Addis Ababa University

Addis Ababa Institute of Technology

School of Electrical and Computer Engineering

Telecommunication Engineering Graduate Program

Fault Prediction in Optical Transport Network using Machine Learning Algorithms - Case of Ethio Telecom

By

Yonas Bekele

Advisor

Yalemzewd Negash (PhD)

A Thesis Submitted to School of Electrical and Computer Engineering in Partial Fulfillment of the Requirements for the Degree of Master of Science in Telecommunication Network Engineering.

January 2022

Addis Ababa, Ethiopia

Addis Ababa University

Addis Ababa Institute of Technology

School of Electrical and Computer Engineering

Telecommunication Engineering Graduate Program

Fault Prediction in Optical Transport Network using Machine Learning Algorithms - Case of Ethio Telecom

By: Yonas Bekele

Approval by Board of Examiners

Bisrat Derebssa (PhD)_____	_____	_____
Chairman / School Dean	Signature	Date
Yalemzewd Negash (PhD)__	_____	_____
Advisor	Signature	Date
Murad Ridwan (PhD)_____	_____	_____
Examiner	Signature	Date
Yihenew Wondie (PhD)____	_____	_____
Examiner	Signature	Date

Declaration

I, the undersigned, declare that this thesis is my original work, has not been presented for a degree in this or any other university, and all sources of materials used for the thesis have been properly acknowledged.

Yonas Bekele_____ _____

Name Signature

_____

Date of submission

Addis Ababa

This thesis has been submitted for examination with my approval as a university advisor.

Yalemzewd Negash (PhD)__ _____

Advisor's Name Signature

Abstract

The availability of Optical Transport Network (OTN) is an important factor to provide dependable telecom services. This backbone infrastructure of ethio telecom suffers from frequent outages and fault alarm data are exploding in magnitude. Hence, in this research machine learning based fault interarrival time prediction models have been developed using Artificial Neural Network(ANN), Long Short-Term Memory(LSTM) and Gated Recurrent Unit(GRU) algorithms to select the best model that captures the overall fault interarrival pattern and help take preventive actions before its occurrence. Further, the models have been trained using 70/30 and 80/20 training schemes to study their stability and perform models comparison based on five months fault history data. The prediction performance of these models have been evaluated using Mean Squared error (MSE), Mean Absolute Error (MAE), Mean Absolute Percentage Error(MAPE) and R-Square (R^2).

The models prediction performances indicate that the ANN model has predicted the fault interarrival time with 16.08% MAPE while GRU shows 16.23% MAPE and LSTM model has 17.93% MAPE on average. The R^2 score for the ANN model falls between 0.2576 and 0.4923 followed by the GRU model that exhibits its score in between 0.2387 and 0.4912. For LSTM model its R^2 score lies between 0.1583 and 0.4641. Therefore, the ANN model is able to catch the general trend of the fault interarrival time followed by the GRU model. In addition, the average computational time taken per epoch by ANN model is 2.44 ms, GRU model consumes 10.22 ms and LSTM takes 13.94 ms. Hence, ANN model outperforms in terms of prediction performance and computational efficiency.

Keywords- *Fault Prediction, Optical Transport Network, Network Management System (NMS), Machine Learning, Artificial Neural Network (ANN), Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU).*

Acknowledgement

First and foremost, I would like to thank God for giving me the strength and courage to accomplish my study.

I would like to express my sincere thankfulness to my advisor Yalemzewd Negash (PhD) for his unreserved guidance and encouragement on my thesis work. My gratitude also extends to my evaluators Murad Ridwan (PhD) and Yihenew Wondie (PhD) for their constructive comments and feedbacks to help improve this research.

I would like to thank my company, Ethio Telecom, for sponsoring and providing me the opportunity to further enhance my educational career at Addis Ababa University. I am also very grateful to my colleagues at Ethio Telecom for their unlimited support in providing the necessary information and data.

Finally, my love and special gratitude goes to my family for their courage and support.

Table of Contents

List of Figure

List of Tables

List of Acronyms

3G	Third Generation
4G	Fourth Generation
AdaGrad	Adaptive Gradient
Adam	Adaptive Moment Estimation
ALS	Automatic Laser Shutdown
ANN	Artificial Neural Network
API	Application Programming Interface
ARIMA	Auto Regressive Integrated Moving Average
ARMA	Auto Regressive Moving Average
CMSE	Cumulative Mean Square Error
CNN	Convolutional Neural Network
CNTK	Microsoft Cognitive Toolkit
DWDM	Dense Wavelength Division Multiplexing
GARCH	Generalized Auto Regressive Conditional Heteroskedasticity
GRU	Gated Recurrent Unit
IoT	Internet of Things
IP	Internet Protocol
ITU-T	International Telecommunications Union
LSTM	Long Short Term Memory
MAE	Mean Absolute Error
MATLAB	Matrix Laboratory
ML	Machine Learning
MPLS	Multiprotocol Label Switching

MSE	Mean Squared Error
NAR	Nonlinear Auto Regressive
NE	Network Element
NMS	Network Management System
NRMSE	Normalized Root Mean Squared Error
NumPy	Numerical Python
OTN	Optical Transport network
PCA	Principle Component Analysis
QoS	Quality of Service
R^2	R-squared
ReLU	Rectified Linear Unit
RMSE	Root Mean Squared Error
RMSProp	Root Mean Square Propagation
RNN	Recurrent Neural Network
SciPy	Scientific Python
SDH	Synchronous Digital Hierarchy
SLA	Service Level Agreement
SSE	Sum of Squared Error
SST	Total Sum of Squares
STM-N	Synchronous Transport Module level-N
SVM	Support Vector Machine
TanH	Tangent Hyperbolic Function

Chapter 1. Introduction

Fault prediction plays an important role on improving the operational reliability and availability of optical networks. Faults in optical networks cause service interruption and huge data loss. The reliability of optical backbone network is crucial for telecom operators as more traffics being carried on the infrastructure and to provide the required quality of service to customers. To support the growing demand of high capacity transmission links due to the booming data traffic, broadband services and bandwidth intensive services Ethio telecom has deployed Optical Transport Network (OTN) as backbone transmission infrastructure. Despite the best effort to maintain and protect the optical backbone network from multiple faults, the network suffers from frequent faults and unavailability.

A fault in a single optical link may lead to huge economic loss to the operator. The optical network protection algorithms take passive protection actions, which are not capable of preventing the faults in advance. These faults are causing difficulties for the operator to transmit different classes of services with high quality and reliability. The network service interruption can be planned outages that are often scheduled in advance to perform certain tasks on specific date, duration and services to be affected in detail. This gives the possibility to perform certain tasks including upgrading of the system, traffic migration, etc. On the other hand, unplanned outages occur due to failures of Network Equipment (NE) hardware, software, power supply, fiber bend or cut, lossy fiber, optical transponder and transceiver module failures, etc. The availability of backbone optical transmission affects the availability of network elements that are directly connected in the backbone network like IP backbone router that uses the optical network for its transport medium and other indirectly connected access network devices. Hence, a fault that occurs in the optical backbone network can be considered as fault in other connected

network as it can affect their performance and availability. Because of frequent optical backbone network faults, Quality of Service (QoS) is affected and Ethio Telecom loses huge amount of its income and incurs high maintenance cost leading the network to operate in sub optimal state. In telecom networks usually reactive and preventive maintenance are performed. Reactive maintenance is carried out when there is fault in network or service failure alarm is triggered on the Network Management System (NMS). Preventive maintenance is characterized by scheduled maintenance operations in order to avoid optical nodes, links, etc. from failures. Therefore, by performing fault occurrence time prediction in optical network preventive actions can be taken to links and nodes recognized as the most common generators of network faults to reduce the number of reported fault alarms and the operational expenses of the operator. Further, the operator can perform time, work force and resource scheduling to ensure services availability and reliability of the optical transport network.

1.1 Statement of the Problem

Ethio Telecom is facing network unavailability challenges across various networks. From these networks, optical transport network is the main contributor for outages and service interruption due to lossy fiber, software bugs, faults in transponders, transceivers, power supply, etc. Furthermore, due to the frequent fiber cut incidents these crucial telecom infrastructures are affecting the quality of optical transmission network. The faults not only have direct revenue loss to the telecom operator but also increased maintenance cost and degradation of service quality provided to customers. These have made the company to become hugely challenged to provide good Quality of Service (QoS) and meet Service Level Agreement (SLA).

The amount of fault is also associated with the number of nodes, optical channels, topology and varieties of hardware and software that constitute the optical network. Hence, there should be a mechanism for proactive prediction of fault occurrence time that could help the operational and maintenance personnel in quick prevention and removal of possible fault causes to reduce the amount of faults. Therefore, machine learning based fault interarrival time prediction is proposed in this particular research based on the collected data and incidents that occurred in the optical transport network.

1.2 Objective

1.2.1 General Objective

The main objective of the research is to perform machine learning based fault interarrival time prediction in optical transport network using Artificial Neural Network (ANN), Long Short Term Memory (LSTM) and Gated Recurrent Unit (GRU) algorithms.

1.2.2 Specific Objective

The specific objectives of the research are as follows:

- Review fault prediction techniques and algorithms
- Collect and preprocess optical fault data
- Model development using ANN, LSTM and GRU algorithms
- Perform sequential fault time interval prediction using the models
- Evaluate each models with error metrics like Mean Squared Error (MSE), Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE) and R-square(R^2)
- Compare models and recommend the one with better prediction performance

1.3 Methodology

The methodologies used to achieve the general and specific objectives are:

- **Literature Review**

Different literatures i.e. books, articles, conference proceedings, and industry white papers have been reviewed to understand state of the art in fault prediction research area.

- **Data Collection**

Fault alarm data that occurred in the Optical Transport Network (OTN) has been collected from the Network Monitoring System (NMS) for the research.

- **Data Preprocessing**

MS-Excel tool is used to collect, extract and sequentially arrange the fault occurrence time. Jupyter notebook is used to pre-process and perform fault inter-arrival time prediction.

- **Model Development**

To perform the fault interarrival time prediction, models have been developed using the selected Artificial Neural Network (ANN), Long Short Term Memory (LSTM) and Gated Recurrent Unit (GRU) algorithms.

- **Model Evaluation**

The three models prediction of fault interarrival time have been evaluated using the evaluation metrics described in section 4.5.

- **Analysis and Conclusion**

This part includes analysis of the results and conclusion drawn from the research. Finally, recommendations for future work have been suggested and the model that captures the actual fault pattern recommended as the best model.

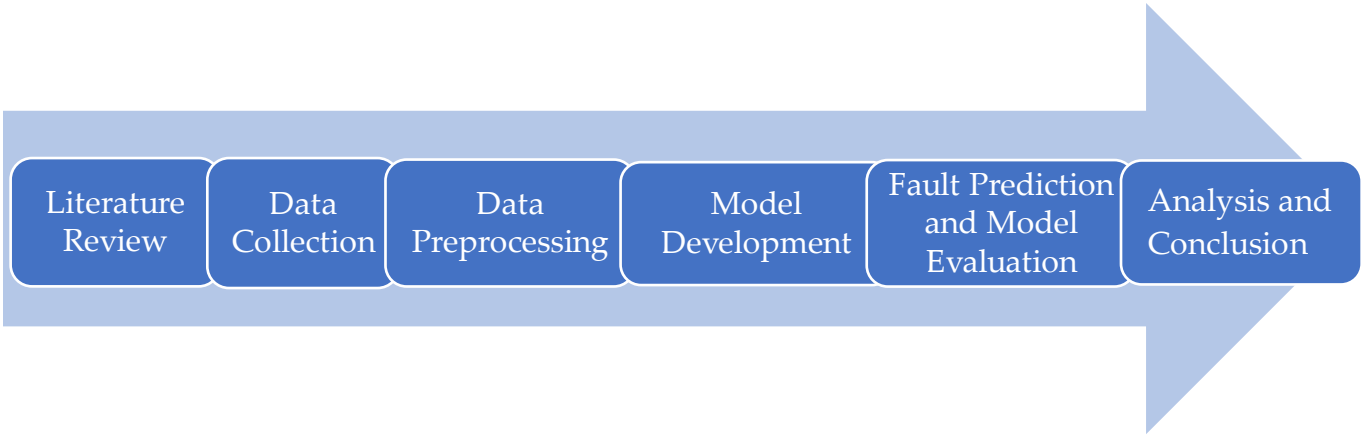

Figure 1. 1: Methodology Summary

1.4 Literature Review

Fault prediction using machine-learning techniques is becoming a good research area in telecommunication networks in recent times. Despite the great efforts to enhance the availability of the optical networks using the protection and restoration schemes, fault occurrence time prediction also aids to proactively reduce the amount of faults expected to occur in the optical network.

The authors in [1] performed fault prediction in nationwide optical backbone network by collecting alarms coming directly from all devices with their topological informations. They extracted critical alarms and the time differences between two consecutive alarms have been accounted for prediction using neural network toolbox in MATLAB. A four layer feed forward neural network trained with Levenberg Marquardt back propagation algorithm showed better convergence of Mean Squared Error (MSE) to 10^{-10}. They also tried Recurrent Neural Network (RNN) of the same structure but the predicted values were not as accurate as feed forward network. After several trials, they were able to predict the future fault occurrence time close to the actual one. The error in prediction of each consecutive fault is less than 0.5 milliseconds, which is a high precision result.

The authors in [2] performed prediction of fault interarrival times in cellular network based on one-month faults data from one of the national mobile operators. They investigated the prediction accuracy of neural network, linear regression and other five models using Normalized Root Mean Squared Error (NRMSE) and Root Mean Squared Error (RMSE) as performance evaluation metrics using MATLAB. The deep neural network with auto encoders and nonlinear autoregressive neural network models attained better prediction accuracy than the others.

The authors in [3] collected ten months wireless network log of a company to predict whether the network will fail in the near future using different time windows. In their experiment, they proposed a hybrid Convolutional Neural Network and Long Short Term Memory (CNN-LSTM) prediction model for the wireless network faults. Further, the hybrid model performance was compared with CNN and Random Forest models. The result shows that the prediction performance of the hybrid CNN-LSTM model is better than the other two models.

The authors in [4] performed fault occurrence time prediction in wireless mobile network used for train-ground communication known as GSM for Railways (GSM-R). They proposed Hidden Markovian Chain (HMC) framework that consists a scheme based on Viterbi algorithm and absorption probability to predict the fault occurrence time. Viterbi algorithm was utilized to estimate system state. A dynamic correction mechanism based on GSM-R quality was used to modify the predicted occurrence time. The dynamic scheme possesses more prediction accuracy than its static version.

The authors in [5] collected log data obtained from a large scale Information Technology (IT) system to predict the time period when a failure is expected to occur using Bayesian learning from log messages and past failure reports. The results show that their method

predicted the occurrence of failure with time interval of 60 minutes before the occurrence of failures with sufficient accuracy.

The authors in [10] applied Auto Regressive Integrated Moving Average (ARIMA) model to find prediction formulae in order to evaluate the expected number of faults at time t with t-1, t-2, t-3, etc. in a broadband network. ARIMA (4, 1, 5) model achieved better result with the least Cumulative Mean Square Error (CMSE) in each fault types studied.

The author in [11] collected five months of 3G mobile sites fault history data to predict fault occurrence time based on Nonlinear Auto Regressive (NAR) neural network time series prediction model. The fault inter-arrival times data were used to train the Neural Network toolbox in MATLAB and achieved 90.71% in prediction accuracy.

1.5 Scope and Limitation

1.5.1 Scope

The scope of the research contains model building, training and testing along with performance evaluation and comparison of the Artificial Neural Network (ANN), Long-Short Term Memory (LSTM) and Gated Recurrent Unit (GRU) models.

1.5.2 Limitation

The research is limited to a fault interarrival time prediction in optical transport network that occurred in single vendor equipments. The fault data exported from the Network Management System (NMS) has a five months duration.

1.6 Contribution

A fault that occurs in the optical transport network can disrupt the availability of other connected network elements as the aggregate traffic being carried by the backbone optical infrastructure. By performing fault occurrence time predictions, proactive actions can be

taken to minimize the amount of faults, maintenance time and cost can be reduced, and preventive maintenance can specifically be performed in optical nodes and links where traffic disruptions are common. Therefore, the availability of the overall optical transport network increases.

1.7 Thesis Layout

The thesis contains six chapters.

Chapter one deals with introduction of the whole paper. It clearly cites statement of the problem, objective of the study, literature review, methodologies used, contribution, scope and limitation of the paper.

Chapter two presents overview of the Optical Transport Network (OTN) along with evolutional stages, OTN Network Management System, fault reasons in OTN and fault severity level description.

Chapter three provides an overview of machine learning algorithms that are suitable for fault prediction purpose.

Chapter four presents about problem formulation/ system model of the research, data collection, data preparation techniques for machine learning, time series concepts and model performance evaluation metrics.

Chapter five presents the details of the experiment, results and analysis of the research. It describes about essential tools and libraries, model building, hyperparameter tuning, model training and testing, performance evaluation, comparison and analysis of the models developed by the selected algorithms.

Chapter six presents the conclusions and future work.

Chapter 2. Optical Transport Network

2.1 Overview

Telecom operators try to satisfy the growing demand for good quality of services and reliable network. The major driver for high capacity optical transport network is the increasing demand for bandwidth intensive services of residential, mobile and business customers. Residential customers current trend of using social media, online gaming, online video streaming, cloud storage of data etc. require high capacity optical transport network to carry the traffic. Furthermore, the Internet Protocol (IP) based data traffic volume flowing in cellular network like 3G and 4G are enforcing service providers to have higher-capacity optical backbone network. Additionally, business customers, governmental offices, banks, insurance companies, aviation, industrial and electrical power companies etc. demand high bandwidth to support their traffic along with strict Service Level Agreement (SLA) agreements with telecom service providers. Further, the high number of connected Internet of Things (IoT) and other automated systems generate huge traffics volumes that must be transported in the optical network of a telecom company. In order to support the booming data traffic a capacity enhanced optical network is required. Therefore, by combining the key benefits of Synchronous Digital Hierarchy (SDH) and Dense Wavelength Division Multiplexing (DWDM), the Optical Transport Network (OTN) has currently emerged as a solution for the backbone network.

OTN contains set of optical network elements interconnected by optical fibers called nodes. These nodal elements perform multiplexing, switching, amplification, performance management and service transport on optical channels. The OTN is specified in International Telecommunications Union (ITU-T) G.709 standard [34]. On the evolutionary timeline, it is regarded as the fourth generation (4G) optical network [36].

DWDM technologies are used in optical transport networks to achieve transparent transmission of multiple services with high capacity. At present, OTN can multiplex up to 80 channels on a single optical fiber. This means that it can transmit 80 channels of different wavelengths having different line rate signals.

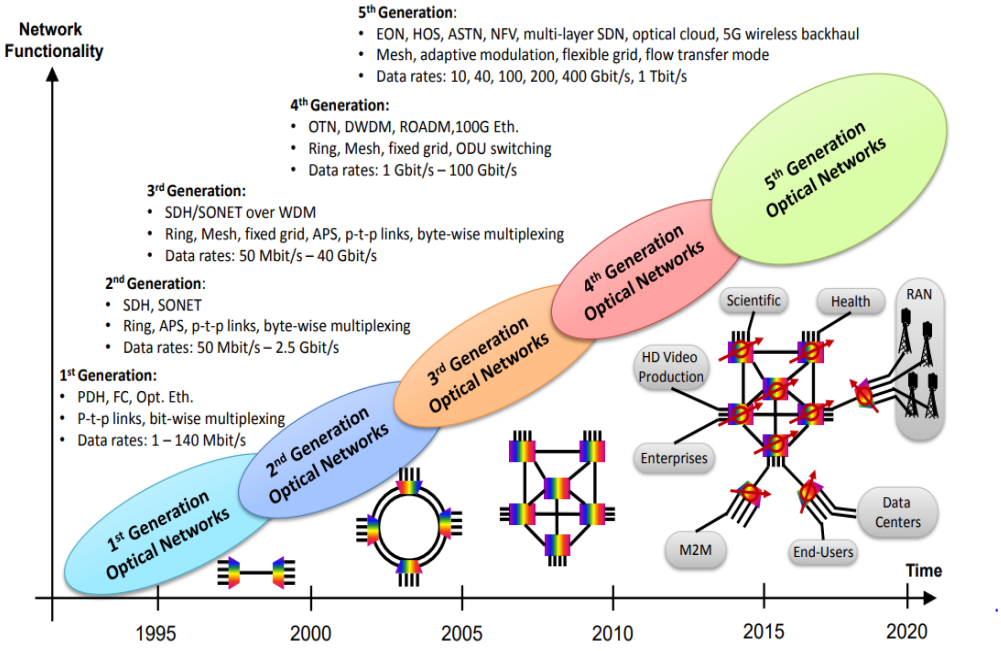

Figure 2. 1: Evolution of Optical Transport Networks [36].

