the norm. A real-world setting is far more complex, often exhibiting diverse fault types that may interact with one another, making single-fault prediction methods inadequate for robust fault management in practice.

Additionally, many existing fault prediction models struggle with accuracy, leaving room for significant improvement. Faults in OTN are inherently complex and can exhibit diverse behaviors, making them challenging to predict accurately using traditional approaches. The complexity of these scenarios demands advanced methods that can dynamically adjust to changing conditions and better model the fault behavior.

To address these problems, we propose a Hybrid Deep Learning approach that combines the advantages of multiple models to effectively capture the complex and dynamic behavior of OTN faults. The hybrid model incorporates different deep learning techniques to handle various aspects of fault prediction and classification more efficiently. By training the model with

sufficient data, including historical fault data, real-time monitoring information, and simulated scenarios, the hybrid model is better equipped to predict fault.

This hybrid method is particularly beneficial in the OTN environment, where fault behavior can shift over time and the nature of faults can evolve. By capturing these evolving patterns, the model can better generalize to unseen data, reducing the likelihood of failure to predict faults accurately. This improves both the accuracy and the timeliness of fault prediction, allowing for more effective preventive measures, minimizing service disruption, and ensuring the reliability of the network.

# CHAPTER THREE

# 3. RESEARCH METHODOLOGY

## 3.1. Overview of the Chapter

OTNs are high-speed telecommunications networks that use optical fiber for communication. These networks are designed to transport an enormous amount of data over vast distances in an efficient manner. However, OTN faults can lead to significant service disruptions and outages, affecting numerous users and businesses that rely on these networks for communication and data transfer. These disruptions can result in lost revenue and decreased productivity. Faults in optical transport networks can result in reduced network performance and quality, including decreased data transfer rates, increased latency, and higher error rates. This can negatively impact various applications, such as video conferencing, real-time gaming, and cloud services. Detecting and resolving faults in optical transport networks can be time-consuming and costly, requiring specialized equipment and trained personnel. This can lead to increased maintenance and repair costs for network operators and service providers. So, providing proactive communication, swift resolution of issues, and transparency in addressing faults can help mitigate the negative effects on customer relationships.

To predict future network behavior and categorize distinct forms of network traffic, prediction and classification in OTN fault require the application of a variety of techniques and methodologies. This is essential for guaranteeing quality of service, maximizing network performance, and identifying abnormalities or possible problems.

This chapter's following sections will cover the research design, data collection techniques, data preprocessing and feature engineering tasks carried out, model selection, dataset overview, design and development resources (hardware and software), and evaluation metrics used to meet the research objectives.

## 3.2. Methodology of the Research

Methodology refers to the process of collecting, analyzing, and selecting appropriate methods for evaluating data to achieve objectives and address research questions. It involves the systematic approach to interpreting data to fulfill the goals of a study. It is the scientific study

of methodically conducting research. Hybrid deep learning methodologies have developed as a promising approach for fault prediction and classification in optical transport networks due to their capability to combine the strengths of deep learning models. This combination allows for improved fault prediction accuracy, reduced computational cost, and enhanced interpretability of the results. To achieve our research objectives the methodology involves the following steps:

- First, understanding the area of the research.
- Second, review different sources related to the selected area; articles, journals, books, and sites.
- Third, the issue in the designated area has been identified and needs to be resolved.
- Fourth, after determining the necessary parameters for the solution model, we designed the suggested solution.
- Fifth, we will have to implement through implementation tools to get the result of the proposed solution.
- Finally, the performance of the suggested approach must be assessed using our metrics about the current methodologies, and the results must be analyzed.

The research process described above is illustrated in the figure 3.1 below.

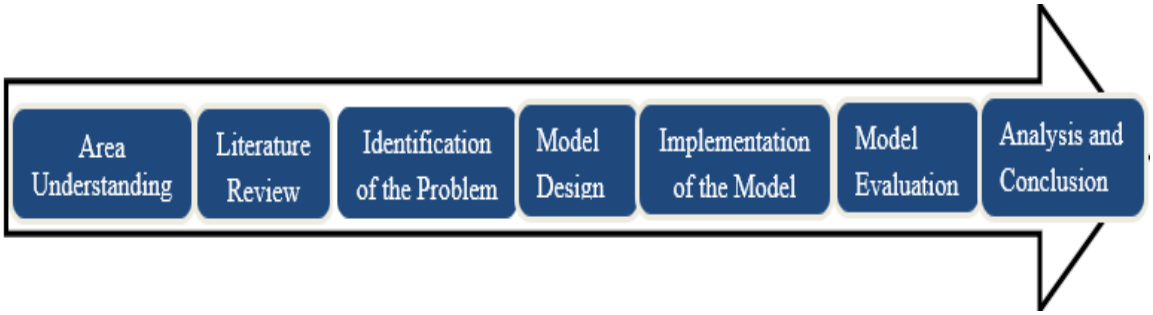


*Figure 3. 1. Flow of Research Design*

### 3.2.1. Research Area Understanding

Research Area Understanding is a broad impression that refers to the ability to comprehend and analyze a specific research area. It involves the identification of research gaps, the analysis of existing literature, and the development of new research questions. The goal of Research Area Understanding is to gain a comprehensive thoughtful of a research area and to identify

new research directions that can contribute to the advancement of knowledge. As a result, we identified network failure management and understood it for our research by studying various network study areas.

### 3.2.2. Literature Review

It aims to identify the research gaps, the challenges, and the future directions of research in this area. Review various works such as journal articles, conference papers, and research papers to achieve the research objective and gather important information for research. To accomplish the study goal and gather relevant information for research, review a variety of publications, including journal articles, conference papers, and research papers.

### 3.2.3. Understanding the Problem Domain

Problems are identified through the review of multiple research articles. Numerous recent publications have discussed this issue.

### 3.2.4. Design Research

The third step in our research is to create a solution to the problems that have been identified. This design is based on an optimized algorithm that is considered the most efficient method of fault prediction. We will use hybrid deep learning to design the solution, which allows computational models composed of multiple processing layers to learn data representations with multiple levels of abstraction. A Hybrid deep learning architecture is a type of machine learning model that combines two or more deep learning models to achieve better performance and accuracy.

### 3.2.5. Methods to Select Model

In recent years, the focus on time series forecasting research has grown significantly. Deep neural networks have demonstrated exceptional accuracy and effectiveness across various application domains, making them one of the most widely used machine learning techniques for addressing big data challenges. These factors make them one of the most popular machine-learning techniques used today to address big data-related issues (Torres et al., 2021). Based on the characteristics of the data and the nature of the problem (sequential/time-series data), models like LSTM, BiLSTM, GRU, CNN-LSTM, CNN-BiLSTM, and CNN-GRU are selected because they are designed to capture temporal dependencies and complex patterns in time-series

data. The hybrid approach typically combines CNNs with LSTM, GRU, or BiLSTM. CNNs help in extracting local features and patterns from the input data, while LSTM, BiLSTM, and GRU capture temporal dependencies, making the combination effective for fault prediction in OTN. The models would be evaluated based on evaluation metrics such as MSE, MAE, RMSE, and $R^2$. Systematically searching through a range of hyperparameters (e.g., number of layers, units per layer, learning rate, batch size, and epoch) to identify the optimal configuration for each model. The model that achieves the best performance on these metrics could be chosen for OTN fault prediction.

Figure 3.2 workflow outlines a systematic approach for predicting OTN faults using hybrid deep learning models. The process begins with the collection of raw optical fault datasets, which are then preprocessed through data preparation and feature selection/engineering to ensure that the data is clean, normalized, and rich in predictive features. Various deep learning models, including GRU, LSTM, BiLSTM, CNN-LSTM, CNN-BiLSTM, and CNN-GRU, are then built to capture both temporal dependencies and complex patterns within the data. These models are subsequently trained using the preprocessed data, and their performance is tested on unseen data to evaluate their generalization ability.

Following the testing phase, the models are evaluated using metrics such as MSE, MAE, RMSE, and $R^2$ to determine their predictive accuracy. If the models do not meet the desired performance criteria, hyperparameter tuning is conducted to optimize model parameters. This step may involve retraining the models as needed. Once the models have been fine-tuned, their performances are compared to identify the best-performing model. The process concludes with the selection of the most effective model for OTN fault prediction, ensuring that the model chosen is both accurate and robust for deployment in real-world network environments.

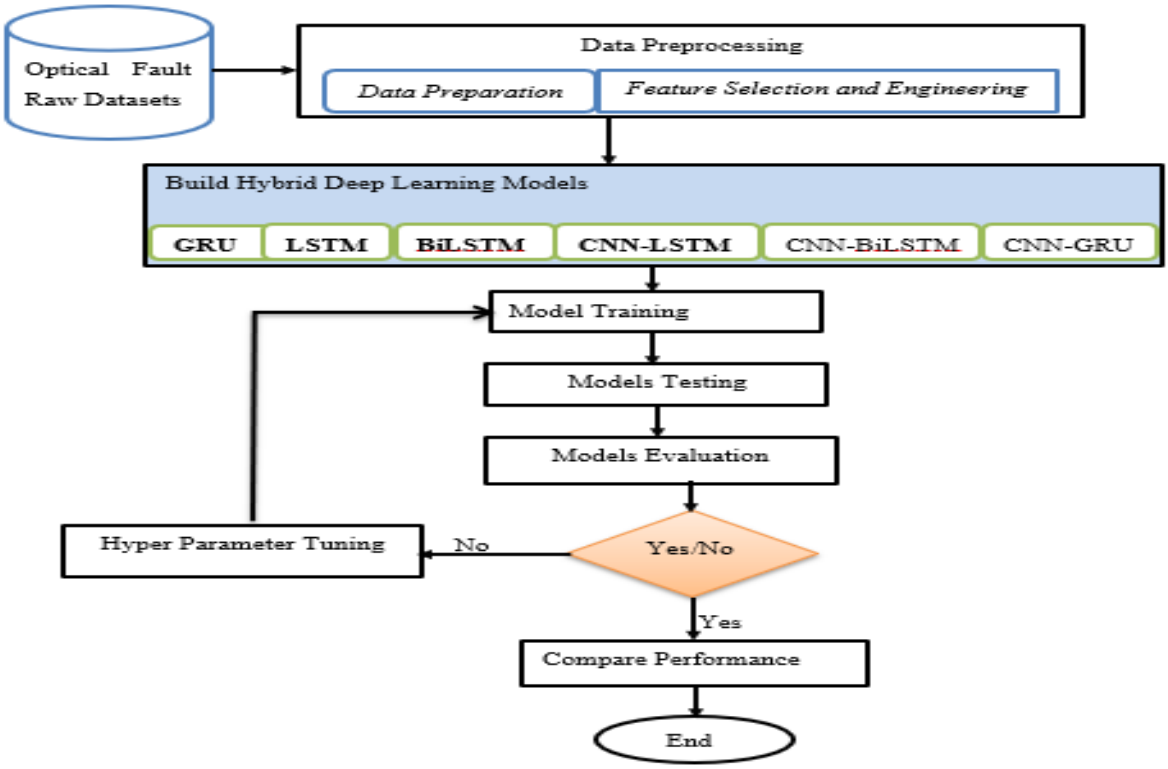


*Figure 3. 2. Flow of Experimental Design*

## 3.3. Dataset Source and Description

[14] The dataset was created at the TeCIP Institute, Scuola Superiore Sant'Anna, utilizing the ARNO testbed in the InRete Lab. Real-world optical network datasets can be found in the optical-failure-dataset repository on GitHub. Periodic variations and soft/hard failures have been emulated in this dataset. SPO1 and SPO2, two Ericsson SPO 1400 devices, make up the testbed. A 100Gb/s Optical Transport Network muxponder fitted in slot 18 is a feature of every device. The muxponder has ten tributary ports and is connected to port 11 of a DWDM optical line. (GitHub - Network-And-Services/optical-failure-dataset: Datasets of a real optical network, in which periodic fluctuations and soft/hard failures have been emulated). Under normal operating settings, coherent metrics (such as BER and OSNR) are continuously gathered from the optical testbed, where periodic failures are emulated.

[14] https://github.com/Network-And-Services/optical-failure-dataset

Amplifiers (Ampli1, Ampli2, Ampli3, and Ampli4) with a total length of 240 km (each measuring 80 km) are part of the testbed configuration and are managed by SPO devices. Every amplifier is set up in constant gain mode, which enables 0dBm of optical power to enter each span. The reverse link presents a 10dB attenuator and is configured in a back-to-back fashion from SPO2 to SPO1.

*Table 3. 1. Description of Dataset*

| No | Feature | Data type | Description |
|---|---|---|---|
| 1 | Timestamp | Int64 | Recorded time value. |
| 2 | Type | Object | Indicate whether infrastructure or devices. |
| 3 | SPO_ID | Object | Identifier of the optical port on the SPO. |
| 4 | BER | Float64 | Quantifies the accuracy of data transmission over a communication channel. |
| 5 | OSNR | Float64 | Represent the ratio of the optical signal to the noise power within a specific bandwidth. |
| 6 | Input Power | Float64 | The input signal power at the transmitter. |
| 7 | Output Power | Float64 | The output signal power at the receiver. |
| 8 | Failure | Float64 | The outcome. |

## 3.4. Preparation Data

Data preparation is a crucial step in building accurate and effective machine-learning models. The objective is to clean, transform, and organize the raw data in a way that makes it suitable for training and testing models. This process involves transforming raw data into a final, refined dataset that is suitable for use as input in machine learning models. Therefore, before using the data as input for the prediction models it must go through a preprocessing phase (Webb, 2017)(Abidin et al., 2020). Here are some data preprocessing we have used.

**Data cleaning:** For time series data involves identifying and handling various issues such as missing values, outliers, inconsistent timestamps, and duplicate records. The goal is to ensure the data is accurate, complete, and suitable for analysis or modeling. It is important to handle missing values appropriately to avoid bias in analysis or modeling. Common approaches

include filling missing values using approaches like forward fill, backward fill, interpolation, or using statistical techniques like mean, median, or regression imputation. The problem we faced in this phase was missing values. These missing values can be handled by the linear interpolation method.

**Data integration:** Data integration for time series data involves combining data from several sources or datasets into a unified, consistent, and coherent dataset. This is necessary in our case because our data is collected individually as hard and soft data datasets. Therefore, the data of different datasets need to be integrated to train a model.

**Data Transformation:** It plays a crucial role in preparing and analyzing time series data, which involves applying mathematical or statistical operations to modify the data in a way that improves its characteristics or facilitates specific analysis techniques. This process may involve tasks such as scaling, normalizing, and converting data types, such as changing the timestamp feature from an object to a datetime format. In our case, we have already converted the timestamp feature's data type from object to datetime format. In this step, there are three techniques.

