



School of Science

M.Sc. Data Analytics and Technologies

University of Bolton

Enhancing and Optimizing Asset and Equipment Lifecycle Management through Predictive Maintenance Techniques

Prepared by Oluwatosin Ayokunle Ojo

Student ID: 2312602

Supervisor: Ibtisam Mogul

**A report submitted in the partial fulfilment for
the award of a Master's degree in Data Analytics and Technologies**

September 2024.

The University of Bolton

Deane Road, Bolton BL3 5AB

<http://www.bolton.ac.uk>

Declaration

This is to certify that this thesis titled the “Enhancing and Optimizing Asset and Equipment Lifecycle Management through Predictive Maintenance Techniques” is an indigenous work and has not been put forward to any other publishing or educational establishment. All sources made use of in this study has been properly cited and referenced.

Acknowledgement

I wish to extend a heartfelt appreciation to the almighty God for his guidance and sustenance all through the course of this study.

I am thankful to my advisor Mrs. Ibtisam Mogul for constantly providing me with valuable insights and directions on how to proceed with my study. I am sincerely appreciative of the effort she put into reviewing and correcting errors while providing nudges on how to improve my writeup especially in producing a proper research design process.

I would also like to thank the University of Bolton for the resources provided as well as appreciate all other professors and colleagues involved along the course of the study for their time and effort into breaking down some other concepts and solutions when needed.

Finally, a heartfelt thank you to my family and friends for being understanding and supportive during the study.

Abstract

The study examines the introduction and integration of Big Data, Machine Learning, Deep Learning and Artificial Intelligence in industrial applications and equipment's to improve Asset and Equipment Lifecycle Management through the proper implementation of a Predictive Maintenance System therefore leading to the optimization of resources and services.

Asset Lifecycle Management (ALM) remains important in maximizing the use of critical assets across multiple sectors, however the traditional approaches adopted by these organizations are mostly reactive in nature leading to prolonged downtimes and increased maintenance and operation cost.

With the advent big data and ML techniques in detecting asset and equipment failures and automating scheduled maintenance, industries can mitigate against these challenges. This report takes a systematic review method in reviewing related literature to understand how predictive maintenance can be used to address the said challenges. It uses a case study of the Metro Railway Stations incorporates an innovative approach using different advanced ML and DL modelling techniques in predicting equipment failure. The key models used in this experiment are Logistic Regression, Random Forest, Decision Tree, Gradient Boosting Machine and Neural Network. The study involves testing the models on different iterations of the dataset (Metro4, Analogue and Digital Dataset), performing feature engineering to decide when failure is registered, a hybrid data sampling to create a balanced dataset, and feature selection prior to initiating the models.

This report shows different experiments done with different data samples and identifies the Analogue dataset as the best dataset results for detecting failures. It also outlines Random Forest out of all the models initiated as the one with the highest performance, achieving an exceptional F1-Score and Accuracy result of 98.9.

Keywords:

Equipment Lifecycle Management, Asset Lifecycle Management, Predictive Maintenance, Time series, Machine Learning, Deep Learning, Big Data, Analysis, Railways, Metro Rail.

Table of Contents

<i>Declaration</i>	<i>i</i>
<i>Acknowledgement.....</i>	<i>ii</i>
<i>Abstract</i>	<i>iii</i>
<i>List of Figures</i>	<i>vi</i>
<i>List of Tables</i>	<i>viii</i>
<i>List of Abbreviations.....</i>	<i>ix</i>
1.0 INTRODUCTION	1
1.1 Background	1
1.2 Problem Statement	5
1.3 Significance of the research	7
1.4 Research Aim and Objectives.....	8
1.4.1 Overall Aim.....	8
1.4.2 Objectives	8
1.5 Research Hypothesis	9
1.6 Research Questions	10
1.6 Scope of Study.....	11
1.7 Thesis Organization	12
2.0 LITERATURE REVIEW	13
2.1 Key Concepts, Theories and Studies	13
2.1.1 Systematic Review Method	14
2.1.2 Key Theories and Studies	15
2.1.3 Related Work and Research Findings	17
2.3 Key Debates and Controversies	23
2.4 Gaps in Existing Knowledge	24
3.0 RESEARCH METHODS AND TECHNIQUES.....	26
3.1 Research Design	26
3.2 Research Philosophy.....	29
3.2.1 Ontology.....	29
3.2.2. Epistemology.....	29
3.3 Research Methodology.....	30

3.3.1 Business Understanding:	31
3.3.2 Data Understanding	33
3.3.3 Data Preparation	45
3.3.4 Data Modelling	50
3.3.5 Evaluation	55
3.3.6 Deployment	58
4.0 IMPLEMENTATION ANALYSIS	59
4.1 Data Analysis Tools	59
4.2 Data Analysis Technique	60
4.3 Statistical Analysis	62
4.3 Discussion of Results and Analysis.	65
4.3.1 Hypothesis Testing Discussion:	65
4.3.2 Analysis Discussion	65
4.4 Potential Challenges and Mitigation Strategies	75
4.5 Research Considerations	78
4.5.1 Practical Implications	78
4.5.2 Ethical Implications	78
5.0 CONCLUSION	80
5.1 Answering Research Questions	81
5.2 Limitations/ Future Works	83
5.3 Research schedule	84
REFERENCE	86
BIBLIOGRAPHY	i
APPENDIX A – GANTT CHART	iii
APPENDIX B – R SCRIPT	x

List of Figures

Figure 1: Asset Lifecycle Process	2
Figure 2: Research Design	27
Figure 3: CRISP-DM	30
Figure 4: Gantt Chart	33
Figure 5: MetroPT-3 Dataset (Davari et al., 2021)	34
Figure 6: Load the dataset	34
Figure 7: Feature Description (Davari et al., 2021)	35
Figure 8: Data Summary	36
Figure 9: No of rows	36
Figure 10: No. of columns	36
Figure 11: Correlation Matrix	37
Figure 12: Normalization Code	37
Figure 13: TP2 Distribution	38
Figure 14: TP3 Distribution	38
Figure 15: H1 Distribution	39
Figure 16: DV_Pressure Distribution	39
Figure 17: Reservoirs Distribution	40
Figure 18: Oil Temperature Distribution	40
Figure 19: Motor Current Distribution	41
Figure 20: COMP Distribution	41
Figure 21: DV_Pressure Distribution	42
Figure 22: TOWERS Distribution	42
Figure 23: LPS Distribution	43
Figure 24: PRESSURE SWITCH Distribution	43
Figure 25: OIL LEVEL Distribution	44
Figure 26: Caudal Impulses Distribution	44
Figure 27: Missing Values	45
Figure 28: Outlier Code	46
Figure 29: Timestamp Boxplot	46
Figure 30: TP3 Boxplot	47
Figure 31: Oil Temperature Boxplot	47
Figure 32: Failure Information (Davari et al., 2021)	48
Figure 33: Feature Engineering Code	48
Figure 34: New Column "Aircompfail"	48
Figure 35: Class Distribution	49
Figure 36: Hybrid Data Sampling Code	49
Figure 37: New Class Distribution	49

Figure 38: Feature Selection Code	50
Figure 39: Features Selected	51
Figure 40: Train_Test_Split Code	51
Figure 41: Train Dataset	51
Figure 42: Test Dataset	51
Figure 43: ML Libraries	52
Figure 44: Logistic Regression Code	52
Figure 45: Decision Tree Code	53
Figure 46: Decision Tree	53
Figure 47: Random Forest Code	54
Figure 48: GBM Code	54
Figure 49: DL Libraries	55
Figure 50: NN Code	55
Figure 51: Evaluation Code	57
Figure 52: Model Evaluation	57
Figure 53: Correlation Coefficient 1	62
Figure 54: Correlation Coefficient 2	63
Figure 55: Correlation Coefficient 3	63
Figure 56: Correlation Coefficient 4	64
Figure 57: Metro4 Evaluation	66
Figure 58: Metro4 RF Accuracy	67
Figure 59: Analogue Sensor Data	67
Figure 60: Analogue Sensor Accuracy	67
Figure 61: Analogue Model Accuracy	68
Figure 62: Digital Sensor Data	68
Figure 63: Digital Model Accuracy	69
Figure 64: Digital LOG Accuracy	69
Figure 65: Evaluation Data frame	70
Figure 66: Analogue Hybrid Model	73
Figure 67: Hybrid Model Performance	73
Figure 68: Model Comparison	74
Figure 69: OECD Principles	79
Figure 70: GDPR Framework	79
Figure 71: Gantt Chart Header	iii
Figure 72: Gantt Chart Page 1	iv
Figure 73: Gantt Chart Page 2	v
Figure 74: Gantt Chart Page 3	vi
Figure 75: Gantt Chart Page 4	vii
Figure 76: Gantt Chart Page 5	viii

Figure 77: Gantt Chart Page 6	ix
-------------------------------------	----

List of Tables

Table 1: Literature Review Table	17
Table 2: Model Comparison.....	71
Table 3: Challenges and Mitigation	75
Table 4: Research Question and Answers	81
Table 5: Research Schedule	84

List of Abbreviations

ML	–	Machine Learning
DL	–	Deep Learning
AI	–	Artificial Intelligence
PdM	–	Predictive Maintenance
PdMs	–	Predictive Maintenance System
ALM	–	Asset Lifecycle Management
ELM	–	Equipment Lifecycle Management
LOG	–	Logistic Regression
DT	–	Decision Tree
RF	–	Random Forest
GBM	–	Gradient Boosting Machine
NN	–	Neural Network
CPS	–	Cyber-Physical Systems
DSS	–	Decision Support System
EDA	–	Exploratory Data Analysis
OECD	–	Organization for Economic Cooperation and Development
GDPR	–	General Data Protection Regulation
APU	–	Air Pollution Unit

1.0 INTRODUCTION

1.1 Background

Across different sectors, the integration of big data solutions and technologies have become increasingly important leading to rapid increase in infrastructures to support daily business operations. Industries have become to recognise the growing necessity of optimizing asset and equipment lifecycle to ensure a continuous, effective and efficient operation by minimizing business and asset failure and downtimes (Gonzalez-Prida et al., 2022). This need has led to the use of predictive maintenance, a data-driven approach offering a proactive solution to addressing equipment-related failures or issues. It leverages advanced big data analytics solutions to identify potential failures by analysing historical data before they lead to high cost breakdowns (Coandă et al., 2020)..