2.1.1 OTN Network Management

The Network Monitoring System (NMS) is used to monitor all the optical transmission equipments in the network. The NMS assists users in the operation, maintenance, and management of optical transmission network. Its management functions are as follows:

Performance Management: monitors the optical performance of the network, collects performance data from the network, stores the performance data, supports threshold management, and provides the current and historical performance data of the network with complete information for the operation and maintenance of the network.

Configuration Management: used to configure and manage interfaces, services, clocks, subnets etc.

Security Management: provides NMS user management, NE login management, security policy management of user rights and domain based management.

Maintenance Management: it collects equipment data to assist maintenance personnel when troubleshooting. It provides loopback, card reset, automatic laser shutdown (ALS) functionalities.

Alarm Management: the network monitoring system promptly notifies the operation and maintenance personnel when the alarm occurs so as to recover the network back to normal working state. The alarm management collects, notifies and counts alarms in real time. Further, it performs fault diagnosis, alarm correlation with NEs, ports, or services.

2.2 Faults in Optical Transport Network

Faults in optical transport network can be attributed to different causes such as fiber bends or cut, power, board, high loss, attenuation, misconfiguration of parameters, amplifier malfunctioning, etc. Moreover, the fault occurrence rate is directly related to number of network elements (NEs), optical channels, transponders, optical transceiver modules, and all the hardware components and softwares that constitute the optical transport network. Faults that occur in component level like the optical transceiver module can occur due to overheating, high input optical power, etc. Network element (NE) level fault can occur due to power system faults occurring in circuit breaker, rectifier and others. Faults can also emerge from interconnected backbone layer devices like Internet Protocol (IP) routers that rely on Optical Transport Network (OTN) for transmitting services. Other access layer devices that are connected to OTN can be reasons for faults exhibited in the network. Service level faults occur when client signals

such as STM-N, IP/MPLS, Ethernet, etc. have high bit error rate, low optical power etc. Hence, the consequences of these faults are poor quality of service, revenue loss, high maintenance cost and sub-optimal state of the optical transport network. Telecom operators utilize reactive approach to maintain and recover the services back to the normal state. This method does not prevent the fault from happening. The reactive approach utilizes fault detection, localization and identification for the fault recovery. Hence, a proactive approach to estimate that a fault is going to occur in a certain amount of time and help to take actions to prevent the fault occurrence is crucial. The Network Monitoring System (NMS) stores optical performance and fault occurrence time data. These informations are stored as time series of events on the NMS. Hence, using time series forecasting models fault interarrival time prediction could be a helpful solution to avoid faults by scheduled preventive maintenance.

2.2.1 Fault Severity Level

Alarm is a notification of a specific fault or abnormal condition that occurs in the network. A prompt message that indicates occurrence of fault is displayed on the screen of the monitoring system. Hence, an operation and maintenance personnel can immediately find out the cause and troubleshoot the faults for proper functioning of a system, quality of service and reliability of the optical transport network. The fault severity level is the potential of a fault to affect services and operational status of the network. It also shows the urgency of the actions to be taken to prevent service from interruption. The alarms can be classified into four levels in the sequence of declining severity levels: critical, major, minor and warning alarms [37]. The critical alarm indicates a fault or an event has occurred that may interrupt services. The major alarm is less severe than the critical alarm but it can cause degradation of quality of services. These two alarms need proper actions as they have the potential to affect the network.

Chapter 3. Machine Learning

3.1 Overview

Machine learning (ML) is a type of artificial intelligence in which computers use set of data to identify and exploit hidden patterns. They utilize algorithms to train themselves, develop their own model that fits these data and make predictions based on the extracted knowledge from the data. ML models perform better at prediction when supplied with more information or data. For prediction accuracy, it is important to consider not only the quantity but also the quality of data. Hence, good quality of data is essential to train and build models on machines. The type of data and the type of activity that needs to be automated affect the choice of algorithms. ML models have proven to influence several industries where it is not able to manually extract information from data. It has been used for business areas to perform sell forecasting, detection of fraudulent credit card etc. ML has influenced health sector in patient diagnosis, disease identification and treatment prescription etc. It has become applicable in foreign language learning and translation in the educational area. ML has also become applicable in security areas for video surveillance, speech and face recognition. Transport sector also uses ML based application for traffic monitoring and vehicle identification, etc.

ML has also a broad range of applications to improve telecommunication network operation, optimization etc. Some of the applications in telecom networks are traffic volume prediction and classification. This is primarily used for congestion control, resource allocation and network routing. ML is also applicable for network performance prediction to evaluate reliability. Further, it has gained success in fault management to locate the root alarm to recover the service back to working state. The other application area in telecom network is fault interarrival time prediction.

According to training supervision, machine-learning systems can be categorized to [25]:

- Supervised learning
- Unsupervised learning
- Reinforcement learning

3.1.1 Supervised Learning

The purpose of supervised learning is to discover the relationships between feature set and the target variables. A model assigns a label to some data by using a training dataset as reference. The input as well as the desired results are included in the training data. In supervised learning, it is critical to construct appropriate training, validation, and test datasets. These learning methods are usually fast and accurate in prediction. The model must be able to provide reasonable output for new inputs not encountered during training. In supervised learning, the machine-learning model task is generating a function that maps the inputs to the desired outputs. It is suitable for classification and regression kind of problems. Neural Networks, Support Vector Machine (SVM) and Logistic Regression are examples of common supervised learning algorithms.

3.1.2 Unsupervised learning

These learning algorithms are opposite to supervised learning that they do not depend on supervisor. The model mainly deals with the unlabeled data and attempts to find patterns and knowledge that was previously undetected. The algorithm receives only the input data and no output data is provided to the algorithm. Its primary target is to explore the hidden patterns and predict the outcome. The algorithm identifies the patterns and learns them. Unsupervised learning aims at clustering, finding association among features, and dimensionality reduction. It is used in text mining, face recognition, etc.

3.1.3 Reinforcement Learning

Reinforcement Learning algorithms are neither supervised nor unsupervised ones. The algorithms try to discover ways how to react to an environment based on the learning system called an agent that observes the dynamic environment to select correct route to reach its goal through trials and errors. It is commonly utilized in a variety of autonomous systems such as intelligent self-driving cars, industrial robotics, gaming, and manufacturing.

3.2 Machine Learning Algorithms

3.2.1 Artificial Neural Network (ANN)

Human brain incorporates billions of neurons that are linked to many different neurons to form a network. Neurons in human brain are capable of performing complex decisions that means they can execute parallel processes. Similarly, neural networks have been developed to create artificially interconnected neurons that function like biological neurons. The simplest computational element is known as node or unit. It takes inputs from outside sources, and each input has a weight assigned to it. The function of the weighted sum of its inputs is computed by the unit. The structure of neural network consists an input layer of neurons, hidden layers and one output layer which are usually interconnected in a feed-forward way.

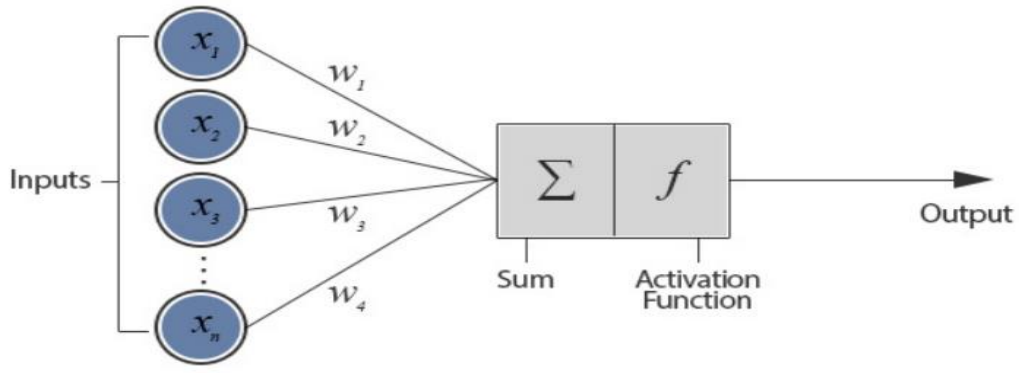

Figure 3. 1: Modeling Artificial Neurons [40]

The neurons proceed to sum all the inputs and calculate an output. The equation 3.1 represents it mathematically [40]:

$$y(x) = f(w \odot x + b) \tag{3.1}$$

Where x, w, b, f, y represent input vector, weight vector, neuron bias, element-wise multiplication, activation function, and neuron output respectively.

Activation functions: are needed to introduce non-linearity and determine the output of neural networks. Four activation functions are available [29].

- **Linear Function**

The linear activation function, which is also known as identity or no activation, does not affect the weighted sum of the inputs. For every layer in our model the activation function of the output layer is a linear function of the input layer. Nonlinear activation functions are the most widely used ones. The linear activation function has an equation similar to a straight line, f(x) = x.

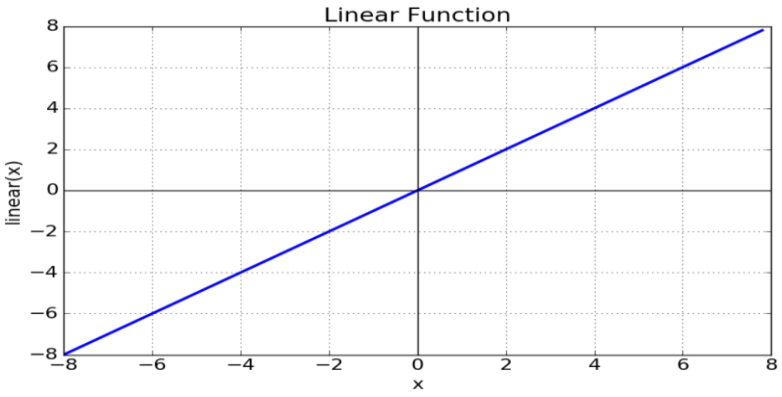

Figure 3. 2: Linear Activation Function [31]

- **Rectified Linear Unit (ReLU)**

For hidden layers, the rectified linear unit (ReLU) is the most commonly used activation function[30]. Since it involves simpler mathematical operations, ReLU is computationally efficient and converges much faster than sigmoid and tanh functions. It is more resistant to vanishing gradients, which inhibits deep learning models from being trained. The rectified linear unit activation function outperforms the sigmoid and tanh activation functions in terms of performance and generalization. Its main limitations is the ability to easily overfit but dropout technique helps to reduce the overfitting.

The equation for ReLU:

$$f(x) = max\ (0,\ x) \tag{3.2}$$

If x is greater than 0, the activation function returns x, else it returns 0. Where x is the sum of the weighted inputs and bias.

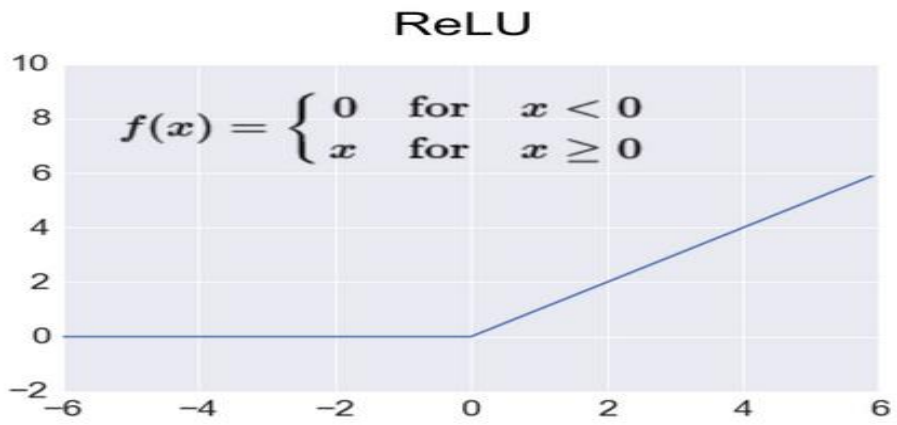

Figure 3. 3: Rectified Linear Unit Activation Function [32]

- **Sigmoid Function**

When the model predicts probability or binary classification, the sigmoid function is used in the output layer. It is also known as logistic function that has an S shape in the range from 0 to 1.

Its major limitations are slow convergence and non-zero centered output that can cause the gradient updates to propagate in different direction.

The sigmoid function is described by:

$$f(x) = \frac{1}{1+e^{-x}} \tag{3.3}$$

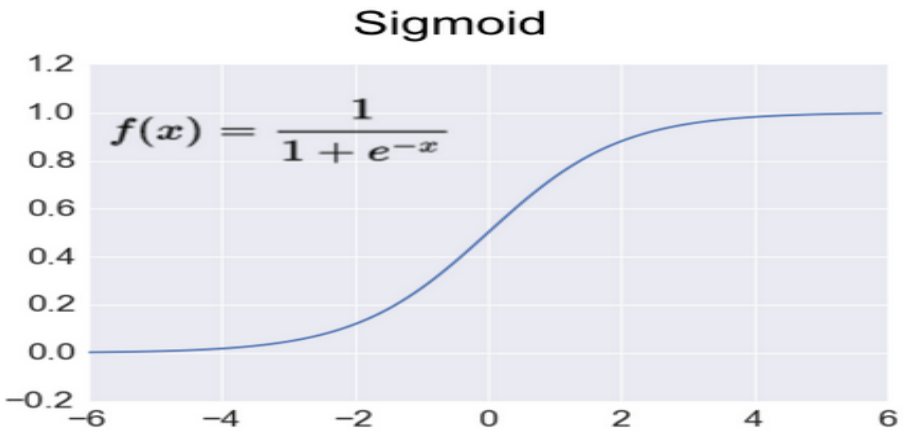

Figure 3. 4: Sigmoid Activation Function [32]

- **Tangent Hyperbolic Function (TanH)**

The hyperbolic tangent function has a zero center and an output value range of –1 to +1. This makes each layer's output to be normalized or centered around 0 which helps to speed up convergence. It is a shifted version of sigmoid function and is an S-shaped activation function.

The hyperbolic tangent function Tanh:

$$f(x) = \frac{2}{1+e^{-2x}} - 1$$

$$= 2\text{Sigmoid}(2x) - 1 \tag{3.4}$$

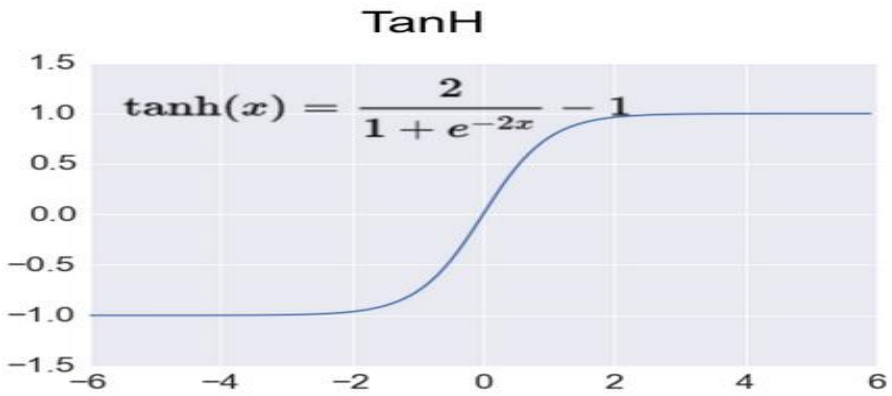

Figure 3. 5: Tangent Hyperbolic Activation Function [32]

3.2.2 Long Short-Term Memory (LSTM)

Long Short Term Memory (LSTM) is a kind of recurrent neural network that can learn and predict sequential data as well as their long-term dependencies. A recurrent neural network (RNN) is a network of repeating modules having very simple structure, such as a single tanh layer, which limits their ability to maintain long-term memory [39]. As a result, by incorporating a memory structure with gates, the LSTM is capable to tackle the drawbacks of short-term memory and vanishing gradients [46]. The LSTM is composed of four interconnected layers that can retain short-term memories for an extremely long time.

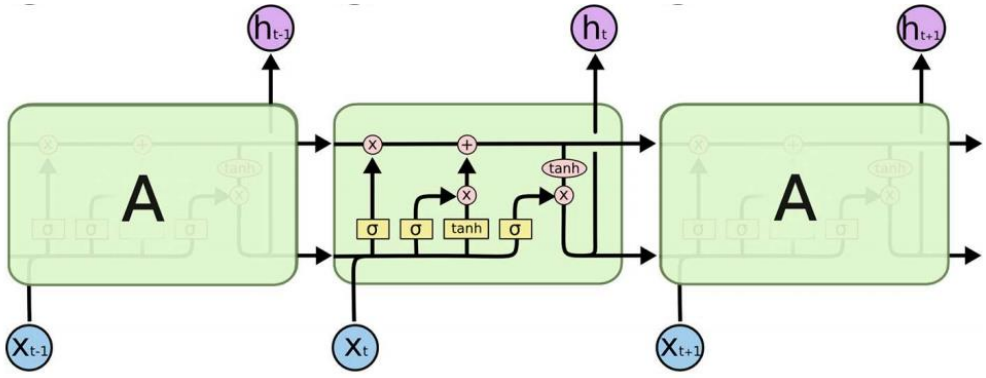

Figure 3. 6: LSTM Modules and Interacting Layers [39]

A cell state, an input gate, an output gate, and a forget gate are all components of an LSTM unit [3]. The information that must be reserved from the previous cell state is determined by the input gate. The forget gate determines which data should be erased. The processed information is used by the output gate to generate the correct output, which is then passed to the next LSTM unit. Cell state is represented by the horizontal line across the top of the diagram. A pictorial representation of a fully connected LSTM unit that contains four layers fed with σ, tanh, the input vector (x_t) and the previous short-term state (h_{t-1}) is depicted in figure (3.7).

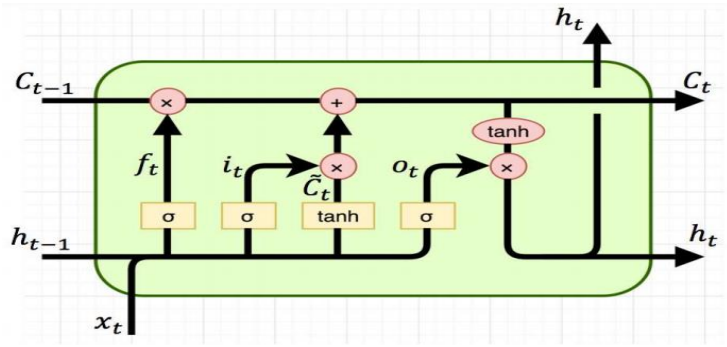

Figure 3. 7: Structure of LSTM Cell [39]

Forget gate: is the first block represented in the LSTM cell structure that erases information that is no longer useful in the cell's current state. The input at specified time (x_t) and the output of the previous cell (h_{t-1}) are fed to the gate, multiplied by weight matrices before bias is added. An activation function is applied to the result, yielding a binary output. If the outcome is close to 0 it means forget, and if it is close to 1 it means keep for future use. The states for each gates of LSTM are computed by equations (3.4-3.9) as described by [3].

$$f_t = \sigma(W_f . [h_{t-1}, x_t] + b_f) \quad (3.5)$$

Input gate: functions as a cell state input. To determine which values will be updated, the previous hidden state (h_{t-1}) and current input (x_t)are passed through a sigmoid function. The two inputs will be fed into the tanh activation in order to regulate the network.

$$i_t = \sigma\,(W_i.[h_{t-1},x_t] + b_i) \tag{3.6}$$

$$\tilde{C}_t = tanh\,(W_C.\,[h_{t-1},x_t] + b_C\,) \tag{3.7}$$

The old state C_{t-1} gets multiplied by f_t to remove the information decided to forget earlier and add the product $i_t \circ \tilde{C}_t$ to update the old cell state C_{t-1} into the new cell state C_t.

$$C_t = f_t \circ C_{t-1} + i_t \circ \tilde{C}_t \tag{3.8}$$

Output gate: the current hidden state (h_t) is controlled by the output gate. The sigmoid function receives the previous hidden state (h_{t-1}) and the current input(x_t). The output is multiplied by the tanh function's output to obtain the current hidden state. The final outputs of the LSTM unit are the current state (C_t) and the present hidden state (h_t).

$$o_t = \sigma\,(W_o.[h_{t-1},x_t] + b_o) \tag{3.9}$$

$$y_t = h_t = o_t \circ tanh(C_t) \tag{3.10}$$

The LSTM cell state (hidden state) splits into short-term state (h_t) and long-term state (C_t). The long-term dependencies between the current hidden state and previous hidden state over time is stored in the long-term state(C_t). The weight matrices of each of the four layers for their connection to the input vector x_t and short-term state (h_{t-1}) are W_i, W_f, W_C, W_o. The bias terms for each of the four layers are b_i, b_f, b_C, b_o and σ represents the sigmoid function described by equation (3.2).

