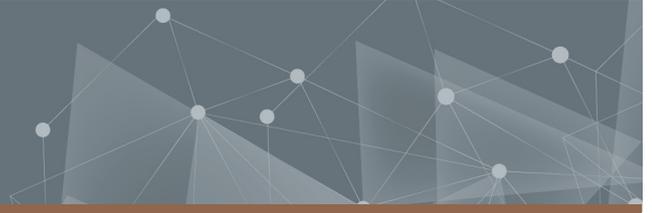




**CHALMERS**  
UNIVERSITY OF TECHNOLOGY



# DC Line Fault Prognosis Using Deep Recurrent Neural Network Over Sensor Data

Master's thesis in Electric Power Engineering

Akhil Venkat Vissakodeti

**DEPARTMENT OF ELECTRICAL ENGINEERING**

CHALMERS UNIVERSITY OF TECHNOLOGY

Gothenburg, Sweden 2023

[www.chalmers.se](http://www.chalmers.se)



MASTER'S THESIS 2023

# DC Line Fault Prognosis Using Deep Recurrent Neural Network Over Sensor Data

Akhil Venkat Vissakodeti



**CHALMERS**  
UNIVERSITY OF TECHNOLOGY

Department of Electrical Engineering  
CHALMERS UNIVERSITY OF TECHNOLOGY  
Gothenburg, Sweden 2023

DC Line Fault Prognosis Using Deep Recurrent Neural Network Over Sensor Data  
Akhil Venkat Vissakodeti

© Akhil Venkat Vissakodeti, 2023.

Supervisor: Arkadiusz Burek, HITACHI ENERGIES  
Advisor: Examiner: Name, Department

Master's Thesis 2023  
Department of Electrical Engineering  
Chalmers University of Technology  
SE-412 96 Gothenburg  
Telephone +46 31 772 1000

Cover: DC faults are identified using the Machine learning techniques on the Baltic sea cable. This would help to predict the faults before it occurs in the original DC cable and reduce the effect of failure of the line.

Typeset in L<sup>A</sup>T<sub>E</sub>X  
Printed by Chalmers Reproservice  
Gothenburg, Sweden 2023

DC Line Fault Prognosis Using Deep Recurrent Neural Network Over Sensor Data  
Akhil Venkat Vissakodeti  
Department of Electrical Engineering  
Chalmers University of Technology

## **Abstract**

The HVDC technology has become prominent because of its increased long-distance bulk power transmission efficiency and facilitation of asynchronous interconnections. The loaded cable can, however, fail due to flashover or short circuit in the power system. As a result, this can cause a grid failure and damage the equipment by introducing a high level of current in the system. To detect fault is therefore considered a cost-efficient and non-destructive technique to monitor the cable operating condition. The main aim of this thesis is to predict faults in a DC cable using measured data from the sensors present in the system. Moreover, this method helps to identify the cable fault before power failure with possible catastrophic consequences occurs.

This thesis examines the prospect of employing deep neural networks to capture the hidden patterns from the time series sensors to predict DC cable fault at early stages. This is justified because deep learning approaches are well suited to incorporating feature extraction into the predictive model. In this regard, long short-term memory (LSTM) is considered to get a remarkable accuracy of 99.93%. A lower Relative value of the absolute error of the signals proves that the model predicts the accurate results for the fixed window size.



## Acknowledgements

I want to give our greatest thanks to the supervisor at Hitachi Energy AB, Arkadiusz Burek, for being significantly invested in helping me to achieve the goals in this thesis and consistently providing us with valuable information and feedback. This thesis would not have been complete without your help.

Secondly, I want to thank the advisor and Examiner, Thomas Hammarström, for giving us great feedback during our meetings and guiding us through this process.

Lastly, I would like to thank my parents, Subba Rao and Lakshmi, for giving me this opportunity to pursue a master's. Further, I would like to thank Janani mudiliyar for her moral support in this thesis.

Akhil Venkat Vissakodeti, Gothenburg, June 2022



# List of Acronyms

Below is the list of acronyms that have been used throughout this thesis listed in alphabetical order:

AC	Alternating Current
AI	Artificial Intelligence
ANN	Artificial neural networks
DC	Direct current
DL	Deep Learning
ETDM	Encapsulated time division multiplier
HVDC	High Voltage Direct Current
KPSS	Kwiatkowski–Phillips–Schmidt–Shin
MACH	Modular Advanced Control for HVDC
MAE	Mean absolute error
MIMS	MACH Integrated Management system
ML	Machine Learning
MSE	Mean Squared error
NLP	Natural language Processing
LSTM	Long short-term memory
RAE	Relative absolute error
RNN	Recurrent neural network
RRSE	Root relative squared error
TSO	Transmission System Operator
VSC	Voltage Source Converter



# Contents

<b>List of Acronyms</b>	<b>ix</b>
<b>List of Figures</b>	<b>xiii</b>
<b>List of Tables</b>	<b>xv</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Background . . . . .	2
1.2 Objective . . . . .	3
1.3 Limitation . . . . .	3
<b>2 Theory</b>	<b>5</b>
2.1 Basics of HVDC system . . . . .	5
2.1.1 HVDC Schemes . . . . .	6
2.2 DC faults . . . . .	7
2.2.1 DC pole to pole short circuit: . . . . .	8
2.2.2 DC line to ground faults: . . . . .	9
2.3 DC fault identification method in HVDC: . . . . .	10
2.3.1 Handshaking method . . . . .	10
2.3.2 Voltage and current derivatives based method . . . . .	11
2.3.3 Transient based method for DC fault detection . . . . .	11
2.3.4 Impedance based method . . . . .	12
2.4 Basics of Machine learning . . . . .	13
2.5 Stationary test . . . . .	14
2.6 Neural Network . . . . .	15
2.6.1 Recurrent neural network . . . . .	15
2.6.2 Long short temporary memory . . . . .	16
2.7 Model Evaluation . . . . .	17
<b>3 Methods</b>	<b>19</b>
3.1 Cleaning Data . . . . .	19
3.1.1 Data generation . . . . .	20
3.1.2 Data merging using Python . . . . .	20
3.2 Feature selection . . . . .	21
3.3 Preprocessing . . . . .	21
3.3.1 Missing Values . . . . .	22
3.3.1.1 Filling the missing values with forwarding fill . . . . .	22

3.3.1.2	Deleting rows with the missing values . . . . .	22
3.3.2	Removing the outliers and smoothing . . . . .	23
3.3.3	Stationary test . . . . .	24
3.3.4	Differentiating the time series data . . . . .	24
3.3.5	Feature Scaling . . . . .	24
3.3.5.1	Normalization . . . . .	25
3.3.5.2	Standardization . . . . .	25
3.4	Train and Test . . . . .	26
3.5	Sliding Window . . . . .	27
3.6	LSTM architecture . . . . .	28
3.6.1	Input Data . . . . .	28
3.6.2	Layers . . . . .	28
3.6.2.1	Hidden layer . . . . .	28
3.6.2.2	Dense layer . . . . .	28
3.6.2.3	Number of layers . . . . .	29
3.6.3	Optimal hyperparameter . . . . .	29
3.6.3.1	Dropout . . . . .	29
3.6.3.2	The number of epochs for training the model . . . . .	30
3.6.3.3	Optimal Cells in each LSTM layer . . . . .	31
3.6.4	Loss function in LSTM . . . . .	31
3.6.5	LSTM optimizer . . . . .	32
3.6.5.1	Error calculation . . . . .	32
<b>4</b>	<b>Results</b>	<b>33</b>
4.1	Prediction of test data after inverse differentiation and denormalization	33
4.1.1	Comparison of voltage prediction with original value of the voltage: . . . . .	33
4.1.2	Comparison of current prediction with the actual value of the current . . . . .	34
4.1.3	Comparison of power prediction with the original value of the power: . . . . .	35
4.1.4	Comparison of resistance prediction with the actual value of the resistance: . . . . .	36
4.2	Future Prediction for the fault . . . . .	37
<b>5</b>	<b>Discussion</b>	<b>39</b>
<b>6</b>	<b>Conclusion</b>	<b>41</b>
6.1	Ethical Aspects . . . . .	41
<b>7</b>	<b>Future Work</b>	<b>43</b>
	<b>Bibliography</b>	<b>45</b>

# List of Figures

2.1	A basic structure of HVDCsystem . . . . .	5
2.2	Monopolar HVDCsystem . . . . .	6
2.3	Bipolar HVDCsystem . . . . .	7
2.4	Homopolar HVDCsystem . . . . .	7
2.5	Capacitor discharge stage . . . . .	8
2.6	capacitor recharging state . . . . .	8
2.7	Diode freewheeling Stage . . . . .	8
2.8	DCline to ground Capacitor discharge stage . . . . .	10
2.9	Grid current feeding stage . . . . .	10
2.10	Voltage and current derivatives based method . . . . .	11
2.11	Transient based method for DC fault detection . . . . .	12
2.12	Impedance based method . . . . .	12
2.13	Venn diagram showing machine learning, AI, Deep Learning etc., correlate[24] . . . . .	14
2.14	Neural network schematic . . . . .	15
2.15	Structure of RNN . . . . .	16
2.16	Structure of LSTM . . . . .	16
3.1	The structure of Data cleaning . . . . .	19
3.2	Data generation methodology . . . . .	20
3.3	Missing value Data set . . . . .	22
3.4	smoothed graph using low pass filter . . . . .	23
3.5	Unsmoothed graph before low pass filtering . . . . .	23
3.6	Differentiating the time series signal . . . . .	24
3.7	Feature scaling diagram as described in [4] . . . . .	25
3.8	Example of Train Split procedure . . . . .	26
3.9	Sliding window process . . . . .	27
3.10	Structure of the three-Dimensional matrix . . . . .	28
3.11	optimal Dropout Number . . . . .	30
3.12	The optimal number of Epochs . . . . .	31
3.13	Optimal number of Cells in each layer . . . . .	31
3.14	Training and validation loss . . . . .	32
4.1	Comparison of voltage prediction with actual value of voltage . . . . .	34
4.2	Comparison of current prediction with actual value of current . . . . .	35
4.3	Comparison of power prediction with actual value of power . . . . .	36
4.4	Comparison of resistance prediction with actual value of resistance . . . . .	37

## List of Figures

---

4.5	Future fault prediction by comparing the voltage current power and resistance . . . . .	38
7.1	Variational AutoEncoder Architecture . . . . .	44

# List of Tables

3.1	Maximum and Minimum Values for Current, Voltage and Power . . .	25
3.2	LSTM Network Architecture . . . . .	29
4.1	Performance measure of the constructed predictive models on validation data( case voltage) . . . . .	34
4.2	Performance measure of the constructed predictive models on validation data( case current) . . . . .	34
4.3	Performance measure of the constructed predictive models on validation data( case power) . . . . .	35
4.4	Performance measure of the constructed predictive models on validation data( case resistance) . . . . .	36



# 1

## Introduction

In the eighteenth century, electric power was first transported using direct current after the invention of electricity. In 1882, direct current was used to transmit and distribute electrical energy. Although this was a wrong step, at that time it was replaced by an alternate current (AC) power transmission system. Since then, it has been widely assumed that alternating current (AC) is the most effective method of transmitting electrical energy across long distances. However, when the distance between load centres and generation grows, the losses in the AC transmission system exceed tolerable.

Nevertheless, with the development of power electronics, DC was adapted towards commercial use with HVDC. The first HVDC project commenced in 1954 in Sweden as a Gotland. As a result of the development in semiconductor technology, high power converter and inverters promoted the idea of HVDC.[54]

DC transmission system has many advantages compared to the AC transmission system in a long distance, which made the world rethink about the DC transmission system. Despite AC transmission, it experiences high charging current of cable capacitance, high losses, absence of asynchronous operation, difficulty in control of power flow, the need for reactive power compensation and having issues of skin and Ferranti effects [23]. Because of these drawbacks HVDC system has increased its demand significantly for long distances power applications [55]-[57]

Initially, a current source converter (CSC)-based HVDC system was employed for power transmission. CSC-based HVDC systems employ thyristors and may achieve high power ratings while exhibiting minimal losses (usually around 0.7%). On the other hand, thyristors can only be switched on and have no turn-off capability. Therefore, they cannot be managed to stop a fault current. Furthermore, a CSC-based HVDC system requires a large filter, which raises the capital cost, and it is susceptible to AC side faults, which might result in commutation failure [58]-[60]. Given these concerns, voltage source converter (VSC)-based HVDC systems have been developed and given the opportunity to connect renewable sources to the Existing grid. Moreover, HVDC-based transmission lines enable the connectivity of generation from wind, solar, and tidal plants [64],[65] from far off-shore or distant sites, reducing transmission losses.

In spite of these, many advantages seen in VSC- based HVDC systems have a drawback towards the DC faults, which experience high losses (typically around 1.6%)

and lower power ratings [61]-[63] . The DC fault current has a high peak and constant value within a few milliseconds, necessitating high-speed fault detection and isolation systems in an HVDC grid. As a result, developing a protection plan for a multi-terminal VSC-based HVDC system is problematic. However, many methods for detecting DC defects have been proposed. As of today, the HVDC companies are looking forward to the protection of the electric grid with the supervision and maintenance procedure during the present critical power systems.

Currently, technology has been rocketing up towards big data and digitalization in the field of maintenance systems. Moreover, the electrical industry is looking forward to an innovative method using data analytics for supervision, which has a broader scope in power grid. There are multiple approaches under the shade of statistical analysis tools like machine learning, artificial intelligence and deep learning. Using these tools has profoundly affected predicting stocks, making decisions, and identifying objects are some of the areas these tools have shown their outstanding performances. By seeing the advantages of these tools, electric grid companies are also inheriting the use cases for their data from the data acquisition systems. Fault detection on the HVDC line help the TSO in detecting the problem before system failure occurs.

### 1.1 Background

Due to the advantages of long-distance and large-capacity power transmission, asynchronous interconnections, and the ability to prevent inadvertent loop flow in an interconnected ac system, the number of HVDC systems has been rapidly increasing because of their stability when compared to the AC grid. This stability is due to VSC HVDC.[54]

Since roughly 2014, when multi-level converter technology became widely used for HVDC projects, cumulative HVDC capacity has expanded significantly because of a regular stream of new VSC HVDC systems with capacities ranging from 700 MW to around 1 GW that have entered operation. However, new protection techniques have been discovered, and some fault techniques are the Impedance based method, voltage and current derivative-based method and handshaking method. Whenever a DC line fault occurs in the system, it is seen in the system that the voltage drops rapidly. In the meantime, the capacitor discharges the current through an inverter and feeds the fault. The AC side also supplies short-circuit current to the problem spot through the diodes after a few milliseconds. Consequently, when the current from the AC side passes through the diode, it becomes an uncontrolled rectifier which will affect the cable's insulation and later destroy the power electronic switches instantly.

During a DC line fault, it is impossible to isolate the multi-terminal HVDC system in its whole without selectivity by tripping all the AC circuit breakers. Else the converter will absorb the reactive power from the AC grid. However, the AC grid cannot supply the reactive power to the converter, which leads to a blackout situa-

tion.

To mitigate this situation, Hitachi energy devolved a product called MIMS. It stores the values of change of derivative voltage( $dv/dt$ ) and derivative of current ( $di/dt$ ) when the fault occurs in the HVDC system. On the other hand, because both the  $dv/dt$  approach and the  $di/dt$  method are dependent on the fault loop impedance, they are susceptible to fault resistance.

Therefore, there is a need to develop a new automation technique which identifies the DC fault with the data obtained from the MIMS from both stations by applying a machine learning technique to identify the pattern with the regression methods.

## 1.2 Objective

The aim of the thesis is to identify the faults in the HVDC line using the ML models used on the condition monitoring data.

- Predicting the trend of the signal using time series forecasting
- Comparing the accuracy of supervised ML algorithms for predicting the voltage, current, power and resistance from the sensor data.
- By seeing the trend and identifying the fault in advance.

## 1.3 Limitation

- The predicted model cannot predict the multiple outputs continuously.
- The project was implemented with a limited input (240 time-steps) because the model can experience a vanishing and exploding gradient. If the input increases continuously, there is a chance of losing data for the later steps.