The integration of big data technologies has led to a change in operational practices across different industries, leading to the use of large infrastructures to support the influx of data being collected and the way it is processed. Effective infrastructure development has thus become the cornerstone of most sectors such as the manufacturing sector, healthcare sector, telecommunication sector, and much more which plays a critical position in ensuring operational efficiency and strategic decision making. As a result, it is imperative that there is an effective equipment lifecycle management which helps to ensure optimal operational performance, reliability and therefore reducing the downtimes experienced (Urbani et al., 2023), (Islam et al., 2024).

Asset Lifecycle Management (ALM)

Asset Lifecycle Management, otherwise known as ALM is the development and operational process business adopt to provide optimal value from the physical assets owned by the business. It is the process used by organisations to ensure a proper running of the business. ALM covers both physical assets and non-physical assets such as infrastructure, equipment's, the staffs, capital etc. owned by the companies (Campbell et al., 2016).

Asset lifecycle management spans several activities from systemic planning to equipment monitoring, and optimization of use from its moment of acquisition to disposal. For industries that makes use of large machineries and equipment’s, such as production and manufacturing plants, data centres, having an efficient equipment lifecycle management system is critical to their business processes (Löwen et al., 2019). This helps them to increase their productivity, reduce overhead costs, mitigate against risks and extend the lifespans of the equipment (West and Pascual, 2015).



Figure 1: Asset Lifecycle Process

Source: (Nandhakumar, 2023)

Figure 1 above shows an asset lifecycle process. It depicts how the Lifecycle process of the assets from the process of making a request for the infrastructure, to its acquisition, integration and adoption, its maintenance procedures put in place till its decommissioned for use.

Equipment Lifecycle Management

Equipment Lifecycle Management (ELM) is a subset of Asset Lifecycle Management as it falls under the physical assets. This refers to the entire cycle of acquisition to discard of an equipment or infrastructure being used (Löwen, Maier, H. Zhao, et al., 2019).

The Role of Predictive Maintenance in Equipment Lifecycle Management

The emergence of predictive maintenance has brought about a proactive approach in the way potential equipment failures are addressed considering the ever-growing complexity and mutual dependence with organizations involved in large scale infrastructure usage. Traditional maintenance approaches used are mostly reactive in nature as it considers the failure of the equipment's and infrastructure before they are fixed or changed. PdM takes a proactive approach which leverages advancements in big data technologies by incorporating data analytics and ML techniques to forecast likely failures. This approach monitors the equipment lifecycle from purchase to use and change, and ultimately ensures the assets and equipment's are properly used and maintained to increase the equipment lifespan therefore leading to a significant decrease in organisational business downtime and maintenance cost (West and Pascual, 2015).

PdM takes historical data of operations, analysing the trends, patterns and performance metrics of the equipment's and infrastructure in use which helps to identify and pinpoint early warning signs of equipment degradation.

The impact of predictive maintenance (PdM) across different sectors cannot be overstated particularly considering industries such as the manufacturing and industrial operation (Yan et al., 2017). While PdM is widely applicable to different settings such as hospitals, data centres, its significance reaches other sectors such as the metro train operations as by analysing data collected from critical components in a metro rails like compressors in Air Production Units (APUs), PdM techniques can be leveraged to optimize performance and mitigate risk.

Predictive Maintenance in Data centres.

Data centres which play a vital role in data collection, storage and processing. Reliability of the performance of the hard drives being used in the data centres is important as their business is impacted by the efficiency and continuity of usage of the hard drives, thereby making regularly scheduled maintenance a priority to optimize performance and less downtime (Amram et al., 2021a). PdM in the context of data centres involves using ML models to monitor and predict the health of the hard drive components, thereby preventing equipment failures (Dwork et al., 2012), (Copeland et al., 2015).

Predictive Maintenance in Metro Rails

Urban means of transportation such as the metro rail which is a major source of transportation in the United Kingdom is another similar business to the data centres. Metro rails are known for their high frequency, and capacity services connection one part of the city to another part which helps reduce traffic congestion and pollution.

Just like the data centres, Metro train systems depend on the seamless operation of the components of their system. It is important to ensure there is a regularly scheduled maintenance for metro rails systems to ensure its welfare and reliability. The implementation of a PdM system will ensure the smooth running of the rails which can enhance the reliability of the critical components such as the APU and minimize downtimes (Davari, Veloso, R. P. Ribeiro, et al., 2021), (Veloso et al., 2022), (Budai et al., 2006).

The Air Production Unit (APU) in metro trains which play a crucial role in maintaining the needed air pressure for different pneumatic systems, thus any damage to the APU can lead to a significant downtime in operation or loss of business.

This document seeks to tackle the challenges by developing an Equipment Lifecycle Management System (ELM) that leverages on predictive maintenance using ML and DL techniques. This document focuses on proactively identifying potential failures and ensuring regular maintenance is scheduled to prevent the failures of the equipment's.

1.2 Problem Statement

Asset Lifecycle Management (ALM) is a major discipline involving a structured set of activities to be performed by an organization in managing its assets and their respective performance, trends, and associated risks. The use of ALM aims to provide maximum derivation of value from the assets in use such as its equipment's and infrastructure. However, many organizations such as the data centres, metro rail companies, industrial plants, etc still struggle with the traditional maintenance approaches which are mostly reactive leading to prolonged downtimes, increased maintenance and operational cost. The traditional approaches do not account for the proper usage and lifespan of the equipment, or account for real-time performance data leading to increased maintenance cost by over maintaining the equipment's, or prolonged downtimes leading to operational loss (Paolanti et al., 2018). Most organizations making use of the conventional methods registered more equipment failures which in turn leads to more overhead cost, and a drop in brand reliability and its reputation. This drop in trust can lead to a loss of customers thereby reducing the organizations market value and financial stance.

Take the hospitals as an example, they make use of different critical assets such as the MRI scanners, CT scanners, etc, and without the right maintenance process put in place, these equipment's can one day encounter a failure during an emergency. Considering the machineries are vital for patient diagnosis and providing treatment, a damage to the equipment can result in downtime in providing quality healthcare services or worse lead to loss of lives, a risk in compliance with health regulation; increase in overhead cost of fixing or replacing damaged machinery and inefficient resource allocation.

Whereas the adoption of a predictive maintenance approach powered by IoT Sensors, machine learning, deep learning and enhanced analytics techniques offers a proactive or driven approach to help mitigate against these challenges. ML and DL techniques enable organizations to use historical data to foresee potential infrastructures failure, which helps to scheduling regular maintenance of the equipment's or replacements based on the suggested infrastructure lifespan. This process would in turn lead to a proper optimization of resources and increase operational efficiency. The proactive approach helps to ensure customer satisfaction, and results in an increase in market values.

Irrespective of the advantages of using the predictive maintenance approach, many organizations still struggle with optimizing and integrating the predictive maintenance techniques and solutions due to issues such as big data, system integration, accurate predictive maintenance techniques and the scalability of the available techniques and solutions in different industrial fields.

This report aims to review the pre-existing predictive maintenance techniques that have been used by other researchers, understand the techniques, and implement an advanced predictive maintenance technique that will enhance and optimize asset and equipment lifecycle management. Real-time data from the MetroPT-3 would be collected, and pre-processed before deploying advanced predictive maintenance techniques which would minimize downtimes, reduce maintenance and operational costs, and extend asset and equipment lifespan.

1.3 Significance of the research

The inspiration behind this report stems from a review of the everyday usage of equipment's. It is predicated upon the idea that equipment lifespan should be serviced regularly to ensure optimum usage of the machinery and equipment's possessed.

Another motivation behind this research became apparent having worked in an IT infrastructure support-based project with a consultancy firm. Being able to accurately predict maintenance schedules and equipment failures increases the competency of the consultant in charge as they can be quick in providing better support to their clients and service equipment as at when due and replace faulty equipment's which in turn reduces client downtime.