- **Min-Max Normalization.**
- **Z-square Normalization.**
- **Min-Absolute Normalization.**

We have used Min-Max normalization techniques for our research case.

**Handling imbalance:** An imbalance in a dataset is when the number of instances of a particular class in classification is significantly higher than the number of instances of the other classes. As a result of an imbalanced dataset, a biased model during training can result in poor prediction performance, particularly for minority classes. A variety of techniques are used to balance datasets, including under-sampling, over-sampling, SMOTE, and ensemble. In our case, we have used two techniques such as over-sampling and SMOTE to find out the best for our dataset.

### 3.4.1. Selection of Features and Engineering

(Al Nuaimi & Masud, 2017) Feature engineering refers to the process of creating new features or extracting valuable insights from existing ones in a dataset to enhance the performance of machine learning models. This can involve techniques such as scaling, normalization, encoding

categorical variables, generating interaction terms, and deriving new features from the data. In contrast, feature selection focuses on choosing a subset of the most relevant features from a dataset to optimize the performance of the machine learning model. This can involve techniques such as feature scaling, normalization, encoding categorical variables, creating interaction terms, and extracting features from data. Feature selection, on the other hand, is the process of selecting a subset of the most relevant features from a dataset to use in a machine learning model. Feature selection helps reduce the dimensionality of the dataset, enhance the interpretability of the model, and lower the risk of overfitting. By focusing on the most relevant features, it can lead to more efficient and accurate models.

**Feature Selection:** (Abdulwahab et al., 2022) There are three general classes of feature selection algorithms: These are Filter methods, wrapper methods, and embedded methods.

**i. Filter Methods**

These techniques are usually applied during the pre-processing phase to prepare the data for optimal model performance. These techniques choose characteristics from the dataset without utilizing any machine learning algorithms. While these methods are effective at removing redundant, correlated, and duplicate features, they cannot address multicollinearity. Despite their high computational efficiency and low cost, they do not solve issues related to multicollinearity. The evaluation of each feature is done separately, which can be beneficial in certain situations when features are independent of one another but ineffective when a combination of features can enhance the model's overall performance. Among the methods employed are:

**Information Gain**: It measures the reduction in entropy values and is defined as the amount of information a feature provides for identifying the target value. When calculating the information gained from each attribute, the goal values for feature selection are considered.

**Chi-square test:** is typically employed to examine the connection between categorical variables. It does this by comparing the observed values of the dataset's many properties to their predicted values.

**Correlation Coefficient:** a metric with values ranging from -1 to 1 that quantifies the relationship's direction and the degree of association between the two continuous variables.

**Variance Threshold:** This approach involves removing features whose variance falls below a specified threshold. By default, it eliminates features with zero variance. The underlying assumption is that features with higher variance are more likely to contain valuable information.

### ii. Wrapper methods

Greedy algorithms are wrapper approaches that use a subcategory of features to train the algorithm iteratively. Features are added and removed based on the findings drawn from training the model beforehand. Typically, stopping criteria are pre-established during model training and include things like the model's performance declining or reaching a certain number of features. The main advantage of wrapper techniques over filter techniques is that they provide the optimal feature set for model training, resulting in higher accuracy. Some commonly used methods include:

**Forward selection:** is an iterative process in which we start with a blank set of features and, after each iteration, continue to add features until we find one that best enhances our model. The model will keep running until adding a new variable does not result in an improvement in the model's performance.

**Backward elimination:** is another iterative method in which all features are originally included, and the least important item is eliminated after each iteration. The stopping criterion is to keep going until the removal of the feature results in no discernible increase in the model's performance.

**Bi-directional elimination:** This approach concurrently uses backward elimination and forward selection methods to reach a single, distinct solution.

**Recursive elimination:** This method selects features by progressively narrowing down the feature set through recursive elimination. The estimator is initially trained on a full set of features, and their relevance is assessed using the feature importance attribute. The least significant features are then removed from the current set, repeating the process until only the desired number of features remains.

### iii. Embedded methods

The feature selection algorithm is included in the learning process in embedded techniques, giving it built-in feature selection capabilities. The disadvantages of filter and wrapper

techniques are met by embedded methods, which combine both of their advantages. These methods are as fast as filter methods, yet more accurate, and they also account for combinations of features. Some of these methods may include:

**Regularization:** This method applies a regularization penalty to the model's parameters to mitigate overfitting. Specifically, it employs Lasso regularization, which penalizes the magnitude of the coefficients. As a result, some coefficients are reduced to zero, indicating that the corresponding features can be excluded from the dataset due to their minimal contribution.

**Tree-based methods:** Methods like Random Forest and Gradient Boosting offer feature importance as a technique for feature selection. This approach highlights which features play a more significant role in influencing the target variable. By assessing feature importance, we can identify the variables that have the greatest impact on the model's predictions. In general, feature selection in machine learning has the following important usage.

- To reduce the dimensionality of feature space.
- To speed up a learning algorithm.
- To enhance the predictive accuracy of a classification algorithm.
- To enhance the clarity and understanding of the learning outcomes.

**Feature Engineering:** Feature engineering is an essential phase in developing machine learning models. It includes creating new features or transforming current ones to boost the model's effectiveness. This process plays a key role in improving model performance by refining the input data. The main goal of feature engineering is to maximize the predictive power of the model by providing it with relevant and informative input data. A model can produce more precise and accurate forecasts the more it comprehends the dynamics and relationship between the instances or observations. As a result, new features can be created to provide the model with more data and insight, enhancing its functionality. According to (Mulu et al., 2024) the two main categories of features that time-series problems create from the original features are the following.

1. **Time-based Features**

Time-based features rely on the time aspect of the data and help identify temporal patterns and trends within the time series. These features can reveal underlying behaviors linked to time progression in the dataset. For instance:

- **Date/Time features.**
- **Moving averages.**

2. **Statistical Features**

Trends and patterns that are not directly tied to time can be identified using statistical features, which rely on the statistical properties of the data.

- **Rolling mean.**
- **Rolling standard deviation.**
- **Rolling minimum and maximum.**

**Splitting Data and Framing into a Supervised Learning Problem:** (Joseph & Vakayil, 2022) It is common to divide a dataset into training and testing portions to develop machine learning models. During the training phase, the model's parameters are estimated or adjusted to fit the data. Following this, the model's accuracy is assessed using a separate testing dataset. This approach prevents overfitting, which occurs when the model learns the training data too well and performs poorly on new, unseen data, leading to inaccurate predictions for future events. Consequently, one way to protect against unforeseen problems brought on by overfitting is to withhold some of the dataset and test the model's performance before putting it into use in real-world scenarios. Holding back a portion of the training set for validation is another frequent practice.

(Šafránková et al., 2010) the validation set can be used to select hyper-parameters or regularization parameters in the model, for example, to optimize the model's performance. In fact, the model may be trained via cross-validation by splitting the training set up into different sets. By repeatedly using the procedure on the training set, our suggested approach for optimally dividing the dataset into training and testing can likewise be applied to these objectives.

The split datasets should be adequate for both training and testing, while there isn't a defined labeled dataset ratio of training to testing. Inadequate datasets impact the training and testing procedures. But the most common ratio for separating data is 70% to 30% or 80% to 20%. In our case, we have used 20% for testing and 80% for training in this study.

## 3.5. Selecting the Model

Model selection involves choosing the most suitable model from a range of candidates for a specific problem or dataset. This process includes evaluating and comparing the performance of various models to determine which one best aligns with the data and meets the desired goals.

The goal of model selection is to find a model that strikes a balance between underfitting (having high bias and poor predictive performance) and overfitting (exhibiting minimal bias but inadequate generalization to new data). Underfitting happens when the model is too easy to capture the underlying patterns and relationships in the data, while overfitting occurs when the model becomes too complex and starts to fit the noise or random fluctuations in the training data. Model selection involves a combination of empirical evaluation, statistical analysis, and understanding of the problem domain. It is an iterative process that may involve trying different algorithms, adjusting hyperparameters, and evaluating the performance until the most suitable model is identified. Based on the above criteria, we identified the following candidate models and evaluated each of them as we have mentioned in chapter two.

**Gated Recurrent Unit:** allows the network to selectively update or retain information from previous time steps, enabling it to capture long-term dependencies in the input sequence which can also address the vanishing gradient problem. Compared to LSTM, GRU has a simpler architecture and fewer gates, which leads to faster training and better results on smaller datasets.

**Long-short Term Memory:** is a type of recurrent neural network that can understand and forecast long-term dependencies between sequential inputs. Recurrent Neural Networks (RNNs) are composed of repeating units with simple architectures, such as a single tanh layer, which limits their ability to retain long-term memory. To overcome the challenges of vanishing gradients and short-term memory, Long Short-Term Memory (LSTM) networks introduce a more sophisticated memory structure with gates. The LSTM's architecture, consisting of four interrelated layers, enables it to maintain short-term memories over extended periods effectively.

**Bidirectional LSTM (BiLSTM): BiLSTM:** Bidirectional LSTM is a variant of the Long Short-Term Memory (LSTM) architecture designed to capture information from both past and future contexts when handling sequential data. It integrates two LSTMs. One processes the input sequence in the forward direction, while the other processes it in the backward direction. This approach allows the model to leverage information from both directions, enhancing its ability to understand and utilize context from both the past and the future. It can efficiently capture temporal patterns and long-term dependencies, which makes it appropriate for fault prediction in historical datasets.

**CNN-LSTM:** A CNN-LSTM hybrid model is a neural network architecture that combines the strengths of CNNs and LSTM networks. It is commonly used for processing and analyzing sequential data that also has temporal dependencies. The CNN-LSTM model uses CNNs to mine historical data for pertinent spatial features, which are then sent to LSTM layers for temporal pattern recognition and prediction.

**CNN-BiLSTM**: is a neural network architecture that combines the strengths of both CNNs and BiLSTM networks. This hybrid model is used for processing and analyzing sequential data, such as text or speech, where both local and contextual information is important. To collect features from historical data and feed them into BiLSTM layers, it uses CNNs to extract spatial features. This allows it to capture temporal dependencies both forward and backward.

**CNN-GRU:** is a hybrid model architecture that combines the power of CNNs and GRUs to address sequential and spatial data analysis tasks. A CNN-GRU model leverages the strengths of both CNNs and GRUs. CNNs are effective in capturing spatial patterns and extracting relevant features from input data. On the other hand, GRUs are a type of RNN that excel in sequential data analysis by capturing temporal dependencies and long-range dependencies within the input. GRUs have gating mechanisms that control the flow of information within the network, enabling them to selectively remember or forget information. By combining CNNs and GRUs, a CNN-GRU model can simultaneously process spatial and sequential information in a unified architecture.

## 3.6. Performance Analysis Evaluation Metrics

A variety of metrics are frequently employed to assess a model's performance. Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), Mean Squared Error (MSE), and coefficient of determination, or R-squared (R2) are the metrics most frequently employed for time series prediction models. A prediction error, or error ($\boldsymbol{ei}$), is the difference between the actual and the expected value. This difference relates to the element of the matching observation that is unanticipated at that particular moment ($i$). The residual between the actual observation and the model-predicted value is another name for the error.

$$e_i = y_i - \hat{y}_{i,} \quad \textit{Equation 3. 1. Prediction Error}$$

### 3.6.1. Prediction Analysis Evaluation

**MAE:** is a metric used to measure the average magnitude of errors in a regression or forecasting model. It calculates the average absolute difference between the predicted values and the true values, indicating how well the model's predictions align with the actual outcomes. This is how it looks mathematically:

$MAE = (1/n) * \Sigma|y - \hat{y}|$, *Equation 3. 2. Mean Absolute Error (MAE*

, where:

- n is the number of data points or observations.
- Σ (sigma) denotes the sum.
- y represents the true values or targets.
- ŷ represents the predicted values.

**MAPE:** calculates the mean of the absolute percentage errors between predicted and actual values. It provides a measure of prediction accuracy as a percentage, where a lower MAPE score signifies a more accurate prediction.

$MAPE = (1/n) * \Sigma (|(y - \hat{y})/y|) * 100$, *Equation 3. 3. Mean Absolute Percentage Error*

**MSE:** measures the average squared difference between the predicted values and the true values. It quantifies the average magnitude of errors while giving more weight to larger errors compared to the Mean Absolute Error (MAE). This makes MSE particularly sensitive to outliers.

$MSE = (1/n) * \Sigma (y - \hat{y})^2$, *Equation 3. 4. Mean Square Error (MSE)*

**RMSE:** is a measure of the average magnitude of errors between the predicted values and the true values, expressed in the same units as the target variable.

$RMSE = sqrt((1/n) * \Sigma (y - \hat{y})^2)$, *Equation 3. 5. Root Mean Square Error (RMSE)*

**$R^2$:** is a statistical measure that represents the proportion of the variance in the dependent variable explained by the independent variables (predictors) in a regression model. It indicates how well the model fits the data. $R^2$ values range from 0 to 1. An $R^2$ value of 0 means the model does not explain any of the variance in the dependent variable. An $R^2$ value of 1 means the model perfectly explains the variance. A high $R^2$ value, close to 1, suggests that a large proportion of the variance in the dependent variable is accounted for by the model, indicating a

good fit. Conversely, a low $R^2$ value, close to 0, implies that the model does not explain much of the variance, indicating a poor fit (Vaid & Ghose, 2020).

## 3.7. Development Tools

### 3.7.1. Software Resources Used

- OS: - Microsoft Windows 10.
- Programming Language: - Python.
- Mendeley Desktop: used for reference management.
- Word processing Software: include Microsoft Word and PowerPoint to write and format our thesis document and presentation.

### 3.7.2. Hardware Resources Used

- Processor: - Intel(R) Core (TM) i5-4200M CPU @ 2.50GHz  2.50 GHz.
- Random Access Memory (RAM): - 8.00GB.
- System Type: - 64-bit operating system, x64-based processor.
- Hard Disk: - 500GB.

# CHAPTER FOUR

# 4. ARCHITECTURE OF THE PROPOSED SOLUTION

## 4.1. Introduction

As we have mentioned about the problem statement, the research methodology, the gap in the literature that we identified, and the literature review. We have also raised a few research questions. This chapter will provide a detailed description of the architecture of the solution we suggested to close the aforementioned gap and address the research objectives posed. We will also discuss the methods and processes used, such as the model selection, and provide an overview of the OTN fault prediction architecture.