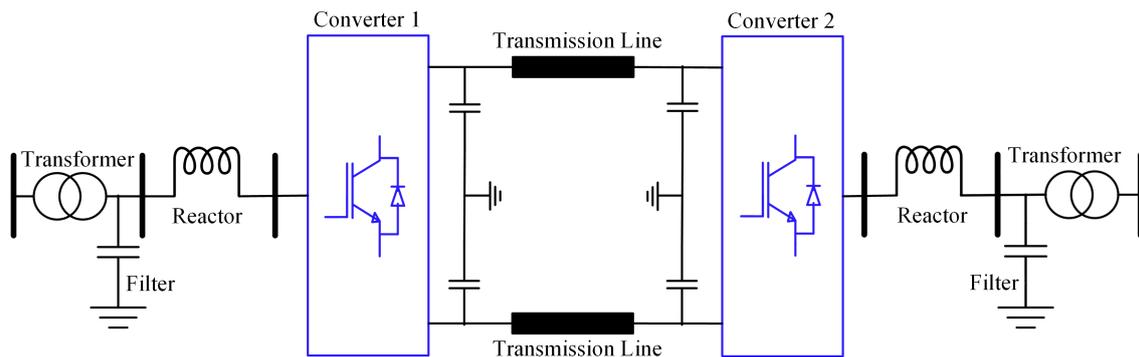
# 2

## Theory

In this chapter, a brief insight of HVDC systems, type of faults in a DC Line, faults identification methods, machine learning model and methods such as stationary test, LSTM model and model evaluations are presented.

### 2.1 Basics of HVDC system

The basic structure of the HVDC system shown in the Figure 2.1, which consist of AC side filter, Converter, transformer, DC side filters, reactors and DC cables.



**Figure 2.1:** A basic structure of HVDC system

Converters:

The converters are critical components of the HVDC transmission system. At the transmitting and receiving ends, it converts from AC to DC like a rectifier and from DC to AC like an inverter, respectively. Converters are linked to the alternating current system by transformers at both the transmitting and receiving ends. Current source converters with line commutated thyristor switches are employed in a simple HVDC converters system. A six-pulse valve bridge is utilized for both conversions in a simple HVDC system, i.e., rectification and inversion. Similarly, twelve pulse converter bridge circuits may be created by connecting two six-pulse bridge circuits.[12]

Transformer:

The transformers connect the alternating current transmission network to the six- or twelve-pulse valve bridge. The transformers change the alternating current voltage to a level appropriate for the converters. The power to be conveyed in the system

influence the design of these transformers.[12]

AC side Filter:

On the AC side, the converters of an HVDC transmission system generate harmonic currents. These harmonic currents enter the alternating current system, which affects the sensitive loads. AC filters on the AC side reduce these harmonic currents. The converters use reactive power from the HVDC system during the conversion process. This reactive power is partially corrected in the filter banks, with the remainder supplied by capacitor banks linked to the HVDC transmission system.[12]

DC filters:

The HVDC transmission system's converters cause a ripple in the DC voltage, affecting the communication between two stations by causing resonance with the ripples. So to avoid the concept of resonance, a DC filter is used in An HVDC system. However, DC filters are not required for pure cable transmission or back-to-back HVDC stations. Nevertheless, if overhead transmission lines are used in an HVDC system, DC filters need to be installed.[12]

DC cables:

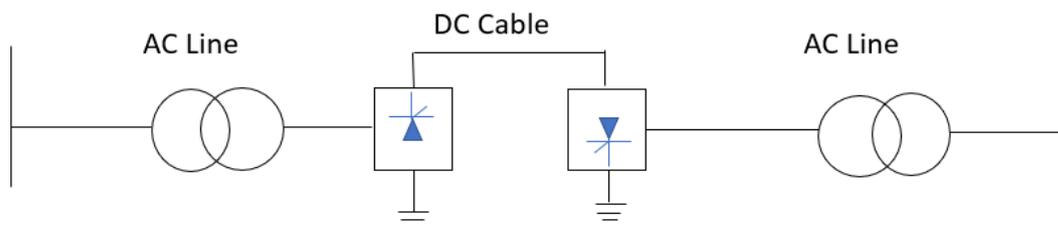
HVDC cables are often employed in underwater and subsurface transmission systems. There are no length restrictions for HVDC cable transmission systems. A back-to-back HVDC transmission system does not require DC cables or overhead wires. Cables are utilized for land connections due to environmental concerns.[12]

### 2.1.1 HVDC Schemes

HVDC systems are designed and selected based on the operation requirements, cost, and reliability issues. So for that reason, the HVDC systems are divided into three schemes depending on the flexibility of demand. The following HVDC configuration schemes are presented in [13]

Monopolar HVDC Scheme:

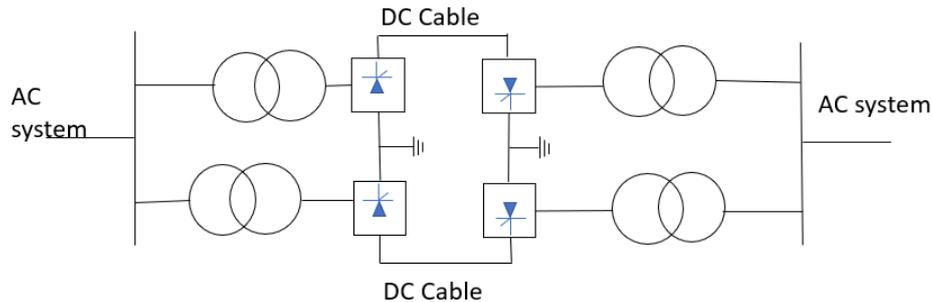
A single DC cable connects the two converters in this scheme. A positive or negative voltage is used for voltage transmission in a monopolar HVDC scheme [13]. The earth, sea, or grounding electrode can be utilized as a return path. Figure 2.2 depicts the monopolar HVDC system.



**Figure 2.2:** Monopolar HVDCsystem

Bipolar HVDC Scheme:

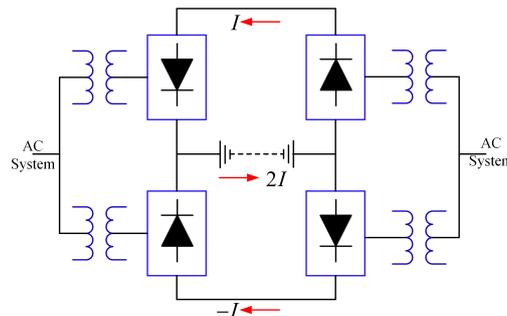
This scheme is a combination of two Monopolar HVDC systems. With this scheme, power transfers between two conductors of opposite polarity. The system's dependability improves because of this combination. However, by removing one of the system's poles, the other part continues to function normally, using the ground as a return channel. Figure 2.3 illustrates this scheme.



**Figure 2.3:** Bipolar HVDCsystem

Homo polar HVDC scheme:

Figure 2.4 is also known as the zero-distance transmission system. In this scheme, two converters are connected without a DC Cable [13]. Figure 2.4 shows this scheme.



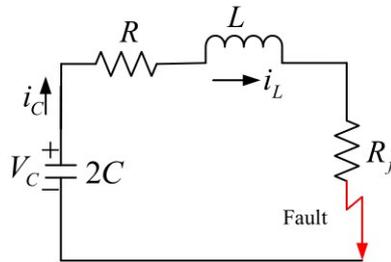
**Figure 2.4:** Homopolar HVDCsystem

## 2.2 DC faults

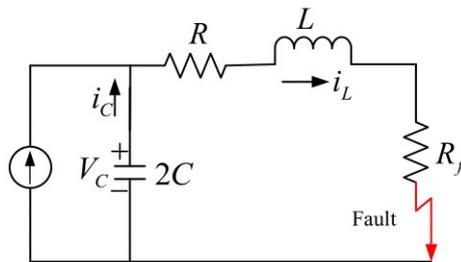
Faults may occur on the DC transmission line in an HVDC transmission system due to external mechanical stress, lightning, and pollution. Moreover, two types of faults appear on the DC transmission line: line to ground faults and pole to pole faults. However, these faults are permanent and take a long time to fix. Whenever there is a fault in the DC transmission line, the converter should be blocked immediately.

### 2.2.1 DC pole to pole short circuit:

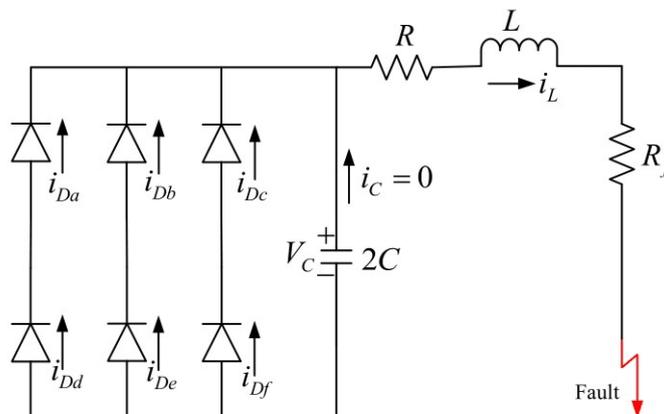
Pole to pole faults occurs due to the insulation breakdown of the positive conductor and the negative conductor of a DC transmission line. Moreover, this fault does not commonly occur in the DC transmission line. However, when this fault occurs in the line, the capacitor quickly discharges and simultaneously, the AC system will be short-circuited through the fault point. When a fault arises on the DC side, the IGBTs can be stopped for self-protection, leaving reverse diodes vulnerable to overcurrent [15]. The problem requires that both converters be disabled. Figure 2.9 depicts the equivalent circuit of a pole-to-pole failure. The DC short-circuit faults are classified into three stages, which are as follows.



**Figure 2.5:** Capacitor discharge stage



**Figure 2.6:** capacitor recharging state



**Figure 2.7:** Diode freewheeling Stage

Capacitor discharge stage:

When the pole-to-pole fault occurs, the DC link capacitor discharges the amount of charge into the transmission line forming an equivalent circuit with the inductor and resistance, as shown in Figure 2.5 5.

Diodes freewheel stage:

When the pole-to-pole fault occurs, the DC line voltage drops below the grid phase voltage, which results in a second stage called the freewheel stage. In this stage, the IGBT switches should be blocked for protection, and the AC side grid supplies the current through the fault by the diode shown in Figure 2.9.

Capacitor recharging state:

In this stage, the dc-link capacitor, cable inductance, and AC side produce a forced reaction during this step, and the capacitor is charged [14]. As a result, the DC voltage rises. The equivalent circuit shown in Figure 2.6.

### 2.2.2 DC line to ground faults:

The dielectric breakdown between a DC conductor and the ground can cause a short circuit. These problems are common and mainly caused by the grounding of the HVDC system. In this scenario, fault resistance significantly affects system responsiveness and should not be overlooked. The paper [16]-[19] presents analytical simulation studies on the effects of residual currents in phase-to-ground faults from various sources such as DC capacitors, reactors, and AC grids. As shown in the figure 2.8, phase-to-ground faults can be evaluated in three phases: capacitor discharge, grid current feeding, and voltage recovery.

DC side capacitor-discharge:

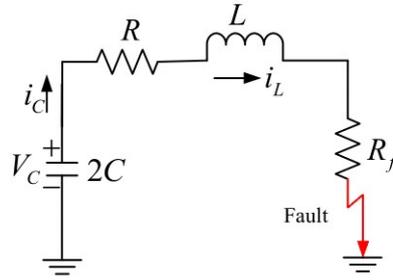
The DC line ground faults generate an equivalent circuit among the fault line capacitor and fault impedance, as shown in Figure 2.8. When a fault occurs in the system, the DC capacitor starts to discharge the amount of charge in the circuit.[14]

Grid- Side current feeding Stage:

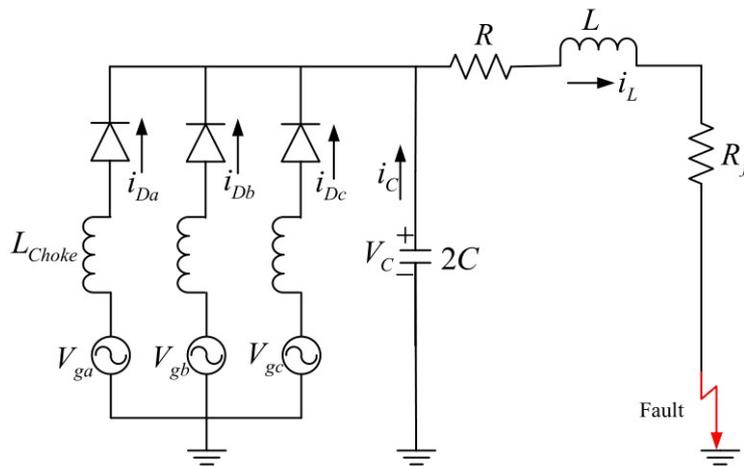
When the line to ground fault occurs in the line, the DC side capacitors start to discharge because the DC voltage drops drastically below the phase voltage. In that case, the system experiences a current feeding from the grid.[14] The equivalent circuit for the grid- side feeding stage shown in figure 2.9.

Voltage recovery Stage:

During discharge, the voltage of the fault pole capacitor falls while that of the non-fault line capacitor rises. The DC voltage returns to a normal level, and the system enters the voltage recovery phase.[14]



**Figure 2.8:** DCline to ground Capacitor discharge stage



**Figure 2.9:** Grid current feeding stage

## 2.3 DC fault identification method in HVDC:

### 2.3.1 Handshaking method

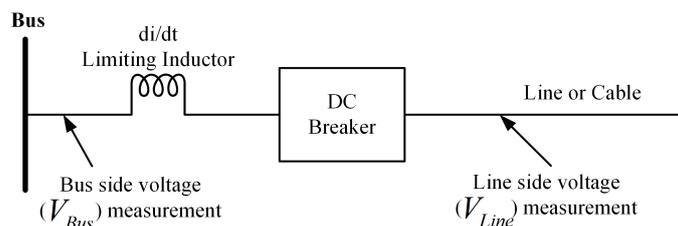
The primary support for a persistent DC fault is terminal sources. However, disconnecting all electrical sources and implementing the protection strategy during the fault in the DC line would protect the voltage source converter.

The Handshaking method comprises an AC circuit breaker with a DC Switch despite the DC circuit breaker. Besides, the AC circuit breaker with the DC switch is cheaper than a DC circuit breaker.

If the DC line fails, the AC circuit breaker opens the line from the AC side, and the DC switch isolates the faulty line by providing the voltage and current value threshold limit. When the current and voltage values are above the threshold, the line has no fault.

### 2.3.2 Voltage and current derivatives based method

The reference paper [20] proposes a voltage derivative ( $dv/dt$ ) and current derivative ( $di/dt$ ) based protection method for the MTDC system. If the rate of DC voltage and current change exceeds the specified threshold, the fault is detected. The work in paper [21] describes fault detection on an earthed HVDC grid using the rate of change of voltage (ROCOV). The DC inductor is connected in series with the DC breaker to keep the peak fault current below the current level rating of the breaker. Figure 2.10 depicts the suggested fault detection approach, using the ROCOV and a current-limiting inductor.  $V_{bus}$  is the bus terminal voltage,  $V_L$  is the voltage across the inductor, and  $V_{line}$  is the transmission line voltage.