3.2.3 Gated Recurrent Unit (GRU)

Gated Recurrent Units are kind of recurrent networks that are emerged as a simplification of LSTMs by merging forget and input gate into update gate z_t. GRU has less complicated structure compared to LSTM that makes it computationally faster as it has only two gates [44]. It is composed of a reset gate r_t and an update gate z_t but lacks an output gate that are used to decide what information should be passed to the output. How the new input is combined with the previous memory is controlled by the reset gate. The nonlinear effect of reset gates r_t is introduced in the relationship between the past and future states. The update gate regulates the amount of previous time's state information that is brought into the current state.

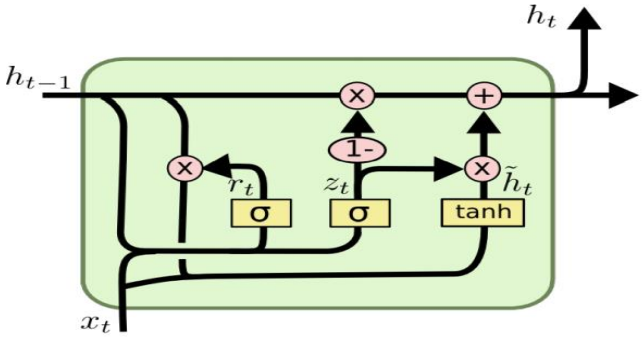

Figure 3. 8: Structure of GRU Cell

The GRU has the following equations for its gates [44]:

$$\text{Update gate:} \quad z_t = \sigma\,(W_z.[h_{t-1}, x_t] + b_z) \tag{3.11}$$

$$\text{Reset gate:} \quad r_t = \sigma\,(W_r.[h_{t-1}, x_t] + b_r) \tag{3.12}$$

$$\text{Cell state:} \quad \tilde{h}_t = tanh(W.[r_t \circ h_{t-1}, x_t] + b) \tag{3.13}$$

$$\text{New state:} \quad h_t = (1\text{-}z_t) \circ h_{t-1} + z_t \circ \tilde{h}_t \tag{3.14}$$

The symbol ∘ indicates element wise multiplication.

Chapter 4. Fault Prediction Approach

This chapter focuses on the problem formulation to perform fault inter-arrival time prediction in optical transport network. It describes about system model, data collection, data preprocessing steps applied on the raw data, time series analysis, and performance evaluation metrics of our models. The overall research workflow is depicted below.

4.1 System Model

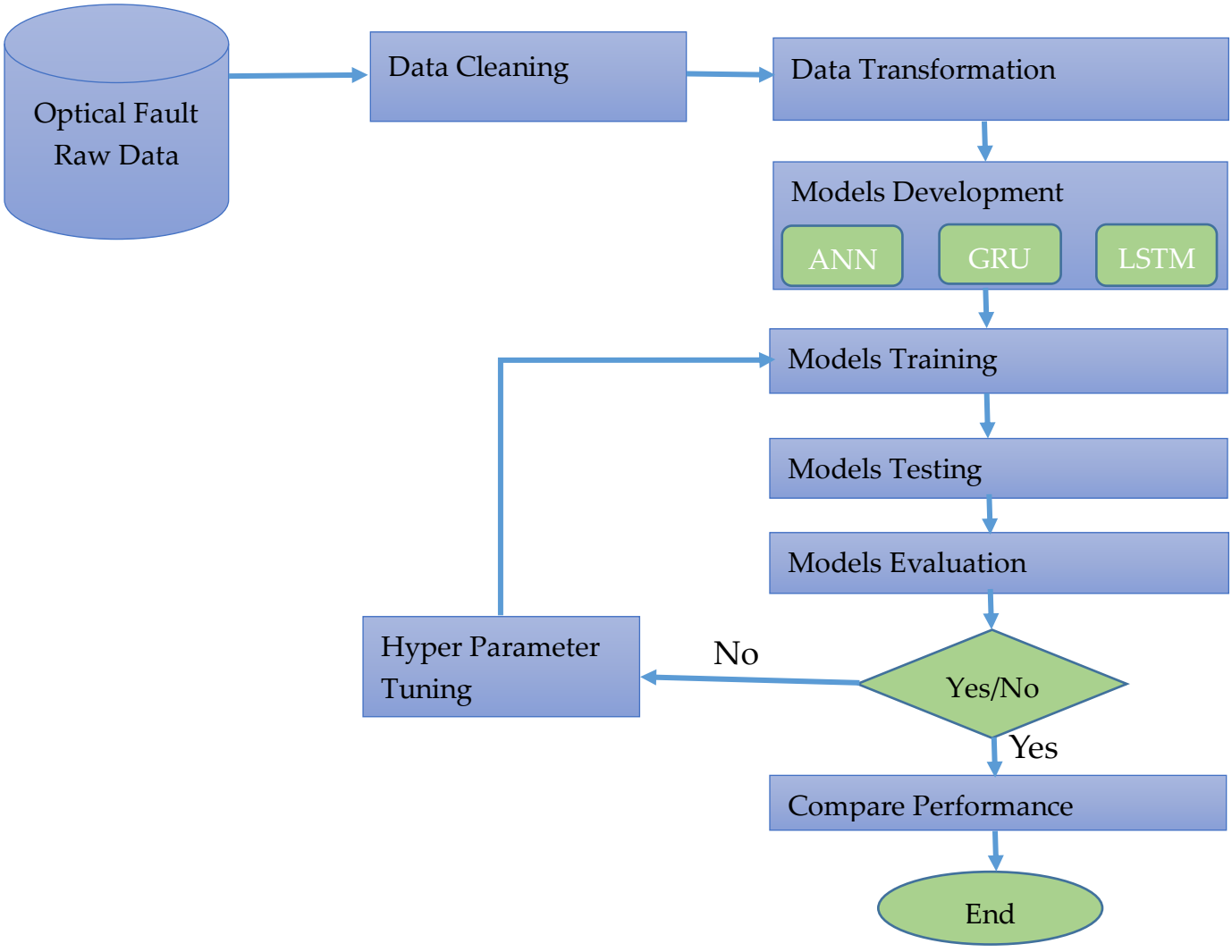

Figure 4. 1: Schematic Diagram of the Workflow

4.2 Data Collection

The data is collected from the Network Monitoring System (NMS) of the optical transport network of Ethio telecom. The collected data is five months of timestamped faults that occurred in the network. The critical alarm has the highest potential to disrupt traffic or services transported by the network and needs immediate action to recover the service back to normal working state. Hence, for this research the critical alarm occurrence time is extracted to predict the fault interarrival time using machine learning algorithms. The alarm data provides informations regarding the alarm name, alarm occurrence time, severity level, alarm source, location information, etc.

Statistics and Density Plot

count	5249.000000
mean	0.561007
std	0.318132
min	0.002200
25%	0.329400
50%	0.524100
75%	0.794400
max	3.470500

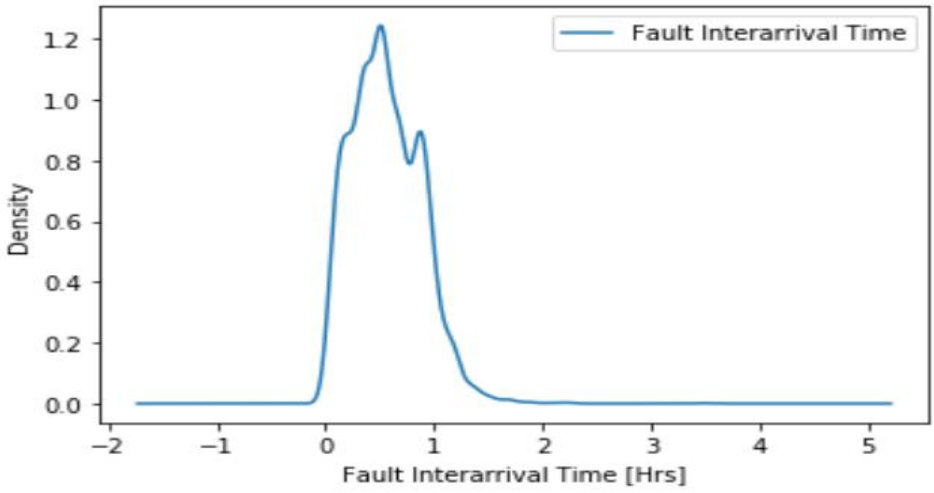

Figure 4. 2: Statistics and Density Plot of Fault Interarrival Time

Autocorrelation Plot

Computing the autocorrelation at varying time lags is used for checking randomness in time series data. If the autocorrelation value is near zero for any time lag, the time series is random else it is non-random if one or more of the autocorrelation is non-zero. Hence, the autocorrelation plot shown in figure 4.3 indicates the randomness of fault occurrence.

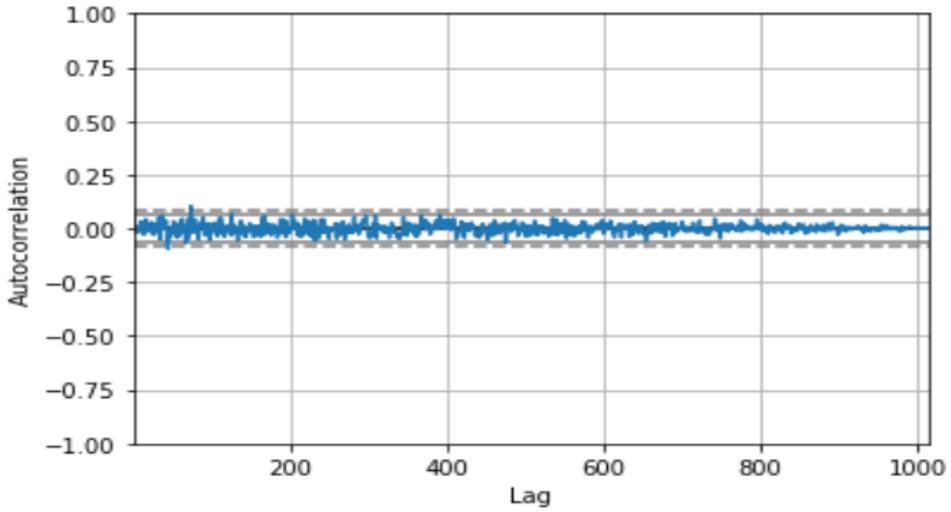

Figure 4. 3: Autocorrelation Plot of Fault Interarrival Time

4.3 Data Preparation

It is a process of preparing raw data into final dataset that is suitable to be fed into machine learning models. It is one of the crucial steps for machine learning as the model accuracy depends on the quality of data. The raw data extracted from the network monitoring system contains data that is not useful for machine learning algorithms. Hence, before using the data as input for the prediction models it must go through a pre-processing phase. For this research, the fault occurrence time is collected and arranged in sequential order to transform the raw data into final dataset.

4.3.1 Data Cleaning

This stage includes removing unnecessary and corrupted data, filling missing values, removing outliers and resolving data inconsistencies. The uncleaned data can lead to unreliable or invalid output. Hence, it negatively affects the prediction performance of the models. The significance of data cleaning is to enhance the quality of data that will be fed into the prediction models.

4.3.2 Data Transformation

The process of converting data from one format or structure to another is known as data transformation. Before using a supervised machine-learning algorithm, preprocessing methods such as data transformation are usually used. These are considered additional data pre-processing techniques that contribute to the success of the mining process.

Normalization and discretization are included in this step.

- **Normalization**

It is done in order to scale all feature values are on the same scale specifically between [0, 1] or [-1, 1] range. Normalization is used to reduce the effect of outliers and optimize the required computational resources. It involves Min-Max scaling or Z-score.

- **Min-Max Normalization**

It is a scaling technique that uses the minimum and maximum feature values to rescale them within a range. Typical normalization in the interval [0, 1] is:

$$x_{norm} = \frac{x - x_{min}}{x_{max} - x_{min}} \quad (4.1)$$

Where x is the original feature value, x_{norm} is the normalized feature value.

x_{min} and x_{max} are minimum and maximum feature values in the training set.

- **Z -score Normalization**

Z − score is a widely used method to perform data standardization and the data will have the properties of a Gaussian distribution with mean, $\mu = 0$ and standard deviation, $\sigma = 1$.

A sample data x's Z score is computed as follows:

$$z = \frac{x - \mu}{\sigma} \quad (4.2)$$

The min-max normalization is used for our research case.

4.4 Time Series Prediction

Time series prediction is a process of forecasting future values based on previous time step values. A number of algorithms have been proposed for time series prediction to model linear and nonlinear time series. ARMA, ARIMA, GARCH etc. are classical time series models that have been widely used for univariate and multivariate time series prediction. However, they show limitations when applied to non-linear time series data. Faults inter arrival time in an optical network are random and non-linear time series. Since machine learning algorithms like ANN, LSTM and GRU have shown promising results exploring easily the nonlinear relationships between inputs and outputs they are selected for the fault interarrival time prediction.

4.5 Performance Evaluation Metrics

There are different metrics widely used for evaluating a model performance. The most commonly used metrics for time series prediction models are Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), Mean Squared Error (MSE) and coefficient of determination or R-squared (R^2).

Error (e_i) is a prediction error which is defined as the difference between actual and predicted value. This difference refers to the unpredictable part of the corresponding observation at the instant time (i). The error is also termed as residual of the actual observation and the model prediction value.

$$e_i = y_i - \hat{y}_i \tag{4.3}$$

Mean Absolute Error (MAE) measures the average absolute difference between actual and predicted values. It measures the mean absolute deviation.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{4.4}$$

Mean Absolute Percentage Error (MAPE) measures the average of the absolute percentage error between actual and forecasted values. The smaller value of MAPE indicates better prediction result.

$$MAPE = \frac{100}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \quad (4.5)$$

Mean Squared Error (MSE) measures the average squared sum of the error between actual and predicted values. Since the error metric squares the errors, the direction of the overall error is lost. The lower the value of MSE, the lower the total squared error and thus the better the model.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \quad (4.6)$$

R-squared (R^2) is coefficient of determination that measures how well the regression predictions approximates the real data points. Its value ranges between 0 and 1. The value 1 indicates that the regression predictions perfectly fit the data. The values outside the range 0 to 1 shows the model poorly fits the data.

Sum of Squared Error (SSE) - residual sum of squared error

$$SSE = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \quad (4.7)$$

Total Sum of Squares (SST) - proportional to the sample variance

$$SST = \sum_{i=1}^{n}(y_i - \bar{y})^2 \quad (4.8)$$

R-squared (R^2) $$R^2 = 1 - \frac{SSE}{SST} \quad (4.9)$$

where y_i-actual value of sample i, $\bar{y}$-mean of the actual values and $\hat{y}_i$-predicted value of sample i and n-sample size.

Chapter 5. Experiment, Result and Analysis

This chapter describes about the overall research conducted. It describes essential tools and libraries, model building using ANN, LSTM and GRU algorithms, hyper parameter tuning, model evaluation and analysis. The developed models are evaluated based on the performance evaluation metrics defined in section 4.5. The models comparison is also performed to find a better model to represent the fault interarrival time.

5.1 Tools and Libraries

Machine learning relies heavily on the following tools and libraries [23]:

Jupyter Notebook: is a computing environment with cells that allow users to write, edit and run codes. It supports programming languages such as Python, R, and others. It is a great tool for data preprocessing, visualization and analysis.

Python: is a powerful interpreted language that is translated and executed by an interpreter. The interpreter is a program that reads statement from the source code, translates in to the machine code and then executes it.

SciPy: is an open source Python library that stands for Scientific Python. It is a set of functions for scientific computing in Python that are based on the NumPy library. It supplements the NumPy library with a number of important high-level science and engineering modules. It contains built-in functions for solving optimization, linear algebra, integration, interpolation, and other mathematical problems. It supports data manipulation, visualization, and image processing.

NumPy: stands for Numerical Python, an open source module that performs fast mathematical computations on arrays and matrices. It has multidimensional array-oriented computing capabilities that are intended for high-level mathematical functions

and scientific computation. NumPy array is the fundamental data structure for Scikit-learn that accepts array data [23].

Pandas: is a data manipulation and analysis library. It is commonly used to preprocess data specifically to load, prepare, manipulate, and analyze structured data. It combines the NumPy library's high performance properties with the ability to modify and operate data in spreadsheets.

Matplotlib: is a graphing and plotting library. It provides functions for making quality visualizations such as line charts, histograms, scatter plots, etc. It is the best suit to obtain a visual interpretation of the dataset and prepare for model building.

Scikit-Learn: is a free and open source library that is built based on NumPy, SciPy, and Matplotlib[26]. Scikit-Learn is used for data preprocessing, as variants of neural networks are sensitive to unscaled data. It provides a transformer called 'StandardScaler' to transform our data that will have the properties of a Gaussian distribution with mean, $\mu = 0$ and standard deviation, $\sigma = 1$. Scikit-Learn also provides a transformer called 'MinMaxScaler' in order to scale all feature values on the same scale. It uses the minimum and maximum feature values to rescale them with in a range [0, 1]. Furthermore, Scikit-Learn provides metrics like mean squared error (MSE) and R- Squared (R^2) to evaluate our prediction models.

Keras: is a deep-learning framework of Python that allows easily defining and training almost any type of deep-learning model. It is a high-level neural network application-programming interface (API) that can be used on platforms such as TensorFlow, Theano, and Microsoft Cognitive Toolkit (CNTK) platforms [22]. It includes a number of parameters and layers that can be used to build, configure, train, and evaluate neural networks.

Hence, Keras is used as a key tool in this research on the TensorFlow backend to build our deep-learning models. Figure 5.1 shows the different backend engines that can be plugged in seamlessly into Keras along with their associated hardware and software stacks.

Figure 5. 1: Deep-Learning Software and Hardware Stack [22]

5.2 ANN Model Building

In this section we will build, train, and evaluate our models using the popular Python library Keras. Building a neural network model requires to determine parameters such as number of hidden layers and neurons, loss function, optimizer, activation functions, learning rate etc. These important parameters are described as follows.

Hidden Layers and Neurons: the number of hidden layers as well as the number of neurons in the hidden layers are critical factors in building an efficient multilayered neural network. Starting with one hidden layer and gradually increasing the layers is a widely accepted rule for the hidden layers. Few neurons can result in an under fitted model because it might fail to learn the patterns of the data. On the contrary, many neurons can result in an over-fitted model as it has the ability to capture patterns that might be noise for the given training dataset. For the number of units in a layer we start with some neurons and try increasing gradually until the network starts overfitting.

Activation Functions: since a neuron performs a linear transformation of the input fed using weight and bias term, an activation function is applied to perform non-linear transformation of the weighted sum and bias term of a neuron. The activation function's output is sent to the next hidden layer, where similar process is repeated. This makes a neuron to learn and perform more tasks that are complex. Rectified Linear Unit (ReLu) activation function has been selected for our models due to its computational efficiency and better performance compared to the sigmoid and tanh activation functions.