Some of the significance of performing predictive maintenance in Equipment lifecycle management include:

1. **Minimal downtime:** Using the predictive maintenance system to detect equipment failure, organizations can proactively take the necessary actions to address the issues encountered preventing abrupt downtime leading to business disruptions therefore leading to increased effectiveness and efficiency.
2. **Increased Lifespan:** Integrating the predictive maintenance system can help organizations adequately provide the necessary checks and balances for their equipment. By regularly monitoring the performance of the equipment, regular maintenance can be scheduled therefore increasing the lifespan of the assets.
3. **Resource Allocation Optimization:** The introduction of the predictive maintenance system allows organizations to properly divert their attention of the assets that require optimum attention. The system gives insights into asset health therefore ensuring longevity of equipment, enabling scheduled repairs and replacements when needed.
4. **Improved Decision Making:** Using ML algorithms and visualization tools, organizations can proactively make informed decisions based on the insights derived.

1.4 Research Aim and Objectives

1.4.1 Overall Aim

The aim of this report is to investigate, build up and integrate existing technologies into industrial use that enhance Asset and Equipment Lifecycle Management System by implementing a predictive maintenance system that makes use of varying machine learning and deep learning techniques to optimize resource usage, reduce operational maintenance cost and reduce system downtime.

1.4.2 Objectives

After extensively reviewing the dataset and various literature, some of the objectives of the documentation are highlighted below.

1. Organize a comprehensive literature review to analyse and synthesise various technological advancements in big data, ML, DL, AI, time series forecasting and its impact in optimizing Asset and Equipment Lifecycle Management across various industries.
2. Identify the issues and limitations about the making use of traditional reactive maintenance approaches and exhibit how predictive maintenance approach can be used in optimizing Asset and Equipment Lifecycle Management.
3. Design, develop, and implement a model system infusing different advanced machine learning and deep learning techniques in predicting potential asset and equipment failures.
4. Evaluate different iterations of the MetroPT-3 dataset, assess performance of the ML and DL models used, identify their merits and weaknesses, and suggest areas for future advancements or considerations.

1.5 Research Hypothesis

The idea of this project is to develop a ML and DL model to be used for forecasting purposes. After defining the goal and analysing the dataset collected, the following hypothesis was conceived.

1. Null Hypothesis: There exists a significant correlation between TP2 and DV_Pressure.
Alternative Hypothesis: There is no significant correlation between TP2 and DV_Pressure.
2. Null Hypothesis: There exists a significant correlation between MPG and Pressure Switch.
Alternative Hypothesis: There is no significant correlation between MPG and Pressure Switch
3. Null Hypothesis: There exists a significant correlation between H1 and TP2.
Alternative Hypothesis: There is no significant correlation between H1 and TP2.
4. Null Hypothesis: There exists significant correlation between TP2, DV_Pressure and Aircompfail.
Alternative Hypothesis: There is no significant correlation between TP2, DV_Pressure and Aircompfail.

1.6 Research Questions

The following research questions have been meticulously formulated and will be addressed in this study.

Some of the proposed research questions for this research include:

1. What contemporary ML and DL techniques are prevalent in addressing Asset and Equipment Lifecycle Management via Predictive Maintenance Systems?
2. What equipment features and attributes of equipment are the most indicative for forecasting potential failures in Metro Railways?
3. How can predictive maintenance models be optimized using feature engineering, data sampling and feature selection to accurately forecast equipment failures?
4. Which ML or DL modelling technique is most effective for predicting equipment failures and what performance metric should be prioritized in evaluating models?
5. How will the implementation and integration of a predictive maintenance system benefit Metro Railways and other industries and what impact would it have on their decision-making process?

1.6 Scope of Study

The focus of this thesis comprises of:

1. Reviewing books, journals, articles, etc and any other form of publication relating to proposed topic.
2. Conduct a detailed analysis of technologies and methodologies adopted in previous research to identify the best methods that can be adopted for this thesis.
3. Explore the application of different machine learning and deep learning techniques to determine those that produce the optimal results.
4. Identify potential limitations encountered during the study and suggest features to be considered for future research.

1.7 Thesis Organization

The thesis is organized as follows:

1. Chapter 1 - Introduction: This provides a background information on the study, speaks about the problem statements, highlights the significance of the research, its aim, objectives, while defining the research hypothesis, questions and project scope.
2. Chapter 2 - Literature Review: This provides insight into the past research done in this field or related field of study. It speaks on the approach used to perform the literature review, the findings, the algorithms, methodologies, techniques and limitations of the previous research conducted.
3. Chapter 3 - Research Methods and Techniques: The section centres on the research philosophy, methodology and processes used in achieving the required results.
4. Chapter 4 – Implementation Analysis: This section of the thesis focuses on the implementation procedures used. It involves building models, testing the models and checking for their levels of accuracy. It also focuses on discussing about the results of the experiments confirming the hypothesis and providing answers to some research questions while also providing insight into the challenges associated with the project, its mitigation strategies, and the research considerations.
5. Chapter 5 - Conclusion: This section speaks on the results of the research. It summarizes the experiments conducted, the process used, the results obtained during this experiment, and limitations of the project while also providing answers to the research questions.

2.0 LITERATURE REVIEW

2.1 Key Concepts, Theories and Studies

Machine Learning (ML)

Machine Learning is the use of computer-based algorithms and models to model human behaviour. It refers to the ability of a machine or computer to learn without being explicitly told to do so (Shanthamallu and Spanias, 2022).

Deep Learning (DL)

Deep learning is another branch of machine learning very similar but however uses models and algorithms to review and learn from its past mistake and errors. It uses a layer of neural networks running different algorithms while learning from its previous iterations (Shanthamallu and Spanias, 2022), (Goodfellow et al., 2023).

Time-Series Forecasting

Time-series forecasting entails the use of statistical analysis and ML techniques in making predictions on historical data with different data points (Waheeb et al., 2019), (Sujjaviriyasup and Pitiruek, 2017), (Hasri et al., 2023)

Predictive Maintenance

Predictive maintenance refers to the act of identifying operational issues, challenges and potential equipment or infrastructure failures in timely manners therefore helping to adequately schedule maintenance processes and timely repairs of damaged equipment's. PdM's approach is to help minimize cost over a real time-based preventive process (Ersöz et al., 2022).

2.1.1 Systematic Review Method

This segment of the report adopts a systematic review method. The Systematic review process is a comprehensive process of selection that involves researching relevant journals, articles, books, web pages, etc on ML, AI, and Big data in Predictive Maintenance for Equipment Lifecycle Management (Carvalho et al., 2019), (Cheng et al., 2022). It examines various keywords extracted from the subject topic cross-checking it in different fields and areas to find relevant works of knowledge where information can be obtained from. A list of materials is collected and sorted through to help address the research questions and meet the specified research objectives.

Some key concepts discussed during this study are predictive maintenance, machine learning, assets management, equipment management, data centres, metro rails, etc. These concepts were thoroughly researched on the internet using the keywords, various materials were collected from sources like Google Scholar, Bolton Library, IEEE Explore library, Google Internet search, AI tools for search, and in-depth explanations.

The review involves a detailed and exhaustive process of researching directly and non-directly related materials, extensively reviewing relevant literatures extracted with an aim to understand how best the knowledge in those literatures and documented experiment can help or be useful to the study.

This report emphasizes the importance of the application of predictive machine learning techniques in equipment maintenance, exploring the methodologies and how they can be used to enhance the infrastructure lifespan across various industries. It includes researching on the various machine learning and time series technologies adopted and how to check for their levels of accuracy in forecasting (Amram et al., 2021a). It also involves a mixed research approach as it combines quantitative and qualitative exploration and making relevant deductions from the observations (Daniel, 2016), (Beinschroth, 2022), (Lee et al., 2024), (Dayo-Olupona et al., 2023a).

2.1.2 Key Theories and Studies

In this section of the report, studies that have been deemed relevant to the research topic are reviewed, their implications, strengths and weaknesses are analysed.

Taking (Kaparthi and Bumblauskas, 2020) expatiates on the role of predictive maintenance of equipment. It goes in-depth into reviewing the machine learning techniques adopted and making references to which has better levels of accuracy in predicting failures of which (Amram et al., 2021b) also discusses more about.

(Campbell et al., 2016) provides a comprehensive insight into the contemporary issues in maintenance management, with emphasis on terms, requirements and methodologies. It talks about the strategic importance of maintenance practices in businesses as it improves operational effectiveness and efficiency.

(Davari, Veloso, R. P. Ribeiro, et al., 2021) speaks on predictive maintenance and anomaly detection framework for metro rails using the Autoencoders and Sparse Autoencoder by aggregating the data gotten from the analogue and digital sensor data and providing the results. It also compared the performance of the models against a variational autoencoders using Precision, Recall, F1-score and Accuracy.

(Barros et al., 2020) comes up with an innovative approach of using real time sensor data placed on the APU with rules based on the analysis of peak frequency which records the normality and abnormality of a sensor leading to its failure while (Lee, 2020) uses logistic regression classifier to model the possible behaviour of an APU used in detecting leakages in a trains brake system.

(Chen et al., 2019) talks about using the Long Short-Term Memory (LSTM) based approach for predicting metro rail compressor failures using the aggregated sensory data. It compares the LSTM results against a Random Forest approach and concludes that although Random Forest performs better, the LSTM approach shows a more stable results over extended periods of time which in turn leads to better decision making.

(Wang, 2021) talks about advanced LSTM approach using Internet of Things (IoT) and Cloud computing technologies that overcomes the challenges observed in data distribution diversification

and lack of labelled anomaly data observed. The proposed approach is said to have been tested in a live metro rail environment giving a better understanding and failure detection result.