**Figure 2.10:** Voltage and current derivatives based method

The inductor voltage is given by

$$V_{L(t)} = L \frac{di(t)}{dt} \quad (2.1)$$

By reference of figure 2.10 the voltage of the bus terminal is given by

$$V_{bus(t)} = V_{L(t)} + V_{line(t)} \quad (2.2)$$

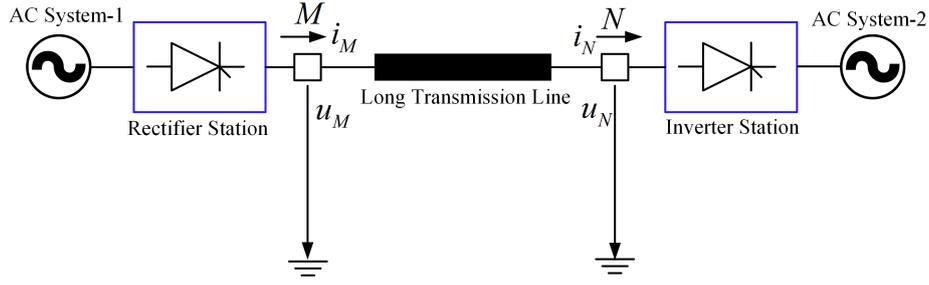
The rate of change of current is obtained from the equation 2.1 and 2.2

$$\frac{di(t)}{dt} = \frac{V_{bus(t)} - V_{L(t)}}{L} \quad (2.3)$$

From equation 2.3,  $L$  is the inductance value. When the fault occurs in a line, the voltage across the voltage bus is constant, and the rate of change of the current at the breaker is mainly dependent on the line voltage. Therefore, the fault location can be achieved by measuring the ROCOV on the line side of the inductor.

### 2.3.3 Transient based method for DC fault detection

According to a reference paper [22], the transient-based method uses the difference in transient energy between the rectifier and inverter end to identify and differentiate faults in an HVDC line. The two terminal HVDC system consists a pair of AC systems with a rectifier station, long transmission line and inverter station shown in figure 2.11. Based on figure 2.11, a relay is placed on the rectifier (M) and inverter (N) sides of the circuit. Using the DC voltage calculates the difference in transient energy, which is currently recorded on both sides. The equations for transmission lines and a distributed parameter model are used to calculate.



**Figure 2.11:** Transient based method for DC fault detection

The equation for the transient energy is provided below:

$$\Delta E_M = \Delta u_M \Delta i_M \Delta t \quad (2.4)$$

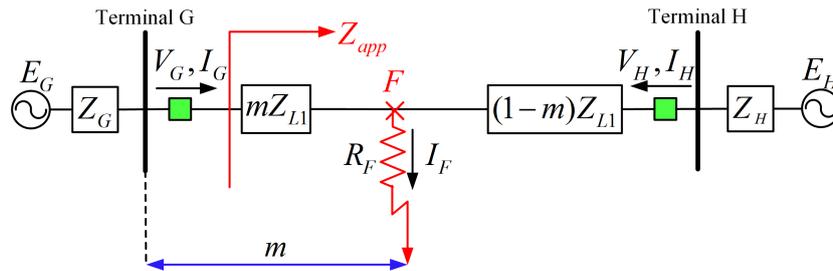
$$\Delta E_N = \Delta u_N \Delta i_N \Delta t \quad (2.5)$$

$$\Delta E = \Delta E_M - \Delta E_N \quad (2.6)$$

From equation 2.5, 2.6 the terms  $\Delta u_M$ ,  $\Delta u_N$ ,  $\Delta i_M$ ,  $\Delta i_N$  are the variation of the DC voltage and current at rectifier and inverter side. Moreover, the continuous period from  $t_1$  to  $t_2$  is denoted by  $\Delta t$ . Under normal conditions, the transient energy ( $\Delta E$ ) fluctuation is zero, but it has a specific value if a fault occurs in the HVDC system. An internal fault occurs when  $|\Delta E| > \Delta E_{set}$  and  $\Delta E < 0$ , but an external fault occurs when  $|\Delta E| > \Delta E_{set}$  and  $\Delta E > 0$ , where  $\Delta E_{set}$  is the threshold value for transient energy change.[23]

### 2.3.4 Impedance based method

From figure 2.12, it is possible to understand the impedance-based method's principle with some assumptions to be considered. The transmission line is homogenous and has the entire positive sequence impedance  $Z_{L1}$  between the two terminals, G and H, illustrated in figure 2.12. The network between the G and H is depicted by their Thevenin analogues having impedances  $Z_G$  and  $Z_H$ . Moreover, when a fault occurs at a distance of  $m$  per unit from terminal G, both sources contribute to the total fault current  $I_F$ .  $V_G$  and  $I_G$  are the voltage and current phasors recorded at terminal G during the fault. Likewise, the voltage and current phasors measured at the terminal during the fault,  $V_H$  and  $I_H$ , represent terminal H. Although measurements are accessible at both ends of the line, one-ended techniques use the voltage and the current acquired at terminal G or terminal H.



**Figure 2.12:** Impedance based method

The voltage drop from terminal G may be described using Kirchhoff's rules as

$$V_G = mZ_{L1}I_G + R_f I_F \quad (2.7)$$

Equation 2.7 states the voltage and the current depending on the fault type in the line. The apparent impedance to the fault ( $Z_{app}$ ) measured at terminal G may be represented by dividing it by  $I_G$ . The equation for the fault impedance is

$$Z_{app} = \frac{V_G}{I_G} = mZ_{L1} + R_f \frac{I_F}{I_G} \quad (2.8)$$

Equation 2.8 serves as the general governing equation in one ended impedance fault identification

## 2.4 Basics of Machine learning

ML is a superset of Artificial Intelligence (AI) devoted to creating computer programs that can learn independently and define the rules that govern this learning process. ML models utilize algorithms to swift through a sizeable input-output database of problems to predict, classify, etc. Machine learning applications have become prominent in robotics, medicine, image classification, computer vision, speech recognition, etc. Machine learning algorithms are further classified by how well they predict accurately. ML has broadly classified as supervised learning, unsupervised learning and reinforcement learning.

### Supervised learning:

A supervised learning algorithm consists of the input features and the output labels or target variables. It tries to learn and map the relation with the input and output labels. This training procedure would help to get the desired output by inputting the labelled data. Moreover, to get a better knowledge of supervised learning, let us consider an example of visual recognition of self-driving cars, which has become an exciting topic in the modern era by training the machine learning model with labelled videos of pedestrians, cyclists and cars. After obtaining the labelled video data, supervised learning algorithms are trained on the data set to get a relationship between the input data and the labelled data. Each algorithm identifies the labelled object in fast-moving scenes. Then the test is done on the different video data to obtain the algorithm's accuracy.

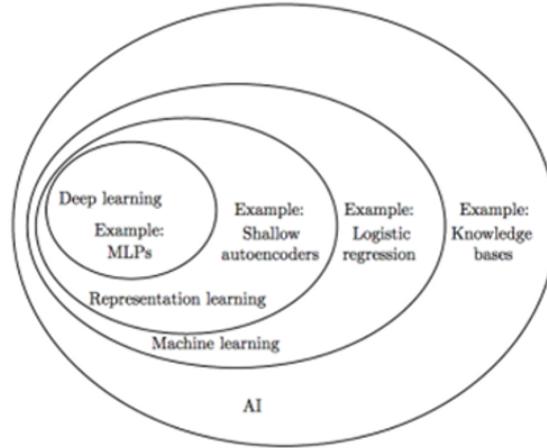
### Unsupervised learning:

Unsupervised learning is a contrast to supervised learning. There would be no pre-labeled output data in unsupervised learning, which means that no target variable is present in the data. The model analyses the pattern in the input data and forms the clusters regarding their functionality without any human intervention. In this case, let us consider an example having a dataset of random images of cats or dogs. The Unsupervised Learning algorithm seeks to learn patterns within the data and cluster inputs with similar "features". In this case, the feature may be recognisable as a distinguishing feature of dogs or cats. However, it is often not identifiable or

explainable.

**Reinforcement learning:**

Agents interact with the environment by performing actions in reinforcement learning. It learns by errors and rewards which was given by the user for every iteration.



**Figure 2.13:** Venn diagram showing machine learning, AI, Deep Learning etc., correlate[24]

Considering the advantages of ML in different fields have become more popular in the power grid application by replacing the conventional statistical approach. Because of the increased availability of large-scale data storage and computational power has allowed ML algorithms to perform mathematical calculations fast and efficiently. This has opened a new trend in power grid application for monitoring, prediction and fault detection to improve the power quality of the grid. Innumerable measuring devices and sensors used in the data acquisition system can measure a continuous stream of data from the sensors. Main goal of analyzing the condition monitoring data is detecting, extracting, and analyzing the fault of the DC line. The fault of the DC line data provides information about the fault and the root cause. Thus, analyzing the fault data would help reduce the probability of the fault in the transmission line before it occurs in the future.

Moreover, numerous papers identify fault detection in motor drives and transformers but not in the transmission line. By taking the inspiration from the research papers [33],[35], the following architecture was designed for this case.

## 2.5 Stationary test

The KPSS test describes the null hypothesis that an observable time series is stationary around the deterministic trend by following a linear regression model with an intercept concept shown in equation 2.9.

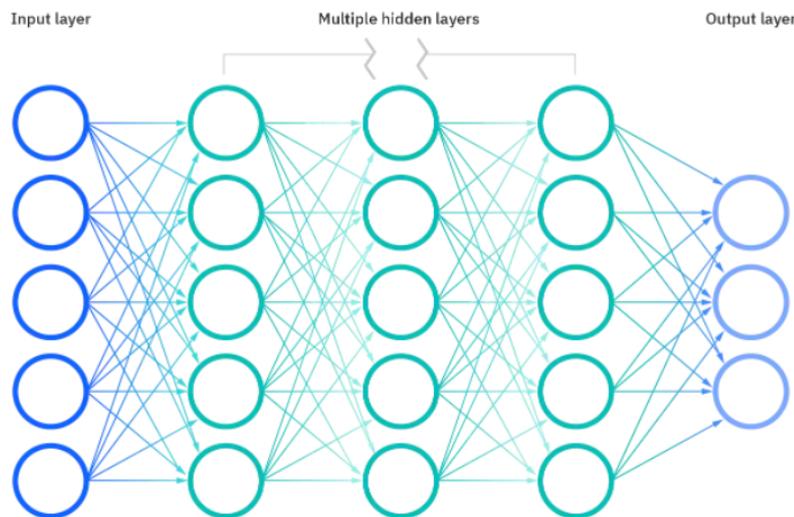
$$X_t = r_t + \beta t + \epsilon_t \quad (2.9)$$

Where equation 2.9 has three terms  $r_t$  is a random walk,  $\beta t$  is a deterministic trend and  $\epsilon_t$  is a stationary error which tells us about the stationary of the time series around a fixed level. While performing this test uses the ordinary least square regression to identify whether the time series is stationary or trend stationarity around the constant mean with a standard deviation of 0. However, most time-series data do not follow the stationary in order to make it stationary there is a method called differencing the time series. By following the equation 2.10 will help to make the time series constant around the mean and make the standard deviation 0.

$$d^{(1)}(t) = X(t) - X(t - 1) \quad (2.10)$$

## 2.6 Neural Network

Neural networks, also known as artificial neural networks (ANNs), are a subset of machine learning foundation deep learning techniques. The human brain inspires its name and form by replicating how real neurons communicate with one another. Artificial Neural Networks (ANNs) consist of nodes containing one or more hidden layers and an output layer. Each node, or artificial neuron, connects to another and has an associated weight and threshold[25]. Suppose the output of any individual node is above the specified threshold value. The node is activated with a function of tanh and sends the data to the next layer of the network. Otherwise, the data is not passed to the network.



**Figure 2.14:** Neural network schematic

### 2.6.1 Recurrent neural network

RNNs are a type of neural network that can remember, making them more comparable to how humans absorb information and providing an effective solution to various scientific issues. Data is processed individually in a loop of neural network cells shown in figure2. Moreover, RNN is a dynamic system of learning sequential

pattern dependencies over time. As a result, it has been used extensively in time series analysis [37]-[39]. However, the RNN has an issue with a long term dependency variable where the neural network will face an issue with vanishing gradients and gradients blown up in training an RNN. So, as a result, to the reference paper [38] has come up with an Idea of LSTM with long storage memory.

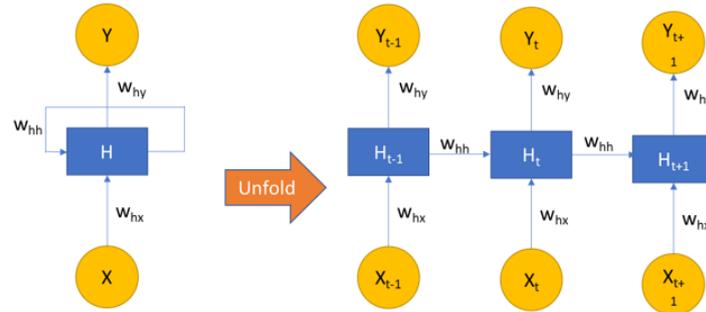


Figure 2.15: Structure of RNN

### 2.6.2 Long short temporary memory

LSTM is also a recurrent neural network that overcomes the long-term dependencies by considering memory cells: the input gate, the output gate, and the forget gate. The memory cells store the previous value, and the gates are responsible for controlling the amount of memory to be exposed.[26]

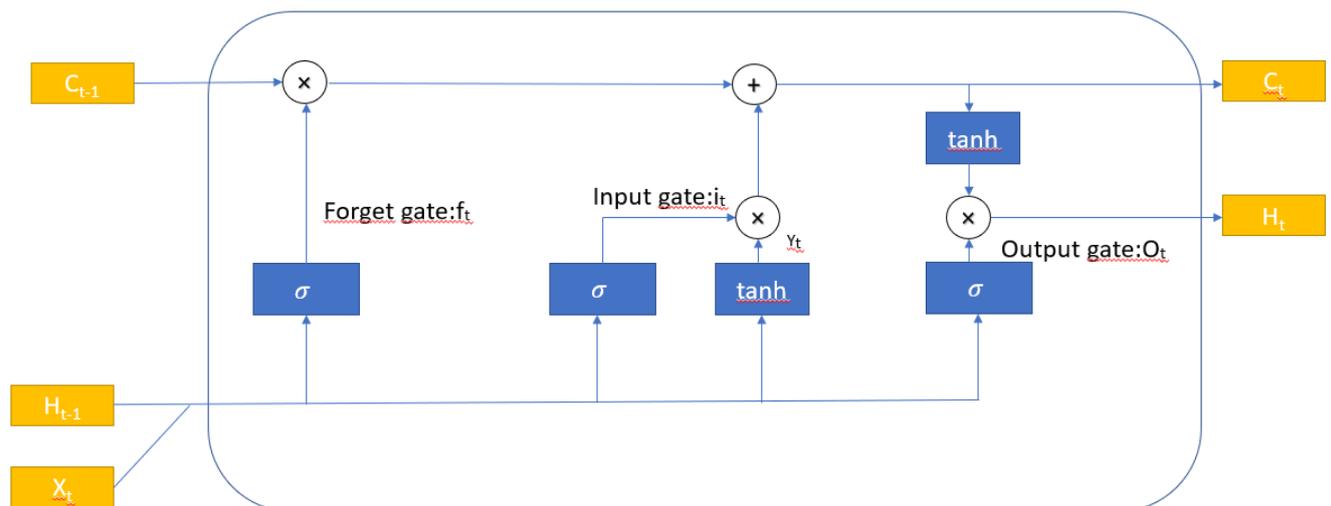


Figure 2.16: Structure of LSTM

#### Input gate:

The function of this gate is to control the input information from  $H_{t-1}$ , and  $X_t$  should be given to the cell shown in figure 4, where  $H_{t-1}$ , and  $X_t$  are the input

vectors taken into the sigmoid function, which gives an output value of 0 and 1. The output value 0 states that the information is not let out from the gate and the output value 1 states that the information is let out from the gate. In the same way, the input vectors passed into the tanh activation function prioritize the input vectors in the form of weights to the values ranging from -1 to 1.

$$i_t = \sigma(\omega_i[H_{t-1}, X_t] + b_i) \quad (2.11)$$

$$Y_t = \tanh(\omega_c[H_{t-1}, X_t] + b_c) \quad (2.12)$$

### Forget gate:

Depending on the sigmoid function, the gate's functionality is to remember the past value or forget the past value by taking the input from the  $H_{t-1}$ , and  $X_t$  shown in figure 3. After the inputs are taken into the sigmoid function, the function outputs 0 and 1, stating that 0 means forget the data and one means remember the past values.