Loss Function: is a cost function that is minimized during training. It indicates how well our model predicts a given set of parameters. The loss function gives the training error using the estimated parameters. To evaluate our models, [22] provides simple guidelines for selecting the appropriate loss functions for common problems like classification and regression. According to the guide Mean squared error (MSE) is a loss function that is commonly used in regression problems. Mean absolute error (MAE) and other standard evaluation metrics are also supported to evaluate our model. These additional evaluation metrics are defined in section 4.5.

Optimizers: are used to minimize a loss function and help to know how to update weights and learning rate of neural network in order to reduce the losses. Keras provides varieties of optimizers that are used to help speed up model training. The most popular optimizers are Adam, AdaGrad and RMSProp. We have selected Adam optimizer as it is fast to train models, computationally efficient and has little memory requirement [38].

Learning rate: is used to control step size for updating the weights. It is also an optimization method to reduce the loss function.

Regularization: is a technique to avoid over-fitting problem that occurs when the model fits the data too well or captures all the noises. The most common ways to prevent overfitting are adding dropout, early stopping and weight (ℓ1 and ℓ2) regularization [27].

- **Dropout**

The most effective and popular regularization technique for neural networks is dropout [33]. It randomly sets output features of a layer to zero during training. This means that neural nodes selected at random are removed from the training process. Dropout rates are typically set between 0.1 and 0.5

- **Early Stopping**

The most popular regularization technique for deep neural networks is dropout [27]. When its performance on the validation set begins to decline, it interrupts the training. That means stopping the training when the validation loss is no longer improving in our model can save time and computational resources.

- **Weight Regularization**

It is an approach to avoid overfitting by penalizing high-valued weights of neural networks to improve the generalization of the model. Large weights in a neural network can indicate that the model has over fitted the training dataset. When making predictions on new data, the model can be unstable and show poor performance. Weight regularization is performed by introducing a cost associated with large weights into the network's loss function. The cost added is proportional to the absolute value of the weight coefficients for ℓ1 regularization and to the square of the value of the weight coefficients for ℓ2 regularization [22].

5.2.1 Hyperparameter Tuning

The most time consuming and challenging task in machine learning process is improving the performance score of a model. This is because machine-learning algorithms require optimized parameters besides clean dataset. Hyperparameter tuning is adjustment of set of parameter values to find an optimal model accuracy. That means the numerical values are tuned until we find reasonable performance score for our model. The selected parameters and values for the ANN, LSTM and GRU models are shown in table 5.1.

Table 5. 1: ANN, LSTM and GRU Model Hyperparameters

Parameters	Value
Number of hidden layers	{2,3,4,5}
Number of neurons in each layer	{8, 16, 32, 64, 128 ,256}
Dropout rate	[0.1, 0.5]
Recurrent dropout rate (LSTM)	[0.1, 0.5]
Hidden layers activation function	{Relu }
Output layer activation function	{Linear }
Regularizers	{1e-2, 1e-3, 1e-4}
Learning rate	{1e-3, 1e-4, 1e-5, 5e-5}
Optimizer	{Adam}
Batch size	{10, 20, 50}
Epoch	{50, 100, 200, 400}

The modules that must be imported from Keras to create neural network model are [22]:

- *Sequential ()* - is a linear stack of layers that initializes the neural network.
- *Dense*- to add densely connected neural network layer.
- *Dropout*- for adding dropout layers to prevent overfitting.

Hence, we have created a sequential model and then added layers connected to the next one in the order of computation. That means all the units in the previous layer are connected to all the neurons in the next layer. Hence, these connected layers are called fully connected or dense layers. The number of hidden layers is varied in the range from 2 to 5 and the number of hidden neurons from 8 to 256, the dropout rate varied from 0.1 to 0.5, and the activation function ReLU is configured in each layer to build the ANN model. The best combination of hyperparameters shown in table 5.2 has been selected based on the training state, loss curves and computational time taken per epoch.

Table 5. 2: Hyperparameters for ANN Model

Neurons	Dropout rate	Activation Function	Optimizer	Learning rate
256,128,64	0.1	ReLu	Adam	5e-5

The snippet shown in figure 5.2 indicates the neural network model summary that has been developed by three hidden layers and with their respective number of neurons to predict the fault interarrival time.

```
Model: "sequential_35"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_86 (Dense)             (None, 256)               1024
_________________________________________________________________
dropout_105 (Dropout)        (None, 256)               0
_________________________________________________________________
dense_87 (Dense)             (None, 128)               32896
_________________________________________________________________
dropout_106 (Dropout)        (None, 128)               0
_________________________________________________________________
dense_88 (Dense)             (None, 64)                8256
_________________________________________________________________
dropout_107 (Dropout)        (None, 64)                0
_________________________________________________________________
dense_89 (Dense)             (None, 1)                 65
=================================================================
```

Figure 5. 2: ANN Model Summary

5.2.2 ANN Model Results and Analysis

The ANN model constructed with three hidden layers and their respective number of neurons 256, 128, 64 shown in table 5.1 has been trained using different learning rate, epochs, batch sizes etc. The optimizer used is Adam as it converges faster than other optimizers like AdaGrad, RMSProp and takes less time to train the model [38]. The learning rate that has produced the best prediction performance was 5e-5. The average test values were selected after performing iterative tests on each data samples. The actual and predicted fault interarrival time prediction of one-month data trained on 70/30 and 80/20 scheme using the ANN model are shown in figure 5.3 and 5.5 respectively. The x-axis shows occurrence of the sequential alarm samples and the y-axis indicates the time difference between two consecutive alarms.

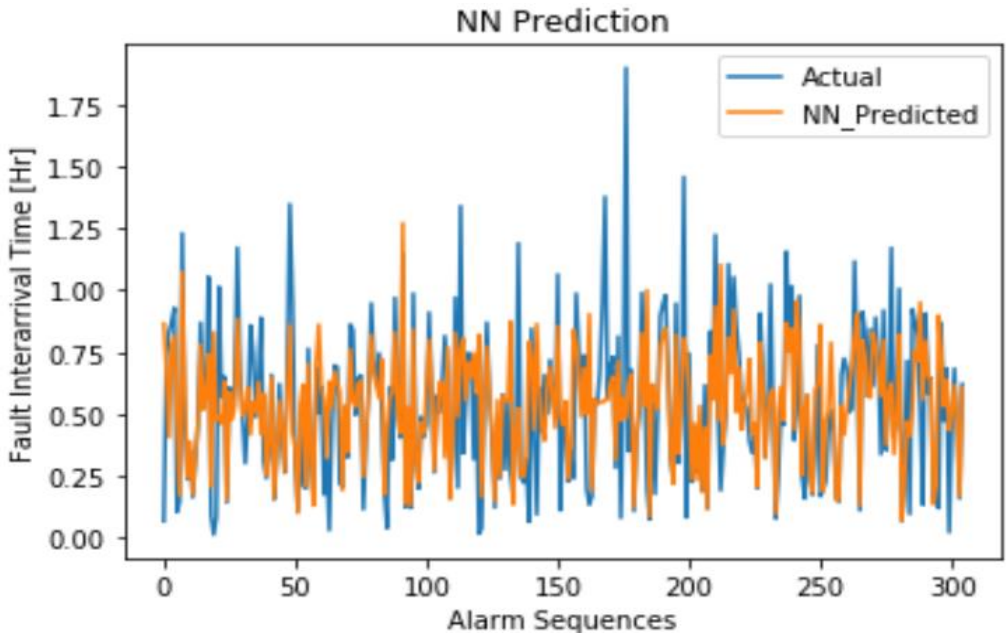

Figure 5. 3: ANN Prediction (70/30)

The loss curve for 70/30 training of the ANN model is shown in figure 5.4. The loss function configured to train the ANN model is the mean squared error (MSE). The number of batch sizes of 20 found better to train the model and outperform in prediction performance. The early stopping criterion with patience value of 5 stops further training of the model if the validation loss remains unchanged. The dropout rate with 0.1 in each layers and weight regularizer 1e-2 helped for the results. The training loss started to decline with the number of epochs and the loss curve remains in the same level beyond epoch 70 that was stopped further training by the early stopping criterion to save the computational resources and time.

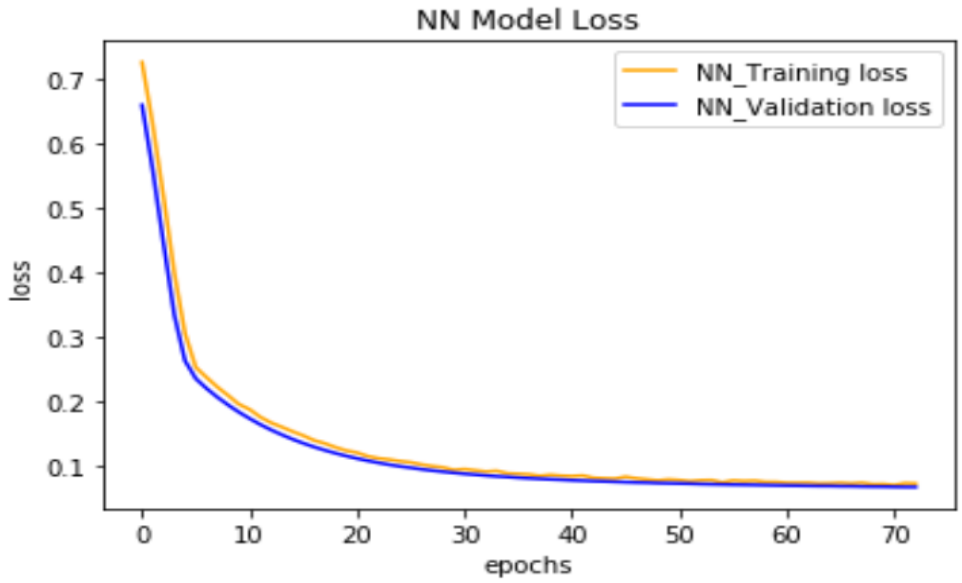

Figure 5. 4: ANN Model Loss (70/30)

The same experiment has been performed using 80/20 training scheme to evaluate the models stability and prediction performance. The average MAPE obtained for the 80/20 training is 17.0448 and while the average MAPE for the 70/30 training is 17.3681 as shown in table 5.3.

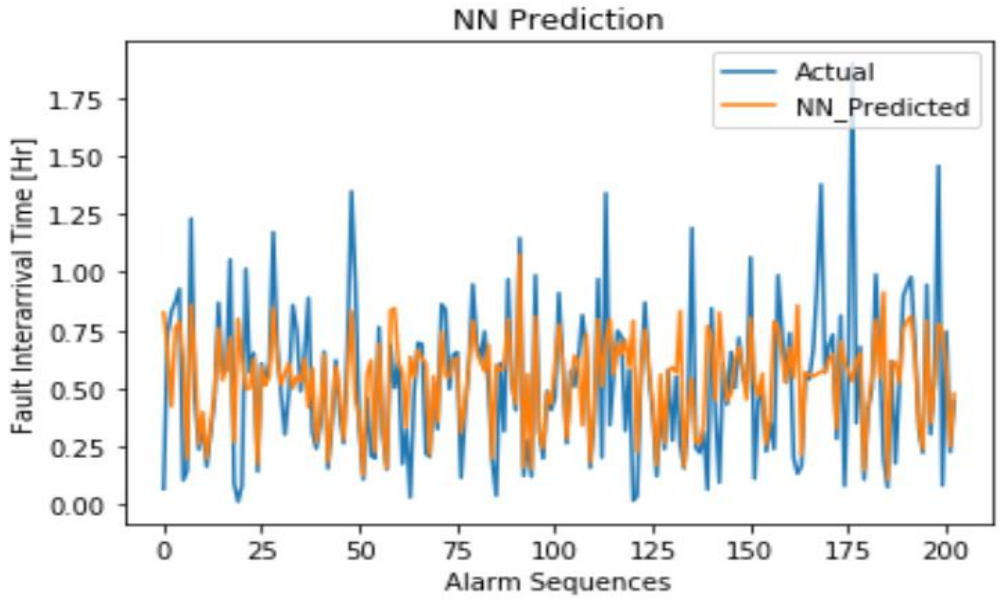

Figure 5. 5: ANN Prediction (80/20)

The loss curve for the 80/20 training is shown in figure 5.4. The average MSE obtained is 0.0768 as shown in table 5.3. From the loss curve, we can see that the model MSE started to decline with the number of epoch and converges after 100 iterations (epoch). The early stopping patience set here is also 5 to stop the training if the loss value remains constant.

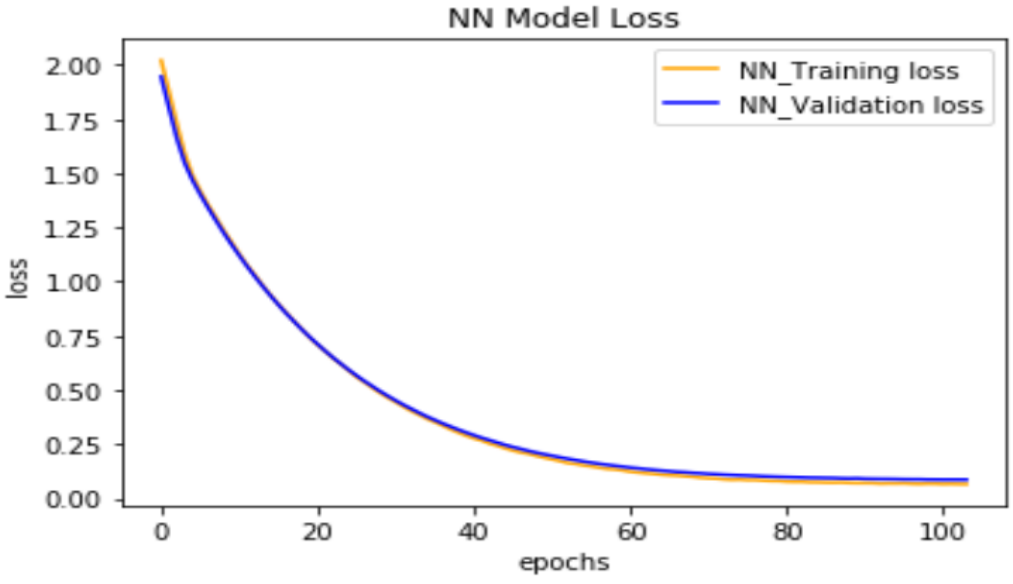

Figure 5. 6: ANN Model Loss (80/20)

The model performance for both the 70/30 and 80/20 training schemes is shown in table 5.3. The MAE varies in between 0.2815 and 0.339 irrespective of the training schemes. The MSE values vary between 0.047 and 0.0835 for the two prediction horizons. The minimum MAPE is 13.9003% and the maximum MAPE value is 18.3116%, which means the least accuracy of the model is around 81%. Therefore, a fault occurrence time can be predicted using the ANN model with such an accuracy level using the ANN model. The MAPE difference for a specific data trained in both 80/20 and 70/30 indicates less than 1 %. This indicates that the models stable performance for the various dataset trained.

Table 5. 3: ANN Model Performance

Data	**Train-Test**	**MAE**	**MSE**	**MAPE**	**R^2**	**Comp.time/epoch[ms]**
January	70/30	0.2815	0.0570	15.3930	0.3845	3
	80/20	0.2823	0.0535	14.6192	0.3893	2
February	70/30	0.3238	0.0761	17.5078	0.3850	3
	80/20	0.3390	0.0835	18.3116	0.3714	2
March	70/30	0.3230	0.0528	13.9003	0.4923	2
	80/20	0.3266	0.0587	14.6904	0.4398	3
April	70/30	0.2939	0.0535	16.1902	0.4240	5
	80/20	0.3018	0.0470	14.4266	0.4885	2
May	70/30	0.2973	0.0720	17.3681	0.2728	3
	80/20	0.2956	0.0768	17.0448	0.2576	3
2 Months	70/30	0.2834	0.0680	18.2535	0.2692	2
	80/20	0.2841	0.0704	18.0029	0.2620	2
3 Months	70/30	0.2984	0.0546	15.4157	0.4372	2
	80/20	0.2987	0.0589	15.9894	0.4102	2
4 Months	70/30	0.3000	0.0560	14.5604	0.4348	2
	80/20	0.3011	0.0571	15.2508	0.4335	2
5 Months	70/30	0.2993	0.0640	16.1520	0.3860	2
	80/20	0.2949	0.0653	16.4357	0.3748	2

5.3 LSTM Model Building

LSTM is a kind of recurrent neural network (RNN) that is considered adaptable to non-linear time series forecasting. The essential parameters that are required to build LSTM model are described in section 5.2. These parameters include activation function, optimizer, learning rate, loss function, metrics, regularization, epoch, batch size etc. In this section, we only introduce additional parameters that is necessary to build LSTM model. In Keras, every recurrent layer has two parameters that are useful to regularize a model. These are *dropout* and *recurrent dropout*. Dropout removes some neural elements of a layer at random. Recurrent dropout in LSTM model removes inputs between time steps but not inputs between layers. It has an effect to regularize and prevent overfitting as the regular dropout. To build the LSTM model the imported modules from Keras are:

- *Sequential (), LSTM* and *Dense* layers
- *Dropout*- for adding dropout layers

The models built by ANN, LSTM and GRU algorithms have been set the same number of hidden layers and neurons in each layers as they have produced comparable level of prediction performance rather than other combination of number of layers and neurons. Further, to compare and select a model with the best performance it would be reasonable to make the same number of hidden layers and neurons.

Table 5. 4: Hyperparameters for LSTM Model

Neurons	Dropout rate	Activation Function	Optimizer	Learning rate
256,128,64	0.1	Relu	adam	5e-5

The three hidden layers with their corresponding number of neurons 256,128 and 64 respectively have been set as shown in LSTM model summary in figure 5.7. The dense layer is used as output layer and the dropout rate in each layer used is 0.1.

```
Model: "sequential_38"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm_33 (LSTM)               (None, None, 256)         264192
_________________________________________________________________
dropout_114 (Dropout)        (None, None, 256)         0
_________________________________________________________________
lstm_34 (LSTM)               (None, None, 128)         197120
_________________________________________________________________
dropout_115 (Dropout)        (None, None, 128)         0
_________________________________________________________________
lstm_35 (LSTM)               (None, 64)                49408
_________________________________________________________________
dropout_116 (Dropout)        (None, 64)                0
_________________________________________________________________
dense_92 (Dense)             (None, 1)                 65
=================================================================
```

Figure 5. 7: LSTM Model Summary

5.3.1 LSTM Model Results and Analysis

The actual and predicted fault interarrival time prediction of one-month data trained on 70/30 and 80/20 scheme using LSTM model are shown in figure 5.8 and 5.10 respectively.

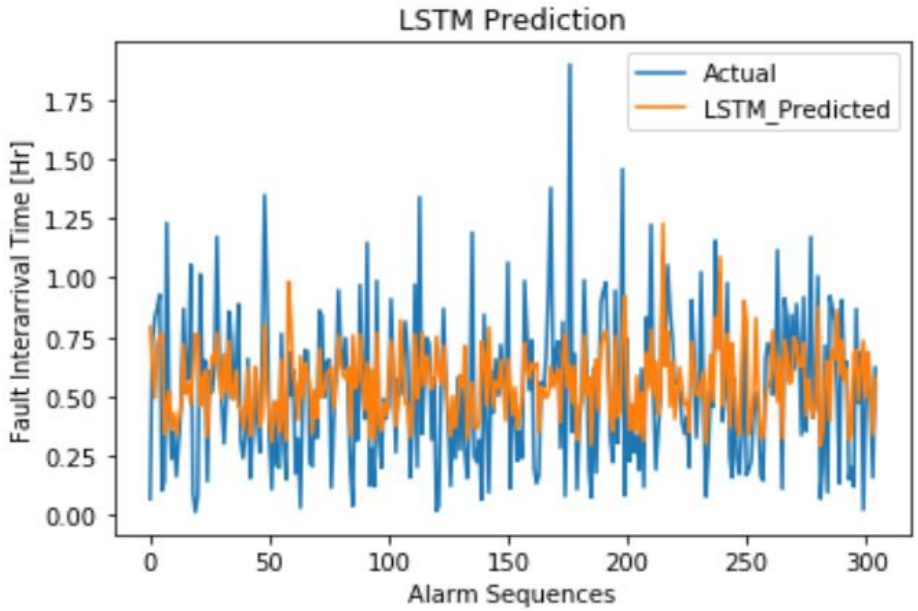

Figure 5. 8: LSTM Prediction (70/30)

The loss curve in training the LSTM model is shown in figure 5.9, the selected activation function is rectified linear unit (ReLU), and the optimizer is Adam with learning rate 5e-5. The batch size of 20 used to train the model. The loss curve declines for both the training and validation sets until the MSE becomes constant. The early stopping criterion with patience value of 5 stops further training of the model around 50 iteration (epoch) to save computational time and resources.