## 4.2. Proposed Model Architecture for OTN Fault Prediction

Our suggested method predicts possible faults or irregularities before they happen based on the OTN historical data. Hybrid deep learning models are commonly used across various domains due to their capability to leverage the strengths of different neural network architectures. By combining multiple types of models, they can address diverse challenges more effectively and enhance overall performance. When it comes to OTN fault prediction, a hybrid model can effectively capture complex patterns and dependencies in the data. A potential hybrid deep learning architecture for OTN fault prediction could consist of different deep learning neural networks as shown in the previous chapter of Figure 3.2.

A multivariate time-series forecast is carried out at this stage. Data integration is used to merge the two datasets into a single data frame after each dataset has been loaded separately as a hard and soft dataset. Next, additional preparation operations are carried out, including data balancing, scaling, data type conversion, data transformation, and handling missing values. Feature engineering tasks, such as generating new time-based features and rolling window statistics, are carried out. The sliding window technique is then employed to convert the data into a supervised learning format. Finally, the preprocessed data is used to train the hybrid deep learning model to estimate future values for OTN.

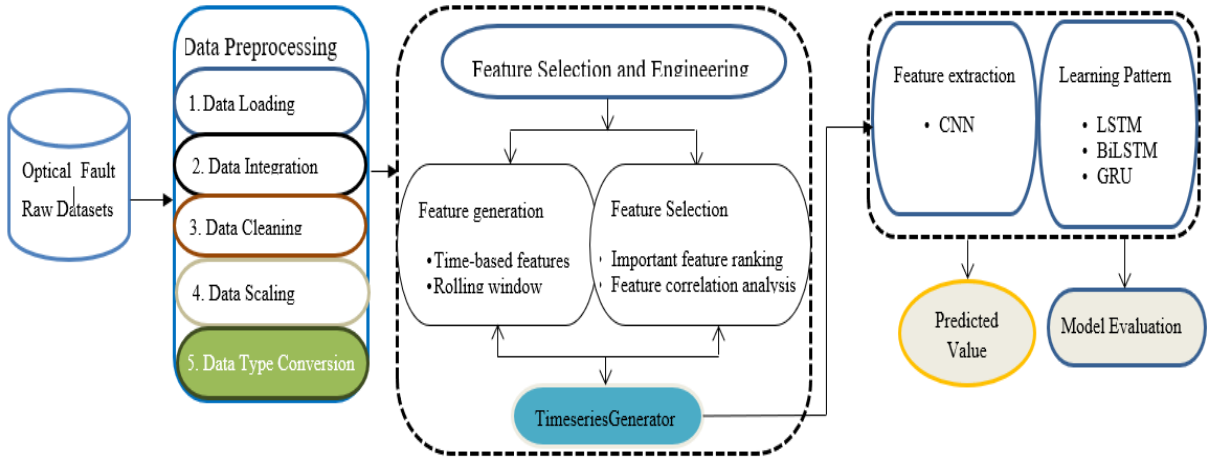


*Figure 4. 1. The Proposed Solution Architecture for OTN Fault Prediction*

Figure 4.1 shows a comprehensive workflow for processing optical fault raw datasets and applying hybrid DL techniques to predict and evaluate outcomes. The process begins with data preprocessing, where raw datasets are loaded and prepared for analysis. This involves several steps; loading, integrating, cleaning, and converting data types into appropriate formats. Once the data is preprocessed, the next phase is important feature selection and engineering.

The final step focuses on learning and model evaluation, where deep learning models are applied to extract meaningful patterns and make predictions. A CNN is used for feature extraction, while different learning models LSTM, BiLSTM, and GRU are employed to learn from the data. The models output a predicted value, which is then assessed in the model evaluation stage to determine the accuracy and effectiveness of the predictions.

### 4.2.1. Data Preprocessing Stage

Data preprocessing is an important step in the data analysis process. It involves taking raw data, which can be messy, incomplete, and inconsistent, and transforming it into a format that can be understood and analyzed by machine learning algorithms. The goal of data preprocessing is to clean, organize, and prepare the data so that it can be effectively used for analysis. During the data preprocessing phase, various techniques are applied to ensure that the data is of high quality and suitable for further analysis. These techniques include data cleaning, which involves correcting errors, handling missing values, and removing irrelevant or noisy data. Proper data preprocessing is especially critical in machine learning as it can influence the performance and

accuracy of models trained on the data (Chernykh et al., 2021; Maharana et al., 2022). Generally, it involves a series of steps aimed at enhancing the quality and reliability of the data.

In our research, data are multivariate time series. A multivariate time series is a type of data that consists of multiple variables or features observed over time. Unlike a univariate time series, which only tracks a single variable, a multivariate time series records the values of several related at each time point.

Time-series data refers to data collected over time at regular intervals. This type of data exhibits temporal dependence, meaning that observations are interdependent and influenced by previous observations in the series.

## 4.3. The flows of the Suggested Model

Hybrid deep learning models are a powerful approach for tackling complex problems, particularly in the realm of multivariate time series analysis. These models leverage the strengths of different deep learning architectures, combining two or more deep learning models or architectures to improve the performance of the machine learning model which is more effective than using a single deep learning model (Dang et al., 2021).

One common hybrid deep learning approach for multivariate time series data involves the integration of RNN, with CNNs. The RNN component can effectively capture the temporal dependencies and long-range interactions within the time series, while the CNN part can extract meaningful features from the multivariate time-series data. The model can work as stated below:

- To extract the key characteristics from the input time series data, a CNN layer is initially fed the data. This layer produces a collection of high-level feature maps as its output.
- An LSTM layer is subsequently given the CNN layer's output to learn the temporal relationships present in the data. Using the feature maps as input, it generates a series of output vectors.
- The fully connected layer that produces the final forecasts receives the LSTM layer's output vectors.

## 4.4. Optimization of Hyperparameters

Hyperparameter optimization is a decisive step in the development of high-performing hybrid models for multivariate time series analysis. Hyperparameters are established before the training procedure and can have a significant impact on the model's performance. Proper hyperparameter tuning is essential to achieve the best possible results (Hutter, 2017).

### 4.4.1. Activation Function

(Xiangyang et al., 2023) In deep learning, an activation function is a mathematical function applied to the output of a neuron (or node) in a neural network. Its primary purpose is to introduce non-linearity into the model, enabling the network to learn and represent complex patterns in the data. Without activation functions, a neural network would function similarly to a linear regression model, no matter how many layers it has, and would be unable to capture intricate patterns in the data. We have used the following three Activation Functions in our thesis while implementing the proposed model.

**Sigmoid Function**: This activation function is the most commonly utilized since it is non-linear. The sigmoid function is a probability-like output that may be used to modify values between 0 and 1, which makes it appropriate for binary classification tasks.

**Hyperbolic Tangent (tanh) Function**: The function has an output value range of -1 to +1, with zero at its center. This characteristic helps accelerate convergence by normalizing or centering the output of each layer around zero. The tanh function is an S-shaped activation function and can be considered a shifted variant of the sigmoid function, providing advantages in terms of centering the data and facilitating more effective learning.

**Rectified Linear Unit (ReLU)**: This is the hidden layer activation function that is most frequently utilized. ReLU converges substantially quicker than sigmoid and tanh functions and is computationally efficient since it uses simpler mathematical procedures. It is more resilient to vanishing gradients, which prevent the training of deep learning models. In terms of performance and generalization, the rectified linear unit activation function performs better than the sigmoid and tanh activation functions (Bai, 2022).

### 4.4.2. Loss Function

(Akbari et al., 2021) The loss function, which is a mathematical function used in machine learning and deep learning to measure the difference between a model's predicted output and the actual target values, is additionally known as an objective function or cost function. It serves as a crucial guide for the training process, providing a measure of how well the model is performing. During training, the model's parameters are adjusted iteratively to minimize the loss function, thereby improving the model's accuracy and generalization capabilities. (Q. Wang et al., 2022) different types of loss functions are used depending on the specific task. The choice of loss function can significantly impact the model's performance and convergence, making it a fundamental aspect of the training process. By minimizing the loss function, the model learns to create more accurate predictions, ultimately leading to better performance on unseen data.

**Learning Rate:** The learning rate is a vital hyperparameter in the training process of deep learning and machine learning models. It controls the size of the steps taken during gradient descent, the optimization algorithm most commonly used to update the model’s parameters (weights and biases) to minimize the loss function. Essentially, the learning rate determines how quickly or slowly a model learns from the data.

### 4.4.3. Regularization Methods

(Badola et al., 2020) A technique in machine learning is employed to prevent overfitting. Overfitting occurs when a model learns the training data too well, capturing noise and random fluctuations instead of the underlying patterns. (Soumare et al., 2021) This leads to poor performance on unseen data. Regularization addresses this issue by adding a penalty term to the loss function. This penalty discourages complex models and promotes simpler models that generalize better. The two regularizations we have used in our thesis are Dropout and Early toping techniques.

**Dropout:** Arbitrarily sets a fraction of input units to zero at each update during training. Prevents complex co-adaptations among neurons and is effective for deep neural networks

**Early Stopping:** Monitors the model's performance on a validation set during training. When the validation set performance begins to decline, the training is stopped. Prevents overfitting by stopping training before the model becomes too complex

### 4.4.4. Optimizer

(Shulman, 2023) in deep learning, an optimizer is an algorithm that adjusts the parameters of a neural network to minimize a cost function. This process is essential for training the model to learn from the data and make accurate predictions. The main purpose of an optimizer is to find the optimal values for the model's parameters by iteratively adjusting them in the direction that minimizes the loss function. The loss function is a measure of how well the model is executed on the training data, and the goal is to find the set of parameters that minimize this loss.

### 4.4.5. Batch Size

The batch size refers to the number of samples processed by the model before updating its parameters. Smaller batch sizes (e.g., 32 or 64) can lead to more stable gradients and faster convergence, but they may also produce more noisy updates. Larger batch sizes (e.g., 256 or 512) can result in more stable updates and better hardware resource utilization, though they might converge more slowly. The optimal batch size depends on factors such as the size of the dataset, model complexity, and available hardware resources. Both batch size and number of epochs are crucial hyperparameters in machine learning models, significantly affecting the model's performance.

**Epochs:** In deep learning, an epoch refers to one complete pass of the neural network through the entire training dataset. During each epoch, the model processes all samples in the training dataset, and its weights are updated based on the gradients accumulated through the backpropagation process. The number of epochs indicates how many times the model will traverse the entire training dataset during the training phase. It suggests ceasing to increase the epochs for improved error reductions if the reduction error for subsequent epochs is not improving.

# CHAPTER FIVE

# 5. IMPLEMENTATION OF THE PROPOSED SOLUTION

## 5.1. Overview

In this chapter, we examine the implementation of the proposed model for the OTN fault prediction using a hybrid deep learning model. We also looked at setting up and using the Jupyter Notebook implementation environment.

## 5.2. Setting Up the Working Environment

A working environment is essential to have a working implementation for any research or system development to ensure that the suggested system can be properly trained, evaluated, and tested. To accomplish our objectives various sets of tools, hardware, and software resources have been used to develop, run, and test the proposed system as stated below.

**Python:** is a high-level, general-purpose programming language that's used in deep learning algorithm development.

**TensorFlow**: TensorFlow is an open-source machine learning framework developed by Google that provides a flexible ecosystem of tools, libraries, and community resources to help developers build and deploy machine learning models.

**Keras**: This is a Python-based high-level neural network API that can be used with TensorFlow.

**Scikit-Learn:** A Python module that facilitates the implementation of numerous machine learning algorithms and neural network models.

**Pandas:** For data manipulation and analysis.

**NumPy:** Allows for the manipulation of massive, multi-dimensional arrays and matrices, as well as a broad range of powerful mathematical operations.

**Seaborn:** For statistical data visualization.

**Matplotlib:** For creating static, animated, and interactive visualizations.

**Jupyter Notebook:** web-based interactive computing notebook environment that is used to plot, write code, clean, manipulate, and visualize data.

### 5.2.1. Hardware and Software Tools

To implement our proposed solution, we have used the hardware and software tools mentioned below.

**Laptop**

- Operation System: Windows 10 Pro.
- Type of System:  64-bit OS, x64-based processor.
- Memory: 8.00 GB.
- Processor speed: Intel(R) Core (TM) i5-4200M CPU @ 2.50GHz  2.50 GHz.

**Desktop**

- Operating system: Windows 10 Pro.
- System type: 64-bit operating, x64-based processor.
- Processor: Intel(R) Core (TM) i3-9100 CPU @ 3.60GHz 3.60 GHz.
- Memory size: 8.00GB.

## 5.3. OTN Fault Prediction and Implementation

### 5.3.1. Preprocessing of Dataset

Preprocessing the dataset for OTN fault prediction involves several key steps to ensure the data is suitable for training predictive models. Initially, the dataset undergoes a thorough cleaning process to handle missing values, outliers, and inconsistencies. This ensures data quality and reliability, crucial for accurate predictions. Then, feature selection and engineering are performed to extract relevant information and create meaningful predictors. This involves transforming raw data into useful features that capture the underlying patterns indicative of faults. Techniques such as normalization or standardization are applied to scale numeric features appropriately, ensuring they contribute equally to model training without biasing the results. Temporal aspects are critical in fault prediction, requiring the creation of sequences or time windows of data. This allows models to learn from historical patterns leading up to a fault occurrence. Sequencing involves selecting an appropriate time window (time_steps), ensuring

it captures sufficient past data while balancing computational feasibility and memory constraints.

Moreover, handling class imbalance is essential in datasets where fault instances are rare compared to normal operations. Techniques such as oversampling minority classes or adjusting class weights during model training can help address the class imbalance and enhance predictive performance. Additionally, dividing the dataset into training, validation, and test sets is crucial. The model is trained on the training set, fine-tuned using the validation set, and evaluated on the test set to assess its ability to generalize to unseen data.

Generally, preprocessing for OTN fault prediction involves cleaning, integrating, feature engineering, temporal sequencing, addressing class imbalance, and careful dataset splitting to prepare data effectively for training and evaluating predictive models. These steps collectively enhance the model's ability to accurately forecast faults in an optical transport network.

```
#import libraries
import pandas as pd
import numpy as np
from statsmodels.tsa.seasonal import seasonal_decompose
from matplotlib import pyplot as plt
from datetime import datetime
import seaborn as sns
from statsmodels.tsa.stattools import adfuller
import statsmodels.api as sm
```

*Figure 5. 1. Import Preprocessing Libraries*

```
#Loding the two datasets into the environment
hf=pd.read_csv('HardFailure_dataset.csv',delimiter=',',index_col='Timestamp',parse_dates=True)
sf=pd.read_csv('SoftFailure_dataset.csv',delimiter=',',index_col='Timestamp',parse_dates=True)
```

*Figure 5. 2. Loading Dataset*

```
concat_hfsf = pd.concat([hf, sf], axis=0)
```

*Figure 5. 3. Data Integration*

```
# Scale the features using MinMaxScaler
scaler = MinMaxScaler()
data[features] = scaler.fit_transform(data[features])
```

*Figure 5. 4. Scaling Dataset*

**Balancing dataset:** An extremely unbalanced dataset can enable a biased model to be trained, which could lead to inferior prediction results, particularly for the minority classes. The number for the normal class in our sample was significantly higher than the number for the fault instance. Figure 5.5 illustrates that 87.1% of data are recorded as normal, or labeled as 0, while 12.9% are faulty, or labeled as 1. We tested with over-sampling and SMOTE to see which worked best for our dataset to solve the data misbalancing problem.