$$f_t = \sigma(\omega_f[H_{t-1}, X_t] + b_f) \quad (2.13)$$

### Output gate:

The functionality of this gate is to give out the future values as the output by taking the values from the input gate, and the memory gate decides the output. But the sigmoid function decides the output of the gate. Whether to activate the gate or not to activate the gate depends on the input vector, which provides output values 0 and 1. If the sigmoid provides an output value of 1, then the output from the sigmoid function is multiplied by the output function of the tanh and provides the output. Else the output gate will not provide the output.

$$O_t = \sigma(\omega_o[H_{t-1}, X_t] + b_o) \quad (2.14)$$

$$H_t = O_t \times \tanh(y_t) \quad (2.15)$$

## 2.7 Model Evaluation

Model evaluation plays a vital role in the performance of the ML model by comparing the accuracy of the predicted value (From the model) to the actual values (sensor data). The evaluation metric used in this thesis is as follows:

### Mean absolute error:

The mean absolute error is the average difference between the observations (actual values) and model output (predictions) [27]. The sign of these discrepancies is disregarded, preventing cancellations between positive and negative numbers. If we didn't disregard the sign, the computed MAE would be far smaller than the genuine difference between model and data.

$$MAE = \frac{\sum_{i=1}^n |Y_i - Y_{pred}|}{n} \quad (2.16)$$

where  $Y_i$  is the actual output,  $Y_{pred}$  is the model predicted output and  $n$  are the total number of samples

### Mean Squared error:

The degree of inaccuracy in statistical models is measured by mean squared error (MSE). The average squared difference between the observed and expected values is calculated. When there is no error in a model, the MSE is 0. As model inaccuracy rises, so does this value.

$$MSE = \frac{\sum_{i=1}^n (Y_i - Y_{pred})^2}{n} \quad (2.17)$$

### Relative absolute error:

The Relative Absolute Error is a ratio that compares a mean error (residual) to errors produced by a simplistic or naive model. This method is used to calculate the model's performance and accuracy.

$$RAE = 100 \times \frac{\sum_{i=1}^n |Y_i - Y_{pred}|}{\sum_{i=1}^n |Y_i - Y_{mean}|} \quad (2.18)$$

### Root relative squared error:

The Root Relative Squared Error (RRSE) is the square root of a predictive model's sum of squared errors normalized by the sum of squared errors of a simple model.

$$RRSE = 100 \times \sqrt{\frac{\sum_{i=1}^n (Y_i - Y_{pred})^2}{\sum_{i=1}^n (Y_i - Y_{mean})^2}} \quad (2.19)$$

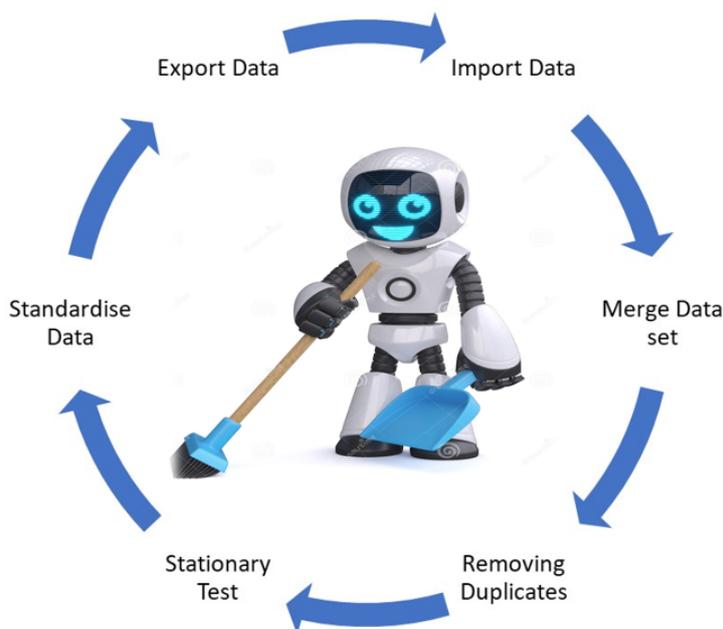
# 3

## Methods

In this chapter, the work flow for building fault model is explained. This comprises of the introduction to data cleaning, Data generation, Merging Data, Feature selection, Preprocessing, LSTM Architecture and the optimal values of the hyperparameters.

### 3.1 Cleaning Data

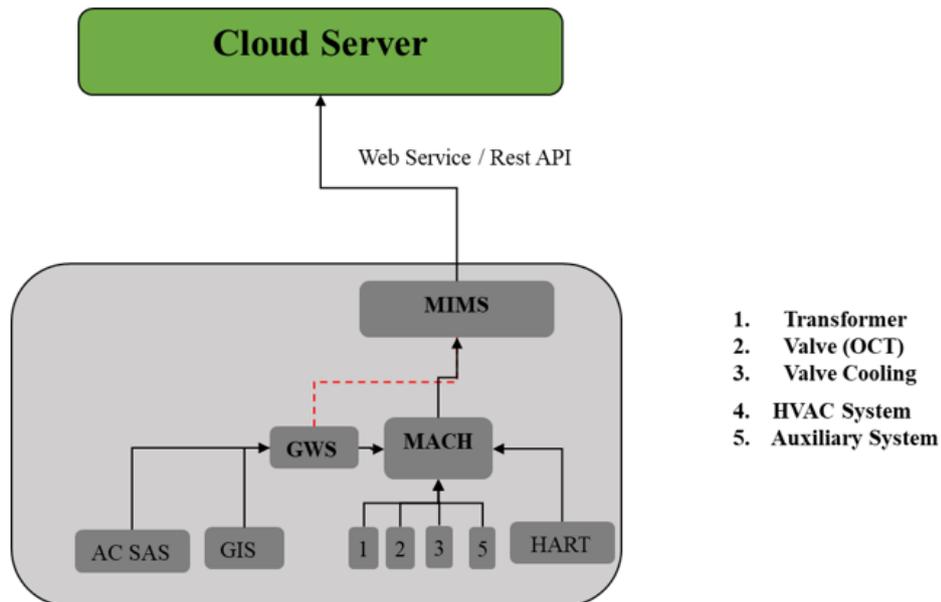
Dean Abbot, Smarter HQ's co-founder and Chief Data Scientist, believes that "no data is pure, but most are relevant." The first and most critical step is to cleanse the data before commencing any further research to get insights into the data. It is the process of recognising data from various files/folders and changing or updating it according to the requirements[1]. By doing this procedure, the data has not been tampered with but extracted the pertinent data from the cloud storage and combined for the future study. As the ML engineer, the issue with the data is to understand and comprehend the data before deciding what data is significant and required to provide helpful insights.



**Figure 3.1:** The structure of Data cleaning

### 3.1.1 Data generation

The data in this thesis is in the form of a time series. Each number represents a readout from the HVDC system sensors at a specific point in time. The data comes from the *MACH<sup>TM</sup>* system (Hitachi Energy’s internal system), which is the brain of the HVDC system. The telemetry data comes from the sensors of transformers, valve, valve cooling, HVAC system and auxiliary systems values are generated and connected through the MACH 3 input-output boards. The operation data (conditional monitoring data) communicated to the MIMS system through the ETDM protocol from the Input Outboard. Moreover, the operation data reaches the MIMS system, converts the data into the text format and sends the text format data into the cloud storage device by the web services. As for a clear understanding, refer to figure 3.2.



**Figure 3.2:** Data generation methodology

### 3.1.2 Data merging using Python

Python is one of the most popular computer programming languages [52]. It provides many libraries that provide excellent functionality for handling mathematics, statistics and scientific operations. Moreover, in the present era, the python libraries have significantly impacted the field of Data sciences and ML with the numerous amount of open-source libraries that have created an enormous impact on the manipulation of the data with the pandas library.

However, due to more data available from different sources, the pandas library can combine all the Data frames but cannot navigate to the different paths. So python has an open library called glob. It can navigate and provide information about the data frame available in the different paths by providing the exact path of the data. Moreover, the discussed python libraries are used in this thesis.

Where the data is stored, the cloud has a folder named Station 1. Under station 1, there are several subfolders, and each subfolder contains the time series data CSV file for each HVDC control and protection system. To access this folder in python, one needs to install the package of OS. Using the function 'Os.listdir.join' to get all the directories and files in a single list by providing the directory of the station 1 folder. After reading the path directory, the subfolders are converted into the list and stored in variable a. The next step is to install the glob library, which helps to say whether the CSV files are present in each subfolder or not. Upon completion of the glob process, each CSV file is read into the pandas' data frame and concatenated each time series CSV into one CSV file. In the end, the CSV files merged into a single time-series CSV file. However, the sampling frequency of the time is not equal. To make the sampling frequency equal. The sampling frequency has been resampled by 30 seconds. This is done to increase the data points available in the data set. The other sampling frequency was also tried, but they intended to reduce the data points, which will affect the model accuracy and also lead to the concept of overfitting. Now, the sampling frequency between each time is unique, making it easy to create a sliding window algorithm, a technique used in fault prognosis analysis in the later section of chapter 3.5.

Data merging with python using the above functions has provided with a data of 60517 rows and 1517 columns sent to the feature selection, This is explained in the following sections.

## 3.2 Feature selection

Feature selection plays a prominent role in the machine learning algorithm. It changes from problem to problem as it deals with an issue of fault prognosis in an HVDC line. It was decided to pick the features of voltage, current and power. As a result of considering these features, resistance is calculated and added as a feature to the data set. Due to this, DC lines are more susceptible to faults that occur with changes in voltage and current.

Therefore, by using the process, the data is reduced to 60517 rows and 4 columns. The scaled down data is sent to the preprocessing step.

## 3.3 Preprocessing

Data preparation is the process of converting raw data into a machine-readable format. In many real-world scenarios, the data obtained is either incomplete, prone to mistakes, includes categorical data, separate columns will have different scales as

described [3]. The steps for the data preprocessing are shown below.

#### 3.3.1 Missing Values

Missing values are common in data obtained from real-world applications. This might be due to information loss, data corruption, or a failure to capture data. Because many machine learning algorithms fail to evaluate data with missing values, it is critical that the missing values be addressed before proceeding with the analysis. In our situation, more than 2% of the dataset contained missing values, which we completely disregard for further research. There are several approaches to dealing with missing data [2]. The methods used for filling the missing values with mean mode median, filling the missing values with forwarding fill, deleting rows with the missing values, filling the missing values with interpolate and filling the missing value with the replace method are the methods used to remove the missing value in the data frame. Among of these methods Filling the missing values with forwarding fill, Deleting rows with the missing values are out performed and the rest did not perform well. Because the rest of the methods fills the missing values by taking the mean, median and mode of the data. In the below figure 3.3 tells us about the missing values.

Sno	Volatage	Current	Power
1	400	1200	480000
2	nan	nan	nan
3	410	1250	512500
4	320	1200	362400

Figure 3.3: Missing value Data set

##### 3.3.1.1 Filling the missing values with forwarding fill

Deleting the rows for the nan values will give good results when the nan values are less than 5%. But on the other hand, if the nan values is more than 5% deleting the rows will not be a good idea. Either the missing values are to be forward filled by the previous value will be a good idea for larger datasets.

As a result of utilizing this method, the data is filled with previous values, but all the nan values did not eradicate from the data set. So it was decided to delete the rows and the columns that are there in the data set.

##### 3.3.1.2 Deleting rows with the missing values

One solution to the missing value problem is to delete the whole row of missing values in one or more columns. It is only successful if the percentage of missing values is low, about 5%. This method works better and trains the machine learning model with robust results. Therefore, after using the procedure, the data is reduced to 57367 rows and 4 columns.

### 3.3.2 Removing the outliers and smoothing

The data obtained from the sensor will consist of outliers and some noises in the system, which affect any machine learning model that you train. To mitigate the problem of outliers and the noise in the signal, python has a signal library where you can use all types of smoothing functions such as low pass filter, high pass filter, and a bandpass filter to remove the outliers and smooth your signal. As employed sampling frequency is 0.033 Hz, it was decided to use a low pass filter from the signal library. In this regard, a low pass filter is used in this thesis. Figure 3.4 and Figure 3.5 shows the difference between a smoothed and an unsmoothed graph.