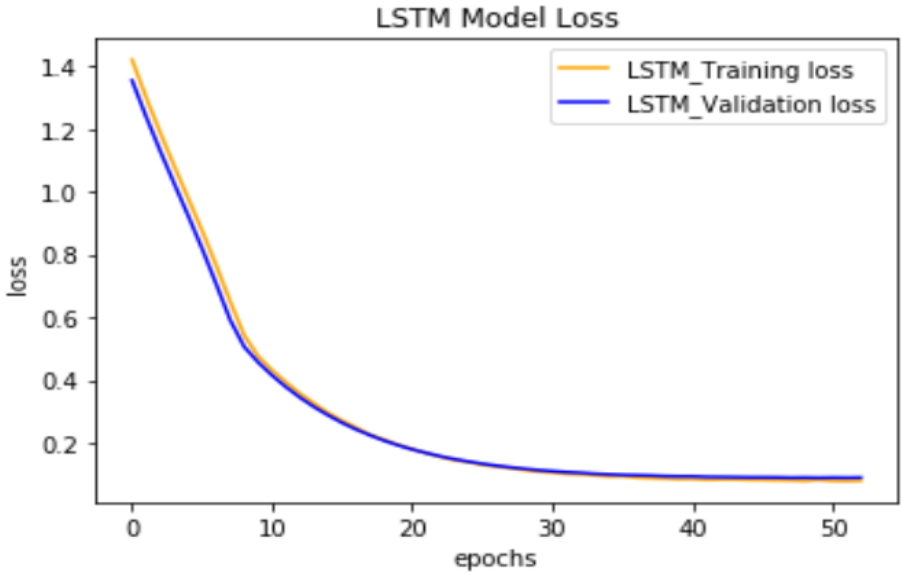

Figure 5. 9: LSTM Model Loss (70/30)

The prediction experiment has been repeated using 80/20 training scheme. The average MAPE obtained for the 80/20 training is 20.5527% while the average MAPE for the 70/30 training is 19.9900% as shown in table 5.5. The MAPE difference between the two training schemes was found to be less than 1%. That means the fault occurrence time prediction using the LSTM model for this particular data is close to each other and the model's prediction performance shows its stability despite the training schemes used. The actual and predicted fault interarrival time is shown in figure 5.10.

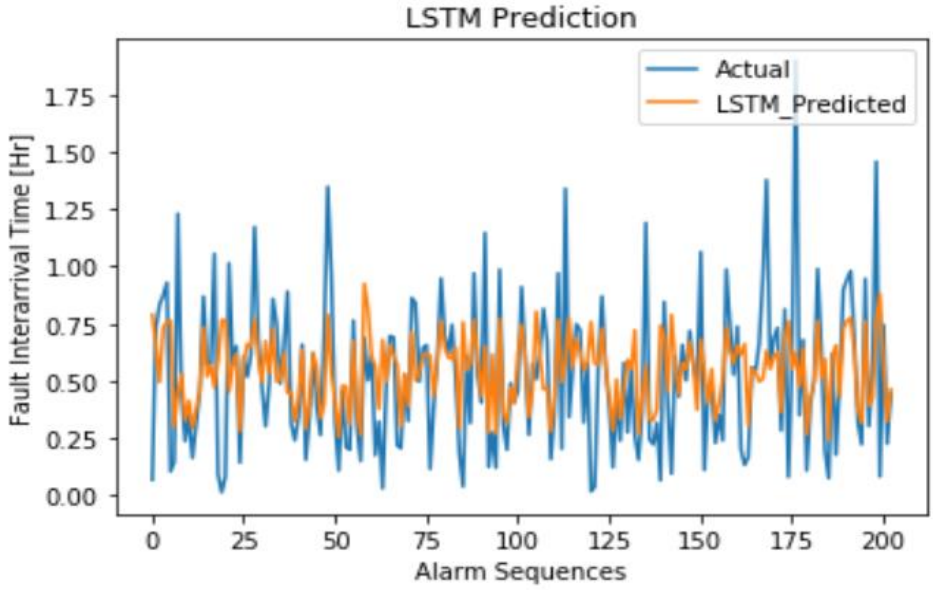

Figure 5. 10: LSTM Model Prediction (80/20)

The loss curve is shown in figure 5.11 that declines for both the training and validation sets until the MSE becomes constant. The early stopping criterion with patience value of 5 stops further training of the model around epoch 50 which is similar with the 70/30 training scheme to save computational time and resources.

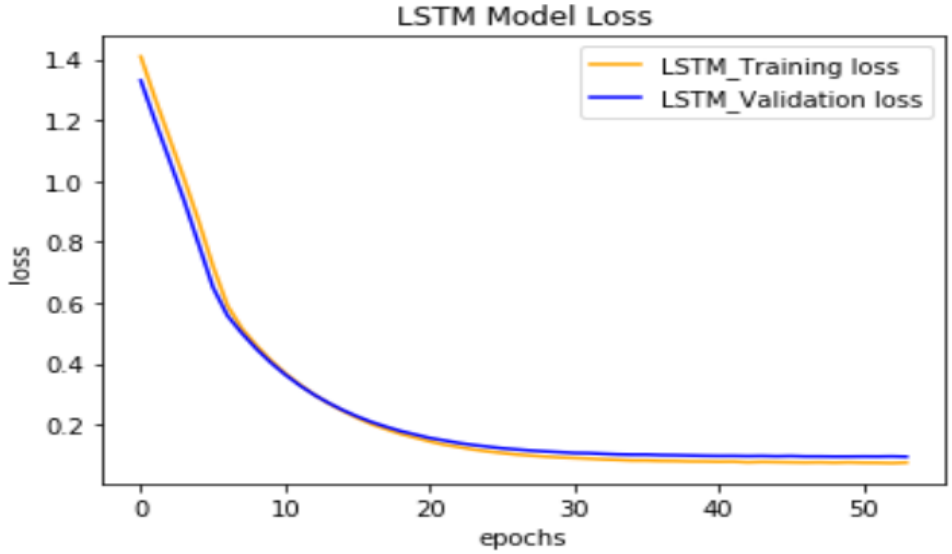

Figure 5. 11: LSTM Model Loss (80/20)

The prediction performance of the LSTM model is shown in table 5.5 for the different dataset used to train the model. The MAE varies between 0.2584 and 0.3293. The MSE fluctuates between 0.0557 and 0.0925. The minimum MAPE is 14.4184% and the maximum MAPE is 21.2047%, which means the least accuracy of the model is around 78.8%. Therefore, a fault occurrence time can be predicted with such an accuracy level using the LSTM model. For each specific data trained the MAPE difference in both 80/20 and 70/30 indicates less than 4 %. This indicates that the model's stable performance for the various dataset trained.

Table 5. 5: LSTM Model Performance

Data	Train-Test	MAE	MSE	MAPE	R^2	Comp.time/epoch [ms]
January	70/30	0.2734	0.0647	17.8805	0.3023	15
	80/20	0.2584	0.0668	19.3658	0.2372	12
February	70/30	0.3293	0.0820	18.2091	0.3376	14
	80/20	0.3268	0.0925	20.8985	0.3036	14
March	70/30	0.3235	0.0557	14.4184	0.4641	11
	80/20	0.3104	0.0705	18.2693	0.3266	15
April	70/30	0.2833	0.0631	18.3370	0.3212	23
	80/20	0.2885	0.0563	16.6278	0.3865	16
May	70/30	0.2792	0.0787	19.9900	0.2053	15
	80/20	0.2764	0.0868	20.5527	0.1606	14
2 Months	70/30	0.2743	0.0717	19.4069	0.2298	12
	80/20	0.2648	0.0803	21.2047	0.1583	14
3 Months	70/30	0.2909	0.0617	16.0429	0.3632	12
	80/20	0.2959	0.0650	16.4256	0.3486	12
4 Months	70/30	0.2920	0.0615	15.8893	0.3793	12
	80/20	0.2967	0.0609	16.2075	0.3963	14
5 Months	70/30	0.2926	0.0676	16.6451	0.3507	13
	80/20	0.2905	0.0674	16.3759	0.3551	13

5.4 GRU Model Building

The essential modules imported from Keras to build GRU model are:

- *Sequential (), GRU* and *Dense* layers
- *Dropout-* for adding dropout layer

The hyper parameters for the GRU model is set to have the same number of layers and neurons with other models for prediction performance comparison purpose.

Table 5. 6: Hyperparameters for GRU Model

Neurons	Dropout rate	Activation Function	Optimizer	Learning rate
256,128,64	0.1	relu	adam	5e-5

The model summary shown in figure 5.12 shows the number of layers and number of neurons in each hidden layers. The dropout rate in each layer is set to be 0.1. The dense layer is used as output layer.

GRU Model Summary

```
Model: "sequential_41"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
gru_33 (GRU)                 (None, None, 256)         198912
_________________________________________________________________
dropout_123 (Dropout)        (None, None, 256)         0
_________________________________________________________________
gru_34 (GRU)                 (None, None, 128)         148224
_________________________________________________________________
dropout_124 (Dropout)        (None, None, 128)         0
_________________________________________________________________
gru_35 (GRU)                 (None, 64)                37248
_________________________________________________________________
dropout_125 (Dropout)        (None, 64)                0
_________________________________________________________________
dense_95 (Dense)             (None, 1)                 65
=================================================================
```

Figure 5. 12: GRU Model Summary

5.4.1 GRU Model Results and Analysis

The actual and predicted fault interarrival time prediction of one-month data trained on 70/30 and 80/20 scheme using GRU model are shown in figure 5.13 and 5.15 respectively. The x-axis shows occurrence of the sequential alarm samples and the y-axis indicates the time difference between two consecutive alarms.

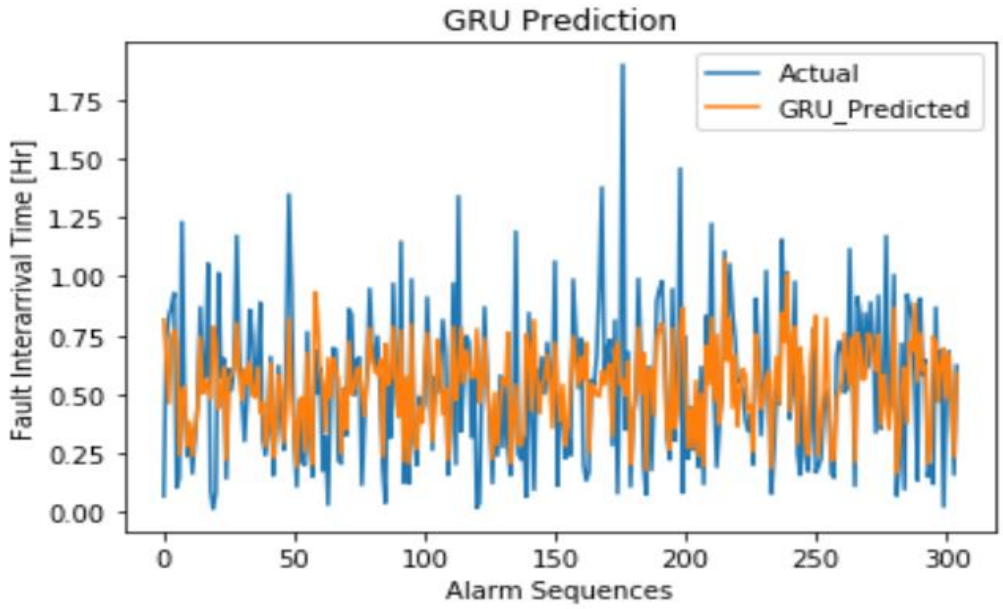

Figure 5. 13: GRU Model Prediction (70/30)

The loss curve of the GRU model trained with 70/30 scheme is shown in figure 5.14, the selected activation function is rectified linear unit (ReLU), and the optimizer is Adam with the same learning rate 5e-5 as the ANN and LSTM models. The batch size of 20 used to train the model. The loss curve declines for both the training and validation sets until the MSE becomes constant. The weight regularizer L2(1e-2) was used in the hidden layers to avoid overfitting and have smoothly declining loss curve. The patience value for early stopping of further training is set to be same as the other models. The validation error for the GRU model has become constant around 60 iteration (epoch) and the further training has been terminated to save computational time and resources.

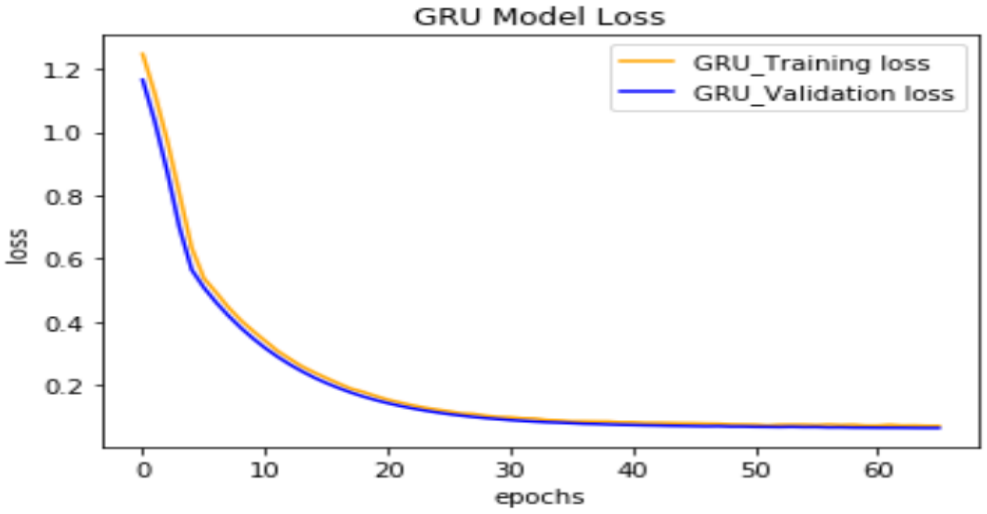

Figure 5. 14: GRU Model Loss (70/30)

Similar experiment has been performed using 80/20 training scheme to evaluate the GRU model's stability and prediction performance. The average MAPE obtained for the 80/20 training is 16.8892 and while the average MAPE for the 70/30 training is 17.1025 as shown in table 5.7.The MAPE difference between the two training schemes is less 1%.

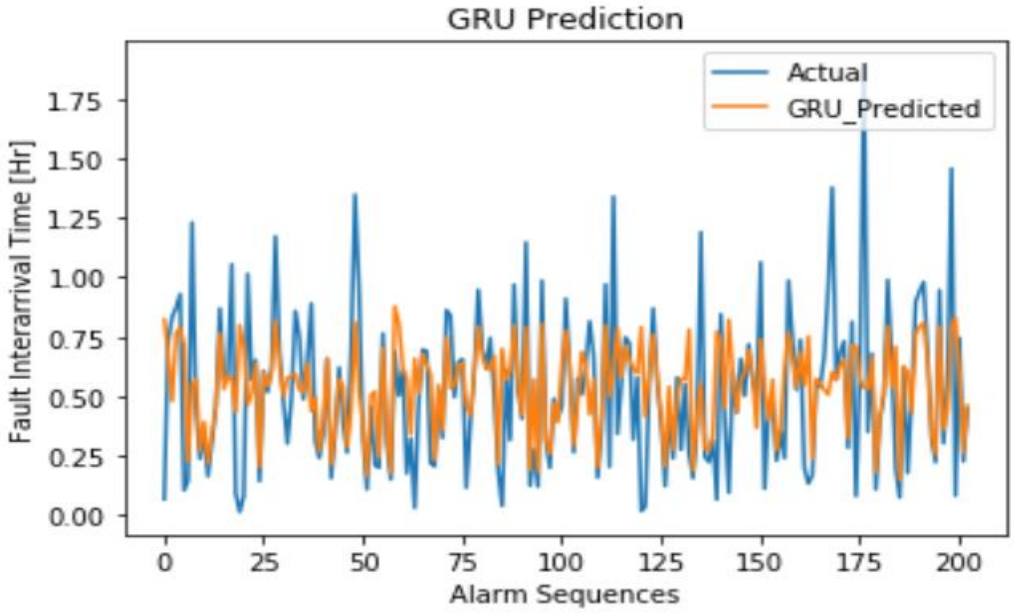

Figure 5. 15: GRU Model Prediction (80/20)

The loss curve for the 80/20 training is shown in figure 5.16. The average MSE obtained is 0.2951 as shown in table 5.7. From the loss curve, we can see that the model MSE started to decline with the number of epoch and converges after 50 iterations (epoch). The early stopping patience set here is also 5 so as to stop the training if the loss value remains constant.

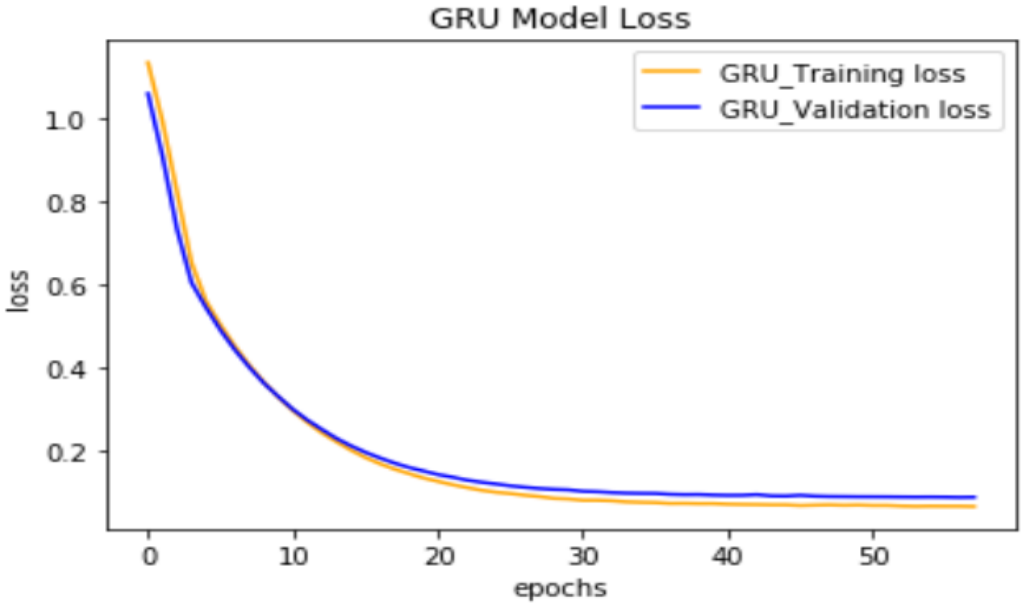

Figure 5. 16: GRU Model Loss (80/20)

The prediction performance of GRU model trained with 70/30 and 80/20 schemes is shown in table 5.7. The MAE varies in between 0.2787 and 0.339 irrespective of the training schemes. The MSE values vary between 0.0557and 0.0925 for the two prediction horizons. The minimum MAPE is 14.1702% and the maximum MAPE value is 18.6743%, which means the least accuracy of the model is around 81%. Therefore, a fault occurrence time can be predicted using the GRU model with an accuracy level of 81%. The MAPE difference for a specific data trained in both 80/20 and 70/30 schemes indicates less than 2 %. This indicates that the models stable performance for the various dataset trained.