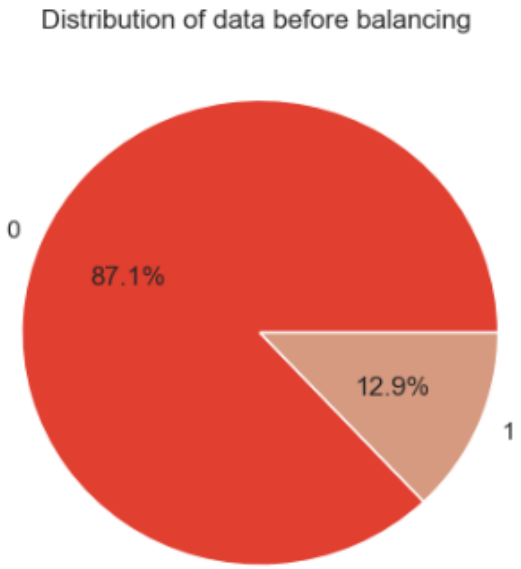


*Figure 5. 5. Imbalanced Dataset*

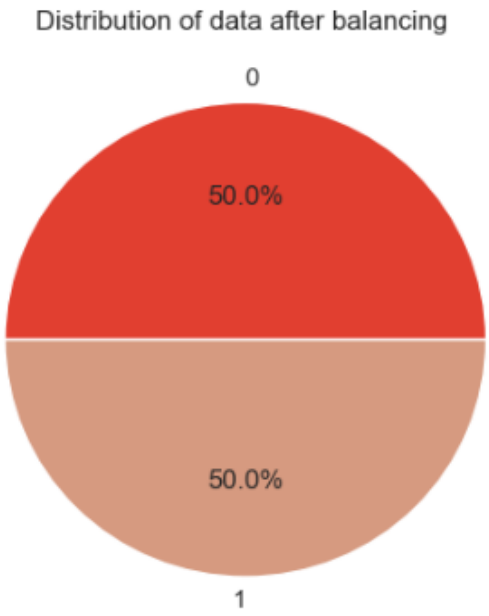


*Figure 5. 6. Balanced Dataset*

### 5.3.2. Feature Engineering and Implementation

(Istiadi et al., 2017) Feature engineering is a crucial step in building machine learning models. It involves creating and refining features to improve the model's performance and predictive accuracy by transforming raw data into more meaningful inputs for the model. It involves creating new features or modifying existing ones to improve the performance of the model. The main goal of feature engineering is to maximize the predictive power of the model by providing it with relevant and informative input data. A model can produce more precise and accurate forecasts the more it comprehends the dynamics and relationship between the instances or observations (Khurana et al., 2018). As a result, new features can be created to provide the model with more data and insight, enhancing its functionality. As mentioned in chapter two we have applied two types of feature engineering. Namely time-based and statistical features and then concatenate the two features to enhance the accuracy of the model.

### 5.3.3. Implementation of the Selected Model

We covered model implementation in this section, which is an important scenario in any research topic. It involves importing essentially libraries and optimization parameters that we used for training the model, as well as an implementation of the chosen model for prediction

and classification of the OTN fault problem. We imported the required libraries before implementing the CNN-LSTM, CNN-BiLSTM, and CNN-GRU.

```
import matplotlib.pyplot as plt
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, LSTM, Bidirectional, GRU, Conv1D, MaxPooling1D, Flatten
from tensorflow.keras.layers import LSTM, Dense, Dropout
from tensorflow.keras.preprocessing.sequence import TimeseriesGenerator
from sklearn.model_selection import train_test_split
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.callbacks import EarlyStopping
from sklearn.metrics import mean_absolute_error, mean_squared_error, mean_absolute_percentage_error
from sklearn.preprocessing import MaxAbsScaler
from sklearn.preprocessing import MinMaxScaler
from tensorflow.keras.callbacks import ReduceLROnPlateau
```

*Figure 5. 7. Import All Needed Libraries*

#### 5.3.3.1. CNN-LSTN Model Implementation

A CNN-LSTM model is a hybrid deep learning architecture that combines the feature extraction capabilities of CNNs with the sequence modeling and long-term dependency learning of LSTM. In the OTN fault prediction domain, this approach can be particularly effective. The CNN component of the model is used to automatically learn and extract relevant features from the OTN time series data. The CNN can capture local dependencies and identify important correlations within the data. Then, the extracted features are given into the LSTM component, which can effectively model the temporal dynamics and long-term dependencies in the OTN performance data. This allows the model to learn complex, non-linear relationships between the historical OTN data and the occurrence of faults. Combining the strengths of the CNN-LSTM architecture can provide a powerful and versatile framework for OTN fault prediction.

Figure 5.8 represents a Python code snippet demonstrating the construction and training of a CNN combined with an LSTM network, incorporating early stopping mechanisms. The model is built using the Sequential API. Initially, three Conv1D layers are defined, each with 128 filters, a kernel size of 2, and 'relu' activation functions. These convolutional layers are followed by MaxPooling1D layers and Dropout layers with a dropout rate of 0.2 to prevent overfitting. The output from the convolutional layers is flattened and then processed through two Dense

layers. The first Dense layer has 64 neurons with a 'relu' activation function, while the second Dense layer has a single neuron with a 'sigmoid' activation function for binary classification.

The model is compiled with the Adam optimizer and the mean squared error loss function. During training, two callbacks are used; EarlyStopping, which monitors the validation loss and stops training when no further improvements are detected, and ReduceLROnPlateau, which reduces the learning rate when a plateau in the validation loss is observed. The training log shows the progress with the loss, validation loss, and learning rate metrics displayed over the epoch. Throughout epochs, the model's performance in terms of validation loss improves slightly, indicating the model is learning effectively without overfitting, as the early stopping and learning rate reduction mechanisms help to optimize the training process.

```
# Build the CNN-LSTM model with early stopping
model = Sequential()
model.add(Conv1D(filters=128, kernel_size=2, activation='relu', input_shape=(sequence_length, len(features))))
model.add(MaxPooling1D(pool_size=1))
model.add(Dropout(0.2))
model.add(Conv1D(filters=64, kernel_size=2, activation='relu'))
model.add(MaxPooling1D(pool_size=1))
model.add(Dropout(0.2))
model.add(LSTM(50, return_sequences=True, activation='relu'))
model.add(Dropout(0.2))
model.add(LSTM(25, return_sequences=False, activation='relu'))
model.add(Dropout(0.2))
model.add(Dense(7, activation='relu'))  #
model.add(Dense(1, activation='sigmoid'))
model.compile(optimizer='Adam', loss='mean_squared_error')
# Define early stopping callback
early_stopping = EarlyStopping(monitor='val_loss', patience=10, restore_best_weights=True)
# Define learning rate reduction callback
reduce_lr = ReduceLROnPlateau(monitor='val_loss', factor=0.2, patience=2, min_lr=0.001)
# Train the model with early stopping and learning rate reduction
history = model.fit(X_train, y_train, epochs=200, batch_size=300, validation_split=0.2, callbacks=[early_stopping, reduce_lr])
```

```
114/114 ━━━━━━━━━━━━━━━━━━━━ 2s 19ms/step - loss: 0.0150 - val_loss: 0.0153 - learning_rate: 1.0000e-04
Epoch 175/200
114/114 ━━━━━━━━━━━━━━━━━━━━ 2s 17ms/step - loss: 0.0145 - val_loss: 0.0156 - learning_rate: 1.0000e-04
Epoch 176/200
114/114 ━━━━━━━━━━━━━━━━━━━━ 2s 19ms/step - loss: 0.0144 - val_loss: 0.0157 - learning_rate: 1.0000e-04
Epoch 177/200
114/114 ━━━━━━━━━━━━━━━━━━━━ 2s 20ms/step - loss: 0.0145 - val_loss: 0.0155 - learning_rate: 1.0000e-04
Epoch 178/200
114/114 ━━━━━━━━━━━━━━━━━━━━ 3s 18ms/step - loss: 0.0156 - val_loss: 0.0152 - learning_rate: 1.0000e-04
Epoch 179/200
114/114 ━━━━━━━━━━━━━━━━━━━━ 2s 16ms/step - loss: 0.0152 - val_loss: 0.0152 - learning_rate: 1.0000e-04
Epoch 180/200
114/114 ━━━━━━━━━━━━━━━━━━━━ 3s 18ms/step - loss: 0.0149 - val_loss: 0.0152 - learning_rate: 1.0000e-04
```

*Figure 5. 8. CNN-LSTM Model Implementation*

### 5.3.3.2. CNN-BiLSTM Model Implementation

CNN-BiLSTM hybrid model integrates the strengths of CNNs and BiLSTM networks to enhance predictive accuracy and efficiency. This hybrid approach leverages CNNs' ability to

extract important features from time series data and BiLSTMs' proficiency in capturing long-term dependencies in both forward and backward directions.

For OTN fault prediction, the hybrid model first employs CNN layers to process the multivariate time series data. CNN's relevant feature extraction helps in reducing noise and focusing on the most relevant aspects of the data. Following the CNN layers, the processed data is fed into BiLSTM layers. The BiLSTM network, with its bidirectional architecture, analyzes the temporal relationships in the data from both past to future and future to past. This bidirectional processing ensures that the model can understand the context of the data more comprehensively, capturing dependencies and trends that might be missed by unidirectional models.

```
# Build the CNN-BiLSTM model with early stopping
model = Sequential()
model.add(Conv1D(filters=128, kernel_size=2, activation='relu', input_shape=(sequence_length, len(features))))
model.add(MaxPooling1D(pool_size=1))
model.add(Dropout(0.2))
model.add(Conv1D(filters=64, kernel_size=2, activation='relu'))
model.add(MaxPooling1D(pool_size=1))
model.add(Dropout(0.2))
model.add(Bidirectional(LSTM(50, return_sequences=True, activation='relu')))
model.add(Dropout(0.2))
model.add(Bidirectional(LSTM(25, return_sequences=False, activation='relu')))
model.add(Dropout(0.2))
model.add(Dense(7, activation='relu'))
model.add(Dense(1, activation='sigmoid'))
model.compile(optimizer=Adam(learning_rate=0.001), loss='mean_squared_error')
# Define early stopping callback
early_stopping = EarlyStopping(monitor='val_loss', patience=10, restore_best_weights=True)
reduce_lr = ReduceLROnPlateau(monitor='val_loss', factor=0.2, patience=2, min_lr=0.001)
# Train the model with early stopping and learning rate reduction
history = model.fit(X_train, y_train, epochs=200, batch_size=300, validation_split=0.2, callbacks=[early_stopping, reduce_lr])
```

```
Epoch 107/200
114/114 ━━━━━━━━━━━━━━━━━━━━ 2s 20ms/step - loss: 0.0153 - val_loss: 0.0165 - learning_rate: 1.0000e-04
Epoch 108/200
114/114 ━━━━━━━━━━━━━━━━━━━━ 3s 24ms/step - loss: 0.0155 - val_loss: 0.0164 - learning_rate: 1.0000e-04
Epoch 109/200
114/114 ━━━━━━━━━━━━━━━━━━━━ 5s 20ms/step - loss: 0.0167 - val_loss: 0.0168 - learning_rate: 1.0000e-04
Epoch 110/200
114/114 ━━━━━━━━━━━━━━━━━━━━ 2s 19ms/step - loss: 0.0156 - val_loss: 0.0163 - learning_rate: 1.0000e-04
Epoch 111/200
114/114 ━━━━━━━━━━━━━━━━━━━━ 2s 19ms/step - loss: 0.0163 - val_loss: 0.0167 - learning_rate: 1.0000e-04
Epoch 112/200
114/114 ━━━━━━━━━━━━━━━━━━━━ 2s 19ms/step - loss: 0.0156 - val_loss: 0.0164 - learning_rate: 1.0000e-04
```

*Figure 5. 9. CNN-BiLSTM Model Implementation*

### 5.3.3.3. CNN-GRU Model Implementation

The CNN-GRU model is another hybrid deep learning architecture that combines the feature extraction capabilities of CNNs with the sequence modeling and temporal dependency learning of GRUs. Similar to the CNN-LSTM model, the CNN-GRU model can be effective for OTN

fault prediction tasks. In this approach, the CNN component is used to automatically extract salient important features from the OTN time series data. The CNN can capture local dependencies and identify important correlations within the data. The extracted features are then given into the GRU component, which is a variant of the LSTM architecture.

By combining the strengths of CNNs and GRUs, the CNN-GRU architecture can provide a powerful and efficient framework for OTN fault prediction. It can learn temporal features from the input data, leading to accurate and robust predictions of future network issues.

```
# Build the CNN-GRU model with early stopping
model = Sequential()
model.add(Conv1D(filters=128, kernel_size=2, activation='relu', input_shape=(sequence_length, len(features))))
model.add(MaxPooling1D(pool_size=1))
model.add(Dropout(0.2))
model.add(Conv1D(filters=64, kernel_size=2, activation='relu'))
model.add(MaxPooling1D(pool_size=1))
model.add(Dropout(0.2))
model.add(GRU(64, return_sequences=True, activation='relu'))
model.add(Dropout(0.2))
model.add(GRU(25, return_sequences=False, activation='relu'))
model.add(Dropout(0.2))
model.add(Dense(1, activation='sigmoid'))
model.compile(optimizer='adam', loss='mean_squared_error')
# Define early stopping callback
early_stopping = EarlyStopping(monitor='val_loss', patience=10, restore_best_weights=True)
# Define learning rate reduction callback
reduce_lr = ReduceLROnPlateau(monitor='val_loss', factor=0.2, patience=2, min_lr=0.0001)
# Train the model with early stopping and learning rate reduction
history = model.fit(X_train, y_train, epochs=200, batch_size=300, validation_split=0.2, callbacks=[early_stopping, reduce_lr])
```

```
114/114 ━━━━━━━━━━━━━━━━━━━━ 3s 22ms/step - loss: 0.0150 - val_loss: 0.0157 - learning_rate: 1.0000e-04
Epoch 164/200
114/114 ━━━━━━━━━━━━━━━━━━━━ 3s 22ms/step - loss: 0.0150 - val_loss: 0.0155 - learning_rate: 1.0000e-04
Epoch 165/200
114/114 ━━━━━━━━━━━━━━━━━━━━ 3s 23ms/step - loss: 0.0159 - val_loss: 0.0154 - learning_rate: 1.0000e-04
Epoch 166/200
114/114 ━━━━━━━━━━━━━━━━━━━━ 5s 19ms/step - loss: 0.0157 - val_loss: 0.0154 - learning_rate: 1.0000e-04
Epoch 167/200
114/114 ━━━━━━━━━━━━━━━━━━━━ 3s 19ms/step - loss: 0.0139 - val_loss: 0.0152 - learning_rate: 1.0000e-04
Epoch 168/200
```

*Figure 5. 10. CNN-GRU Model Implementation*

## 5.4. Model Evaluation

When developing predictive models for OTN fault prediction, it is critical to carefully evaluate the model's performance using appropriate metrics. In this thesis, we have used four evaluation metrics. Namely, MSE, MAE, RMSE, and $R^2$.