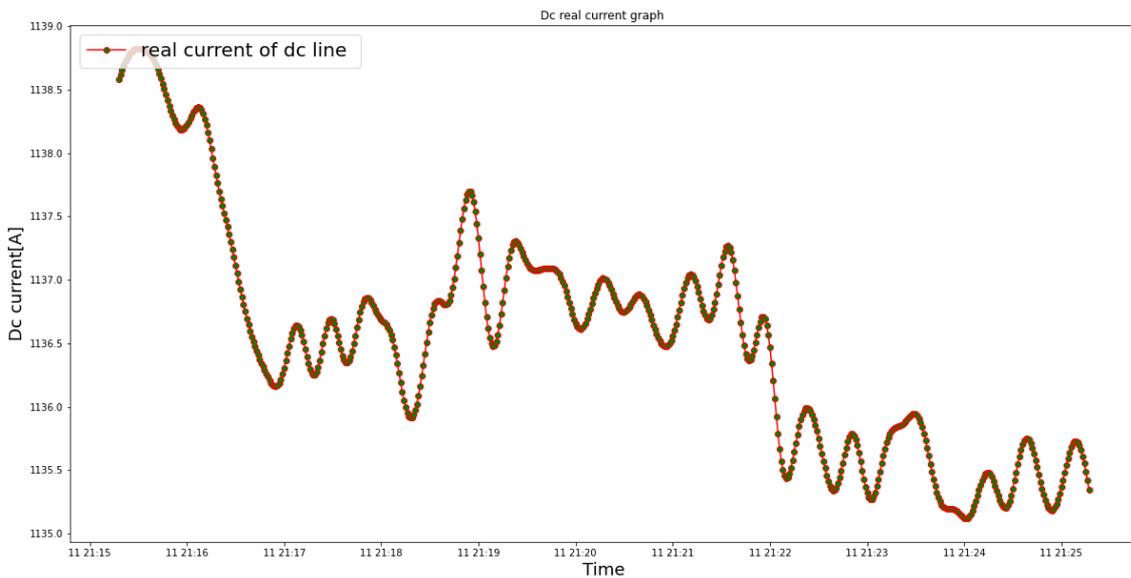


Figure 3.4: smoothed graph using low pass filter

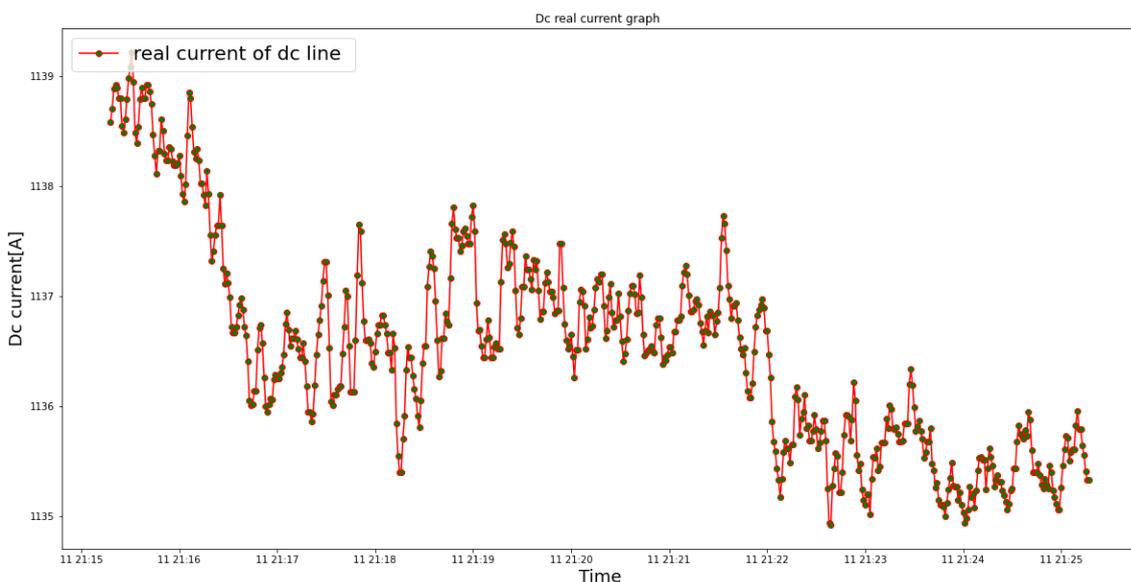
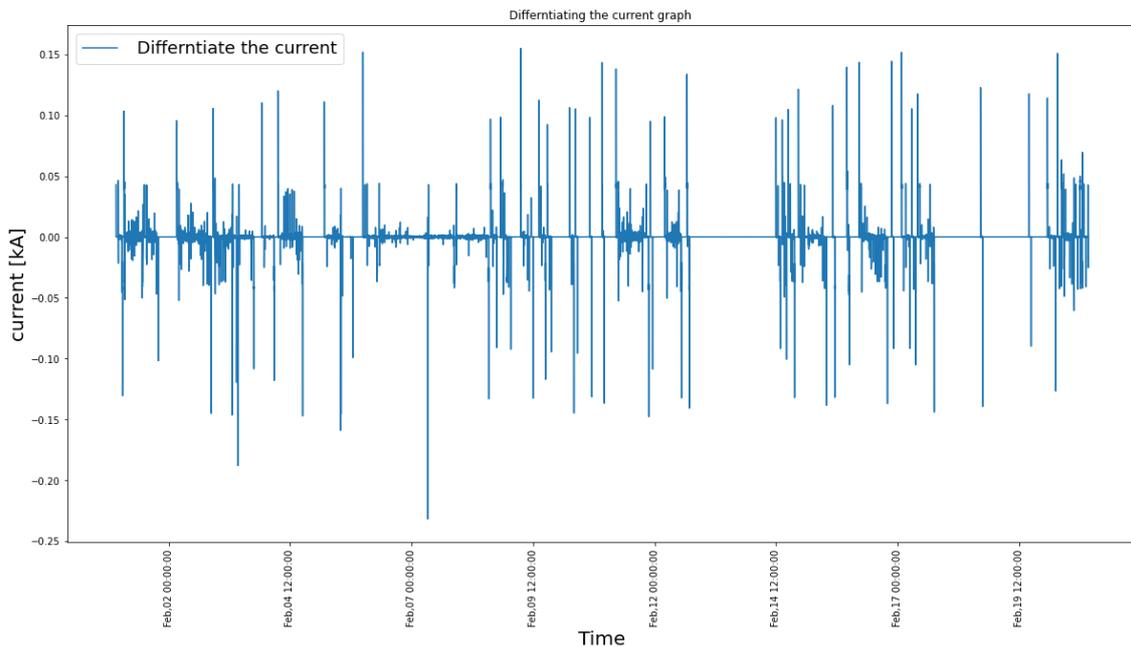


Figure 3.5: Unsmoothed graph before low pass filtering

### 3.3.3 Stationary test

KPSS [53] determines whether or not the time series data are trending and stationary. Therefore, this test performs a regression analysis around the time-series data. The regression analysis enhances the trend and tells us about the stationarity of the signal of the time-series data.

According to this test, the time series data that is used in this thesis is not stationary from figure 3.4. With the reference to the figure 3.4 it is evident that the data points do not have a constant mean, and the standard deviation is not equal to zero. As a result, to make the time series data stationary, there is a method called differentiating the time series data, which is discussed in 3.3.4



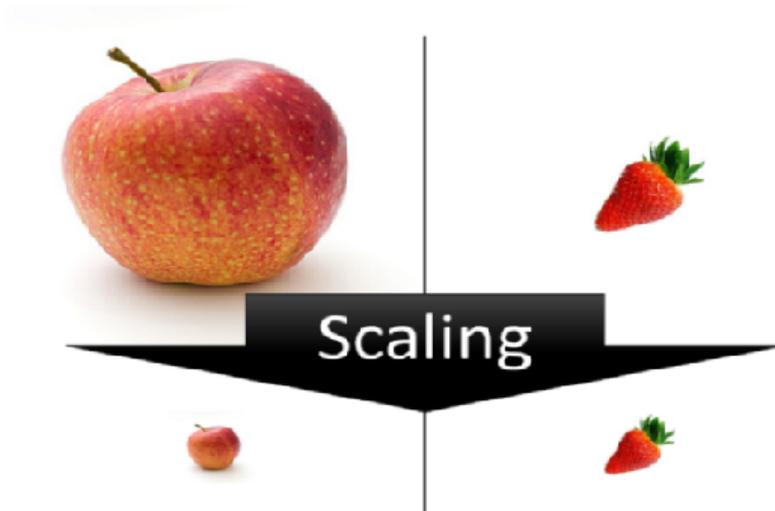
**Figure 3.6:** Differentiating the time series signal

### 3.3.4 Differentiating the time series data

Differentiating the time series data is an important concept by which you can make a non-stationary time series signal into a stationary time series signal by removing the series dependence on time and stabilizing the mean of the time series by eliminating the trend seasonality, which is shown in the Figure 3.6.

### 3.3.5 Feature Scaling

Feature scaling is a prominent methodology when dealing with a machine learning algorithm. It is a strategy for standardizing the independent variables or features within a specific range. Moreover, it helps to reduce the values but also increases the speed of calculation of the algorithm. The most common technique used for feature scaling is discussed below.



**Figure 3.7:** Feature scaling diagram as described in [4]

### 3.3.5.1 Normalization

Normalization is also called as min max scalers where the range of features is rescaled to  $[0,1]$ . The formula of the min max scaler is given by

$$X_{normalization} = \frac{X - \max(X)}{\max(X) - \min(X)} \quad (3.1)$$

The above equation 1 describes the  $\max(X)$  as the maximum value of the feature, and  $\min(X)$  is the minimum value of the feature. However, it depends on the application. The range can potentially be something other than  $[0,1]$ . Whenever the data does not have a Gaussian distribution, normalization is the preferred method of feature scaling. Refer to table 1 for a clear understanding of the maximum value and the minimum value of the signal.

As a result this method is not used in our application because of providing a large error in the model prediction. Instead it was decided to choose the method standardization which is explained in the section 3.3.5.2.

**Table 3.1:** Maximum and Minimum Values for Current, Voltage and Power

Maximum current	1.5172kA
Minimum current	0A
Maximum voltage	443.8294kV
Minimum voltage	0V
Maximum power	605.3441MW
Minimum power	0W

### 3.3.5.2 Standardization

Standardization is putting the diverse variables on the same scale in statistics. This method is used to have the zero mean and unit variance to transform the data into

machine-readable values by calculating the mean and Standardization of the entire feature. The formula for determining each value of the data point in a feature after calculating the mean and standard deviation is,

$$X_{standardization} = \frac{X - \bar{X}}{\sigma} \quad (3.2)$$

$\bar{X}$  is the mean of the feature, and  $\sigma$  is the standard deviation of the feature. Standardization is usually favored when the data has a Gaussian distribution, but it is not always required. Even if there are outliers in the data, standardization will not influence them. In addition, standardization is the best feature scaling technique used in time series analysis because of its sequence of data where it has a significant variance in the series obtained from the sensors.

However, considered all the feature scaling techniques, standardization has outperformed with a lower training loss than the normalization technique [50],[51]. In this thesis, the standardization technique on the entire data set using equation 3.2. Moreover, the coefficients used to rescale the value of the original values after training the model.

### 3.4 Train and Test

Standardization data should be separated into the train set, the validation set, and a test set before the sliding window procedure. Because when you don't split the data into train and test sets. The future values that you want to predict will be stored in the training set and affect the model. So the train split procedure plays a prominent role in the model evaluation. The example of train split procedure is shown in Figure 3.8

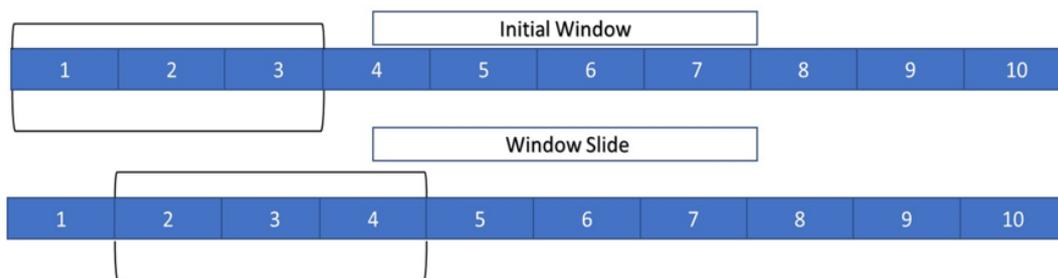


**Figure 3.8:** Example of Train Split procedure

The time-series data cannot be split randomly as other machine learning algorithms because of the sequential way that data obtained from the sensors will lose their sequence and lead to an unbalance in the future prediction. So to avoid that situation, data set is divided by taking the length of the data set. Since the data set used in this thesis has 57636 rows, 60% of them are divided into training data, and the remaining 40% are split into test data and validation data.

### 3.5 Sliding Window

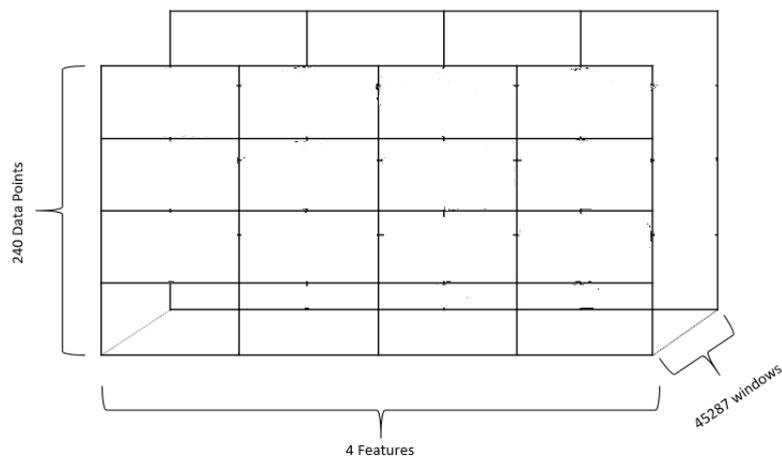
The sliding window algorithm has become prominent in time series applications such as medical, weather, and finance [47]-[49]. This methodology is applied to the fault prognosis of the HVDC line. The sliding window algorithm is a temporal approximation of the actual value of the time-series data [8]. As the window size and the segment size increase [9], the selection of the first segment and the next segment is selected from the end of the first segment [8]. This process continues until all-time series data have been segmented. The process is explained in Figure 3.9 with an example of 10 numbers with a window size of 3. The sliding window accumulates the historical time series data [10] to predict the succeeding 1-minute current, voltage, power, and resistance of the DC line. Figure 3.9 explains the process with window size 3 [11]. Each number in figure 3.9 tells about the time series data obtained for every minute. Once the window size is set, the window starts to consume 3 minutes of the historical data to predict the next minute's current, voltage resistance and power. After consuming the time series data for the first window, the window begins to slide towards the right for one minute to cover the next 3 minutes of the data. Because the step size was set to 1, this process continues until the end of the time series data.



**Figure 3.9:** Sliding window process

The experiment was done on the voltage, current, power and resistance on the training data set with a window size of 240. It stores the past data of the sequence in windows and divides it into a three-dimensional array (Windows×Data points in a Window×number of features). As soon as it is placed in the sliding window, the train data of size(46107,4) appears in a three-dimensional array/ matrix (45287,240,4) as show in figure 3.10.

The figure 3.10 describes about the data representation when the sliding window algorithm technique is applied on the sensor data. As a result the X axis of the figure 3.10 mark out the features of data such as resistance, voltage, power and current. While the Y axis of reports data points obtained from the window size and the Z axis tells us about the windows that are obtained by the sliding window algorithm.



**Figure 3.10:** Structure of the three-Dimensional matrix

## 3.6 LSTM architecture

In this section, the process of building an LSTM network starts from the Input, layers, optimal layers selections and the optimal hyperparameters for training the network discussed below.

### 3.6.1 Input Data

The data prepared from the sliding window section is fed directly to the LSTM input layer as the input to the network. The network takes the input as a 3-dimensional array in the shape of  $(W \times L \times F)$ , where  $W$  is the number of windows or data points,  $L$  is the length of the window, and  $F$  is the input features.

### 3.6.2 Layers

#### 3.6.2.1 Hidden layer

The hidden layer is present between the input and output layers of the LSTM network. In each hidden layer, there are cells which have the functionality to add weights to the input and direct them through an activation function as the output. Besides, the cells used in the hidden layer change with the applications used in a realtime. In this thesis, I have successfully introduced a hidden layer with 256 cells to get better output from the network.

#### 3.6.2.2 Dense layer

In an LSTM network, the dense layer is known as the output layer, which connects the cells present in the hidden layer to the output layer cell.

### 3.6.2.3 Number of layers

The LSTM model consisting of three layers, two hidden layers and one dense layer. Initially, the train data from the sliding window approaches the input layer of the LSTM network. The output of the input layer connects the hidden layer through cells and trains the model by adding the weights to the data. After preparing the model in the hidden layer, the data is transferred to the next hidden layer using the return sequence equal to true. Upon receiving the data from the first hidden layer, the second hidden layer also trains the model with the appropriate weights and sends the data to the dense layer. Refer to Table 3.2 to see how the flow of the network. Table 3.2 tells us the network flow, but in the middle of each hidden layer, a dropout function is used after every hidden layer to prevent the risk of overfitting [5].

**Table 3.2:** LSTM Network Architecture

Layers	Architecture
Input Layer	$45867 \times 240 \times 4$
LSTM Hidden layer 1	(128,256)
LSTM leaku Relu	(128,256)
Dropout	(128,256)
LSTM Hidden layer 2	(128,256)
LSTM leaku Relu	(128,256)
Dropout	(128,256)
Dense Layer	(128,256)

### 3.6.3 Optimal hyperparameter

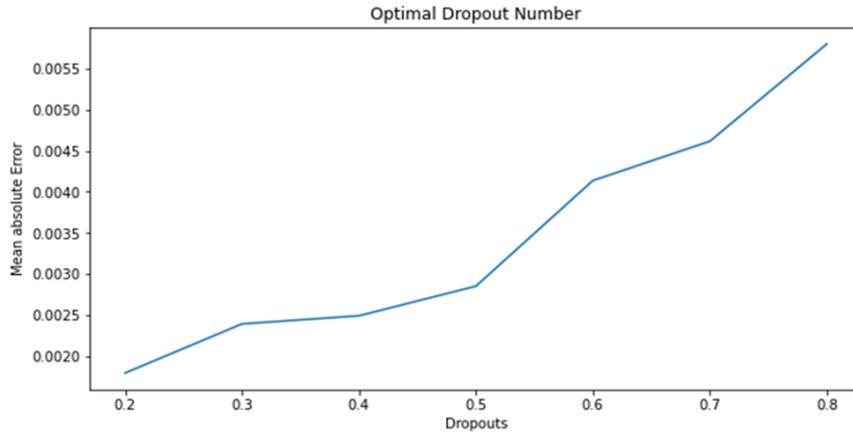
The hyperparameter plays a vital role in predicting accuracy while testing your model in any mathematical network. As referring to many papers, the paper by [5] has motivated to find the optimal hyperparameter, which would lead to higher accuracy and minimise the risk of overfitting data. When testing the Table 3.2 network, the default hyperparameters are provided for the test used in the other papers.

After testing the model, each hyperparameter is tweaked and trained to obtain the optimal value. Furthermore, an optimal value for an LSTM can be achieved by providing the test data to the network. Calculating the mean absolute loss from the predicted value will help us to provide an optimal value for each hyperparameter.

#### 3.6.3.1 Dropout

Dropout is a tunable parameter that reduces the overfitting of the data by choosing the random cells from the layer based on the probability specified and setting their output to 0 [6]. The dropout function applied to every hidden layer can be seen in Table 2. The model tested for every dropout value to obtain the optimal value shown in figure 3.11. In the figure 3.11, the X axis describes about the mean absolute error which was obtain by each iteration with the change of the dropout value. On the Y

axis the dropout values are considered shown in figure 3.11. From figure 3.11, it is evident that the optimal value for the dropout is 0.2 because the MAE loss is low compared to the other dropout values.