Table 5. 7: GRU Model Performance

Data	Train-Test	MAE	MSE	MAPE	R^2	Comp.time/epoch [ms]
January	70/30	0.2787	0.0647	16.0902	0.3707	11
	80/20	0.2804	0.0668	14.9723	0.3917	9
February	70/30	0.3230	0.0820	18.5571	0.3416	10
	80/20	0.3397	0.0925	18.6743	0.3602	10
March	70/30	0.3196	0.0557	14.1702	0.4912	9
	80/20	0.3269	0.0705	14.3930	0.4315	11
April	70/30	0.2956	0.0631	16.3411	0.3873	18
	80/20	0.3027	0.0563	14.5691	0.4785	11
May	70/30	0.2964	0.0787	17.1025	0.2705	11
	80/20	0.2951	0.0868	16.8892	0.2387	10
2 Months	70/30	0.2847	0.0717	17.8343	0.2687	9
	80/20	0.2866	0.0803	17.9576	0.2532	10
3 Months	70/30	0.3039	0.0617	15.0453	0.4217	10
	80/20	0.3009	0.0650	16.0963	0.3935	10
4 Months	70/30	0.3005	0.0615	14.8649	0.4192	8
	80/20	0.3042	0.0609	15.3389	0.4264	9
5 Months	70/30	0.2989	0.0676	16.4415	0.3720	10
	80/20	0.2931	0.0674	16.8150	0.3740	8

5.5 Models Comparison

The three models prediction performance shown in table 5.3, 5.5 and 5.7 are referred for model comparison purpose.

- **70/30 Training**

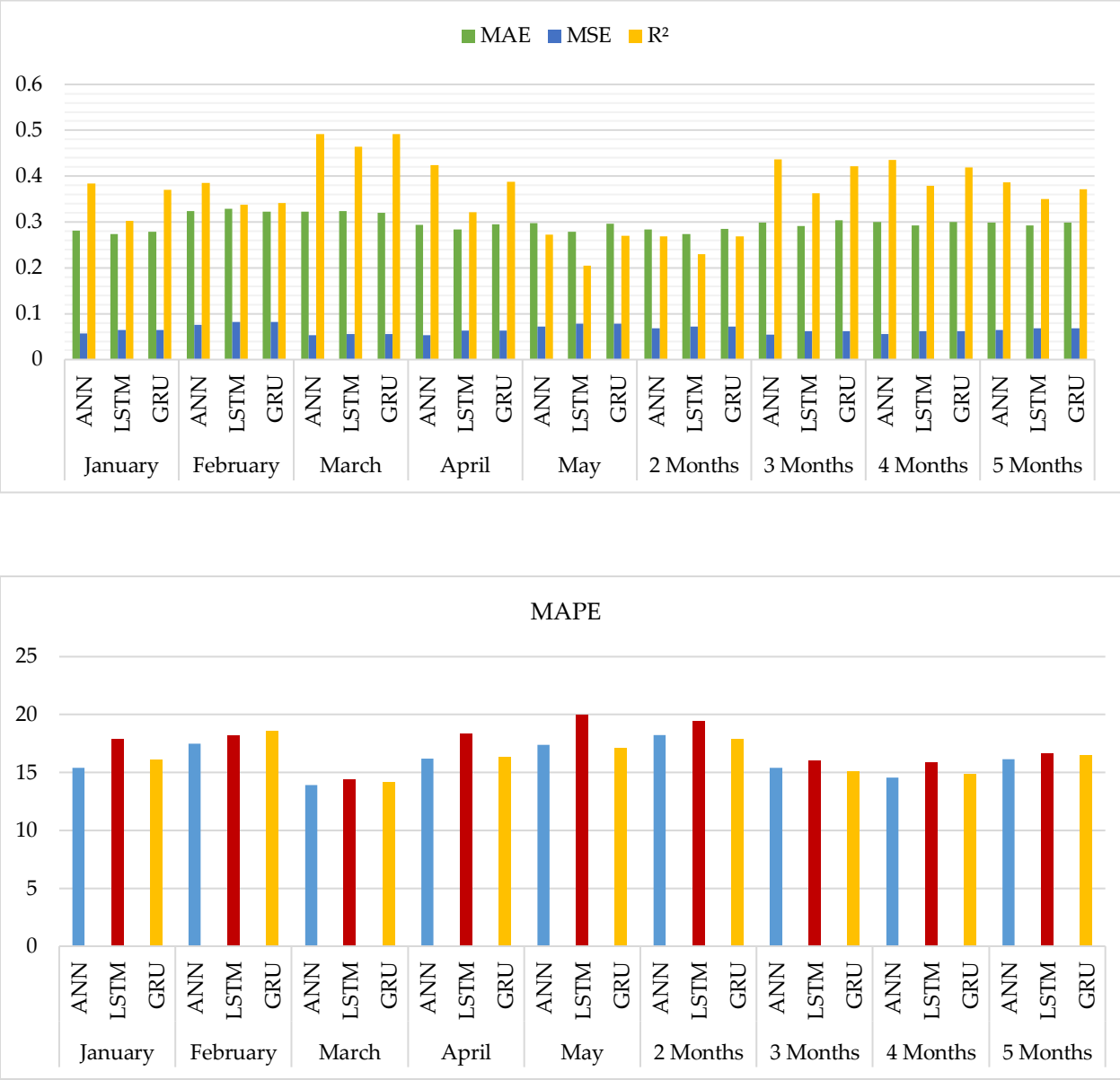

Figure 5. 17: Models Prediction Performance of 70/30 Training Scheme

Training the models with 70/30 scheme, the ANN model produces the least MSE. The coefficient of determination (R^2) of ANN model is the highest as compared with the other two models. Further, the ANN model has the least MAPE followed by the GRU model as shown in figure 5.17. The GRU model generates less MSE error and better coefficient of determination as compared with the LSTM model as shown in figure 5.17. The LSTM model is found to generate the highest MAPE. Additionally, the average computational time taken per epoch by the ANN model is the least one as compared with the other two models as shown in figure 5.18.

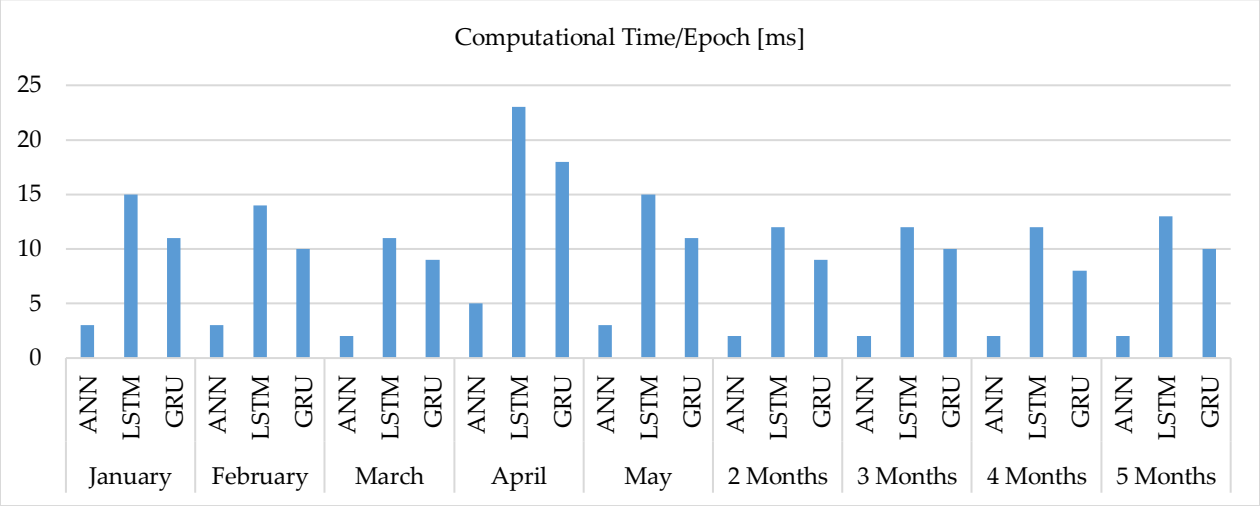

Figure 5. 18: Computational Time of 70/30 Training Scheme

- **80/20 Training**

Similarly, the 80/20 training scheme shows that the MSE of the ANN model is the least one. The coefficient of determination (R^2) of ANN and GRU models show competitive results except the ANN model outperforms in some trained data as shown in figure 5.19. The LSTM model generates the highest MAPE in the 80/20 training horizon too. The ANN model shows the minimum MAPE followed by the GRU model.

Further, the average computational time taken per epoch by the ANN model is the least one followed by the GRU model that consumes less time than the LSTM model as shown in figure 5.20.

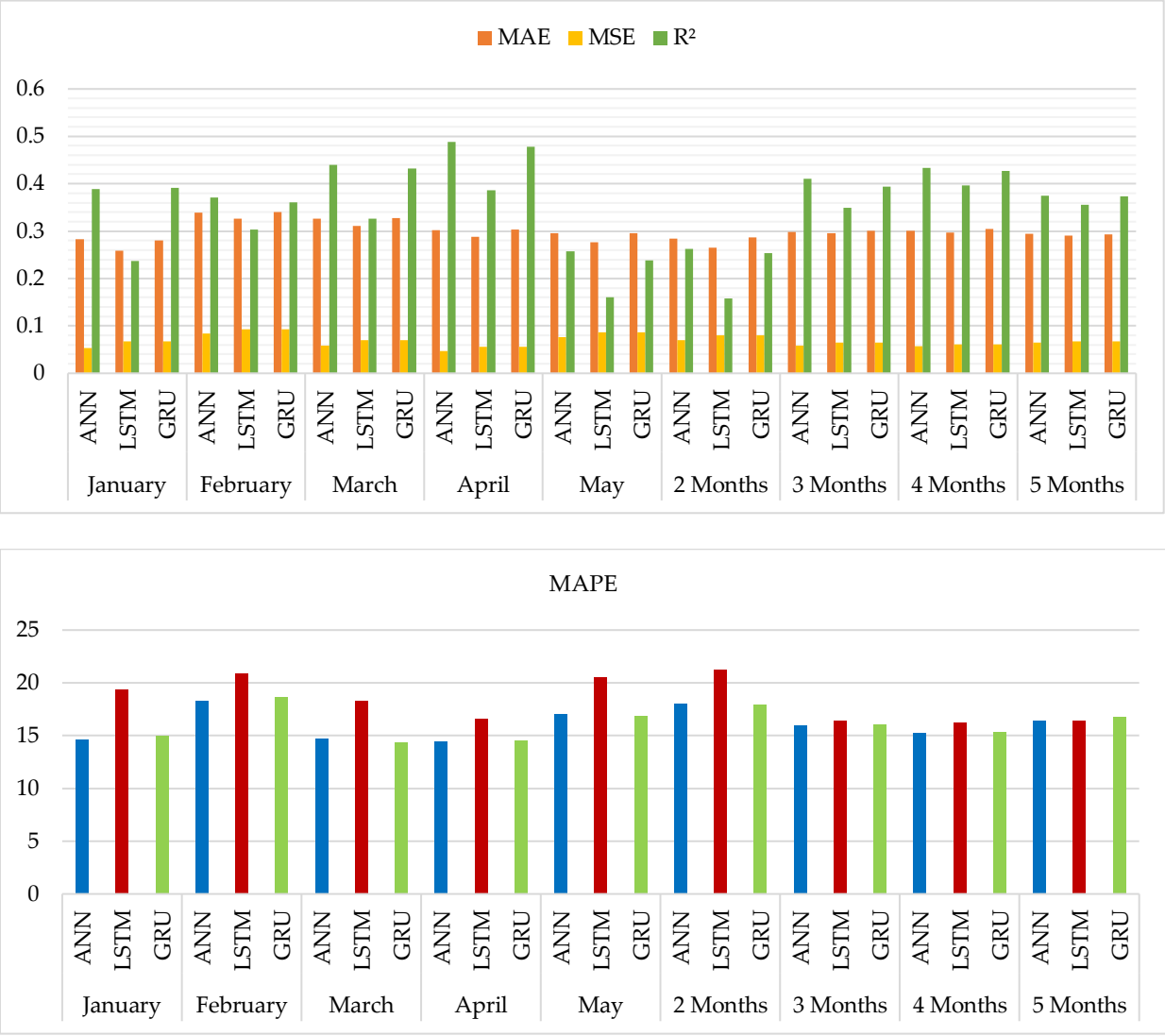

Figure 5. 19: Models Prediction Performance of 80/20 Training Scheme

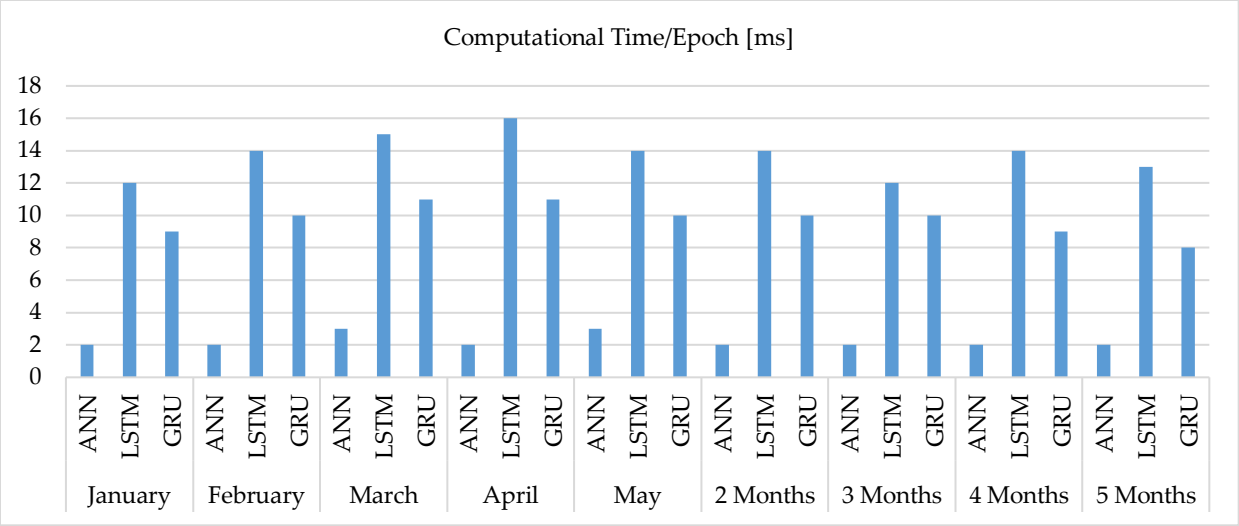

Figure 5. 20: Computational Time of 80/20 Training Scheme

Hence, in every training schemes and data samples considered for models comparison, ANN model outperforms in both prediction performance and computational efficiency. The three models prediction combined plot trained with 70/30 scheme is shown in figure 5.21.

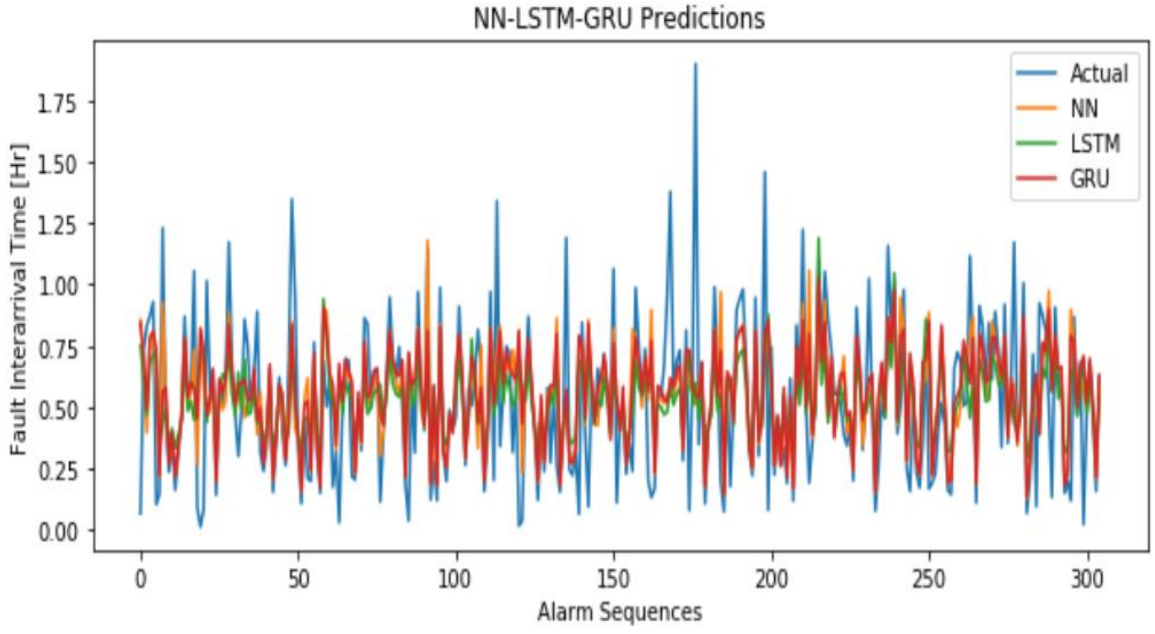

Figure 5. 21: ANN-LSTM-GRU Models Prediction

Chapter 6. Conclusion and Future work

6.1 Conclusion

In this research, machine-learning algorithms have been used to build fault prediction models for optical transport network. The prediction models used fault interarrival time history of the network to be trained. The three models built by the ANN, LSTM and GRU algorithms have been trained with 70/30 and 80/20 training schemes to study the fault interarrival pattern in the network and select the best model in prediction performance. Even though the dataset is nonlinear and the fault arrival is random, the models prediction results indicate that the ANN model has predicted the fault interarrival time with 16.08% MAPE while GRU shows 16.23% MAPE and LSTM prediction error is 17.93% MAPE on average. The R^2 scores for the ANN model falls between 0.2576 and 0.4923. It implies that the model is able to catch the general trend of the fault interarrival time followed by the GRU model with R^2 scores in between 0.2387 and 0.4912. The LSTM model performs the least to catch the fault pattern having R^2 scores in between 0.1583 and 0.4641. In addition, the average computational time taken per epoch by ANN model is 2.44ms, GRU model consumes 10.22 ms and LSTM takes 13.94 ms. So, the ANN model takes less time per epoch.

The nonlinear trend of the data has made the models not easily learn the fault interarrival pattern and refrain them from great success in prediction. Here, the ANN model is better to learn the irregular nature of the data followed by GRU and then LSTM model. Further, since the models try to predict the fault occurrence of the whole optical transport network instead of a single node or optical channel the prediction error has become higher. Hence, ANN outperforms in terms of prediction performance and computational efficiency.

6.1 Future work

Similar approach can be applied to the other vendor optical network. Further, same approach can be used on network element (NE) and link level that exhibit frequent fault occurrence to take proactive measures and maintain the faults promptly. Minimizing the prediction error can be another task that could help expect the fault interarrival as close as the actual occurrence time so that preventive actions can be taken to remove the fault before affecting services. Hybrid model designing using deep learning algorithms could also help further refine the results. Moreover, designing optimal number of layers and neurons that can learn the non-linearities of fault interarrival time can be a future work.

References

[1] M. Jaudet, N. Iqbal and A. Hussain, "Neural Networks for Fault Prediction in a Telecommunications Network," 8th International Multitopic Conference, 2004. Proceedings of INMIC 2004, 2004, pp. 315-320.

[2] Y. Kumar, H. Farooq and A. Imran, "Fault Prediction and Reliability Analysis in a Real Cellular Network," 2017 13th International Wireless Communications and Mobile Computing Conference (IWCMC), 2017, pp. 1090-1095.

[3] Z. Tan and P. Pan, "Network Fault Prediction based on CNN-LSTM Hybrid Neural Network," 2019 International Conference on Communications, Information System and Computer Engineering (CISCE), 2019, pp. 486-490.

[4] M. Chang, Y. Ji, W. Dong and Z. Zhang, "Hidden Markovian Chain based Occurrence Time Prediction for GSM-R Down Fault," 2011 Chinese Control and Decision Conference (CCDC), 2011, pp. 75-80.

[5] M. Sonoda, Y. Watanabe and Y. Matsumoto, "Prediction of Failure Occurrence Time based on System Log Message Pattern Learning," 2012 IEEE Network Operations and Management Symposium, 2012, pp. 578-581.

[6] C. Wrench, F. Stahl, G. D. Fatta, V. Karthikeyan and D. Nauck, "A Rule Induction Approach to Forecasting Critical Alarms in a Telecommunication Network," 2019 International Conference on Data Mining Workshops (ICDMW), 2019, pp. 480-489.

[7] Z. Wang, M. Zhang, D. Wang, C. Song, M. Liu, J. Li, L. Lou, and Z. Liu, "Failure Prediction using Machine Learning and Time Series in Optical Network," Optics Express 25, 18553-18565 (2017).