Deep learning has also been used and shown promising results in weather forecasting due to its ability to capture complex nonlinear relationships. Its focus on machine learning fault diagnosis is significant however it might also be considered a weakness as it limits the depth of the analysis conducted (Zhu et al., 2023). (Chen et al., 2020) also talks about using a new approach called COX proportional hazard deep learning (CoxPHDL) which is used to address the challenges of data sparsity and censoring. This approach is used by industrial plants to reduce downtimes and costs associated with maintaining it.

Another suggested method to use is the Hilbert-Huang Transform or HHT, which is a powerful tool for analysing nonlinear relationships and non-stationary time series data. This was used for the time-series forecasting due to its ability to draw out valuable information from provided data and its accuracy in prediction, however it doesn't address limitations that the HHT method might have in non-meteorological forecast application (Huang and Wu, 2008).

2.1.3 Related Work and Research Findings

The table below gives an overview of the literature reviewed during the project.

Table 1: Literature Review Table

Source	Relevance	Methodology	Details	Strengths	Weakness	Impact
(Alamr and Artoli, 2023)	4	Experimental – Transformers encoder	Transformer encoder anomaly detection using ECG5000 and MIT-BIH Arrhythmia datasets.	Outperforms other traditional models by achieving high levels of Accuracy, Precision and F1-score.	Lower performance on MIT-BIH Arrhythmia dataset.	Transformer models offer improved performance levels for detecting anomalies.
(Zhao et al., 2020)	4	Experimental - Holt-Winters (HW) method and LSTM Networks	IT makes use of a hybrid prediction time series decomposition model in passenger flows and different urban railway traits.	Enhanced accuracy superior prediction performance compared to other traditional models	Increase in model complexity and need for more computational power and high expertise.	It improves the short-term prediction for the urbanized railway system
(Amram et al., 2021b)	4	Experimental – Logistic Regression,	Initiates various ML models and checks its	Improves prediction accuracy which can be	Sole focus on decision tree technique.	Adopts a strategic and scalable solution to detecting failures

		Random Fores, Decision Tree	performance in predicting failures.	integrated into various systems.		using decision tree algorithms.
(Chauhan and Vig, 2015)	5	Experimental – Deep RNN with LSTM	Predicting ECG behaviours and detect anomalies observed from the predictions.	LSTM doesn't require previous knowledge of abnormal signs and makes it easy to process ECG.	Doesn't account for a wide variety of arrhythmias and possibly require extensive validation of various datasets and arrhythmias.	Limited preprocessing required in automating ECG detection and detecting anomalies.
(Gupta, 2023)	5	Experimental - Autoencoders	Makes use of sparse and variational autoencoders on the digital and analogue sensor data.	Increase in performance levels of detecting anomalies and a reduction of false alarms.	Limits in exploring analogue dataset and comparing other models on the datasets involved.	Shows the importance of DL in predictions and detecting anomalies.
(Kanawaday and Sane, 2017)	4	Experimental – ARIMA Model	Predictive modelling using ARIMA modelling with timestamps	Effectiveness of timestamp in optimizing manufacturing processes.	Sole reliance on ARIMA model.	Talks about the importance of IoT ML techniques in optimizing

						manufacturing processes.
(Pan et al., 2023)	4	Experimental - CNN	Using CNN for a Multivariate database for anomaly detection	Handling anomaly detection in structural health monitoring (SHM).	Requires extensive well labelled bridge dataset	Improves the performance of detecting anomalies in SHM for bridges.
(Berroukham et al., 2023)	3	Review	Comparison of different models, available datasets, and evaluating existing works with performance metrics	Structured detailed comparison of different models and methods used.	Anomaly detection in videos can be regarded as subjective.	Advancement in understanding the complexity in video anomaly detection.
(Li and Jung, 2023)	4	Experimental – LSTM and Autoencoders	LSTM and Autoencoders are used to identify abnormality of timepoints, intervals, and series	Structured comprehensive approach making use of diverse DL techniques used in addressing complex time-series data.	Interpreting the DL models such as LSTM and Autoencoders used and its results can be complex.	Contribution to the field of anomaly detection.

(Tatineni, 2020)	3	Review	Uses IoT and ML in predictive maintenance in manufacturing industries.	Increased reliability and minimized downtime.	Experimentation needs to be done to confirm effectiveness.	Enhances data reliability, yield efficiency gains, and fosters cost savings
(Tran et al., 2021)	2	Review	A thorough review on the recent ML developments adopted in PdM where research is classified based on the algorithm.	Detailed explanation on different models and algorithms used and its how they have improved over the years.	Experimentation needs to be done to confirm effectiveness.	Improves equipment reliability, reduces maintenance costs, and extends equipment lifespan through proactive maintenance strategies based on predictive insights.
(Achouch et al., 2022)	3	Review	Talks about a novel approach that makes use of asset management and smart maintenance practices that help in	Novel approach that integrates different sensors and maintenance approaches used for different machineries	The research is still relatively new and has a lot of ground to cover and practical use is not that applicable yet. Generalized findings	Offers a complete view into the predictive maintenance and decision-making process.

			point of failure detection and support decision making process.		might not be applicable to all industries.	
(Abusitta et al., 2022)	4	Experimental – Denoising autoencoder, Logistic Regression Classifier	Speaks on the detecting anomalies in different heterogeneous environments	Improved anomaly detection with improved accuracy levels	Requires extra data validation.	Offers an advanced approach in detecting anomalies.
(Carvalho et al., 2019),	3	Experimental	Talks about early fault detection in in equipment using specific predictive maintenance techniques.	Specialized model for reducing downtimes in hard drives	No extensive validation of model selected.	Targeted solution for enhancing hard drive lifespan.
(Abhari, 2024)	3	Experimental - XGB	XGB used as preferred model for engine detection.	Increase accuracy using ML detection	Insufficient anomaly recognition for different data sources	Unified system for anomaly and intrusion detection.

				algorithm to fuse the different data sources.		
(Cheng et al., 2022).	3	Review	Talks about the effect of predictive maintenance in the manufacturing field.	Provides a specialized predictive maintenance approach used in manufacturing sectors	Insufficient studies conducted	Offers detailed insight into predictive maintenance in the manufacturing sector and talks about area of future works.
(Kaparthi and Bumblauskas, 2020)	4	Review	The use of predictive maintenance in optimum maintenance decision making in the service management industry	Insight into using predictive maintenance for an agricultural company interested in reducing their downtime.	Limited discussion on practical implementation considerations or scalability issues.	Improvement on the proposed strategies used in predicting equipment failures and therefore minimizing downtimes.

2.3 Key Debates and Controversies

There may be conversations regarding the ethical implications surrounding the extensive use data decision making and safeguards put in place against them.

It can also be argued that there is an excessive reliance on predictive models which can lead to neglects of other equipment related management activities. While this is possible, the solution being developed is to be used in conjunction with human expertise. It is of benefit to have a model that helps predict failures in equipment which workers in adequately doing their jobs.

2.4 Gaps in Existing Knowledge

The current body of literature in predictive maintenance highlights several gaps that necessitate further exploration. One significant area of improvement is the integration of additional sensor data, environmental data, and human error metrics into predictive maintenance models. These factors are crucial for accounting for the various conditions that might influence equipment failures, thus enhancing the accuracy and reliability of predictions. Presently, many models do not sufficiently incorporate these variables, which can lead to incomplete assessments and suboptimal maintenance strategies.

Including a broader range of data sources would allow for a more nuanced understanding of the factors contributing to equipment failures. For instance, environmental conditions such as humidity, dust levels, and temperature fluctuations can have substantial impacts on equipment longevity and performance. By integrating these elements into predictive models, it is possible to develop a more holistic view of the operational environment and its effects on equipment. This comprehensive approach can lead to more accurate predictions and timely interventions, ultimately reducing downtime and maintenance costs.

Moreover, the existing literature often overlooks the potential application of advanced predictive maintenance models across different sectors and use cases. The versatility of these models can be significantly enhanced by testing and validating them in diverse industrial contexts. Different sectors may present unique operational challenges and failure patterns giving insights into the generalizability and robustness of predictive models. For example, predictive maintenance techniques proven effective in the manufacturing sector might also offer benefits in healthcare, transportation, or energy industries.

Future research should prioritize these areas to develop more comprehensive and versatile predictive maintenance solutions. By integrating a wider range of data and exploring applications in different contexts, researchers can pitch in to the development of more fruitful and universally applicable maintenance methodologies. This expansion not only enhances the precision of failure predictions but also ensures that the developed models are robust enough to be deployed across various industries with different operational dynamics.

In summary, addressing these gaps in existing knowledge is crucial for advancing the field of predictive maintenance. Incorporating more sensor data, environmental variables, and temperature metrics, along with exploring applications in diverse sectors, can significantly improve the accuracy and applicability of

predictive maintenance models. Such advancements will ultimately lead to more efficient maintenance practices, reduced downtime, and optimized operational performance across a range of industries.

3.0 RESEARCH METHODS AND TECHNIQUES

3.1 Research Design

Research design is the strategy or framework adopted by a researcher while conducting a study. It refers to the various techniques, models, and techniques that will be used during a research process to provide answers to the questions. A good research design covers the hypothesis and research questions, process through which data is gathered, the experiments to be done, models and techniques to be used, and method of evaluating and interpreting the proffered results. It is a blueprint of steps carried out which involves the amalgamation of various steps adopted to reach a resulting and beneficial conclusion (Marczyk et al., 2005), (Ogu et al., 2016), (Khanday and Khanam, 2023).