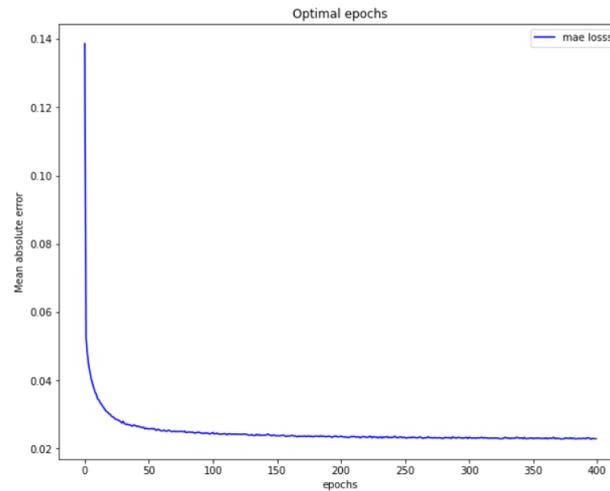
**Figure 3.11:** optimal Dropout Number

Despite doing this test, the value of epochs was set to 150 and the LSTM cells considered in each layer were fixed to 256,256,1, with a learning rate of 0.0001.

#### 3.6.3.2 The number of epochs for training the model

The number of epochs is a hyperparameter that specifies how many times the learning algorithm will run over the whole training dataset. An epoch represents a single training data sample, which updates the model parameters at the end of each epoch. As soon as the data passes through the network, it divides the data according to the batch size. The batch size used in this thesis is 256, which divides the training set data into 237 batch samples. Once the batch samples are divided, then the training of the network is done by the 237 batch samples for one iteration. This process continues until the end of the epoch.

The X axis of the figure 3.12 marks out the mean absolute error which was obtained by changing the epochs. On the Y axis different epochs values are considered, for each epoch value on the Y axis an iteration is done on the network to obtain the mean absolute error. After testing all the epochs value on the Y axis shown in the figure 3.12 describes the optimal epoch value is 200. Because at the epoch value of 200 has the lowest MAE value. However, While performing this test the LSTM cells are considered in each layer were fixed to 256,256,1, with a learning rate of 0.0001.

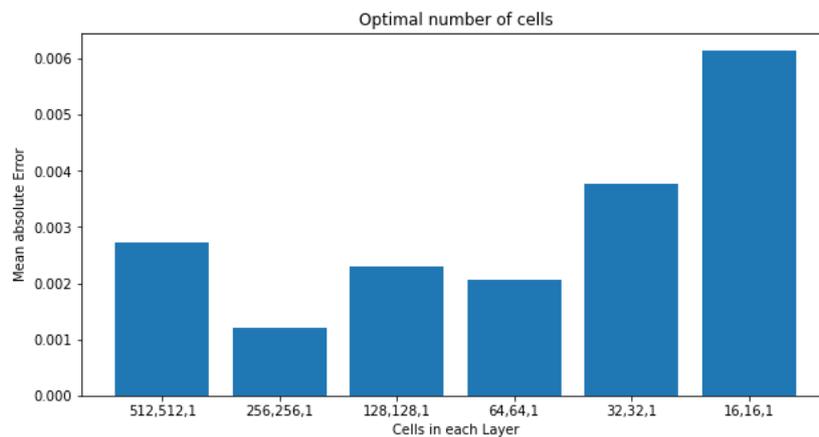


**Figure 3.12:** The optimal number of Epochs

### 3.6.3.3 Optimal Cells in each LSTM layer

In a neural network, hidden and dense layers consist of cells in each layer which would help to train the LSTM network. We consider different cell values on X axis shown in the figure 3.13 and on the Y axis different values of mean absolute error values are considered by training the model with a different value of cell. As a result to determine the optimal value of cells present in the hidden layers. Figure 3.13 clearly shows that cells in each layer have 256,256,1 since the number has the lowest MAE compared to the other cell's variety.

While doing this test, the value of the epochs set to 150 and the dropout set to 0.2 with a learning rate of 0.001.

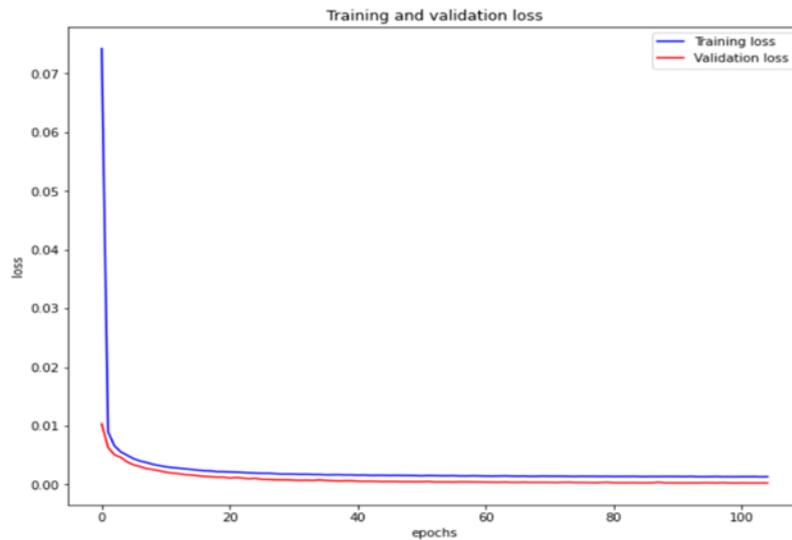


**Figure 3.13:** Optimal number of Cells in each layer

### 3.6.4 Loss function in LSTM

An LSTM loss function measures the difference between the predicted and absolute values for every epoch of the training data. The 20% of the validation data used to evaluate the model's training. When the model's training starts, you need to see

that the training loss should reduce simultaneously, and the validation loss should be reduced with the previous value and store the best value of the model. Moreover, the validation loss stored value should be less than the training loss, as shown in Figure 3.14. In the figure 3.14 the blue line marks about the training loss and the red line talks about the validation loss of the model. As these losses are obtained when the model is trained and tested with the optimal hyper-parameter. However, if the model is not trained with optimal hyperparameters, this would lead to a case of overfitting when the validation loss is not lower than the training loss.



**Figure 3.14:** Training and validation loss

The loss function used in this model is the Mean absolute error mainly used in the fault prognosis analysis.

#### 3.6.5 LSTM optimizer

RMSprop and Adams are the optimizer used in this thesis with a learning rate of 0.001. By comparing both methods, Adam has high performance and convergence compared to the RMSprop optimizer.

##### 3.6.5.1 Error calculation

After the optimizer selection it needed to calculate the error while training the model. The error used in this thesis are MAE and RMSE which is explained in the section 2.7.

# 4

## Results

In this section, the validation of the LSTM model is presented with the different conditioning monitoring data of voltage, current, power, and resistance to predict possible fault that would happen in the future. The output from the model is evaluated by the different loss functions explained in the above section[theory], which would result in the accuracy of the model and the error of the predicted value with the original value.

### 4.1 Prediction of test data after inverse differentiation and denormalization

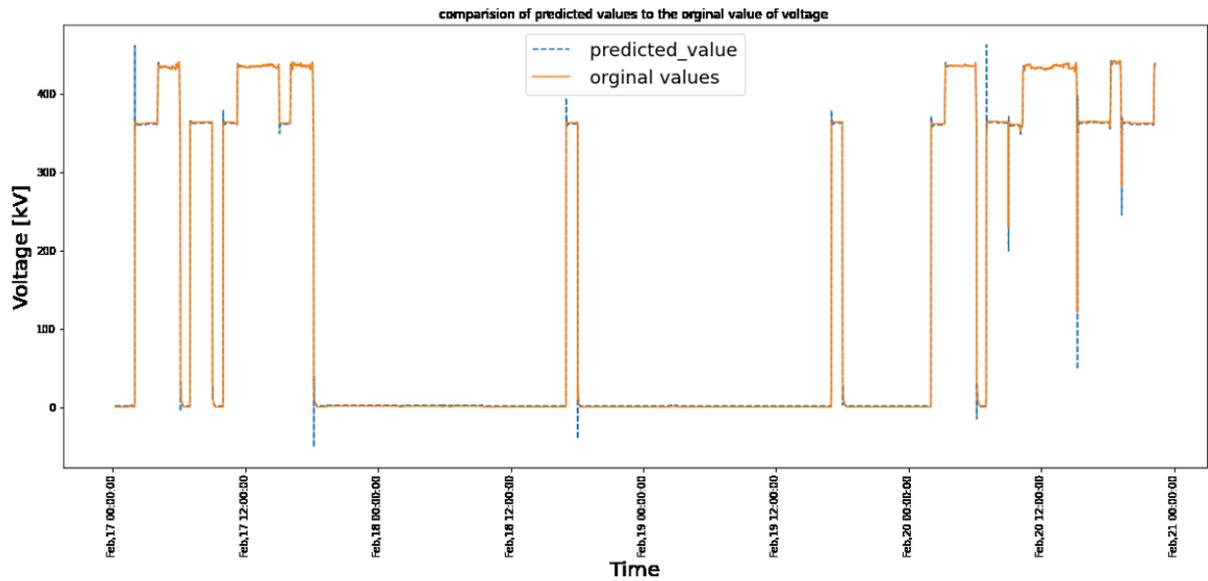
The denormalization of the values plays a vital role when you want to predict the voltage, current, power and resistance value. It would be challenging to understand the value of those features. So, when the values of the features appear after testing the model one should denormalize and do the inverse differentiation to obtain the current, voltage, power, and resistance signal.

#### 4.1.1 Comparison of voltage prediction with original value of the voltage:

Figure 4.1 talks about the LSTM model prediction of the voltage in the blue dashed line and the original voltage values in the solid orange line. Considering figure 4.1, it is evident that the prediction value follows the trend of the actual voltage. With that reference, it is evident that the model fits perfectly with the original values of the voltage. Moreover, the maximum sensor value of the voltage is 443 kV, and the model has predicted a value of 445 kV. Nevertheless, the model has failed to predict the values at 340 kV and 0 kV, which can be seen in figure 4.1. It has happened at every rise and downfall of the voltage values that the model has predicted the high voltage values. This happened because the LSTM requires more correlated features to reduce the peaks. As a result, identify the error between the actual and predicted value shown in table 4.1, the estimated error of the model using MAE (mean absolute error), MSE (mean squared error), RAE (relative absolute error), and RRSE (root relative squared error) variables with 256 samples and a learning rate of 0.0001.

**Table 4.1:** Performance measure of the constructed predictive models on validation data( case voltage)

MSE	MAE	RAE	RRSE	Accuracy
117.333	1.637	0.876%	5.5925%	99.36%

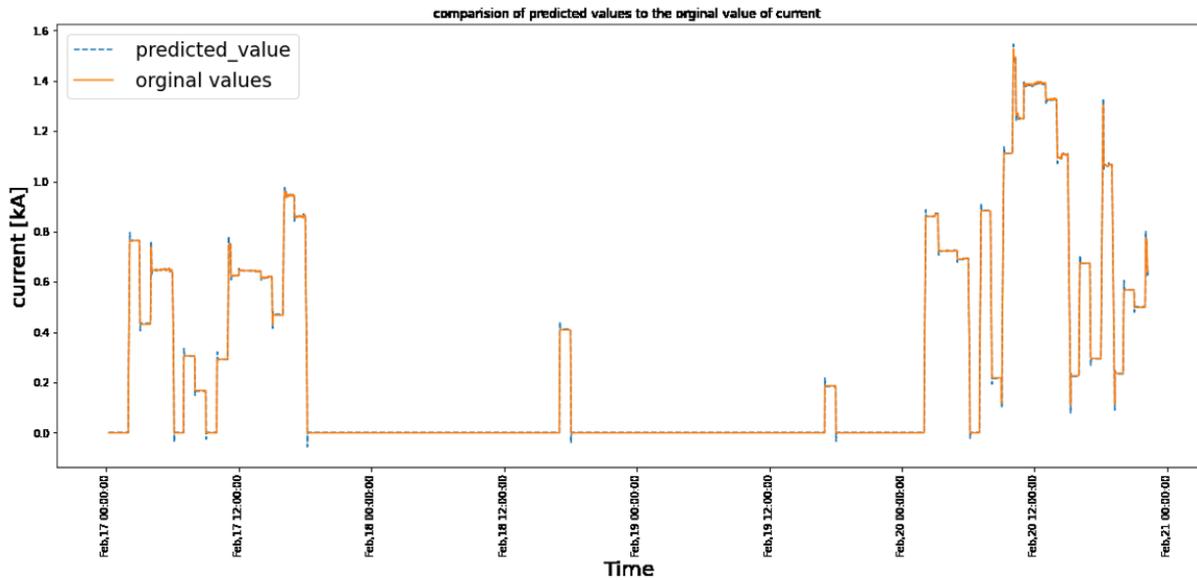
**Figure 4.1:** Comparison of voltage prediction with actual value of voltage

#### 4.1.2 Comparison of current prediction with the actual value of the current

Figure 4.2 depicts the LSTM model forecast of the current in the blue dashed line and the original current values in the solid orange line. Figure 4.2 shows that the forecast value closely matches the pattern of the actual current, Which means that the model fits perfectly with the actual sensor values of the current. But the predicted values of the model are slightly above the actual values of the current at 0 kA and at 0.7 kA. A tiny variation during the current's increase and decrease is inconsequential. As a result, find the difference between the actual and predicted values in 4.2, as well as the model's estimated error using MAE (mean absolute error), MSE (mean squared error), RAE (relative absolute error), and RRSE (root relative squared error) variables with 256 samples and a learning rate of 0.0001.

**Table 4.2:** Performance measure of the constructed predictive models on validation data( case current)

MSE	MAE	RAE	RRSE	Accuracy
3.68912857e-05	0.00213346	0.65475119%	1.54380465%	99.36%



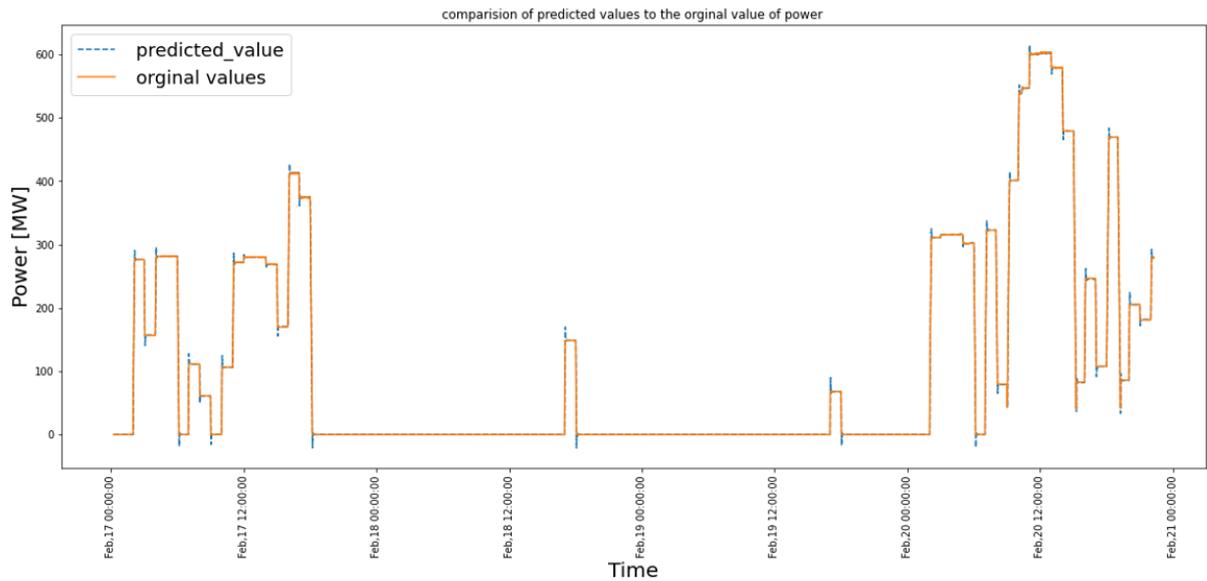
**Figure 4.2:** Comparison of current prediction with actual value of current

### 4.1.3 Comparison of power prediction with the original value of the power:

Figure 4.3 shows the power predictions of the LSTM model as dashed blue lines and the actual power values as solid orange lines. Figure 4.3 shows that the predicted values are very close to the actual power pattern and small deviations between power increases and decreases are not significant which can be seen between the time feb 17 00:00 to feb 17 12:00. However, by inspecting the figure 4.3 there it can be seen that model predicted value follows the path but there is a small difference at the sudden rise and sudden falls with a change of value of 2%. As a result, to calculate the difference between the actual and predicted values in table 4.3 and the MAE (mean absolute error), MSE (mean squared error), RAE (relative absolute error), and RRSE (root). (Relative squared error) variables with 256 samples and a learning rate of 0.0001.