[8] G. Weiss "Predicting Telecommunication Equipment Failures from Sequences of Network Alarms" Handbook of Knowledge Discovery and Data Mining, vol. U. K. Oxford: Oxford University Press, 2002.

[9] C.Zhang, M.Wang, M.Zhang, D.Wang, C.Song, L.Guan and Z.Liu "Adaptive Failure Prediction Using Long Short-term Memory in Optical Network," 24th OptoElectronics and Communications Conference (OECC) and International Conference on Photonics in Switching and Computing (PSC), 2019, pp. 1-3.

[10] S. Yuhensky, R. Munadi and Hafiddudin, "Forecasting Formulation Model for amount of Fault of the CPE Segment on Broadband Network PT. Telkom using ARIMA method," 2016 International Conference on Control, Electronics, Renewable Energy and Communications (ICCEREC), 2016, pp. 185-191.

[11] A.Tesfay "Neural Network based 3G Mobile Sites Fault Prediction: A Case Study in Addis Ababa, Ethiopia" Master Thesis, Addis Ababa University, 2018.

[12] L. H. Cui, Y. Zhao, B. Yan, D. Liu, and J. Zhang, "Deep-learning based Failure Prediction with Data Augmentation in Optical Transport Networks" in 17th International Conference on Optical Communications and Networks (ICOCN2018), 2019.

[13] M. Zhang and D. Wang, "Machine Learning Based Alarm Analysis and Failure Forecast in Optical Networks," 2019 24th Optoelectronics and Communications Conference (OECC) and 2019 International Conference on Photonics in Switching and Computing (PSC), 2019, pp. 1-3.

[14] B. Zhang, Y. Zhao, Y. Li and J. Zhang, "Transfer Learning aided Concurrent Multi-Alarm Prediction in Optical Transport Networks," 2020 Asia Communications and Photonics Conference (ACP) and International Conference on Information Photonics and Optical Communications (IPOC), 2020, pp. 1-3.

[15] J. Cavalcante, A. Patel and J. Celestino, "Performance Management of Optical Transport Networks through Time Series Forecasting," 2017 IEEE 31st International Conference on Advanced Information Networking and Applications (AINA), 2017, pp. 152-159.

[16] M. Ghobadi and R. Mahajan, "Optical Layer Failures in a Large Backbone" in Proceedings of the 2016 Internet Measurement Conference, 2016.

[17] W. Ji, S. Duan, R. Chen, S. Wang and Q. Ling, "A CNN-based Network Failure Prediction Method with Logs," 2018 Chinese Control And Decision Conference (CCDC), 2018, pp. 4087-4090.

[18] A. Altay, O. Ozkan, and G. Kayakutlu, "Prediction of Aircraft Failure Times using Artificial Neural Networks and Genetic Algorithms," Journal of Aircraft, vol. 51, no. 1, pp. 47–53, 2014.

[19] G. Scalabrini Sampaio, A. R. de A. Vallim Filho, L. Santos da Silva, and L. Augusto da Silva, "Prediction of Motor Failure Time using an Artificial Neural Network," Sensors, vol. 19, no. 19, p. 4342, 2019.

[20] C. C. Aggarwal, Neural Networks and Deep Learning: A Textbook. Cham, Switzerland: Springer Nature, 2019., pp 271-298.

[21] J. Han, M. Kamber, and J. Pei, Data Mining: Concepts and Techniques, 3rd edition, Oxford, England: Morgan Kaufmann, 2012, pp 103-127.

[22] F. Chollet, Deep learning with python. New York, NY: Manning Publications, 2017, pp. 56-67,196-222.

[23] S. Guido and A. C. Mueller, Introduction to Machine Learning with Python: A Guide for Data Scientists. Sebastopol, CA: O'Reilly Media, 2016. pp 7-15, 104-115.

[24] J. Sheng Tan, C. Kuan Ho, A. Hui Lan Lim, and M. Rizal bin Mohd Ramly,"Predicting Network Faults using Random Forest and C5.0," International Journal of Engineering & Technology., vol. 7, no. 2.14, p. 93, 2018.

[25] M. Swamynathan, Mastering Machine Learning with Python in Six Steps: A Practical Implementation Guide to Predictive Data Analytics using Python. Berlin, Germany: APress, 2019, pp. 67-77.

[26] C. Albon, Machine Learning with Python Cookbook: Practical Solutions from Preprocessing to Deep learning. O'Reilly Media, 2018, pp.297-325.

[27] A.Géron, Hands on Machine Learning with Scikit-Learn and Tensor Flow Concepts, Tools and Techniques to Build Intelligent Systems, pp.252-273,293-310,396-405.

[28] H. Zhuang, Y. Zhao, X. Yu, Y. Li, Y. Wang and J. Zhang, "Machine-Learning-based Alarm Prediction with GANs-based Self-Optimizing Data Augmentation in Large-Scale Optical Transport Networks," 2020 International Conference on Computing, Networking and Communications (ICNC), 2020, pp. 294-298.

[29] G. Zargarian ''Unsupervised Machine Learning for Mining Alarm Logs of a Large Telecommunication Network''Master Thesis, Politecnico di Torino, 2018.

[30] F. K. Oduro-Gyimah, J. Q. Azasoo and K. O. Boateng, "Statistical Analysis of Outage Time of Commercial Telecommunication Networks in Ghana," 2013 International Conference on Adaptive Science and Technology, 2013, pp. 1-8.

[31] L. Lou, M. Zhang, D. Wang, J. Li, X. Tang and L. Ai, "Alarm Compression Based on Machine Learning and Association Rules Mining in Optical Networks," 2018 23rd Opto-Electronics and Communications Conference (OECC), 2018, pp. 1-2.

[32] "KDnuggets," Kdnuggets.com. [Online]. Available: https://www.kdnuggets.com. [Accessed: 20-July-2021].

[33] M.M.Cambra "Using Recurrent Neural Networks to Predict the Time for an Event" Master Thesis, Universitat De Barcelona,2018.

[34] A. Schubert,"G.709-The Optical Transport Network (OTN) " 2021.

[35] R. K. Sundararajan, "The Optical Transport Network Building the Future of Transport, "2019.

[36] S. Aleksic,"Towards Fifth-Generation (5G) Optical Transport Networks," in 2015 17th International Conference on Transparent Optical Networks (ICTON), 2015.

[37] The International Telegraph and Telephone Consultative Committee, Ed., "Information Technology Open Systems Interconnection Systems Management: Alarm Reporting Function, " ITU, 1992.

[38] Keras Team, "Keras: the Python deep learning API, " Keras.io. [Online]. Available: https://keras.io., [Accessed: 5-Aug-2021].

[39] "Understanding LSTM Networks," Github.io. [Online]. Available: http://colah.github.io/posts/2015-08-Understanding-LSTMs/. [Accessed: 12-Aug-2021].

[40] T.Me, "The Project Spot," Theprojectspot.com. [Online]. Available: https://www.theprojectspot.com/tutorial-post/introduction-to-artificial-neural-networks. [Accessed: 19-Aug-2021].

[41] T. Liu, K. Kang and H. Sun, "Fault Prediction for Satellite Communication Equipment Based on Deep Neural Network," 2018 International Conference on Virtual Reality and Intelligent Systems (ICVRIS), 2018, pp. 176-178.

[42] R.Pant, P.H.Ghetia and A.Gupta "Telecom Analytics: WAN Outage Prediction "Conference Paper, 2020.

[43] D. A. C. Almeida ''Modelling and Forecasting Incidents in Cellular Networks'' Master Thesis, Técnico Lisboa, November 2017.

[44] K. Venugopal, P. Madhusudan and A. Amrutha, "Artificial Neural Network based Fault Prediction Framework for Transformers in Power Systems," 2017 International Conference on Computing Methodologies and Communication (ICCMC), 2017, pp. 520-523.

[45] T. Kibatu ''Recurrent Neural Network-based Base Transceiver Station Power System Failure Prediction'' Master Thesis, Addis Ababa University,2019.

[46] P.T. Yamak, L. Yujian, and P. K. Gadosey, "A Comparison between ARIMA, LSTM, and GRU for Time Series Forecasting" In Proceedings of the 2019 2nd International Conference on Algorithms, Computing and Artificial Intelligence (ACAI 2019), Association for Computing Machinery, New York, NY, USA, 49–55.

[47] F. M. Butt, L. Hussain, A. Mahmood, and K. J. Lone "Artificial Intelligence based Accurately Load Forecasting System to Forecast Short and Medium-term Load Demands " Mathematical Biosciences and Engineering, vol. 18, no. 1, pp. 400–425, 2020.

[48] B. Yan et al., "First Demonstration of Imbalanced Data Learning-Based Failure Prediction in Self-Optimizing Optical Networks with Large Scale Field Topology," 2018 Asia Communications and Photonics Conference (ACP), 2018, pp. 1-4.

[49] A. Stepchenko, J. Chizhov, L. Aleksejeva, and J. Tolujew, "Nonlinear, Non-stationary and Seasonal Time Series Forecasting using Different Methods Coupled with Data Preprocessing," Procedia Comput. Sci., vol. 104, pp. 578–585, 2017.

[50] K. Celikmih, O. Inan, and H. Uguz, "Failure Prediction of Aircraft Equipment using Machine Learning with a Hybrid Data Preparation Method," Sci. Program., vol. 2020, pp. 1–10, 2020.

[51] M. Jaudet, N. lqbal, A. Hussain and K. Sharif, "Temporal Classification for Fault-Prediction in a Real world Telecommunications Network," Proceedings of the IEEE Symposium on Emerging Technologies, 2005., 2005, pp. 209-214.

[52] R. Chandra, S. Goyal and R. Gupta, "Evaluation of Deep Learning Models for Multi-Step Ahead Time Series Prediction," in IEEE Access, vol. 9, pp. 83105-83123, 2021.

[53] J. F. Torres, D. Hadjout, A. Sebaa, F. Martínez-Álvarez, and A. Troncoso, "Deep Learning for Time Series Forecasting: A survey," Big Data, vol. 9, no. 1, pp. 3–21, 2021.

[54] Ž. Deljac, M. Kunštić and B. Spahija, "A Comparison of Traditional Forecasting Methods for Short-term and Long-term Prediction of Faults in the Broadband Networks," 2011 Proceedings of the 34th International Convention MIPRO, 2011, pp. 517-522.

Annex

Fault Prediction in Optical Transport Network case of Ethio Telecom

Yonas Bekele
School of Electrical and Computer Engineering
Addis Ababa Institute of Technology
Addis Ababa University
Addis Ababa, Ethiopia
yonniget@gmail.com

Yalemzewd Negash (PhD)
School of Electrical and Computer Engineering
Addis Ababa Institute of Technology
Addis Ababa University
Addis Ababa, Ethiopia
yalemzewdn@yahoo.com

***Abstract*- The availability of optical transport network (OTN) is an important factor to provide dependable telecom services. This backbone infrastructure suffers from frequent outages due to multiple reasons and fault alarm data are exploding in magnitude. Hence, fault interarrival time prediction would help take preventive actions before its occurrence and service interruption. In this research, machine learning based fault interarrival time prediction models have been developed using Artificial Neural Network (ANN), Long Short Term Memory (LSTM) and Gated Recurrent Unit (GRU) algorithms. A five months fault history data exported from Network Management System (NMS) has been used to predict fault interarrival time in the network. The prediction performance of these models have been evaluated using Mean Squared error (MSE), Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE) and R-Square (R^2).**

Since fault occurrence is random and the dataset is nonlinear, the models prediction results indicate that the ANN model has predicted the fault interarrival time with 16.08% Mean Absolute Percentage Error (MAPE) while GRU shows 16.23% MAPE and LSTM prediction error is 17.93% MAPE on average. The R^2 score for the ANN model falls between 0.2576 and 0.4923 followed by the GRU model that exhibits its score in between 0.2387 and 0.4912. The R^2 score for LSTM model lies between 0.1583 and 0.4641. Therefore, the ANN model is able to catch the general trend of the fault interarrival time followed by the GRU model. In addition, the average computational time taken per epoch by ANN model is 2.44 ms, GRU model consumes 10.22 ms and LSTM takes 13.94 ms. Hence, ANN model outperforms in terms of prediction performance and computational efficiency.

Keywords: ***Fault Prediction, Optical Transport Network, Network Management System (NMS), Machine Learning, Artificial Neural Network (ANN), Long Short Term Memory (LSTM), Gated Recurrent Unit (GRU).***

I. INTRODUCTION

Fault prediction plays an important role on improving the operational reliability and availability of optical networks. Faults in optical networks cause service interruption and huge data loss. The reliability of optical backbone network is crucial for telecom operators as more traffics being carried on the infrastructure and to provide the required quality of service to customers. To support the growing demand of high capacity transmission links due to the booming data traffic, broadband services and voice calls Optical Transport Network (OTN) has become solution as backbone infrastructure. Despite the best effort to maintain and protect the optical backbone network from multiple faults, the network suffers from frequent faults and unavailability. The optical network protection algorithms take passive actions which are not capable

of preventing the faults in advance. These faults may occur due to Network Equipment (NE) hardware, software, power supply, fiber bend or cut, lossy fiber, optical transponder and transceiver module failures, etc. The availability of backbone optical transmission affects the availability of network elements that are directly connected in the backbone network like IP router that use the optical network for its transport medium and other indirectly connected access network devices. Hence, a fault that occurs in the optical backbone network can be considered as fault in other connected network as it can affect their performance and availability. Because of frequent optical backbone network faults, Quality of Service (QoS) is affected, the operator loses huge amount of its income and incur high maintenance cost. In telecom networks usually reactive and preventive maintenance are performed. Reactive maintenance is carried out when there is fault in network or service failure alarm is triggered on the Network Management System (NMS). Preventive maintenance is characterized by scheduled maintenance operations in order to avoid optical nodes, links, etc. from failures using preventive maintenance procedure. Therefore, by performing fault occurrence time prediction in optical transport network preventive actions can be taken to links and nodes recognized as the most common generators of network faults to reduce the number of reported fault alarms and the operational expenses of the operator. Further, the operator can perform time, work force and resource scheduling to ensure services availability and reliability of the optical transport network.

Hence, the main objective of the research is to perform machine learning based fault inter-arrival time prediction in optical transport network using deep learning algorithms and compare their prediction performance.

II. Relevant Work

Fault prediction using machine-learning techniques is becoming a good research area in telecommunication networks in recent times. Much of the researches done in optical network focus on connection resilience using the protection and restoration schemes. Despite these great efforts to enhance the availability of the optical networks, fault occurrence time prediction also aids to proactively reduce the amount of faults expected to occur in the optical network.

The authors in [1] performed fault prediction in nationwide optical backbone network by collecting alarms coming directly from all devices with their topological informations. They extracted critical alarms and the time differences between two consecutive alarms are accounted for prediction using neural network toolbox in MATLAB. A four layer feed forward neural network trained with Levenberg Marquardt back propagation algorithm shows a better convergence of Mean Squared Error (MSE) to 10^{-10}. They also tried Recurrent Neural Network (RNN) of the same structure that slowly approached to the desired MSE of 10^{-10} but the predicted values were not as accurate as feed forward network. After several trials, they were able to predict the future fault occurrence time close to the actual one. The error in prediction of each consecutive fault is less than 0.5 milliseconds, which is a high precision result.

The authors in [2] performed prediction of fault inter-arrival times in cellular network based on one-month faults data from one of national mobile operators. They investigated the prediction accuracy of Neural Network, Linear Regression and other five models using Normalized Root Mean Squared Error (NRMSE) and Root Mean Squared Error (RMSE) metrics on MATLAB. The Deep Neural Network with Auto encoders and Nonlinear Autoregressive Neural Network models attained better prediction accuracy than the others.

The author in [3] collected ten months wireless network log of a company to predict whether the network will fail in the near future

using different time windows. In their experiment, they proposed a hybrid Convolutional Neural Network and Long Short Term Memory (CNN-LSTM) prediction model for the wireless network faults. Further, the hybrid model performance is compared with CNN and Random Forest models. The result shows that the prediction performance of the hybrid CNN-LSTM model is better than the other two models.

The authors in [4] performed fault occurrence time prediction in wireless mobile network used for train-ground communication known as GSM for Railways (GSM-R). They proposed Hidden Markovian Chain (HMC) framework that consists a scheme based on Viterbi algorithm and absorption probability to predict the fault occurrence time. Viterbi algorithm was utilized to estimate system state. A dynamic correction mechanism based on GSM-R quality was used to modify the predicted occurrence time. The dynamic scheme possesses more prediction accuracy than its static version.

The author in [5] collected log data obtained from a large scale IT system to predict the time period when a failure is expected to occur using Bayesian learning from log messages and past failure reports. The results show that their method predicted the occurrence of failure with time interval of 60 minutes before the occurrence of failures with sufficient accuracy.

The authors in [6] applied Auto Regressive Integrated Moving Average (ARIMA) model to find prediction formulae in order to evaluate the expected number of faults at time t with t-1, t-2, t-3, etc. in a broadband network. ARIMA (4, 1, 5) model achieved better result with the least Cumulative Mean Square Error (CMSE) in each fault types studied.

III. Problem Formulation

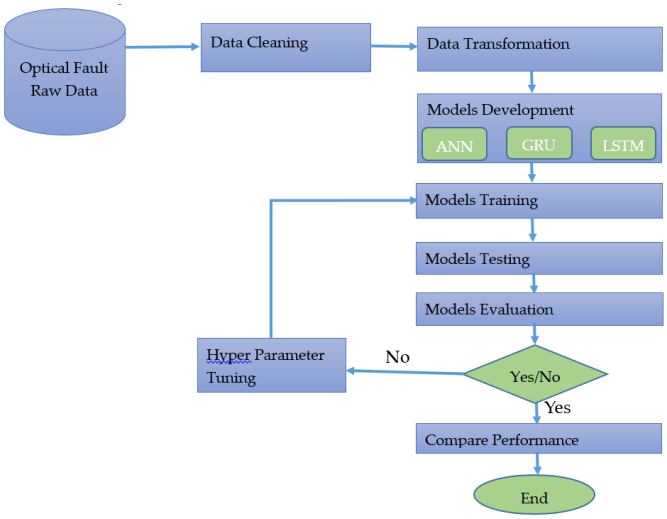

Fig. 1: Schematic Diagram of the Workflow

Data Collection

Alarm data of critical severity level that occurred in the Optical Transport Network (OTN) are collected from the Network Monitoring System (NMS) for the research.

Data Preprocessing

MS-Excel tool is used to collect, extract and sequentially arrange the fault occurrence time. Jupyter notebook is used to pre-process and perform fault inter-arrival time prediction of the optical transport network.

Model Development

To perform the fault interarrival time prediction models will be developed using the selected Artificial Neural Network (ANN), Long Short Term Memory (LSTM) and Gated Recurrent Unit (GRU) algorithms.

IV. Machine Learning Algorithms

Artificial Neural Network (ANN)

Human brain contains billion of neurons that are connected to many other neurons to form a network. Neurons in human brain are capable of making very complex decisions, so this means they run many parallel processes. In a similar manner, neural network

was designed to make artificial interconnected neurons like biological neurons. The basic computational element is called a node or unit. It receives inputs from external sources. Each input has an associated weight. The unit computes function of the weighted sum of its inputs. A neural network is usually structured into an input layer of neurons, one or more hidden layers and one output layer which are usually interconnected in a feed-forward way.

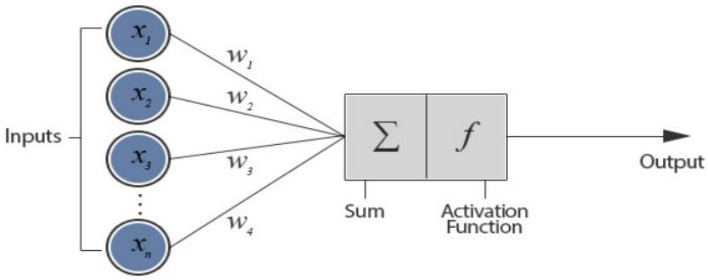

Fig.2: Modeling Artificial Neurons [17]

The working of a neuron can be represented mathematically by the equation [17]:

$$y(x)=f(w\Theta x+b) \quad (1)$$

Where x, w, b, f, y represent input vector, weight vector, neuron bias, element-wise multiplication, activation function, and neuron output respectively.