MSE is a widely used metric that measures the average of the squared differences between predicted and actual values. It is calculated by summing the squared differences and dividing

by the total number of data points, which results in greater penalties for larger errors due to the squaring of the differences. In contrast, Mean Absolute Error (MAE) measures the average of the absolute differences between predicted and actual values, offering a more straightforward interpretation of model performance by representing the average magnitude of errors without regard to their direction.

RMSE is derived by taking the square root of the MSE, thus providing a metric with the same units as the target variable. RMSE is useful for understanding the magnitude of errors relative to the original scale of the data and indicates how well the model fits the data.

Additionally, $R^2$ represents the proportion of the variance in the dependent variable explained by the independent variables within a regression model. This metric helps assess the explanatory power of the model.

We have evaluated various models, including LSTM, BiLSTM, GRU, CNN-LSTM, CNN-BiLSTM, and CNN-GRU, using MAE, MSE, RMSE, and $R^2$ to measure their performance.

# CHAPTER SIX

# 6. RESULTS AND DISCUSSION

## 6.1. Introduction

This chapter presents the findings from our proposed models for OTN fault prediction and classification. We have evaluated three models; CNN-LSTM, CNN-BiLSTM, and CNN-GRU. Based on their performance comparison in fault prediction and classification, we can examine their results and determine which model is the best. Additionally, we go over the significance of the features and how they rank in terms of statistical value contribution for target values.

## 6.2. Importance of Features and Selection Outcomes

Feature importance and feature selection are critical aspects of building predictive models, in deep learning. Feature importance quantifies the significance of each feature in contributing to the prediction of the dependent variable. This concept is crucial because it helps identify which features have the most impact on model performance and which ones can be disregarded without compromising the model's accuracy. In optical fault prediction we have been generating time-based features and statistical features to enhance model performance. In considering this, we created features with rolling windows and time-based functionality and then selected features to find more pertinent features.

### 6.2.1. Feature Correlation Analysis

Feature selection involves choosing a subset of pertinent features from the entire feature set to improve model performance, reduce computational cost, and mitigate overfitting. One common approach to feature selection is using Pearson correlation, which measures the linear relationship between features and the target variable. Pearson correlation helps in assessing how strongly a feature is related to the target by calculating a correlation coefficient. A high absolute value of this coefficient shows a strong relationship, while a value close to zero suggests a weak or no linear relationship. Features with high correlations to the target are considered more informative and thus prioritized during model training. Conversely, features with low or negative correlations may be less useful or redundant. By incorporating Pearson correlation in

the feature selection process, one can ensure that the chosen features are not only relevant but also contribute positively to the model's predictive power.

*Table 6. 1. Feature List Selected Based on Correlation Scores*

| no | Feature name | Results |
|---|---|---|
| 1 | Hour | 0.701330 |
| 2 | OSNR_std | 0.641185 |
| 3 | OSNR_max | 0.434949 |
| 4 | OSNR_min | 0.381263 |
| 5 | OutputPower_max | 0.377163 |
| 6 | OutputPower_mean | 0.261878 |
| 7 | BER_std | 0.258510 |
| 8 | OutputPower_min | 0.175600 |
| 9 | OutputPower | 0.171049 |
| 10 | BER_max | 0.152488 |
| 11 | BER_min | 0.118429 |
| 12 | InputPower_std | 0.116868 |
| 13 | InputPower_max | 0.101945 |
| 14 | InputPower_min | 0.083910 |
| 15 | BER_mean | 0.021742 |
| 16 | OSNR_mean | 0.019727 |
| 17 | BER | 0.017617 |
| 18 | OSNR | 0.015035 |
| 19 | OutputPower_std | 0.011573 |
| 20 | Minute | 0.006067 |
| 21 | InputPower_mean | 0.000581 |
| 22 | Second | 0.000570 |
| 23 | InputPower | 0.000449 |

The feature importance ranking score is shown in Table 6-1, and it is derived from the Pearson Coefficient experiment. Therefore, we understand that these features are more important than other features that are available for training our models and fitting well performance.

*Table 6. 2. Important Features Selected Based on XGBClassifier*

| no | Feature name | Scores |
|---|---|---|
| 1 | InputPower_mean | 0.125368 |
| 2 | OSNR_min | 0.088833 |
| 3 | InputPower_std | 0.066318 |
| 4 | InputPower_max | 0.064618 |
| 5 | OutputPower_mean | 0.059915 |
| 6 | OSNR | 0.058390 |
| 7 | OutputPower_max | 0.055136 |
| 8 | OSNR_mean | 0.051577 |
| 9 | Hour | 0.045762 |
| 10 | InputPower | 0.040459 |
| 11 | OSNR_max | 0.035123 |
| 12 | InputPower_min | 0.034268 |
| 13 | BER_std | 0.030425 |
| 14 | Minute | 0.029700 |
| 15 | Second | 0.029295 |
| 16 | BER_min | 0.028557 |
| 17 | OutputPower | 0.027641 |
| 18 | OutputPower_min | 0.026263 |
| 19 | BER_mean | 0.026121 |
| 20 | BER_max | 0.020511 |
| 21 | OSNR_std | 0.019512 |
| 22 | BER | 0.019161 |
| 23 | InputPower_std | 0.017048 |

The outcomes of the XGBClassifier are shown in Table 6.2. The gradient boosting approach is implemented by XGBClassifier. Feature importance scores are produced by the classifier depending on how frequently a feature is utilized for splitting the data among all of the machine learning model's trees. A higher frequency of usage for separating attributes makes them more valuable.

*Table 6. 3. Selected Feature Based on Chi_Scores*

| s.no | Feature name | Scores |
|---|---|---|
| 1 | OSNR_std | 7312.637549 |
| 2 | Hour | 2887.945202 |
| 3 | BER_std | 981.114795 |
| 4 | OSNR_min | 851.115393 |
| 5 | OutputPower_min | 739.40151 |
| 6 | OutputPower | 349.195469 |
| 7 | OutputPower_mean | 346.577797 |
| 8 | OSNR_max | 309.495821 |
| 9 | OutputPower_max | 291.896550 |
| 10 | BER_max | 271.404250 |
| 11 | BER_min | 217.853486 |
| 12 | InputPower_std | 114.035901 |
| 13 | InputPower_min | 15.143184 |
| 14 | InputPower_max | 8.789848 |
| 15 | BER_mean | 4.407248 |
| 16 | BER | 3.926403 |
| 17 | OutputPower_std | 2.014539 |
| 18 | OSNR | 0.815703 |
| 19 | OSNR_mean | 0.80666 |
| 20 | Minute | 0.345026 |
| 21 | Second | 0.002963 |
| 22 | InputPower_mean | 0.000288 |
| 23 | InputPower | 0.000265 |

Table 6-3 indicates the ranking score of feature importance and the ranking score is generated depending on the Chi-square(X2) test. Therefore, such listed features are deemed more relevant for our proposed (hybrid DL) model performance.

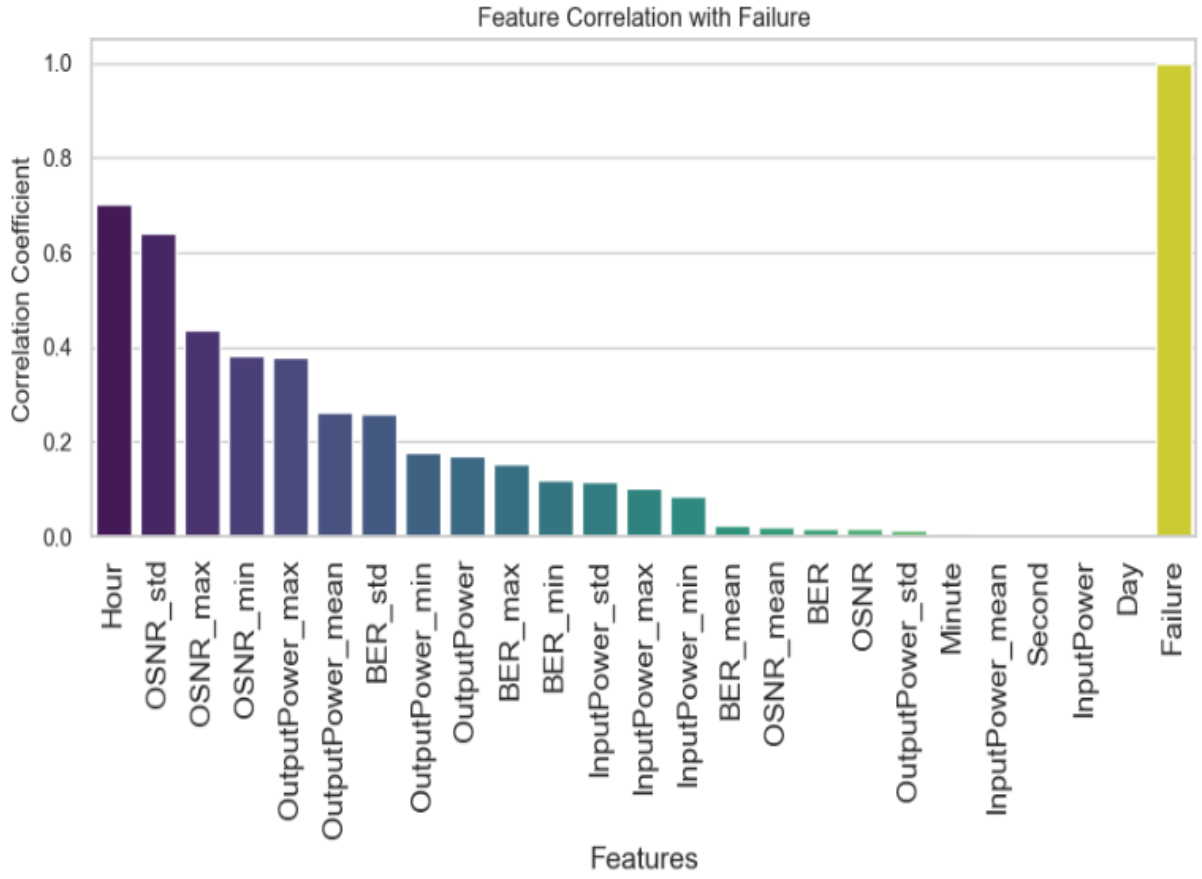


*Figure 6. 1. The Results of Correlation Analysis*

The bar plot displayed in Figure 6.1 represents the correlation coefficients between various features in a dataset and the target feature Failure. The x-axis indicates the different features in the dataset. These features are labeled and displayed along the x-axis and each bar in the plot represents the correlation score between relevant features and to dependent variable. Among the features, Hour exhibits the strongest correlation with Failure, followed by OSNR_std, OSNR_max, and OutputPower_max, which also show relatively strong correlations. Features such as Day, Minute, and InputPower_mean show much weaker correlations, with values close to zero. This plot is a useful tool for identifying which features are most relevant for predicting OTN faults.

## 6.3. The Results of the Proposed Model

This section covers the experiment we performed using six models. Namely, CNN-LSTM, CNN-BiLSTM, and CNN-GRU were the models that were tested for OTN fault prediction and classification using optical failure datasets.

### 6.3.1. Experiment Four (CNN-LSTM)

In this experiment, both the training loss and validation loss are initially high, indicating poor model performance at the start of the training process. As the number of epochs increases, both losses decrease, demonstrating that the model is learning and improving its performance over time, as illustrated in Figure 6.2 below.

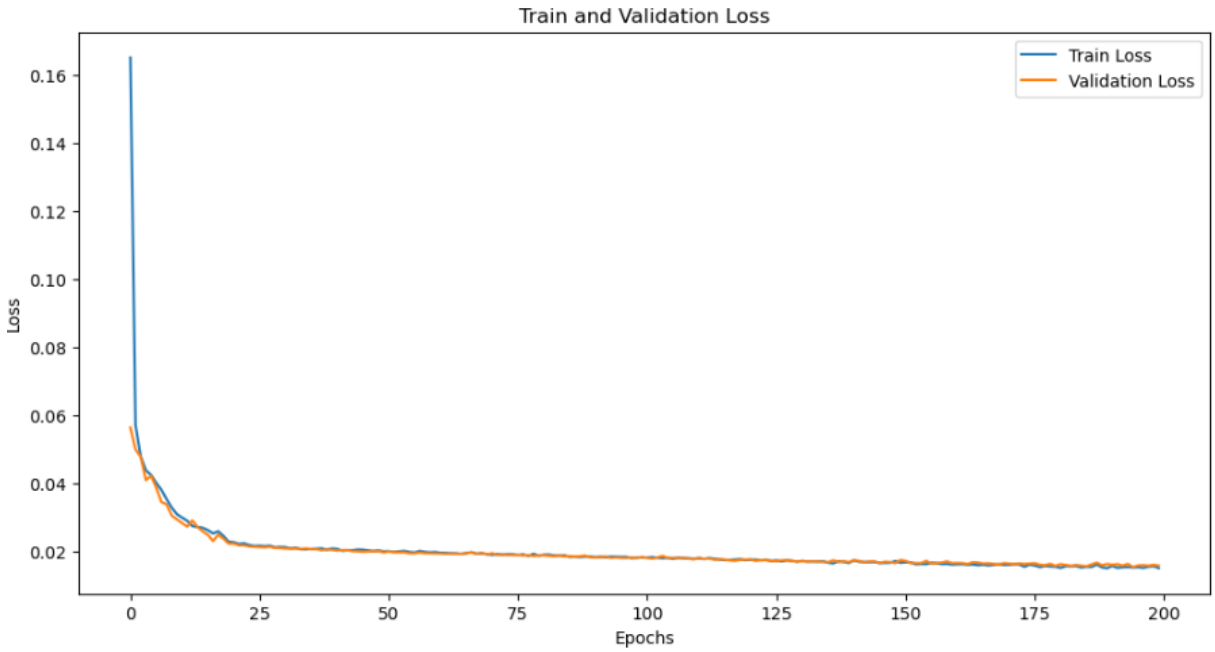


*Figure 6. 2. CNN-LSTM Training and Validation Loss*

### 6.3.2. Experiment Five (CNN-BiLSTM)

Figure 6.3 shows the trends of Train Loss and Validation Loss over 100 epochs for a CNN-BiLSTM model. The experiment demonstrates that both the Train Loss and Validation Loss start relatively high and then decrease rapidly during the early epochs, demonstrating that the model is learning and improving its performance. As training progresses, both the training loss and validation loss continue to decrease. The close alignment between these losses suggests that the model is not overfitting to the training data and is generalizing well to the validation data.