**Table 4.3:** Performance measure of the constructed predictive models on validation data( case power)

MSE	MAE	RAE	RRSE	Accuracy
8.29800211	1.03718855	0.77797587%	1.75467877%	99.36%



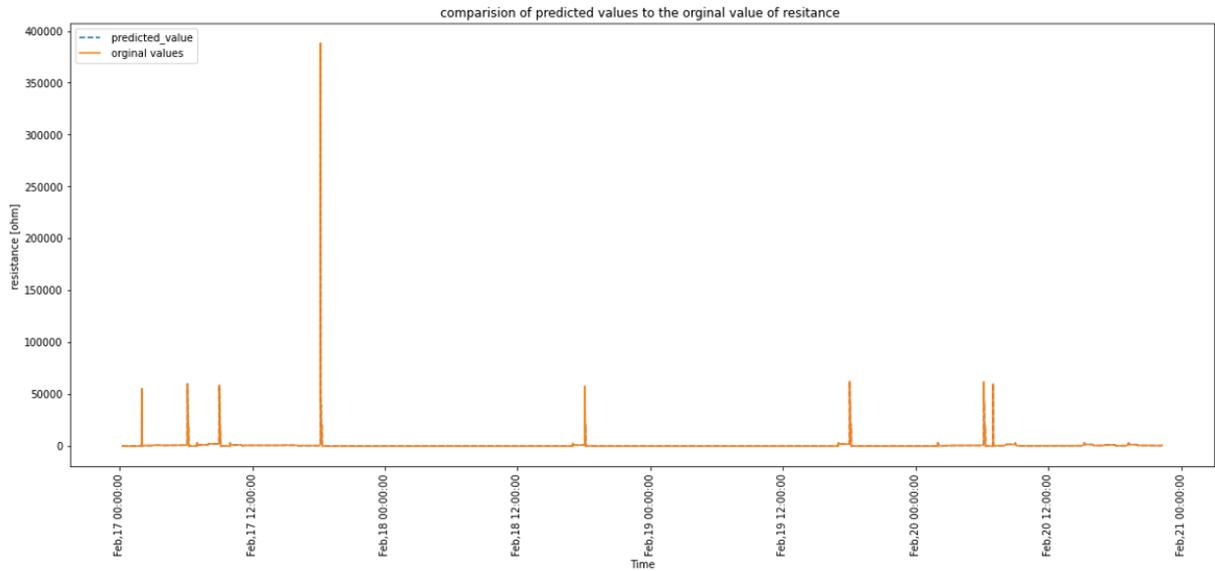
**Figure 4.3:** Comparison of power prediction with actual value of power

#### 4.1.4 Comparison of resistance prediction with the actual value of the resistance:

Figure 4.4 depicts the LSTM model’s resistance predictions as dashed blue lines and the actual resistance values as solid orange lines. Figure 4.4 indicates that the predicted values are quite similar to the actual resistance pattern, where the predicted value of the resistance is inscribed below the orange dashed line. By inspecting the figure 4.4 it says that it a not overfitting the graph and concludes that it was a good fit. However, by looking the figure 4.4 there could be a strange behaviour between the time feb 17 12:00 to feb 18 00:00 there is a large spike with 38000 ohms. This is happened due to the system current tended to zero then a infinite amount of resistance can be observed and also time between feb 17 00:00 to feb 17 12:00 there are three short spikes which discuss about the sudden change in the current obtained in the system. Moreover this spikes are identifeid by the model accurately.As a consequence, to check the difference between the actual and projected values in table 4.4, as well as the MAE (mean absolute error), MSE (mean squared error), RAE (relative absolute error), and RRSE (relative squared error), were calculated (root). Variables with (Relative Squared Error) 256 samples and a learning rate of 0.0001.

**Table 4.4:** Performance measure of the constructed predictive models on validation data( case resistance)

MSE	MAE	RAE	RRSE	Accuracy
6.29800211	1.43718855	0.70747587%	1.54607877%	99.3%



**Figure 4.4:** Comparison of resistance prediction with actual value of resistance

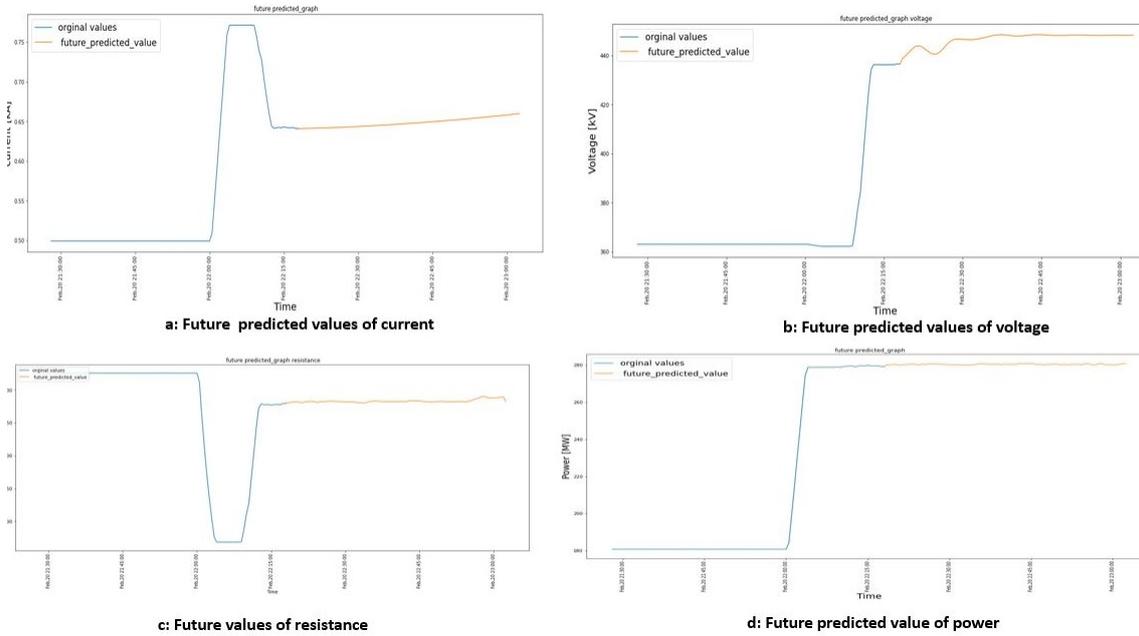
## 4.2 Future Prediction for the fault

The top left corner of figure 4.5a describes the future predicted current, where the blue line discusses the past values of the current, and the orange line describes the future values of the current. Looking deep into the figure 4.5a shows a slight rise in future value at feb 21 22:45 is 0.67 kA. This rise has happened due to the current model's prediction shown in the figure 4.2 has a RAE of 0.65%. However, the slight rise can be reduced by choosing appropriate hyperparameters. Now let us consider the figure 4.5b at the top right corner, which describes the future predicted voltage. In this graph, the blue line tells us about the past values, and the orange line describes the future values of the voltage. By inspecting the voltage graph, a small heap appeared at feb 21 21:15 in the voltage. The model's predicted voltage graph shown in the figure 4.1 had predicted the voltage as very high at the sharp corners. So for that reason, there is a heap in the future voltage values. The future predicted power graph can be seen right below the voltage graph in the figure 4.5d . This graph consists of a blue line about past power values and an orange line about future power values. Inspecting the graph tells us that the future values of the power have a minor error which can be negligible similarly if we look at the graph of future predicted resistance situated at the bottom left corner shown in the figure 4.5c. This graph has a blue line representing the past values of the resistance and an orange line representing the future values of the resistance. However, the resistance graph also has a minor error that can be seen from the future predicted resistance graph 4.4 with a RAE 0.70747%.

From figure 4.5, it is evident that future voltage value is around 441 kV which is constant, while the current is around 0.65kA, the power is around 280 MW, and the resistance is 650 ohm. This value states that no fault appears in the next 30

## 4. Results

minutes of the time interval. However, if the fault appears in the DC line, then one can see the change in the voltage would reach the value of 0V, and the current reaches a higher value above the rating of the converter. So, then there would be a large amount of short circuit current flowing in the system and causes a black out situation of the HVDC grid.



**Figure 4.5:** Future fault prediction by comparing the voltage current power and resistance

# 5

## Discussion

As part of this thesis, the HVDC (High Voltage Direct Current) system provides fault signals from the sensor in the form of time-series data. These signal parameters of voltage, current, resistance and power are estimated using the time series measurement. To estimate the signal parameters, one needs to divide the known signal parameters with a long history in statical signal processing [35]. The division of the signal into long segments is done by using a method called a sliding window, explained in the above section.

In this study, I used to set of purely data-driven techniques to capture the hidden patterns of the fault time series without any signal modelling of DC faults. In this regard, I have considered an LSTM model to better understand the time-series sequence compared to the machine learning algorithms such as (support vector regression, multi-layer perceptron and linear regression). Nevertheless, it is not intended to say that the LSTM model performed well compared to the other Machine learning models in the fault prognosis context. There are no specific guidelines in the machine learning community to say that this algorithm suits this application.

Moreover, in this study, I have observed that predicting the fault of the DC line based on the time series is a more challenging problem than predicting the regular operation of the DC line. From the HVDC line perspective, we may justify the fault of the DC line via an unexpected change in the voltage and the current when a line to a ground fault has appeared in the system. When these faults appear in the system, they try to absorb more current from the converter to feed the fault and may affect the line's resistance. While this process continues, the line resistance changes and may lead to damage to the line. Since the damage to the line is dependent on the rise of current and the fall of voltage values. These values are taken in with respect to time. However, most faults appear in the DC line when the pole is blocked, so it is difficult to identify the faults if there is no change in the voltage and current. Hence, I faced more challenges in estimating the faults than the line's regular operation.

The trained LSTM architecture in this thesis will estimate the DC line voltage, the current, the power, and the resistance is shown in figure 4.5 using the recorded HVDC system waveform in the last 1 hour. In addition, by analyzing the HVDC system data over the last hour, the model should be able to detect the fault at an early stage or become suspicious when the value changes in the voltage lesser than the limit and the value of the current greater than the limit coined as the faults in

the system.

Furthermore, solving the problem by the categorical variable [35] way might be difficult because the data obtained from the sensors will not have the output labels of the fault and not the fault. As a result, the problem was formulated into a regression problem. The aim is to estimate the change in the continuous variable of voltage, current, power and resistance. A regression prediction is more complex than classification [36]. In practice, however, it enables practitioners to interpret fault situations based on pole faults and line to ground faults rather than imposing our definition for "faulty" conditions. Due to a lack of consistency in definitions of problematic conditions, we are experiencing this problem. In such a case, predicting the signs of a defect (like a change of voltage or current) would be the most appropriate way to describe the problem.

One of the practical points in the utility of our predictive model relates to the length of time needed for decision-making [35]. On the one hand, the 1-hour length of the collected voltage, current, power and resistance waveform required for decision-making by the trained LSTM is impressively brief. On the other hand, no model is 100% accurate compared to the real-time monitoring system connected to the HVDC line.

# 6

## Conclusion

The thesis aims to predict the DC cable fault using the ML model trained on the condition monitoring data of the DC cable. Section 3.1.1 describes the data generated from the sensors of the DC cable. After generating the dataset by the sensors of the DC line, it was identified that it has less than 5% faults compared to the entire dataset. However, while going through the dataset, it has been identified that most line-to-ground faults appear in the pole reversal. This means that when the power direction has changed from HVDC station 1 to HVDC station 2, the line fault locator has triggered, stating that there is a fault.

Nevertheless, there is no voltage and current fluctuation when the fault has appeared. In addition, line-to-ground faults appeared in the system when the power requested by HVDC station 2 was equal to 0MW. Then the line fault locator sensor has triggered and stated about the fault. Nonetheless, there was no change in the voltage and current values because the power was equal to 0MW. By considering all these criteria, the data generated from the sensors were modified using the standard scaler, discussed in section 3.3.5.2.

The modified data sent to the sliding window algorithm discussed in the section 3.5 converts the data into supervised learning and fed into the baseline model of the LSTM. However, due to the recent advances in machine learning, deep learning may rely more heavily on data-driven models, including all the above model generation steps. In this regard, the author relied exclusively on LSTM architecture in the current study to identify the patterns from the voltage, current and power signals while building up the predictive model to detect the faults in the cable. In other words, a distinct feature extraction stage during model creation was skipped entirely. A brute-force model selection with optimal hyperparameters and the LSTM architecture examined resulted in training for predicting the fault in the DC line.

This approach led us to predict the model's performance calculated using the Relative absolute error of the voltage, current, power, and resistance are deficient and led to an accuracy of 99.93%. By seeing this accuracy, it is easy for a TSO to identify the fault prior and can take precautionary steps to avoid the fault.

### 6.1 Ethical Aspects

The project presents the following risks in regards to the IEEE code of ethics.

**To improve the understanding by individuals and society of the capabilities and societal implications of conventional and emerging technologies, including intelligent systems.**

Possible risks: Hiding the disadvantages of the project and only promoting the advantages.

Preventive measures: The limitations of the project must clearly be understood and stated. The sustainable impacts of the project must also be promoted to increase awareness and interest.

**To avoid injuring others, their property, reputation, or employment by false or malicious actions, rumors or any other verbal or physical abuses.**

Possible risks: Implementing the services in available wind turbines without further research or testing. This can lead to damage to property and reputation.

Preventive measures: Presenting all available data and results, while stressing on the need for more testing and development of technology.

**To support colleagues and co-workers in following this code of ethics, to strive to ensure the code is upheld, and to not retaliate against individuals reporting a violation.**

Possible risks: Not encouraging fellow team members, leaving them unnoticed are some possible risks.

Preventive measures: Setting a time plan and helping each other in technical aspects. Involving everyone in the team in discussions and decisions. Conducting peer reviews and providing constructive suggestions will allow for individual growth and contribute to the team's objectives.

# 7

## Future Work

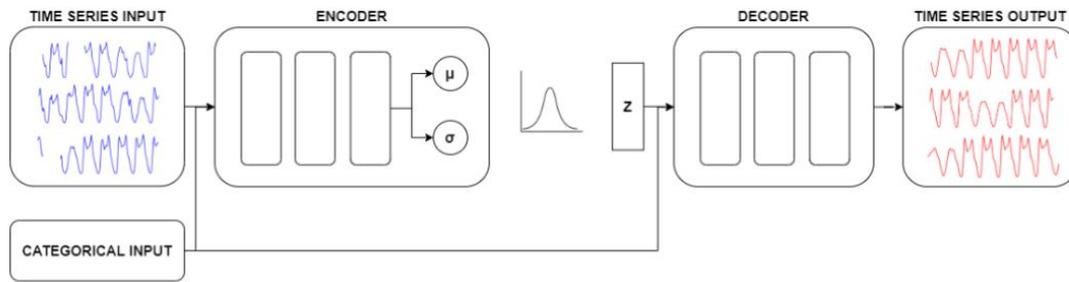
This thesis considers four parameters that influence the fault of the DC line. These parameters generate the training samples and predict the outcomes from the ML model. Nevertheless, some other parameters can be included in the DC line fault. Moreover, to improve the concept of multivariate and multi-step time series prediction with LSTM. Furthermore, NLP can also be used to identify the faults in the DC line. In addition to NLP, two other methods are identified for multi-step development.