Long Short-Term Memory (LSTM)

Long Short Term Memory (LSTM) is a type of recurrent neural network with a strong ability to learn and predict sequential data and their long-term dependencies more precisely than traditional recurrent neural networks. Recurrent Neural Network (RNN) has the form of a chain of repeating modules of neural network. This repeating module has a very simple structure such as a single tanh layer and is limited in maintaining long-term memory. Therefore, the LSTM has the ability to overcome the problem of short-term memory and vanishing gradients by adding memory structure with gates. LSTM contains four interacting layers that can remember short-term memories for a very long time.

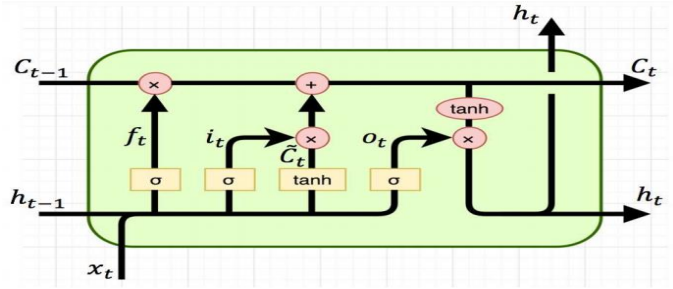

Fig. 3: Structure of LSTM Cell [13].

The LSTM has the following equations for its gates [3]:

Forget gate: $f_t = \sigma(W_f.[h_{t-1},x_t] + b_f)$ (2)

Input gate: $i_t = \sigma(W_i.[h_{t-1},x_t] + b_i)$ (3)

Output gate: $o_t = \sigma(W_o.[h_{t-1},x_t] + b_o)$ (4)

Intermediate Cell State: $\tilde{C}_t = \tanh(W_c.[h_{t-1},x_t] + b_c)$ (5)

Cell State (next memory input): $C_t = f_t \circ C_{t-1} + i_t \circ \tilde{C}_t$ (6)

New State: $h_t = o_t \circ \tanh(C_t)$ (7)

$$\sigma(x) = \frac{1}{1+e^{-x}} \quad (8)$$

$$\tanh(x) = \frac{e^{x}-e^{-x}}{e^{x}+e^{-x}} \quad (9)$$

Where f_t, i_t, o_t, h_t, C_t are forget gate, input gate, output gate, intermediate output and cell state. W_f,W_i, W_o and Wc are the weight matrices of each of the four layers. b_f, b_i, b_o and b_c are the bias terms for each of the four layers, σ represents the sigmoid function.

Gated Recurrent Unit (GRU)

Gated Recurrent Units are kind of recurrent networks that are emerged as a simplification of LSTMs by merging forget and input gate into update gate z_t. GRU has less complicated structure compared to LSTM that makes it computationally faster as it has only two gates [11]. It has a reset gate r_t and an update gate z_t but

lacks an output gate that are used to decide what information should be passed to the output. The reset gate defines how to combine the new input with the previous memory. Reset gates r_t introduces nonlinear effect in the relationship between past state and future state. Update gate is used to control the degree to which the state information of the previous time is brought into the current state.

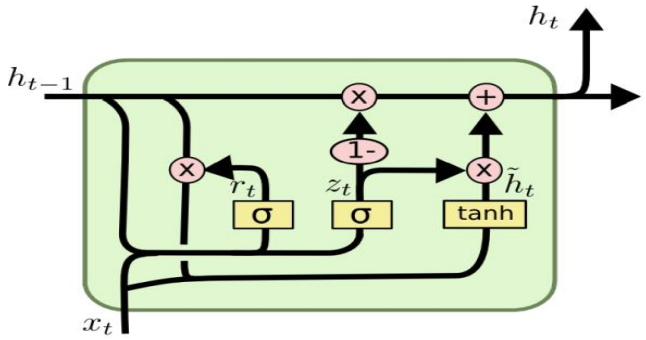

Fig. 4: Structure of GRU Cell [13].

The GRU has the following equations for its gates [11]:

Update gate**:** $z_t = \sigma(W_z.[h_{t-1}, x_t] + b_z)$ (10)

Reset gate: $r_t = \sigma(W_z.[h_{t-1}, x_t] + b_r)$ (11)

Cell state: $\tilde{h}_t = tanh(W.[r_t \circ h_{t-1}, x_t] + b)$ (12)

New state: $h_t = (1-z_t) \circ h_{t-1} + z_t \circ \tilde{h}_t$ (13)

The symbol ∘ indicates element wise multiplication.

The activation function used in the experiment is the ReLU function which is fast in convergence [3].

V. Performance Evaluation Metrics

The most commonly used metrics for time series prediction models are Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), Mean Squared Error (MSE) and coefficient of determination or R-squared (R^2).

Mean Absolute Error (MAE): measures the average absolute difference between actual and predicted values. It measures the mean absolute deviation.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \quad (14)$$

Mean Absolute Percentage Error (MAPE): measures the average of the absolute percentage error between actual and predicted values. The smaller value of MAPE indicates better prediction result.

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \quad (15)$$

Mean Squared Error (MSE) measures the average squared sum of the error between actual and predicted values. Since the error metric squares the errors, the direction of the overall error is lost. The lower the value of MSE, the lower the total squared error and thus the better the model.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \quad (16)$$

R-squared (R^2): is coefficient of determination that measures how well the regression predictions approximates the real data points. Its value ranges between 0 and 1. The value 1 indicates that the regression predictions perfectly fit the data.

Sum of Squared Error (SSE) - residual sum of squared error.

$$SSE = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \quad (17)$$

Total Sum of Squares (SST) - proportional to the sample variance.

$$SST = \sum_{i=1}^{n}(y_i - \bar{y})^2 \quad (18)$$

R-squared

$$R^2 = 1\text{-}SSE/SST \quad (19)$$

where y_i is actual value of sample i, $\bar{y}$ is mean of the actual values and $\hat{y}_i$ is predicted value of sample i and n-sample size.

VI. Result and Analysis

Hyperparameters for ANN Model

Neurons	Dropout Rate	Activation Function	Optimizer	Learning Rate
256,128,64	0.1	Relu	Adam	5e-5

The ANN model is trained using different learning rate, epochs, etc. The loss function is MSE and the activation function ReLU. The number of hidden layers is varied in the range from 2 to 4 and the number of hidden neurons from 16 to 256. After tuning several hyper-parameters, the best performing hyper parameters combination is selected. The actual and predicted fault interarrival time prediction of one-month data trained on 70/30 scheme using ANN model is shown below. The x-axis shows occurrence of the sequential alarm samples and the y-axis indicates the time difference between two consecutive alarms.

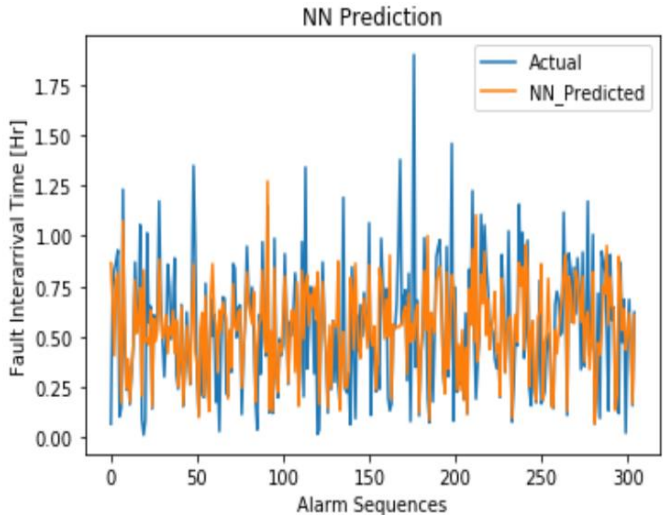

Fig. 5. ANN Prediction

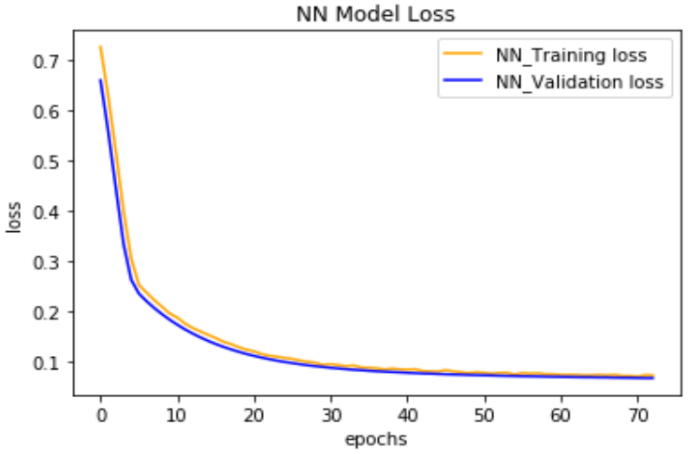

Fig. 6 : ANN Model Loss

Hyperparameters for LSTM Model

Neurons	Dropout Rate	Activation Function	Optimizer	Learning Rate
256,128,64	0.1	Relu	Adam	5e-5

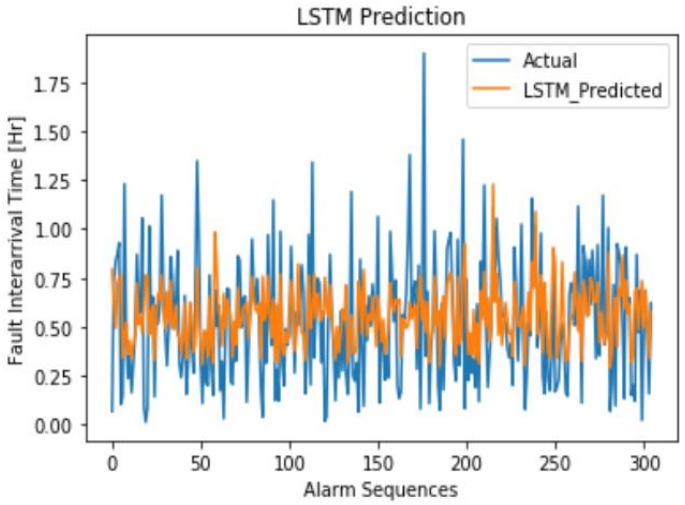

Fig. 7. LSTM Model Prediction

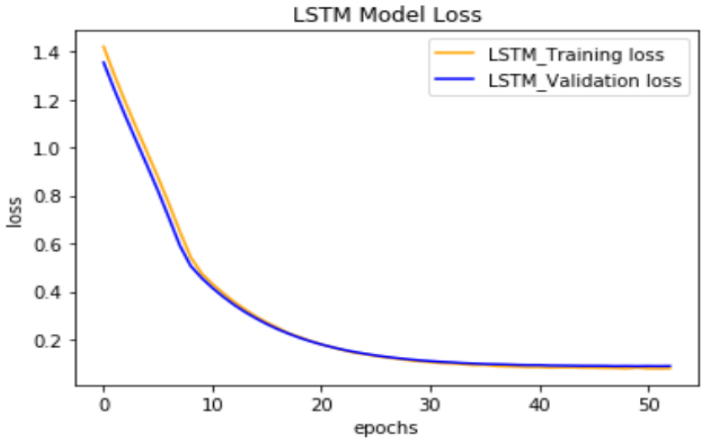

Fig. 8: LSTM Model Loss

HYPERPARAMETERS FOR GRU MODEL

Neurons	Dropout Rate	Activation Function	Optimizer	Learning Rate
256,128,64	0.1	Relu	Adam	5e-5

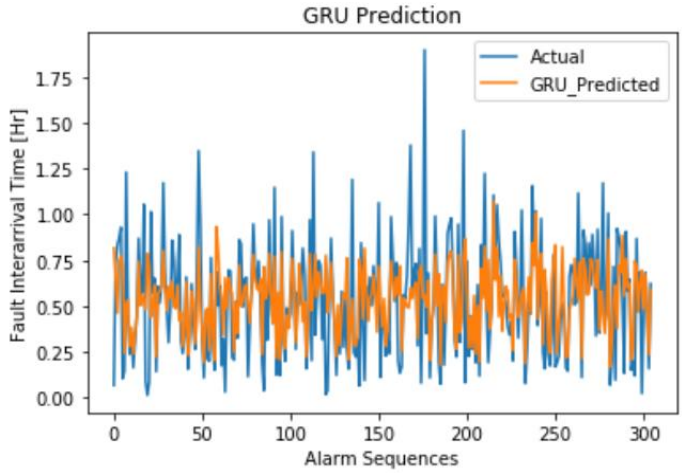

Fig. 9. GRU Model Prediction

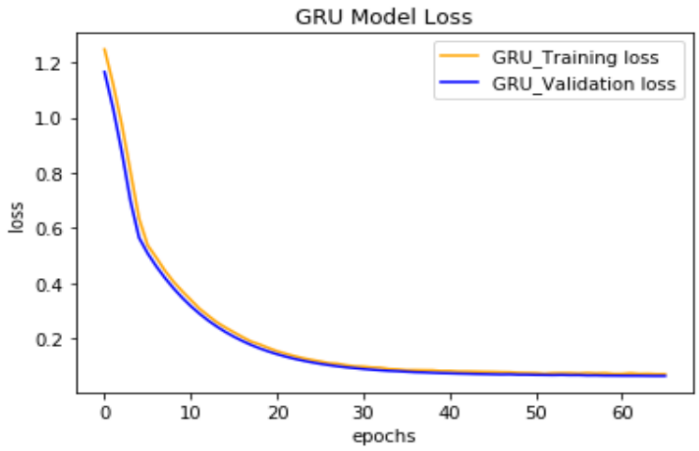

Fig. 10. GRU Model Loss

The combined plot of the three models prediction trained with 70/30 scheme of the one-month data is shown below.

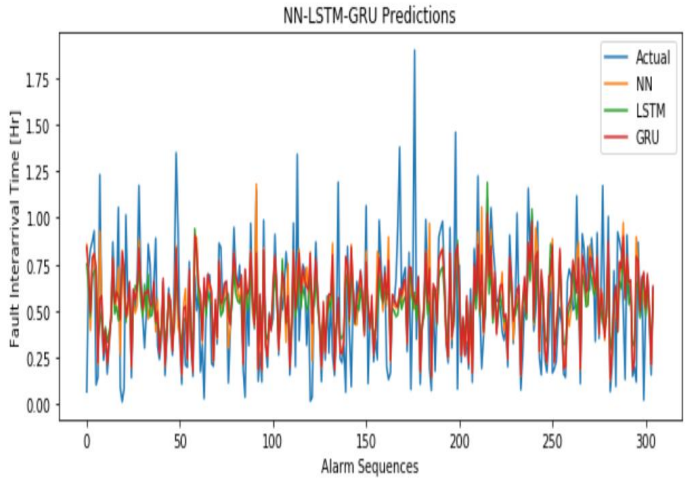

Fig. 11: NN-LSTM-GRU Models Prediction

VII. CONCLUSION

In this article, machine-learning algorithms are used to build fault prediction models for optical transport network. The prediction models used fault interarrival time history of the network to be trained. The three models built by the ANN, LSTM and GRU algorithms have been trained with three different training schemes to study the fault interarrival pattern in the network and select the best model in prediction performance. Even though the dataset is nonlinear and the fault arrival is random, the models

prediction results indicate that the ANN model has predicted the fault interarrival time with 16.08% MAPE while GRU shows 16.23% MAPE and LSTM prediction error is 17.93% MAPE in average. The nonlinear trend of the data has made the models not easily learn the fault interarrival pattern and refrain them from great success in prediction. Here, the ANN model is better to learn the irregular nature of the data followed by GRU and then LSTM model. Further, since the models try to predict the fault occurrence of the whole optical transport network instead of a single node or optical channel the prediction error has become higher. In addition, the average computational time taken per epoch for the ANN model is 2.44ms ms, GRU model consumes 10.22 ms and LSTM takes 13.94 ms. So ANN model takes less time per epoch. Hence, ANN outperforms the in terms of prediction performance and computational efficiency.

VIII. FUTURE WORK

Similar approach can be applied to network element (NE) or node level that can be used to localize the faults which helps to troubleshoot and maintain the faults promptly. Additionally, minimizing the prediction error can be another task that could help expect the fault arrival as close as the actual occurrence time so that proactive actions can be taken to remove the fault before affecting services. Hybrid model designing using deep learning algorithms could also help further refine the results. Moreover, designing optimal number of layers and neurons that can learn the non-linearities of fault interarrival time can be a future work.

REFERENCES

[1] M. Jaudet, N. Iqbal and A. Hussain, "Neural networks for fault-prediction in a telecommunications network," 8th International Multitopic Conference, 2004. Proceedings of INMIC 2004., 2004, pp. 315-320.

[2] Y. Kumar, H. Farooq and A. Imran, "Fault prediction and reliability analysis in a real cellular network," 2017 13th International Wireless Communications and Mobile Computing Conference (IWCMC), 2017, pp. 1090-1095.

[3] Z. Tan and P. Pan, "Network Fault Prediction Based on CNN-LSTM Hybrid Neural Network," 2019 International Conference on Communications, Information System and Computer Engineering (CISCE), 2019, pp. 486-490.

[4] M. Chang, Y. Ji, W. Dong and Z. Zhang, "Hidden Markovian Chain based occurrence time prediction for GSM-R down fault," 2011 Chinese Control and Decision Conference (CCDC), 2011, pp. 75-80.

[5] M. Sonoda, Y. Watanabe and Y. Matsumoto, "Prediction of failure occurrence time based on system log message pattern learning," 2012 IEEE Network Operations and Management Symposium, 2012, pp. 578-581.

[6] S. Yuhensky, R. Munadi and Hafiddudin, "Forecasting formulation model for amount of fault of the CPE segment on broadband network PT. Telkom using ARIMA method," 2016 International Conference on Control, Electronics, Renewable Energy and Communications (ICCEREC), 2016, pp. 185-191.

[7] Z. Wang, M. Zhang, D. Wang, C. Song, M. Liu, J. Li, L. Lou, and Z. Liu, "Failure prediction using machine learning and time series in optical network," Opt. Express 25, 18553-18565 (2017).

[8] C.Zhang, M.Wang, M.Zhang, D.Wang, C.Song, L.Guan and Z.Liu "Adaptive Failure Prediction Using Long Short-term Memory in Optical Network," 24th OptoElectronics and Communications Conference (OECC) and International Conference on Photonics in Switching and Computing (PSC), 2019, pp. 1-3.

[9] G. Weiss "Predicting Telecommunication Equipment Failures from Sequences of Network Alarms'' Handbook of Knowledge Discovery and Data Mining, vol. U. K. Oxford: Oxford University Press, 2002.

[10] C. Wrench, F. Stahl, G. D. Fatta, V. Karthikeyan and D. Nauck, "A Rule Induction Approach to Forecasting Critical Alarms in a Telecommunication Network," 2019 International Conference on Data Mining Workshops (ICDMW), 2019, pp. 480-489.

[11] P.T. Yamak, L. Yujian, and P. K. Gadosey, "A Comparison between ARIMA, LSTM, and GRU for Time Series Forecasting" In Proceedings of the 2019 2nd International Conference on Algorithms, Computing and Artificial Intelligence (ACAI 2019), Association for Computing Machinery, New York, NY, USA, 49–55.

[12] F.Chollet, Deep learning with python. New York, NY: Manning Publications, 2017, pp. 56-67,196-222.

[13] "Understanding LSTM Networks," Github.io. [Online]. Available: http://colah.github.io/posts/2015-08-Understanding-LSTMs/. [Accessed: 12-Aug-2021].

[14] A. Essien and C. Giannetti, "A Deep Learning Framework for Univariate Time Series Prediction Using Convolutional LSTM Stacked Autoencoders," 2019 IEEE International Symposium on INnovations in Intelligent SysTems and Applications (INISTA), 2019, pp. 1-6

[15] R. Chandra, S. Goyal and R. Gupta, "Evaluation of Deep Learning Models for Multi-Step Ahead Time Series Prediction," in IEEE Access, vol. 9, pp. 83105-83123, 2021.

[16] F. M. Butt, L. Hussain, A. Mahmood, and K. J. Lone, "Artificial Intelligence based accurately load forecasting system to forecast short and medium-term load demands," Mathematical Biosciences and Engineering, vol. 18, no. 1, pp. 400–425, 202