The research design process adopted in this section of the report is the mixed research design approach. The mixed research design approach combines both the qualitative analysis and quantitative exploration (Daniel, 2016), (Kandel, 2020). The use of numerical data, extracting insights and use of statistical analysis are all part of the quantitative analysis method. Quantitative analysis makes use of data collected and explores the patterns, trends and relationships between the variables in the data collected which it uses to derive insights used for decision making.

On the other hand, the qualitative approach follows a thorough process of understanding the past methods used in a similar field, or contemporary methods adopted. The qualitative approach involves identifying previous methods adopted by other researchers by undergoing a systematic review for a detailed presentation or interpretation (Beinschroth, 2022), (Lee et al., 2020), (Dayo-Olupona et al., 2023b).

The figure below gives a depiction of the research methodology proposed for this project. It gives a brief visual breakdown of the models to be used, and the modes of evaluation adopted.

It shows a breakdown of the activities and tasks of each phase right from the business understanding to System Deployment.

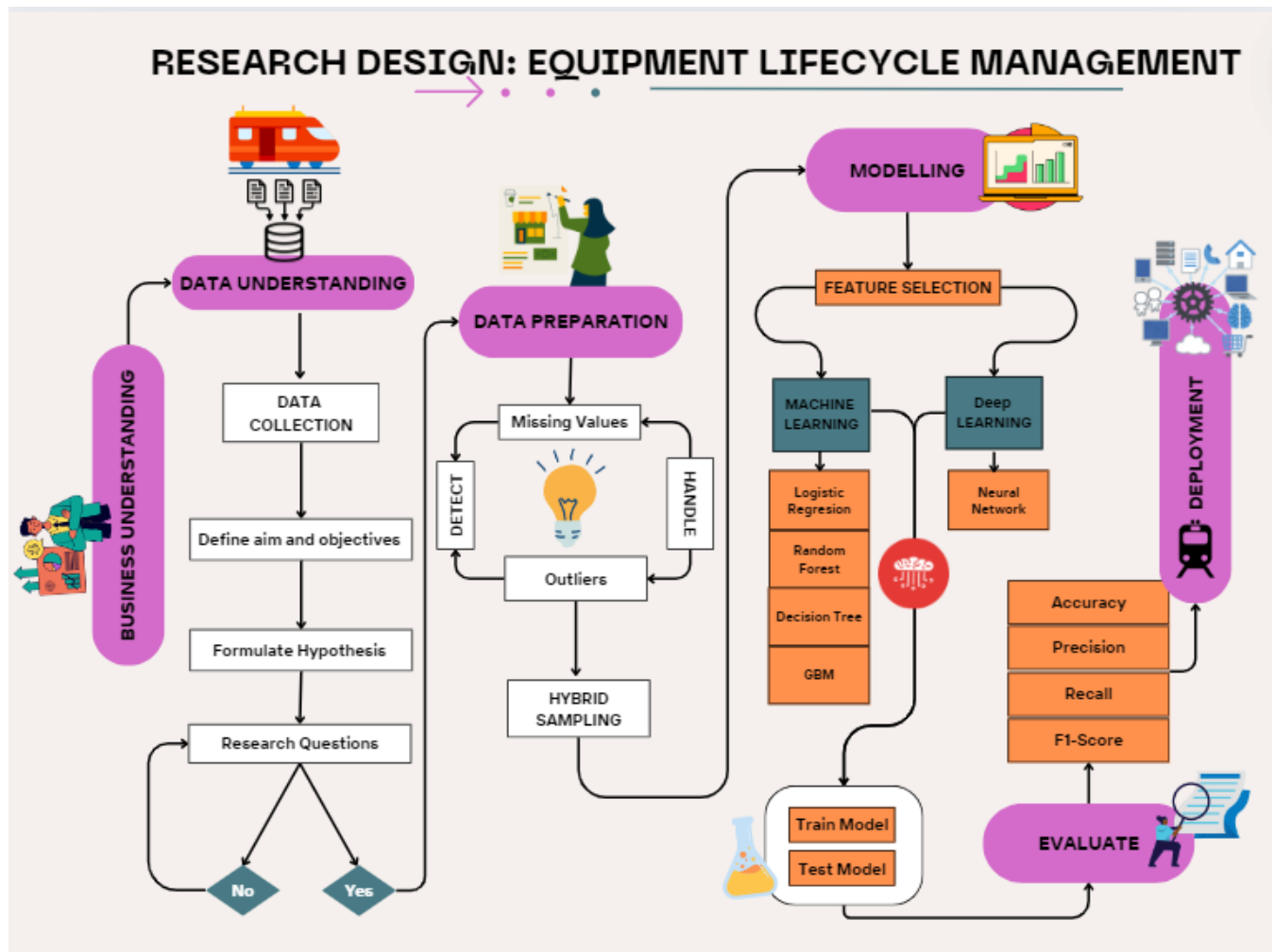


Figure 2: Research Design

Research Design Process

Detailed below are some of the research design processes adopted during this report as shown in the figure above.

1. **Data collection:** This refers to the process involved in collection of data and what type of data is being collected. Historical data containing analogue and digital sensor data will be collected from Metro PT-3. The sensors were installed on the train compressor and was collected in real-time from February 2020 to august 2020. It contains 15 columns and 1516948 rows from an APU, which includes readings of pressure, temperature, motor current, and air intake valves, providing valuable insights for predictive maintenance.

2. **Data pre-processing:** Data pre-processing is vital in data analysis as this is the process that ensures data reliability and usability. Some of the steps performed in this section involve handling missing values, checking for outliers, feature generation if needed and performing data sampling techniques (Zhenhui, 2020).
3. **Feature selection:** Due to the volume of the dataset which comprises multiple columns and rows, the relevant features, and attributes necessary would be identified for the predictive modelling. The step of feature selection is applied to both the analogue sensor data and the digital sensor data to select the best features with the best predictive capabilities (Sharma and Gupta, 2023).
4. **Model development:** This is another crucial step involving initiating different machine learning models such as logistic regression, decision tree, random forest, etc and running a test using the models. After running the machine learning models, neural network deep learning model is initiated (Musa et al., 2023).
5. **Evaluation:** The performance of the implemented models for each iteration of the dataset will be gauged using metrics such as accuracy, recall, precision, and f1-score.
6. **Integration:** Upon finding the models that improve the levels of accuracy, it would be seamlessly integrated into other Equipment Lifecycle Management Systems or ALM systems.

3.2 Research Philosophy

Research philosophy refers to the view of the researcher about how data is collected, analysed, used, and adopted. A research philosophy correlates with the assumptions a researcher has about a work, the knowledge base involved, and the nature of the research to be carried out (Bergmann, 2024), (Mbanaso et al., 2023). The key constituents of the research philosophy include ontology, epistemology, and methodology.

3.2.1 Ontology

Ontology refers to the researcher's view of the world and the theories and assumptions deduced based on worldly nature and reality (Keet, 2018). There are four schools of ontology which are: Realism; Internal Realism; Relativism; and Nominalism. In this research document, the researcher's ontology view takes a realism and relativism school of thought as it adopts examining and observing the worldly view of the researcher.

3.2.2. Epistemology

Epistemology refers to the preferred way the researcher decides to investigate the worldly view and reality (Sol and Heng, 2022). This research document prioritizes inquiries, hypotheses, and scientific evidence which in turn aligns with the positivist school of thought which believes in scientific studies and objectivity.

The research philosophy approach being adopted for this methodology is the realism approach as it's a quantitative based approach which involves performing experiments and making inference based on the results of the experiments.

3.3 Research Methodology

Research Methodology refers to the adopted framework of techniques and principles adopted by the researcher as properly elucidated below (Dawson, 2019). It spans the entirety of the project from the formulation of the idea, development of project schedule, formulation of research hypotheses and questions, initiating the models and techniques to use and deriving insights to be reported.

CRISP-DM Methodology

“Cross Industry Standard Process for Data Mining” is regarded as a standard data science approach adopted to provide structure to the planning process involved in the projects, organizing of the project, and its implementation of the project (Hotz, 2018), (Schröer et al., 2021), (Cazacu and Titan, 2020). The figure below elucidates the 6 phases involved in CRISP-DM Methodology from start to finish. It is an agile development process for data mining and shows how each phase related to one another.