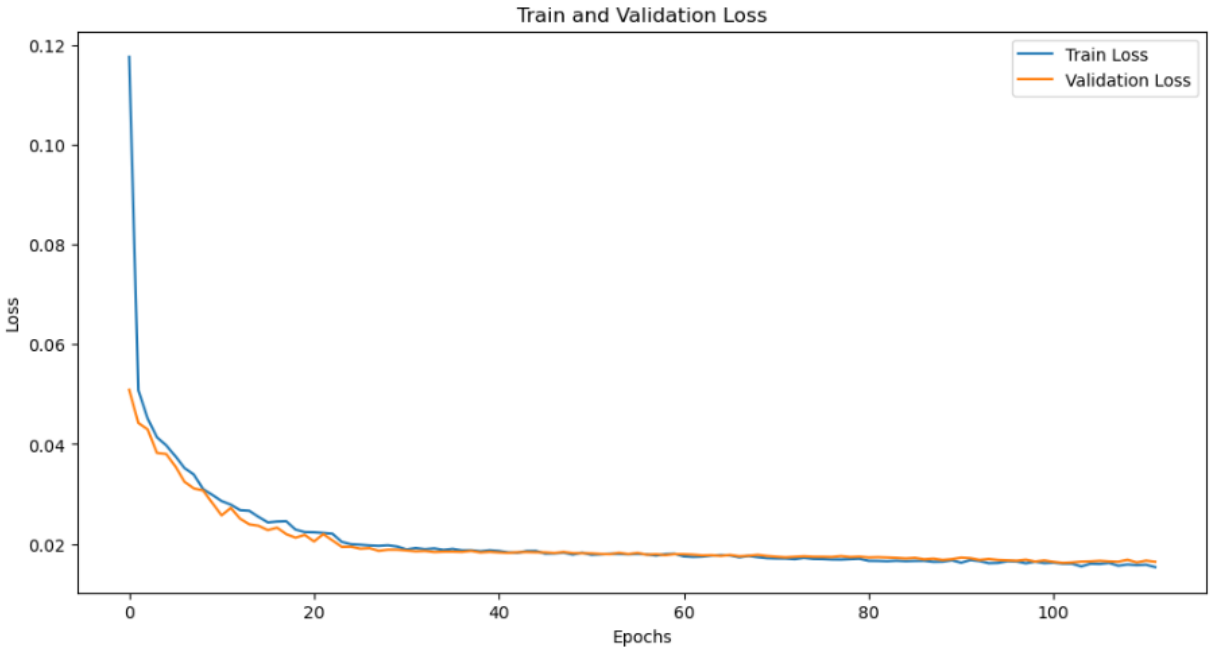


*Figure 6. 3. CNN-BiLSTM Training and Validation Loss*

### 6.3.3. Experiment Six (CNN-GRU)

This experiment depicts the trends of Train Loss and Validation Loss throughout 150 training epochs for a machine-learning model. Figure 6.4 illustrates that both the train loss and validation loss begin at relatively high levels and decrease rapidly during the early epochs, signaling that the model is learning and enhancing its performance. As training continues, the losses continue to decline, albeit at a slower rate. The close alignment between the train loss and validation loss indicates that the model is generalizing well and not overfitting to the training data.

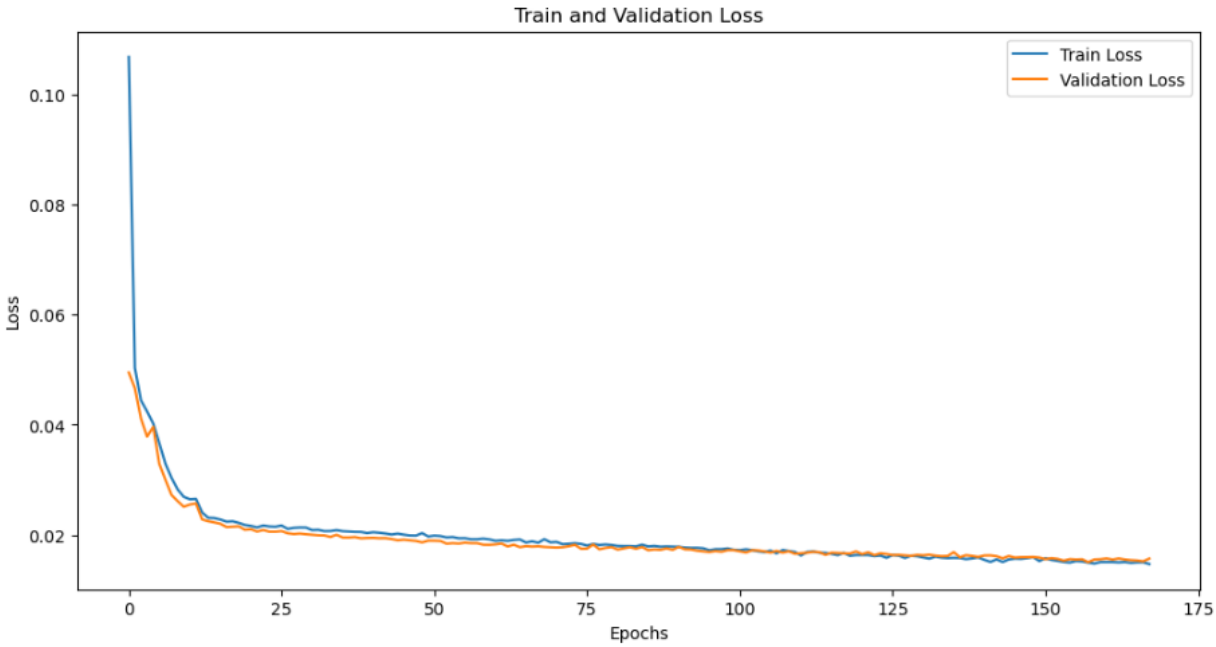


*Figure 6. 4. CNN-GRU Training and Validation Loss*

Figure 6.5 presents a comparison of training and validation loss curves for three different hybrid models CNN-LSTM, CNN-BiLSTM, and CNN-GRU over 200 epochs. All models exhibit a rapid decrease in both training and validation loss during the initial epochs, indicating efficient learning. The training loss for CNN-LSTM (blue solid line) drops quickly and then stabilizes, demonstrating its ability to learn effectively. The CNN-BiLSTM (red solid line) follows a similar pattern, showing a comparable rate of loss reduction and stabilization. Likewise, the CNN-GRU (green solid line) also experiences a rapid initial decrease, though with a slightly more gradual descent, eventually reaching a similar performance to the other models.

The validation loss curves, represented by dashed lines, reflect how well the models generalize to unseen data. For all models, the validation loss decreases alongside the training loss, and the close alignment of these curves suggests that overfitting is minimal. Both CNN-LSTM and CNN-BiLSTM demonstrate strong generalization with their validation loss following the training loss closely. The CNN-GRU model shows similar behavior, with its validation loss also converging effectively.

Overall, the graph highlights that all three models perform well in terms of learning and generalization, with their loss curves converging to stable values. The consistent performance

across the models shows that they are effectively trained and have a similar capacity for generalization, making them suitable for the OTN fault prediction task.

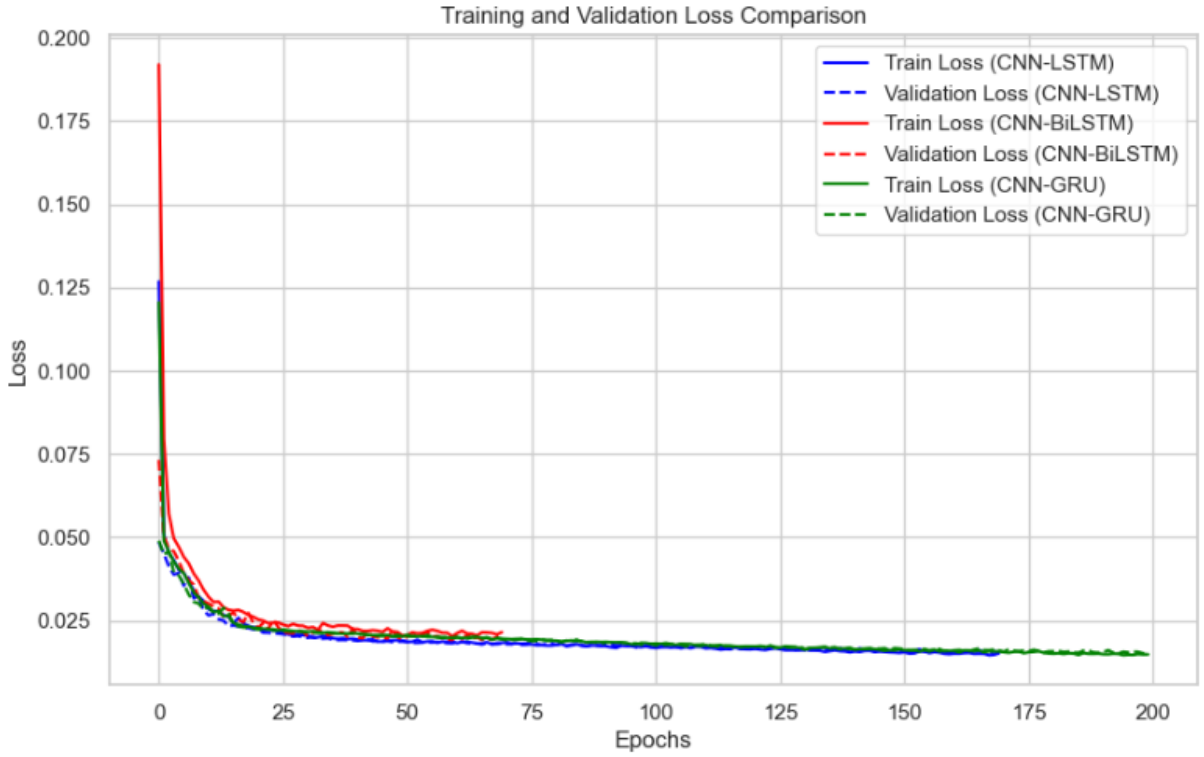


*Figure 6. 5. Training and Validation Loss Comparison*

## 6.4. Outcomes for the Proposed Model

### 6.4.1. Evaluation Analysis Results

Our objective is to test the CNN-LSTM, CNN-BiLSTM, and CNN-GRU proposed models using common performance measures including MAE, MSE, RMSE, $R^2$, validation loss, and training loss. The evaluation and comparison of these models using a 0.2 training testing split are displayed in the table below. Due to the frequent occurrence of fault behaviors, this experiment was conducted to test the model's performance on both fully unseen data and dynamic data input. The Table 6.4 comparison makes it clear that all three models CNN-LSTM, CNN-BiLSTM, and CNN-GRU perform well in terms of multiple evaluation metrics. Nonetheless, there are a few minor differences in how well each model performs.

## 6.5. Comparison of Evaluation Metrics

1n Table 6.4 appears to be presenting performance metrics for six deep learning models using MAE, MSE, RMSE, and $R^2$ evaluation metrics as shown below.

*Table 6.4. Performance Evaluation Metrics Outcomes for Proposed Model*

| Models | Metrics | | | |
|---|---|---|---|---|
| | MAE | MSE | RMSE | R-squared($R^2$) |
| **CNN-LSTM** | **2.7686** | **1.5203** | **1.2330** | **0.9392** |
| CNN-BiLSTM | 2.9538 | 1.5437 | 1.2424 | 0.9382 |
| **CNN-GRU** | **2.7890** | **1.5240** | **1.2345** | **0.9390** |

Figure 6.5 presents a comparison of three hybrid deep learning models CNN-LSTM, CNN-BiLSTM, and CNN-GRU evaluated using four metrics MSE, MAE, RMSE, and $R^{2,}$ and for Optical Transport Network fault prediction. Among these models, the CNN-LSTM model shows superior performance, achieving the lowest values across all three metrics, with MSE of **1.5203**, MAE of **2.7686**, RMSE of **1.2330**, and highest R-squared($R^2$) of **0.9392**. This shows that the integration of CNN with LSTM enhances the model's capability to capture complex patterns in the data, making it the most effective model for OTN fault prediction. The CNN-BiLSTM and CNN-GRU models also perform well, as the CNN-LSTM model. CNN-GRU has slightly lower MAE than CNN-BiLSTM, but both models have similar MSE and RMSE values and CNN-GRU is higher in $R^2$, indicating that these hybrid models are effective, though not as robust as CNN-LSTM.

In summary, the comparison shows that the CNN-LSTM model is the most effective for OTN fault prediction, offering the best balance between low error metrics.