1. This method uses a brute force approach to extract the feature from the data set and send them to an LSTM Autoencoder. LSTM Autoencoder comprises an encoder part and a decoder part. In an encoder part, a fast Fourier transform mathematical tool is applied to the dataset, which converts the data into complex data. After obtaining the complex data is sent to the LSTM model.

The decoder also consists of an LSTM model, which converts the complex data to the original size and applies an inverse Fast Fourier transform with an appropriate hyperparameter to tune the model.

2. This technique extracts the feature from the data set via brute force and sends it to an LSTM cell, which serves as the encoder. It takes as input 3D sequences formed by concatenating time series data with a categorical feature embedding. It generates a 2D output, as do all encoders in a VAE architecture, which estimates the mean and variance of the latent distribution. To construct 3D sequences, the decoder samples from the 2D latent distribution upsampling. The resulting sequences are concatenated with the original category embeddings and fed into an LSTM cell to recover the original time series data sequence.

A VAE (Variational AutoEncoder) is trained by minimising a loss that is the combination of two parts. A reconstruction component (scaled mean squared error), which indicates the model reproduces the target, and a regularisation component (Kullback Leibler divergence), works as a restriction to make the latent space more akin to a normal distribution. For a clear understanding, refer to the figure 7.1.



**Figure 7.1:** Variational AutoEncoder Architecture

These are the two methods that can reduce this thesis's limitation and provide great results in identifying the faults for multiple steps.

# Bibliography

- [1] Z. Guan, T. Ji, X. Qian, Y. Ma and X. Hong, "A Survey on Big Data Preprocessing," 2017 5th Intl Conf on Applied Computing and Information Technology/4th Intl Conf on Computational Science/Intelligence and Applied Informatics/2nd Intl Conf on Big Data, Cloud Computing, Data Science (ACITCSII-BCD), 2017, pp. 241-247, doi: 10.1109/ACIT-CSII-BCD.2017.49.
- [2] <https://towardsdatascience.com/7-ways-to-handle-missing-values-in-machine-learning-1a6326adf79e>
- [3] J. Abasova, J. Janosik, V. Simoncicova and P. Tanuska, "Proposal of Effective Preprocessing Techniques of Financial Data," 2018 IEEE 22nd International Conference on Intelligent Engineering Systems (INES), 2018, pp. 000293-000298, doi: 10.1109/INES.2018.8523922.
- [4] <https://medium.com/technofunnel/what-when-why-feature-scaling-for-machine-learning-standard-minmax-scaler-49e64c510422>
- [5] Nils Reimers and Iryna Gurevych. Optimal hyperparameters for deep lstm-networks for sequence labeling tasks. arXiv preprint arXiv:1707.06799, 2017.
- [6] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.
- [7] Hota, H.S., Handa, R., and Shrivastava, A.K. (2017). Time Series Data Prediction Using Sliding Window Based RBF Neural Network.
- [8] Yahmed, Y.B., Bakar, A.A., Razak Hamdan, A., Ahmed, A., and Abdullah, S.M.S. (2015). Adaptive sliding window algorithm for weather data segmentation. *Journal of Theoretical and Applied Information Technology*, 80(2), 322-333.
- [9] Mozaffari, L., Moxaffari, A. and Azad, N.L. (2015). Vehical speed prediction via a sliding-window time series analysis and an evolutionary least learning machine: A case study on San Francisco urban roads. *Engineering science and technology, an international journal*, 18, 150-162
- [10] Yu, Y., Zhu, Y., Li, S. and Wan, D. (2014). Time Series Outlier Detection Based on Sliding Window Prediction. Hindawi Publishing Corporation, 2014.
- [11] Vafaeipour, M., Rahbari, O., Rosen, M.A., Fazelpour, F. and Ansarirad, P. (2014). Application of sliding window technique for prediction of wind velocity time series. *International journal of Energy and environmental engineering (springer)*, 5, 105-111.
- [12] A. K. Khaimar and P. J. Shah, "Study of various types of faults in HVDC transmission system," 2016 International Conference on Global Trends in Signal

- Processing, Information Computing and Communication (ICGTSPICC), 2016, pp. 480-484, doi: 10.1109/ICGTSPICC.2016.7955349.
- [13] Koganti Sri Lakshmi , G. Sravanthi, L. Ramadevi, “ A Review Paper on Technical Data of Present HVDC Links in India”, International Journal on Recent and Innovation Trends in Computing and (IJRITCC)
- [14] Yi Wang, Ziguang Zhang, Yuag Hei, Xiangyu Zhang,“ Pole-to-Ground Fault Analysis in Transmission Line of DC Grids Based on VSC ”, IEEE 8th International Power Electronics and Motion Control Conference (IPEMC-ECCE Asia), pp.2028-2032, May 2016.
- [15] Danhua Liu, Tongzhen Wei, Qunhai Huo, Lixin Wu, “ DC Side Line to-line Fault Analysis of VSC- HVDC and DC- fault-clearing Methods”, IEEE 5th International Conference on Electric Utility Deregulation and Restructuring and Power Technologies, pp.2395-2399, November 2015
- [16] Bucher, M. K., and Franck, C. M. (2013a). Contribution of fault current sources in multiterminal HVDC cable networks. *IEEE Transactions on Power Delivery*,28(3), 1796–1803.
- [17] M. K. Bucher, and C. M. Franck. "Analysis of transient fault currents in multiterminal HVDC networks during pole-to-ground faults." presented at the International Conference on Power Systems Transients (IPST2013), Vancouver,Canada, 2013b.
- [18] Bucher, M. K., and Franck, C. M. (2015). Analytic approximation of fault current contributions from capacitive components in HVDC cable networks. *IEEE Transactions on Power Delivery*, 30(1), 74–81.
- [19] Bucher, M. K., and Franck, C. M. (2016). Analytic approximation of fault current contributions from AC networks to MTDC networks during pole-to-ground faults. *IEEE Transactions on Power Delivery*, 31(1), 20–27.
- [20] J. Wang, B. Berggren, K. Linden, J. Pan, and R. Nuqui, "multi-terminal DC system line protection requirement and high speed protection solutions," in Proceedings of the 2015 CIGRE Symposium, Cape Town, South Africa, 2015,26–30.
- [21] Seneath, J., and Rajapakse, A. D. (2016). Fault detection and interruption in an earthed HVDC grid using ROCOV and hybrid DC breakers. *IEEE Transactions on Power Delivery*, 31(3), 973–981.
- [22] Zheng, X., Nengling, T., Guangliang, Y., and Haoyin, D. (2012). A transient protection scheme for HVDC transmission line. *IEEE Transactions on Power Delivery*, 27(2), 718–724.
- [23] Muniappan, M. A comprehensive review of DC fault protection methods in HVDC transmission systems. *Prot Control Mod Power Syst* 6, 1 (2021). <https://doi.org/10.1186/s41601-020-00173-9>
- [24] Tanya Singh. How is artificial intelligence different from machine learning, 2017. URL <https://www.linkedin.com/pulse/how-artificial-intelligence-different-from-machine-learning-singh>[Accessed: 2018-03-28].
- [25] What are Neural Networks? | IBM. <https://www.ibm.com/cloud/learn/neural-networks>
- [26] <https://medium.com/tech-break/recurrent-neural-network-and-long-term-dependencies-e21773defd92>
- [27] <https://insidelearningmachines.com/meanabsoluteerror/>

- 
- [28] Wojciech Zaremba, Ilya Sutskever, and Oriol Vinyals. Recurrent neural network regularization. arXiv preprint arXiv:1409.2329, 2014.
- [29] Felix A Gers, Jürgen Schmidhuber, and Fred Cummins. Learning to forget: Continual prediction with lstm. 1999.
- [30] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," in Proc. Int. Conf. Mach. Learn., Jun. 2013, pp. 1310–1318.
- [31] N. Sharma, S. Ali and G. Kapoor, "Fault Detection in Wind Farm Integrated Series Capacitor Compensated Transmission Line Using Hilbert Huang Transform," 2018 International Conference on Computing, Power and Communication Technologies (GUCON), 2018, pp. 765-769, doi: 10.1109/GU-CON.2018.8675116.
- [32] A. R. Abbasi, M. R. Mahmoudi and M. M. Arefi, "Transformer Winding Faults Detection Based on Time Series Analysis," in IEEE Transactions on Instrumentation and Measurement, vol. 70, pp. 1-10, 2021, Art no. 3516210, doi: 10.1109/TIM.2021.3076835.
- [33] Y. Huang, C. -H. Chen and C. -J. Huang, "Motor Fault Detection and Feature Extraction Using RNN-Based Variational Autoencoder," in IEEE Access, vol. 7, pp. 139086-139096, 2019, doi: 10.1109/ACCESS.2019.2940769.
- [34] M. Alhussein, K. Aurangzeb and S. I. Haider, "Hybrid CNN-LSTM Model for Short-Term Individual Household Load Forecasting," in IEEE Access, vol. 8, pp. 180544-180557, 2020, doi: 10.1109/ACCESS.2020.3028281.
- [35] A. Zollanvari, K. Kunanbayev, S. Akhavan Bitaghsir and M. Bagheri, "Transformer Fault Prognosis Using Deep Recurrent Neural Network Over Vibration Signals," in IEEE Transactions on Instrumentation and Measurement, vol. 70, pp. 1-11, 2021, Art no. 2502011, doi: 10.1109/TIM.2020.3026497
- [36] S. M. Weiss and N. Indurkha, Predictive Data Mining: A Practical Guide. San Francisco, CA, USA: Morgan Kaufmann, 1998.
- [37] C.-I.-L. Liu, W.-H. Hsaio, and Y.-C. Tu, "Time series classification with multivariate convolutional neural network," IEEE Trans. Ind. Electron., vol. 66, no. 6, pp. 4788–4797, Jun. 2019
- [38] M. Längkvist, L. Karlsson, and A. Loutfi, "A review of unsupervised feature learning and deep learning for time-series modeling," Pattern Recognit. Lett., vol. 42, pp. 11–24, Jun. 2014.
- [39] R. Chandra, "Competition and collaboration in cooperative coevolution of elman recurrent neural networks for time-series prediction," IEEE Trans. Neural Netw. Learn. Syst., vol. 26, no. 12, pp. 3123–3136, Dec. 2015.
- [40] J. Liu, N. Tai and C. Fan, "Transient-Voltage-Based Protection Scheme for DC Line Faults in the Multiterminal VSC-HVDC System," in IEEE Transactions on Power Delivery, vol. 32, no. 3, pp. 1483-1494, June 2017, doi: 10.1109/TPWRD.2016.2608986
- [41] <https://search.abb.com/library/Download.aspx?DocumentID=POW-0038&LanguageCode=en&DocumentPartId=Action=Launch>
- [42] Wang, Tao and Guan, Lin. (2014). Probabilistic risk assessment applied to maintenance schedule of HVDC. 1-5. 10.1109/PMAPS.2014.6960608.
- [43] C. Yuan, D. Dai, J. Qiu, L. Chen, H. Qu and J. Cao, "An Optimal Preventive Maintenance Method for MMC based VSC-HVDC," 2020 2nd International

- Conference on Smart Power and Internet Energy Systems (SPIES), 2020, pp. 13-16, doi: 10.1109/SPIES48661.2020.9242935.
- [44] Das, Swagata and Santoso, Surya and Gaikwad, Anish and Patel, Mahendra. (2014). Impedance-based Fault Location in Transmission Networks: Theory and Application. *IEEE Access*. 2. 1-1. 10.1109/ACCESS.2014.2323353.
- [45] Wang, Y., Zhang, Z., Fu, Y., Hei, Y., and Zhang, X. (2016). Pole-to-ground fault analysis in transmission line of DC grids based on VSC. 2016 IEEE 8th International Power Electronics and Motion Control Conference (IPEMC-ECCE Asia), 2028-2032.
- [46] S. S. Dessouky, M. Fawzi, H. A. Ibrahim and N. F. Ibrahim, "DC Pole to Pole Short Circuit Fault Analysis in VSC-HVDC Transmission System," 2018 Twentieth International Middle East Power Systems Conference (MEPCON), 2018, pp. 900-904, doi: 10.1109/MEPCON.2018.8635237.
- [47] Gürsakal, Necmi Yilmaz, Fırat Ugurlu, Erginbay. (2020). Finding opportunity windows in time series data using the sliding window technique: The case of stock exchanges.
- [48] Piyush Kapoor, Sarabjeet Singh Bedi, "Weather Forecasting Using Sliding Window Algorithm", *International Scholarly Research Notices*, vol. 2013, Article ID 156540, 5 pages, 2013.
- [49] Abdulkader Helwan, Dilber Uzun Ozsahin, "Sliding Window Based Machine Learning System for the Left Ventricle Localization in MR Cardiac Images", *Applied Computational Intelligence and Soft Computing*, vol. 2017, Article ID 3048181, 9 pages, 2017.
- [50] Abdulkader Helwan, Dilber Uzun Ozsahin, "Sliding Window Based Machine Learning System for the Left Ventricle Localization in MR Cardiac Images", *Applied Computational Intelligence and Soft Computing*, vol. 2017, Article ID 3048181, 9 pages, 2017.
- [51] Sezer, Omer Gudelek, Ugur Ozbayoglu, Murat. (2019). Financial Time Series Forecasting with Deep Learning : A Systematic Literature Review: 2005-2019.
- [52] 10 Best Visual Studio Code theme for Python in 2022. <https://toastofcode.com/10-best-visual-studio-code-theme-for-python-in-2022/>
- [53] Kokoszka, Piotr Young, Gabriel. (2016). KPSS test for functional time series. *Statistics*. 50. 1-17. 10.1080/02331888.2015.1128937.
- [54] Iqbal, Muhammad Muzaffar, Salman Amin, Intisar Ali Sajjad and Saaqib Haroon. "HVDC its Scope and Future Trends." (2018).
- [55] Islam, M. R., Guo, Y., Zhu, J. (2014). A review of offshore wind turbine nacelle: Technical challenges, and research and development trends. *Renewable and Sustainable Energy Reviews*, 33, 161–176.
- [56] Perveen, R., Kishor, N., Mohanty, S. R. (2014). Offshore wind farm development: Present status and challenges. *Renewable and Sustainable Energy Reviews*, 29, 780–792.
- [57] Wu, J., Wang, Z. X., Wang, G. Q. (2014). The key technologies and development of offshore wind farm in China. *Renewable and Sustainable Energy Reviews*, 34, 453–462.
- [58] Kamakshaiyah, S., Kamaraju, V. (2013). HVDC Transmission. McGraw Hill.

- [59] J. P. Rodriguez, “Dynamic averaged models of VSC-based HVDC Systems for Electromagnetic Transient Programs,” Ph.D. dissertation, University of Montreal, Canada, 2013.
- [60] Franck, C. M. (2011). HVDC circuit breakers: A review identifying future research needs. *IEEE Transactions on Power Delivery*, 26(2), 998–1006.
- [61] Jovcic, D., Ahmed, K. (2015). High voltage direct current transmission: Converters, systems and DC grids. John Wiley Sons.
- [62] Xu, L., Yao, L., Sasse, C. (2007). Grid integration of large DFIG-based wind farms using VSC transmission. *IEEE Transactions on Power Systems*, 22(3), 976–984.
- [63] J. Canadelaria, and J. D. Park, “VSC HVDC System Protection: A review of Current Methods,” in *Proc. Power Systems Conference and Exposition (PSCE)*, Phoenix, USA, 2011, pp.1–7.
- [64] Meyer, C., Hoing, M., Peterson, A., et al.: ‘Control and design of dc grids for offshore wind farms’, *IEEE Trans. Ind. Appl.*, 2007, 43, (6), pp. 1475–1482.
- [65] Kirby, N. M., Xu, L., Lockett, M., et al.: ‘HVDC transmission for large offshore wind farms’, *Power Eng. J.*, 2002, 16, (3), pp. 135–141.



DEPARTMENT OF SOME SUBJECT OR TECHNOLOGY  
CHALMERS UNIVERSITY OF TECHNOLOGY  
Gothenburg, Sweden  
[www.chalmers.se](http://www.chalmers.se)



**CHALMERS**  
UNIVERSITY OF TECHNOLOGY