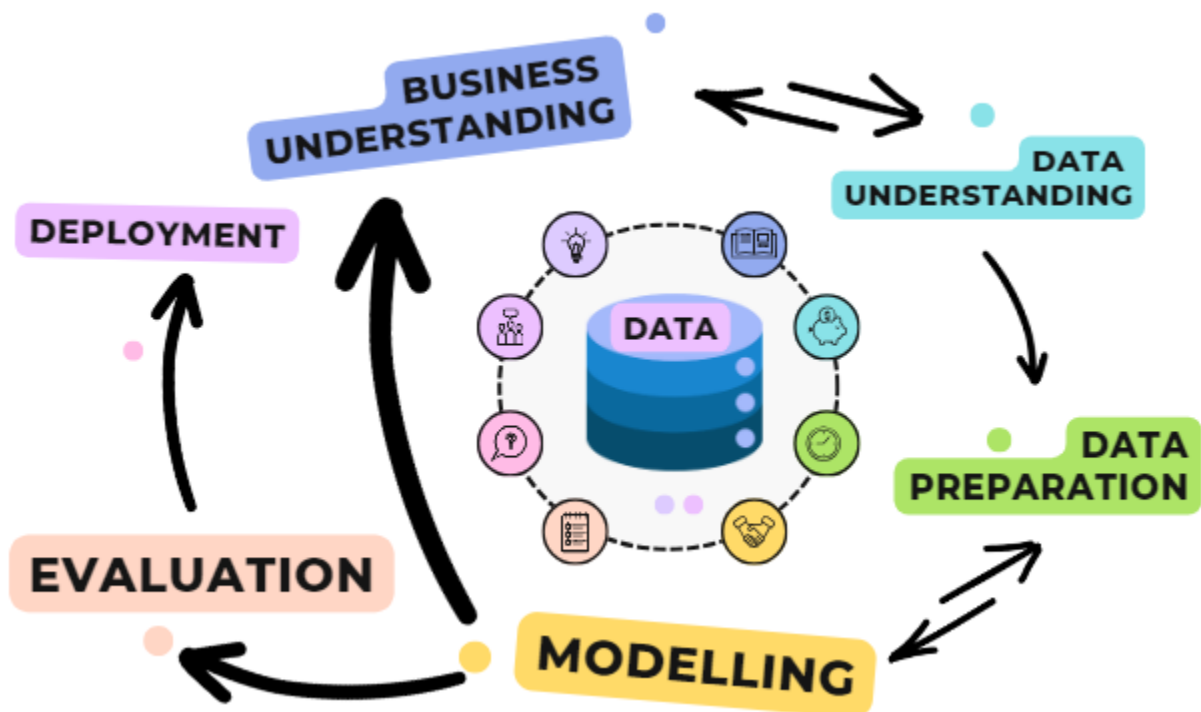


Figure 3: CRISP-DM

3.3.1 Business Understanding:

Business understanding refers to having critical understanding of the business problem, its aims and objectives, and the proposed forecasting goal. With a focus on Metrorail in this documentation, the industry problem has been identified, its aim and objectives have been deduced and the expected outcomes have been suggested.

3.3.1.1 Stakeholder Management

The stakeholder management confers about the project stakeholders, their respective roles and how they influence the project.

1. **Researcher - Primary Stakeholder (Project Owner)**

Impact: participation in improving the Predictive maintenance field and establishing a good reputation for a successful and novel approach

2. **University of Bolton - Primary Stakeholder (Project collaborator)**

Impact: Direct participation in ML development and establishing a good reputation via mentorship of researcher and a successful partnership in publication.

3. **Metro Railway System - Secondary Stakeholder (Direct beneficiaries)**

Impact: improved PdM would be integrated into the metro rails to improve failure detection and help schedule proper maintenance for its assets and equipment's.

4. **Manufacturing Industry - Secondary Stakeholder (Direct beneficiaries)**

Impact: will adopt the existing system used by metro rails and integrate the solution into other industrial sectors.

5. **Project Supervisor - Secondary stakeholder (Project leadership)**

Impact: in charge of ensuring the project timeline is adhered to, implementation procedures and guidelines are followed as well as taking credit for successful completion of project.

3.3.1.2 Project Schedule

One of the most important parts of undergoing a study is the project schedule. This is critical to the business understanding as a project schedule is needed when making plans on how the project will be executed. It shows the proposed timeline, actions and agreed deliverables while assigning tasks to those involved.

Figure above indicates an illustration of the proposed project methodology, schedule and its timeline which has been broken down into four phases namely: Prepare Phase; Explore phase; Design Phase; and Report phase. Each data science process has been designed to fit into one or more phase as this adequately helps ensure project planning and adherence to the project timeline.

1. **Prepare Phase:** The prepare phase is the first phase that takes four days of the project as it involves the steps of choosing a research topic, finding a base paper to support the research topic picked, approving the selected research topic and the data collection process.
2. **Explore Phase:** The explore phase consists of the data understanding step which takes 6 days. This involves the process of formulating hypothesis from the data collected, formulating research questions, performing extensive literature search and review, and approving the research questions. The explore phase is one of the most important parts of the project as it involves the literature review. Literature review is major in a project as it explores the proposed field, contributions made and areas of future work available. It ultimately helps narrow down the extent of the research to be done.
3. **Design Phase:** The design phase is also known as the implementation phase which will last for 24 days. This involves data preparation and initiating the selected methodologies. The data preparation step includes handling missing values and checking for outliers, before moving forward to perform data sampling and feature selection. Upon concluding the data preparation step, we initiate the ML and DL techniques and move on to train and test the models. The result of the model is evaluated and if better accuracy is achieved, it is accepted and deployed.

4. **Report:** The Report Phase involves documenting the results from the experiments concluded which will last for 10 days. This step involves documenting the outcomes observed from all the previous phases, answering the hypothesis and research questions.

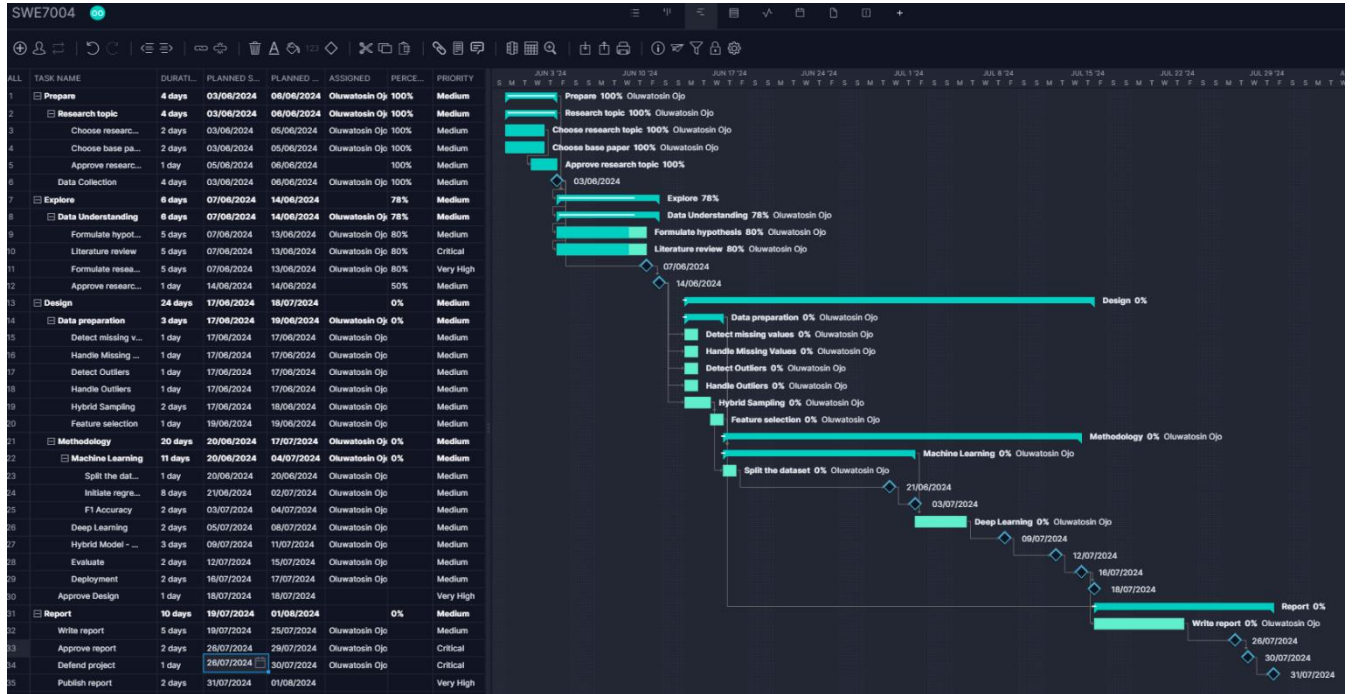


Figure 4: Gantt Chart

3.3.2 Data Understanding

Data understanding as the name suggests refers to having an in-depth understanding of the dataset collected or provided (Schröer et al., 2021). Data exploration is done in this step of the project as it involves cross referencing the variables with one another and verifying the quality and authenticity of the dataset. The dataset is provided on the MetroPT-3 platform with over a million rows and 16 columns. The columns contain a timestamp with analogue and digital sensor data.