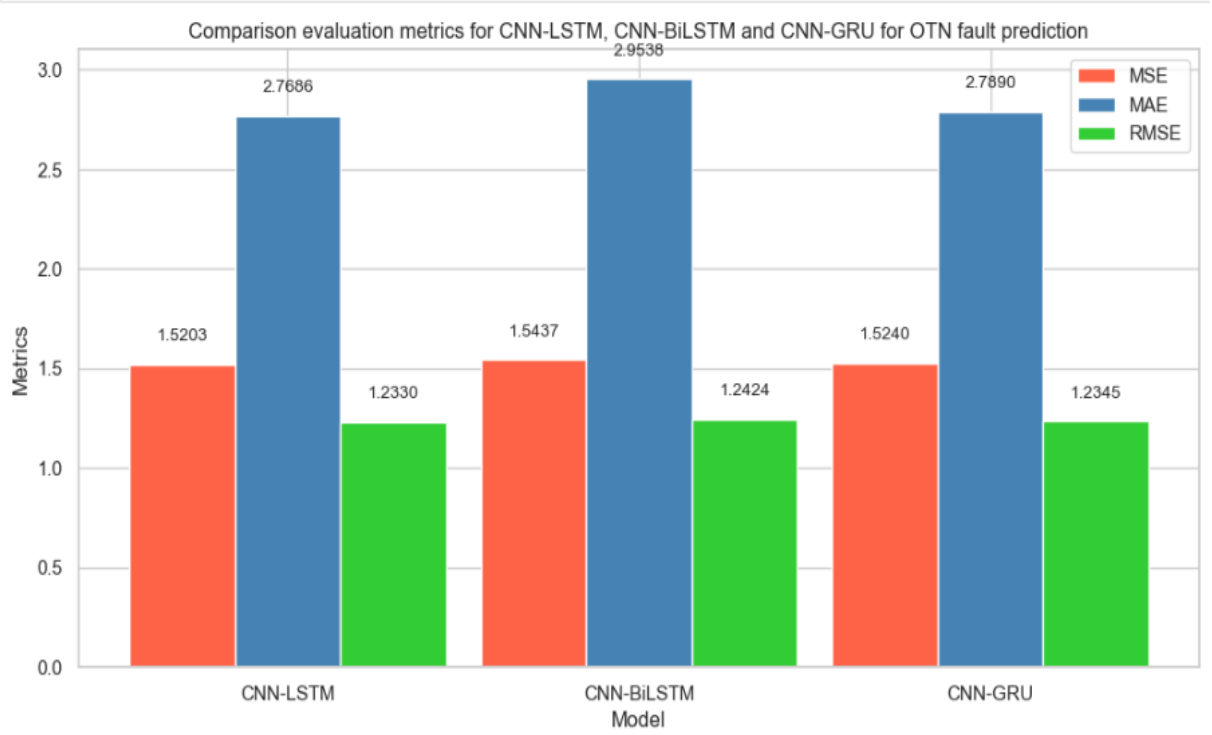


*Figure 6. 6. Comparison Evaluation Metrics for OTN Fault Prediction*

## 6.6. Result of Discussion on Proposed Model

OTNs are essential parts of current telecom networks because they allow high-capacity data transmission across long distances. The management and maintenance of OTNs are essential to ensure reliability and minimize downtime. Predicting faults and classifying potential issues in OTNs can greatly enhance operational efficiency and reduce costs. Leveraging hybrid DL models for this purpose offers a promising approach due to their capability to handle complex, multivariate time-series data and extract meaningful patterns from vast datasets.

In comparing CNN-LSTM, CNN-BiLSTM, and CNN-GRU models for OTN fault prediction, each offers distinct advantages. CNN-LSTM balances complexity and performance, effectively capturing long-term dependencies. CNN-BiLSTM enhances accuracy by processing data in both directions, making it ideal for capturing intricate dependencies but requiring more computational resources. CNN-GRU, with its simpler architecture, provides faster computation and comparable accuracy to CNN-LSTM, making it appropriate for real-time applications in OTN. The CNN component in these models is important in feature extraction by identifying and capturing local patterns in the time series data. This pre-processing step allows the

subsequent LSTM, BiLSTM, or GRU layers to focus on modeling temporal dependencies more effectively. Combining CNNs with LSTM, BiLSTM, and GRU offers distinct advantages. CNN-LSTM leverages the strength of LSTMs in capturing long-term dependencies, CNN-BiLSTM benefits from bidirectional processing for enhanced context understanding, and CNN-GRU combines efficient computation with solid temporal modeling. This synergy enhances the overall predictive accuracy and efficiency of the models in OTN fault prediction.

During training models, CNN-LSTM, CNN-BiLSTM, and CNN-GRU exhibited signs of overfitting. To mitigate overfitting, dropout was employed to randomly deactivate units during training, preventing the model from becoming overly reliant on specific neurons. Additionally, early stopping was utilized to halt training when the model's performance on the validation set began to decline, thereby helping to prevent overfitting.

Generally, hybrid DL models combine different types of neural network designs to capitalize on their strengths. In the OTN fault prediction, a typical hybrid model we have used is CNNs for capturing patterns and local features from time-series data and LSTM, BiLSTM, and GRU for learning temporal dependencies and long-term patterns in multivariate time-series data. The combination of these neural networks allows the model to efficiently learn from temporal dynamics inherent in OTN data, leading to improved prediction accuracy. The application of a hybrid DL model for the prediction of faults in Optical Transport Networks has proven to be highly effective. The capacity of the model to learn from temporal data has resulted in superior fault prediction and classification outcomes, offering a valuable tool for maintaining the reliability and efficiency of OTNs.

## 6.7. Comparison of Proposed Model Against Baseline

The results from the baseline paper (M. F. Silva et al., 2022) indicate that the LSTNet model outperformed both RNN and LSTM models in forecasting tasks, achieving significantly lower MSE and higher $R^2$ values. Specifically, LSTNet demonstrated a 95% performance improvement over the RNN and LSTM models, with an average $R^2$ value of 0.9033. In contrast, our proposed models CNN-LSTM, CNN-BiLSTM, and CNN-GRU exhibited notably lower error rates. The MSE values for our models were scored 1.5203, 1.5437, and 1.5240, respectively, averaging an MSE of 1.5293. This shows an error reduction of 3.470 compared to the baseline study, equating to an accuracy of 98.47% correctly predicted for unseen data.

Moreover, our model achieved $R^2$ of 0.9388, or 93.88%, which is significantly higher than the 90.33% $R^2$ reported in the baseline study which is a 3.55% improvement. These findings underscore the superior predictive accuracy and enhanced error reduction of our approach compared to the LSTNet model. Generally, the proposed model is performing better, with smaller errors in its predictions compared to the baseline.

## 6.8. Discussion on Research Questions

As stated, the first chapter of section 1.4 contains the three research questions for this study. So, in the following order, our study's response to these questions is as follows.

**RQ1. How can a hybrid deep learning model be developed to accurately predict and classify faults in OTN?**

Developing a hybrid DL model for predicting and classifying Optical Transport Network faults involves integrating multiple neural network architectures to leverage their strengths. A typical approach could combine CNNs for feature extraction with LSTM, BiLSTM, and GRU neural networks for temporal pattern recognition. The hybrid model should be trained on a dataset containing relevant OTN features such as BER, OSNR, Input, and Output power levels, along with timestamps to capture the sequential nature of the data and include feature engineering and selection. The development process includes:

**Preprocessing:** In this stage, we have been done cleaning and normalized the dataset, handling missing values, and scaling features to ensure model efficiency.

**Architecture Design:** Combining CNN layers to capture temporal features and LSTM, BiLSTM, and GRU layers to handle temporal dependencies in the OTN failure dataset.

**Training:** Using techniques like early stopping, learning rate, and batch size to optimize the training process of our model.

**Evaluation:** Assessing model performance using metrics such as MAE, MSE, RMSE, $R^2$, Train_loss, and Validation_loss.

**RQ2. Which features are the most relevant for predicting and classifying OTN faults?**

Identifying relevant attributes or features is crucial for accurate prediction of the model. This has been achieved through: -

**Correlation Analysis:** Evaluating the correlation between various features and the target variable (fault occurrence) to identify the most influential attributes.

**Feature Selection Techniques:** XGBClassifier and Chi-Square tests to rank features based on their importance.

**Feature engineering techniques:** Time-based and statistical-based feature engineering have been used.

According to Table 6.3, the importance of various features in predicting faults in Optical OTN is ranked based on their relevance. Among the features, InputPower_mean emerges as the most critical, suggesting that fluctuations in average input power have a significant impact on fault occurrence. Additionally, OSNR_min, which indicates the minimum OSNR, is a key feature, reinforcing the idea that poor signal quality is a major factor in network reliability. The standard deviation and maximum values of input power also rank highly, indicating that not just the average, but variability and extremes in power levels are crucial for fault prediction. Similarly, the OutputPower_mean and OutputPower_max are important, showing that both input and output power levels are vital indicators of potential network issues.

The OSNR_mean and OSNR_max further demonstrate the importance of signal quality, where both average and peak values contribute to predicting faults. Interestingly, temporal features like Hour, Minute, and Second play a smaller but still relevant role, indicating that fault occurrence may vary based on time-dependent factors such as traffic patterns or scheduled maintenance. On the other hand, BER features, though still relevant, have lower importance scores, suggesting that while they are indicative of network issues, power levels and OSNR are more critical in the model used.

In general, the features related to InputPower, OutputPower, and OSNR are the most significant for predicting OTN faults, indicating that maintaining stable power and high signal quality is essential for network reliability. Monitoring these key metrics closely can help in the early prediction of potential failures, allowing for proactive interventions.

**RQ3. How can a hybrid deep learning model be used to improve the efficiency and accuracy of OTN fault prediction?**

A hybrid deep learning model can enhance both efficiency and accuracy by combining the strengths of two DL models. Leveraging the feature extraction capability of CNNs with the multivariate time-series of OTN data handling of LSTMs, BiLSTMs, and GRUs to capture temporal dependencies in the data. The hybrid approach helps in reducing misprediction rates by refining predictions through multiple layers of analysis. The models are better equipped to handle the complexity and high dimensionality of OTN data, ensuring more accurate predictions.

Generally, developing a hybrid DL model for OTN fault prediction involves a systematic approach to model design, feature selection, and training. By combining multiple deep learning techniques, the hybrid model can significantly improve the accuracy and proficiency of fault prediction, ultimately leading to more reliable OTN operations as mentioned in section 6.5 above.

# CHAPTER SEVEN

# 7. CONCLUSIONS AND FUTURE WORKS

## 7.1. Conclusions

In this study, we evaluated OTN faults, their sources, their influence on various network infrastructures, and the techniques used in the literature to detect them. It was found that while a great deal of work has been done on diagnosing and identifying these defects, the majority of the work concentrates on fault detection. Although finding a flaw is helpful, it is a reactive strategy that deals with the issue after it arises. Predicting an issue before it arises is preferable, particularly if it can bring the network to a complete breakdown. This can lower maintenance expenses, avoid expensive downtime, and improve the reliability of vital systems.

In this research, we proposed a proactive method that uses hybrid deep learning models to forecast the possible appearance of faults. To forecast future values, the method involves developing a hybrid CNN-LSTM, CNN-BiLSM, and CNN-GRU model that is trained using a multivariate time series of OTN data.

## 7.2. Contributions of the Study

The study "Prediction and Classification of OTN Faults Using Hybrid Deep Learning Model" makes numerous significant contributions to the field of OTNs and network management as a whole. This thesis undertakes a comprehensive comparison with an emphasis on applying CNN-LSTM, CNN-BiLSTM, and CNN-GRU architectures. By predicting faults before they occur, the model facilitates proactive maintenance strategies, effectively shifting the focus from reactive to preventive maintenance. This shift not only helps to avoid network disruptions but also significantly reduces operational costs associated with emergency repairs. Additionally, improving fault prediction capabilities can further minimize these costs while extending the lifespan of optical network components. The study includes a thorough review of the literature on network faults, their impacts, and the methodologies employed to identify them. Ultimately, this research contributes to the broader objective of enhancing the reliability and performance of OTNs. By reducing the likelihood of faults and improving the speed and accuracy of fault management, the study supports the development of more resilient and efficient optical

networks, which are essential for meeting the growing demands of modern communication infrastructure.

## 7.3. Future Work

In future work, several directions can be explored to further improve OTN management. One promising area is expanding the dataset to contain various and complex real-world fault scenarios. This could involve gathering data from different vendors, such as Huawei, Cisco, and Ciena, in addition to Ericsson, to enhance the generalization capability of the model across various network environments. To improve the hybrid deep learning model, transfer learning could be explored by pre-training the model on one network or vendor's data and fine-tuning it on another.

After successfully predicting faults in OTN, an important next step in enhancing the reliability and efficiency of the network is to classify faults into specific categories. Fault classification involves identifying the type or cause of the faults. This classification can significantly improve fault management by enabling more targeted and efficient troubleshooting, reducing downtime, and preventing future occurrences of similar faults.

# References


Abdulwahab, H. M., Ajitha, S., & Saif, M. A. N. (2022). Feature selection techniques in the context of big data: taxonomy and analysis. *Applied Intelligence*, 13568–13613. https://doi.org/10.1007/s10489-021-03118-3

Abidin, D. Z., Nurmaini, S., Firsandava Malik, R., Erwin, Rasywir, E., & Pratama, Y. (2020). RSSI Data Preparation for Machine Learning. *Proceedings - 2nd International Conference on Informatics, Multimedia, Cyber, and Information System, ICIMCIS 2020*, 284–289. https://doi.org/10.1109/ICIMCIS51567.2020.9354273

Akbari, A., Awais, M., Bashar, M., & Kittler, J. (2021). How Does Loss Function Affect Generalization Performance of Deep Learning? Application to Human Age Estimation. BT - Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event. *Icml*, 141–151. http://proceedings.mlr.press/v139/akbari21a.html

Akpatsa, S. K., Li, X., & Lei, H. (2021). A Survey and Future Perspectives of Hybrid Deep Learning Models for Text Classification. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): Vol. 12736 LNCS* (Issue August). Springer International Publishing. https://doi.org/10.1007/978-3-030-78609-0_31

Al Nuaimi, N., & Masud, M. M. (2017). Toward optimal streaming feature selection. *Proceedings - 2017 International Conference on Data Science and Advanced Analytics, DSAA 2017, 2018-Janua*, 775–782. https://doi.org/10.1109/DSAA.2017.81

Aleksic, S., & Member, S. (2015). *Towards Fifth-Generation ( 5G ) Optical Transport Networks*. 1–4.

Antonio, R. M., Bonenfant, P., Baroni, S., & Wu, R. (2000). Optical data networking: Protocols, technologies, and architectures for next generation optical transport networks and optical internetworks. *Journal of Lightwave Technology*, *18*(12), 1855–1870. https://doi.org/10.1109/50.908757

Badola, A., Nair, V. P., & Lal, R. P. (2020). An Analysis of Regularization Methods in Deep

Neural Networks. *2020 IEEE 17th India Council International Conference, INDICON 2020*, 0–5. https://doi.org/10.1109/INDICON49873.2020.9342192

Bai, Y. (2022). RELU-Function and Derived Function Review. *SHS Web of Conferences*, *144*, 02006. https://doi.org/10.1051/shsconf/202214402006

Bekele, Y. (2022). *Fault Prediction in Optical Transport Network using Machine Learning Algorithms - Case of Ethio Telecom Fault Prediction in Optical Transport Network using Machine Learning Algorithms - Case of Ethio Telecom*. *January*.