Data Source: The data source for this study is solely obtained from MetroPT-3. This is a website dedicated to collecting real-time metro rail data and publishing the results. The MetroPT-3 data is a secondary data, however due to the reputation of the company in which the data was collected from makes me conclude on the reliability of the data provided (Davari et al., 2021)

MetroPT-3 Dataset		
Donated on 3/21/2023		
From a metro train in an operational context, readings from pressure, temperature, motor current, and air intake valves were collected from Unit (APU). This dataset reveals real predictive maintenance challenges encountered in the industry. It can be used for failure predictions, a		
Dataset Characteristics	Subject Area	Associated Tasks
Tabular, Multivariate, Time-Series	Computer Science	Classification
Feature Type	# Instances	# Features
Real	1516948	15

Figure 5: MetroPT-3 Dataset (Davari et al., 2021)

```
Load the dataset
```{r}
Metro <- read.csv("C:/MetroPT3(AirCompressor).csv", header = TRUE)

#DROP THE INDEX COLUMN X
Metro <- Metro[, !names(Metro) %in% "X"]

View(Metro)
```
```

Figure 6: Load the dataset

| s/n | Features | Description |
|-----|----------------------|--|
| 1 | TP2 | Compressor system pressure measure |
| 2 | TP3 | pneumatic panel pressure generated measure |
| 3 | H1 (bar) | Measures the pressure drop caused by the discharge from the cyclonic separator filter. |
| 4 | DV pressure (bar) | Measures the pressure drop generated when the air dryers in the towers discharge; a zero reading indicates that the compressor is operating under load. |
| 5 | Reservoirs (bar) | Measures the downstream pressure in the reservoirs, which should be close to the pneumatic panel pressure (TP3). |
| 6 | Motor Current (A) | Measures the current of one phase of the three-phase motor, with readings close to 0A when off, 4A when offloaded, 7A under load, and 9A at startup. |
| 7 | Oil Temperature (°C) | Measures the oil temperature in the compressor. |
| 8 | COMP | Indicates the electrical signal of the air intake valve in the compressor, active when there is no air intake, meaning the compressor is off or offloaded. |
| 9 | DV electric | Controls the compressor outlet valve; it is active when the compressor is operating under load and inactive when the compressor is off or offloaded |
| 10 | TOWERS | Signals which tower is responsible for air drying and which tower is draining humidity; when inactive, tower one is operational; when active, tower two is in operation. |
| 11 | MPG | Activates the intake valve to start the compressor under load when the pressure in the Air Production Unit (APU) drops below 8.2 bar; it triggers the COMP sensor, which mirrors the MPG sensor's behaviour. |
| 12 | LPS | Detects and activates when the pressure falls below 7 bars. |
| 13 | Pressure Switch | Detects the discharge in the air-drying towers and activates the electrical signal. |
| 14 | Oil Level | Detects the oil level in the compressor and activates the signal when the oil is below the expected level. |
| 15 | Caudal Impulse | Counts the pulse outputs representing the absolute amount of air flowing from the APU to the reservoirs |

Figure 7: Feature Description (Davari et al., 2021)

Descriptive Analysis: This is done to better understand the dataset. The dataset is summarised and helps understand the main features of the dataset and how they relate to one another. The dataset contains 7 analogue features and 8 digital features making a total of 15 features in total with 15169480 rows.

The figures below show the dataset summary and the dataset information.

```

{r}
summary(Metro)

```

| timestamp | TP2 | TP3 | H1 |
|--------------------------------|-----------------|----------------|----------------|
| Min. :2020-02-01 00:00:00.00 | Min. : -0.032 | Min. : 0.730 | Min. : -0.036 |
| 1st Qu.:2020-03-23 05:05:04.50 | 1st Qu.: -0.014 | 1st Qu.: 8.492 | 1st Qu.: 8.254 |
| Median :2020-05-17 08:07:06.00 | Median : -0.012 | Median : 8.960 | Median : 8.784 |
| Mean :2020-05-16 22:58:36.63 | Mean : 1.368 | Mean : 8.985 | Mean : 7.568 |
| 3rd Qu.:2020-07-10 03:07:27.50 | 3rd Qu.: -0.010 | 3rd Qu.: 9.492 | 3rd Qu.: 9.374 |
| Max. :2020-09-01 03:59:50.00 | Max. :10.676 | Max. :10.302 | Max. :10.288 |

| DV_pressure | Reservoirs | oil_temperature | Motor_current | COMP |
|-------------------|----------------|-----------------|---------------|---------------|
| Min. : -0.03200 | Min. : 0.712 | Min. :15.40 | Min. :0.020 | Min. :0.000 |
| 1st Qu.: -0.02200 | 1st Qu.: 8.494 | 1st Qu.:57.77 | 1st Qu.:0.040 | 1st Qu.:1.000 |
| Median : -0.02000 | Median : 8.960 | Median :62.70 | Median :0.045 | Median :1.000 |
| Mean : 0.05596 | Mean : 8.985 | Mean :62.64 | Mean :2.050 | Mean :0.837 |
| 3rd Qu.: -0.01800 | 3rd Qu.: 9.492 | 3rd Qu.:67.25 | 3rd Qu.:3.808 | 3rd Qu.:1.000 |
| Max. : 9.84400 | Max. :10.300 | Max. :89.05 | Max. :9.295 | Max. :1.000 |

| DV_electric | Towers | MPG | LPS | Pressure_switch |
|-------------|--------|-----|-----|-----------------|
|-------------|--------|-----|-----|-----------------|

Figure 8: Data Summary

```

Check number of columns.
{r}
ncol(Metro)

```

[1] 16

Figure 9: No of rows

```

Check number of rows.
{r}
nrow(Metro)

```

[1] 1516948

Figure 10: No. of columns

Exploratory Data Analysis: This is done to better understand the dataset. The dataset is summarised and helps understand the main features of the dataset and how they relate to one another (Corrales et al., 2015).

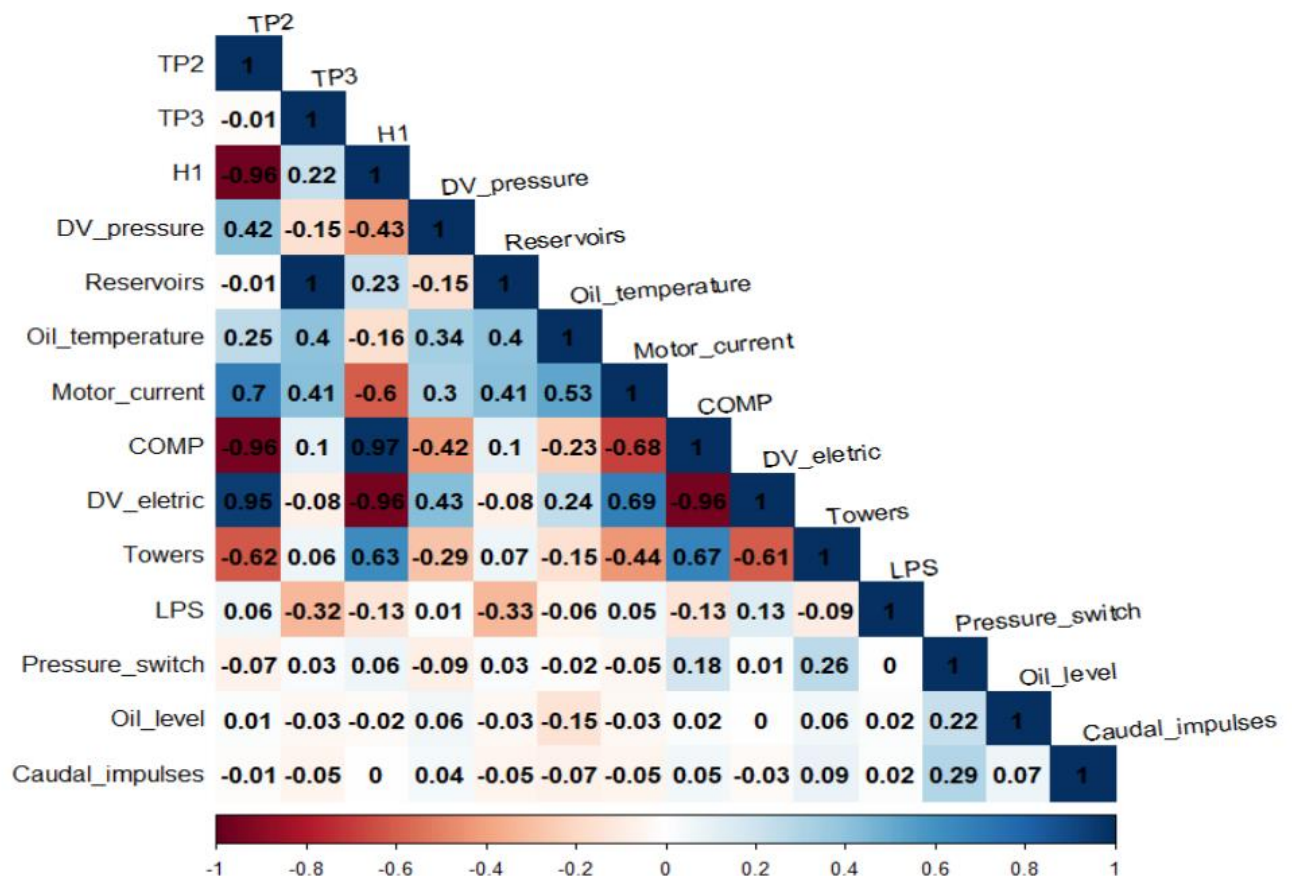


Figure 11: Correlation Matrix

Data Normalization check: The figures below show the data distribution check for the variables in the dataset. This is done to check if the dataset is normally distributed or not.

```
{r}
hist(Metro$TP2, main = "Distribution of TP2", xlab = "TP2")
```