Budka, K. C., Deshpande, J. G., Doumi, T. L., Madden, M., & Mew, T. (2010). *and Design Principles for Smart Grids*. *15*(2), 205–227. https://doi.org/10.1002/bltj

Chen, B., Hong, D., Wang, L., Ou, Y., Tan, Y., Li, X., & Zhong, X. (2021). A high-reliability optical network architecture based on wavelength division multiplexing passive optical network. *Optoelectronics Letters*, *17*(7), 422–426. https://doi.org/10.1007/s11801-021-0174-7

Chen, J., Xiao, W., Li, X., Zheng, Y., Huang, X., Huang, D., & Wang, M. (2022). A Routing Optimization Method for Software-Defined Optical Transport Networks Based on Ensembles and Reinforcement Learning. *Sensors*, *22*(21). https://doi.org/10.3390/s22218139

Chernykh, V., Stepnov, A., & Lukyanova, B. O. (2021). Data preprocessing for machine learning in seismology. *CEUR Workshop Proceedings*, *2930*(October), 119–123.

Cui, L. hua, Zhao, Y., Yan, B., Liu, D., & Zhang, J. (2019). *Deep-learning-based failure prediction with data augmentation in optical transport networks*. *February 2019*, 232. https://doi.org/10.1117/12.2523136

Dang, C. N., Moreno-García, M. N., & De La Prieta, F. (2021). Hybrid Deep Learning Models for Sentiment Analysis. *Complexity*, *2021*, 167–193. https://doi.org/10.1155/2021/9986920

Destras, O., & Nicolescu, G. (2023). *Survey on Activation Functions for Optical Neural Networks*. *56*(2), 1–30. https://doi.org/10.1145/3607533

Djomadji, D., Michel, E., Clovis, T. N., & Christian, T. T. (2023). *Machine Learning-Based Alarms Classification and Correlation in an Machine Learning-Based Alarms Classification and Correlation in an SDH / WDM Optical Network to Improve Network Maintenance*. *February*. https://doi.org/10.4236/jcc.2023.112009

Du, W., Cote, D., Barber, C., & Liu, Y. (2021). Forecasting loss of signal in optical networks with machine learning. *Journal of Optical Communications and Networking*, *13*(10), E109–E121. https://doi.org/10.1364/JOCN.423667

Grobe, K., Optical, A., & Se, N. (2018). *Wavelength Division Multiplexing* (Vol. 1). https://doi.org/10.1016/B978-0-12-803581-8.09471-6

Hewamalage, H., Bergmeir, C., & Bandara, K. (2021). Recurrent Neural Networks for Time Series Forecasting: Current status and future directions. *International Journal of Forecasting*, *37*(1), 388–427. https://doi.org/10.1016/j.ijforecast.2020.06.008

Huang, J., Zhu, B., Zhou, H., Zheng, Q., Chen, Z., Lin, T., Zhao, Y., Dong, Z., Gao, X., & Zhao, Y. (2021). OSNR Prediction Based on Federal Learning in Multi-Domain Optical Networks. *Frontiers in Artificial Intelligence and Applications*, *345*, 612–620. https://doi.org/10.3233/FAIA210454

Hutter, F. (2017). Parameter Optimization. In *Interdisciplinary Mathematical Sciences* (Vol. 19). https://doi.org/10.1142/9789814630146_0014

Istiadi, Effendy, D. U., Sulistiarini, E. B., & Joegijantoro, R. (2017). A knowledge base repository model for multiple domain problems of distributed expert system. *Proceedings - 2017 International Conference on Sustainable Information Engineering and Technology, SIET 2017*, *2018-Janua*, 292–296. https://doi.org/10.1109/SIET.2017.8304151

Jaudet, M., Iqbal, N., & Hussain, A. (2004). Neural networks for fault-prediction in a telecommunications network. *Proceedings of INMIC 2004 - 8th International Multitopic Conference*, 315–320. https://doi.org/10.1109/INMIC.2004.1492896

Jiao, Y., Ho, P. H., Lu, X., You, Y., Tapolcai, J., & Peng, L. (2023). A Novel Framework of Failure Localization in Optical Transport Network. *IEEE Communications Magazine*, *December*, 142–147. https://doi.org/10.1109/MCOM.006.2300243

Joseph, V. R., & Vakayil, A. (2022). SPlit: An Optimal Method for Data Splitting. *Technometrics*, *64*(2), 166–176. https://doi.org/10.1080/00401706.2021.1921037

Khurana, U., Samulowitz, H., & Turaga, D. (2018). Feature engineering for predictive modeling using reinforcement learning. *32nd AAAI Conference on Artificial Intelligence, AAAI 2018*, 3407–3414. https://doi.org/10.1609/aaai.v32i1.11678

Le, X. H., Ho, H. V., Lee, G., & Jung, S. (2019). Application of Long Short-Term Memory (LSTM) neural network for flood forecasting. *Water (Switzerland)*, *11*(7). https://doi.org/10.3390/w11071387

Li, Z. (2022). *Design of an OTN-based Failure / Alarm Propagation Simulator by*.

Liu, P., Ji, W., Liu, Q., & Xue, X. (2023). AI-Assisted Failure Location Platform for Optical Network. *International Journal of Optics*, *2023*. https://doi.org/10.1155/2023/1707815

Liu, T., Mei, H., Sun, Q., & Zhou, H. (2019). *10.23919@Jcc.2019.10.014*. 1–6.

Long, K., Yang, X., Kuang, Y., & Huang, S. (2006). The agile and dynamic optical network architectural evolution and key issues. In *COIN-NGNCON 2006 - The Joint International Conference on Optical Internet and Next Generation Network* (Issue Coin). https://doi.org/10.1109/COINNGNCON.2006.4454465

Maharana, K., Mondal, S., & Nemade, B. (2022). A review: Data pre-processing and data augmentation techniques. *Global Transitions Proceedings*, *3*(1), 91–99. https://doi.org/10.1016/j.gltp.2022.04.020

Manias, D. M., Javadtalab, A., Naoum-Sawaya, J., & Shami, A. (2023). The Role of Optical Transport Networks in 6G and Beyond: A Vision and Call to Action. *Journal of Sensor and Actuator Networks*, *12*(3). https://doi.org/10.3390/jsan12030043

Marín-Tordera, E., Masip-Bruin, X., Sánchez-Lopez, S., Solé-Pareta, J., & Domingo-Pascual, J. (2006). The prediction-based routing in optical transport networks. *Computer Communications*, *29*(7), 865–878. https://doi.org/10.1016/j.comcom.2005.08.008

Mcdermott, T. C. (n.d.). *9:OOam - 9:30am Optical*. *2*, 37–38.

Molefe, M. E., & Tapamo, J. R. (2023). Road-Type Classification with Deep AutoEncoder.

*Computational Intelligence and Neuroscience*, *2023*, 1–14. https://doi.org/10.1155/2023/1456971

Mulu, A., Ketema, S., Gemeda, A., & Janaki, P. (2024). Prediction and classification of IoT sensor faults using hybrid deep learning model. *Discover Applied Sciences*. https://doi.org/10.1007/s42452-024-05633-7

Networks, O. (2008). *Introduction to Optical Networks*. 1–43. https://doi.org/10.1016/B978-0-12-374092-2.50009-6

Nyarko-boateng, O. (2020). *Predicting the actual location of faults in underground optical networks using linear regression*. *July*, 1–13. https://doi.org/10.1002/eng2.12304

Panayiotou, T., Chatzis, S. P., & Ellinas, G. (2018). Leveraging statistical machine learning to address failure localization in optical networks. *Journal of Optical Communications and Networking*, *10*(3), 162–173. https://doi.org/10.1364/JOCN.10.000162

Paraschis, L., Bock, H., Shivakumar, A., & Sharfuddin, S. (2020). System innovations in open WDM DCI networks. *Photonic Network Communications*, *40*(3), 269–280. https://doi.org/10.1007/s11107-020-00888-7

Patell, J. K., Kim, S. U., Su, D. H., Subramaniam, S., & Choi, H. A. (2001). A framework for managing faults and attacks in all-optical transport networks. *Proceedings - DARPA Information Survivability Conference and Exposition II, DISCEX 2001*, *2*, 137–145. https://doi.org/10.1109/DISCEX.2001.932166

Paul, A., Mukherjee, A., Naskar, M. K., & Lake, S. (n.d.). *Fault and Attack Management in Optical Network 1*. 2–3.

Qiu, S., Cui, X., Ping, Z., Shan, N., Li, Z., Bao, X., & Xu, X. (2023). Deep Learning Techniques in Intelligent Fault Diagnosis and Prognosis for Industrial Systems: A Review. *Sensors*, *23*(3). https://doi.org/10.3390/s23031305

Rafidison, M. A., Ramafiarisona, H. M., Randriamitantsoa, P. A., Rafanantenana, S. H. J., Toky, F. M. R., Rakotondrazaka, L. P., & Rakotomihamina, A. H. (2023). Image Classification Based on Light Convolutional Neural Network Using Pulse Couple Neural

Network. *Computational Intelligence and Neuroscience*, *2023*, 1–17. https://doi.org/10.1155/2023/7371907

Rahman, M. R., Mandal, P., Mahbub, A. Al, & Mridul, M. K. (2022). *Designing of the Submarine Cable Backhaul Optical Transport Network and the Prediction of OSNR using Artificial Neural Network*. *4*(5), 1–18.

Šafránková, J., Annual Conference of Doctoral Students (19 2010.06.01-04 Prague), WDS'10 (19 2010.06.01-04 Prague), & Week of Doctoral Students 2010 (19 2010.06.01-04 Prague). (2010). 19th Annual Conference of Doctoral Students, WDS'10 "Week of Doctoral Students 2010", Charles University, Faculty of Mathematics and Physics, Prague, Czech Republic, June 1, 2010 to June 4, 2010 : [proceedings of contributed papers]. Pt. 1 Mathematics and., *1999*(December), 31–36.

Saif, W., Esmail, M. A., Ragheb, A., Alshawi, T., & Alshebeili, S. (2020). *Machine Learning Techniques for Optical Performance Monitoring and Modulation Format Identification : A Survey*. *Ml*, 1–46. https://doi.org/10.1109/COMST.2020.3018494

Sheetal, S. (2024). *Fault Identification in Optical Transport Network Using CNN Related Work :* 2151–2156.

Shulman, D. (2023). *Optimization Methods in Deep Learning: A Comprehensive Overview*. 1–5. http://arxiv.org/abs/2302.09566

Silva, G. K., Souza, R. L., & Ribeiro, S. (2023). *Systematic Review on Methods for OTN Network Planning*. 0–10.

Silva, M. F., Pacini, A., Sgambelluri, A., & Valcarenghi, L. (2022). Learning Long-and Short-Term Temporal Patterns for ML-Driven Fault Management in Optical Communication Networks. *IEEE Transactions on Network and Service Management*, *19*(3), 2195–2206. https://doi.org/10.1109/TNSM.2022.3146869

Skubic, B., Öhlén, P., Rostami, A., Ghebretensaé, Z., Lindqvist, N., & Laraqui, K. (2015). Optical transport and challenges in the Networked Society. *Photonic Network Communications*, *29*(3), 322–329. https://doi.org/10.1007/s11107-015-0501-7

Soumare, H., Benkahla, A., & Gmati, N. (2021). Deep learning regularization techniques to genomics data. *Array*, *11*(May), 100068. https://doi.org/10.1016/j.array.2021.100068

Systems, D. (2019). *ITU-T*. *872*.

Szostak, D., Włodarczyk, A., & Walkowiak, K. (2021). Machine learning classification and regression approaches for optical network traffic prediction. *Electronics (Switzerland)*, *10*(13). https://doi.org/10.3390/electronics10131578

Tan, Z., & Pan, P. (2019). Network fault prediction based on CNN-LSTM hybrid neural network. *Proceedings - 2019 International Conference on Communications, Information System, and Computer Engineering, CISCE 2019*, 486–490. https://doi.org/10.1109/CISCE.2019.00113

Tian, Y., Lian, Z., Wang, P., Wang, M., Yue, Z., & Chai, H. (2024). Application of a long short-term memory neural network algorithm fused with Kalman filter in UWB indoor positioning. *Scientific Reports*, *14*(1), 1–14. https://doi.org/10.1038/s41598-024-52464-y

Torres, J. F., Hadjout, D., Sebaa, A., Martínez-Álvarez, F., & Troncoso, A. (2021). Deep Learning for Time Series Forecasting: A Survey. *Big Data*, *9*(1), 3–21. https://doi.org/10.1089/big.2020.0159

Vaid, K., & Ghose, U. (2020). Predictive Analysis of Manpower Requirements in Scrum Projects Using Regression Techniques. *Procedia Computer Science*, *173*, 335–344. https://doi.org/10.1016/j.procs.2020.06.039

Wang, D., Zhang, C., Chen, W., Yang, H., Zhang, M., Pak, A., & Lau, T. (2022). *A review of machine learning-based failure management in optical networks*. *65*(November).

Wang, P. (2020). *A Deep Learning System for Service Fault Prediction on Optical Transport Networks*.

Wang, Q., Ma, Y., Zhao, K., & Tian, Y. (2022). A Comprehensive Survey of Loss Functions in Machine Learning. *Annals of Data Science*, *9*(2), 187–212. https://doi.org/10.1007/s40745-020-00253-5

Wang, Z., Zhang, M., Wang, D., Song, C., Liu, M., Li, J., Lou, L., & Liu, Z. (2017). Failure

prediction using machine learning and time series in optical network. *Optics Express*, *25*(16), 18553. https://doi.org/10.1364/oe.25.018553

Webb, G. I. (2017). *Data Preparation*.

Xiangyang, L., Xing, Q., Han, Z., & Feng, C. (2023). A Novel Activation Function of Deep Neural Network. *Scientific Programming*, *2023*, 1–12. https://doi.org/10.1155/2023/3873561

Zabin, M., Choi, H. J., & Uddin, J. (2023). Hybrid deep transfer learning architecture for industrial fault diagnosis using Hilbert transform and DCNN–LSTM. *Journal of Supercomputing*, *79*(5), 5181–5200. https://doi.org/10.1007/s11227-022-04830-8

Zhang, B., Zhao, Y., Rahman, S., Li, Y., & Zhang, J. (2020). Optical Fiber Technology Alarm classi fi cation prediction based on cross-layer arti fi cial intelligence interaction in self-optimized optical networks ( SOON ). *Optical Fiber Technology*, *57*(March), 102251. https://doi.org/10.1016/j.yofte.2020.102251

Zhang, M., & Wang, D. (2019). Machine Learning Based Alarm Analysis and Failure Forecast in Optical Networks. *OECC/PSC 2019 - 24th OptoElectronics and Communications Conference/International Conference Photonics in Switching and Computing 2019*, *1*, 1–3. https://doi.org/10.23919/PS.2019.8817991