Figure 12: Normalization Code

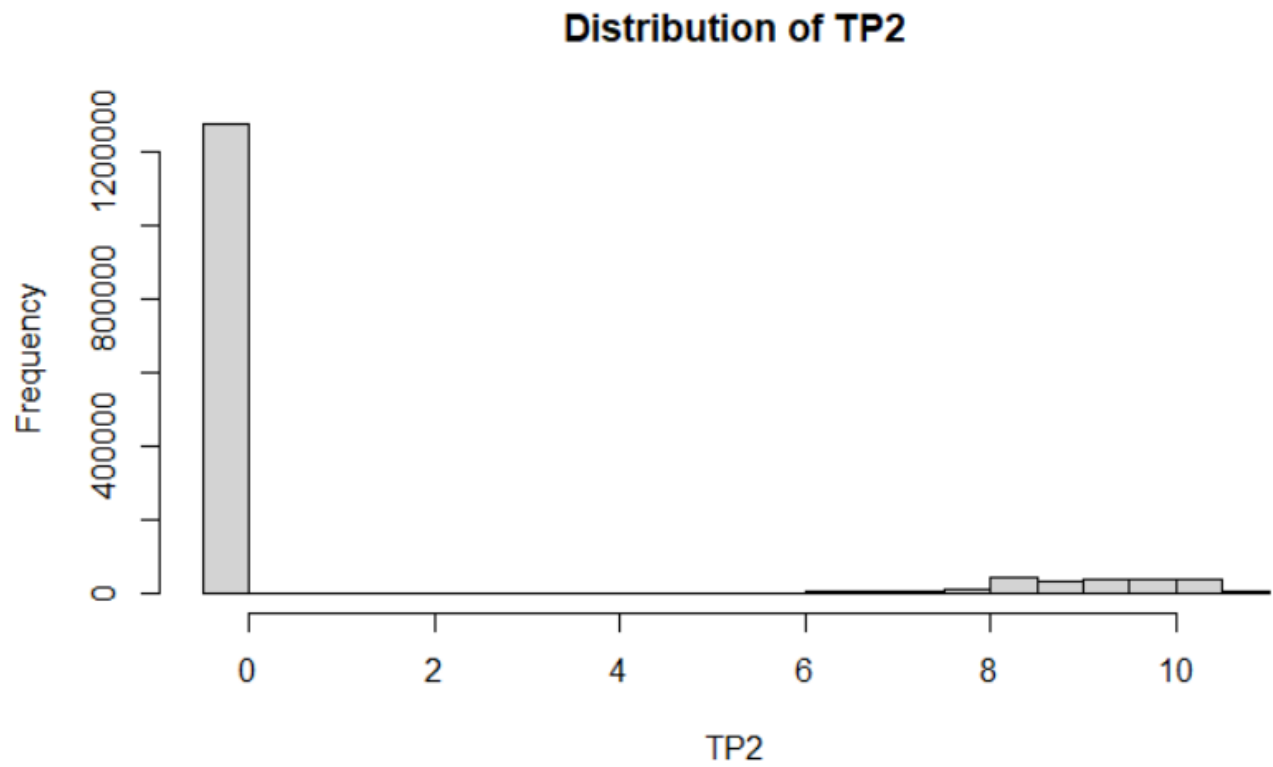


Figure 13: TP2 Distribution

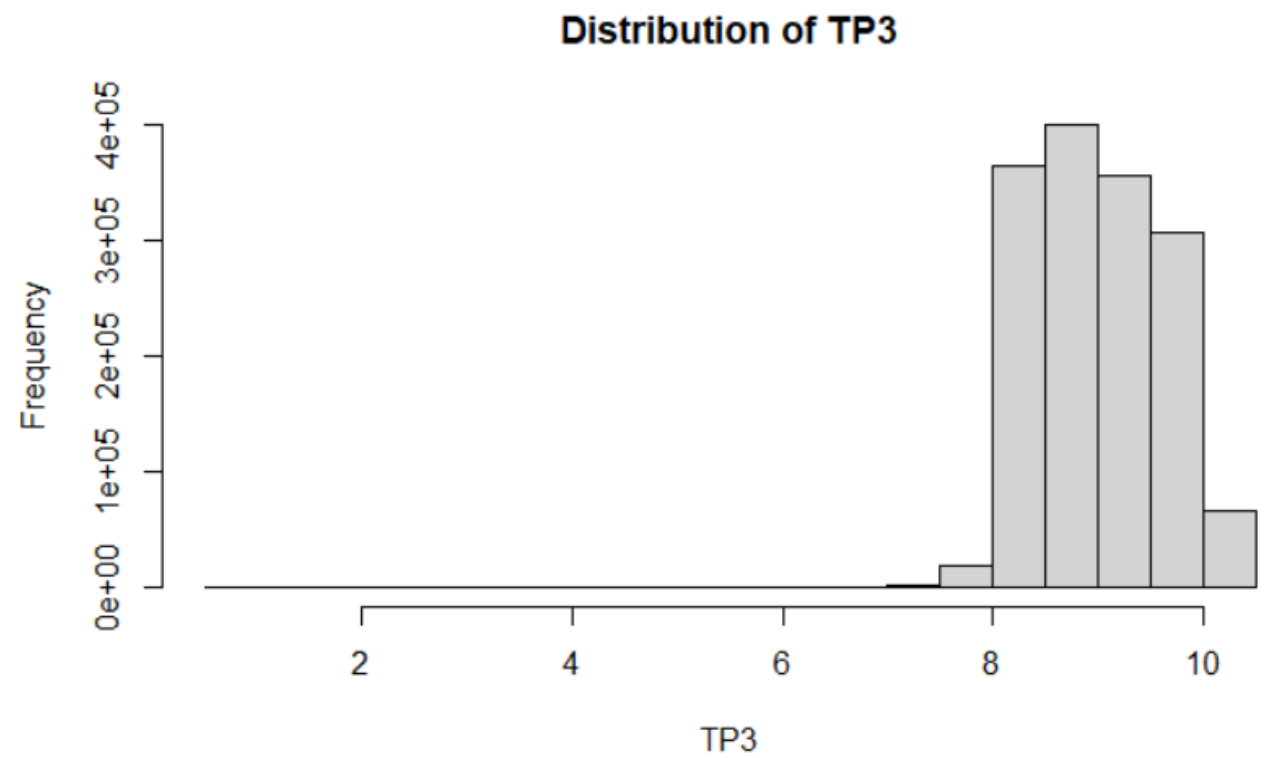


Figure 14: TP3 Distribution

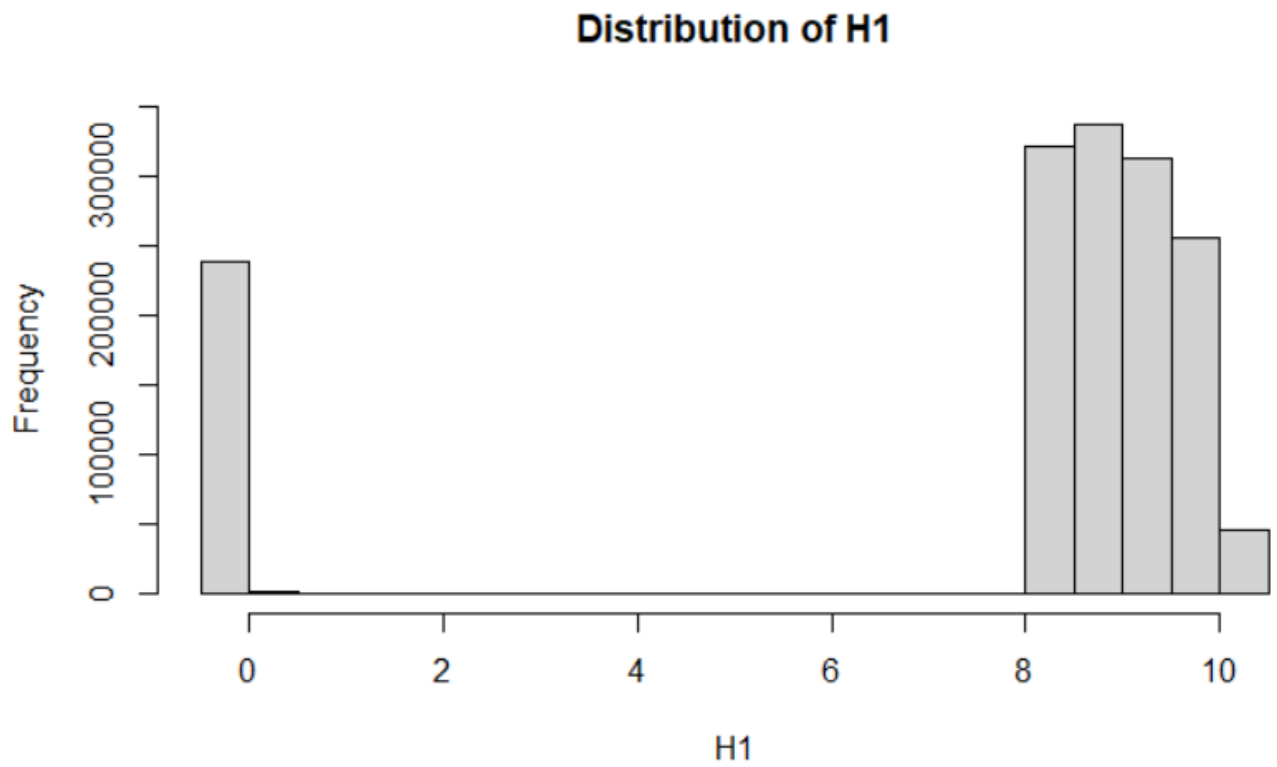


Figure 15: H1 Distribution

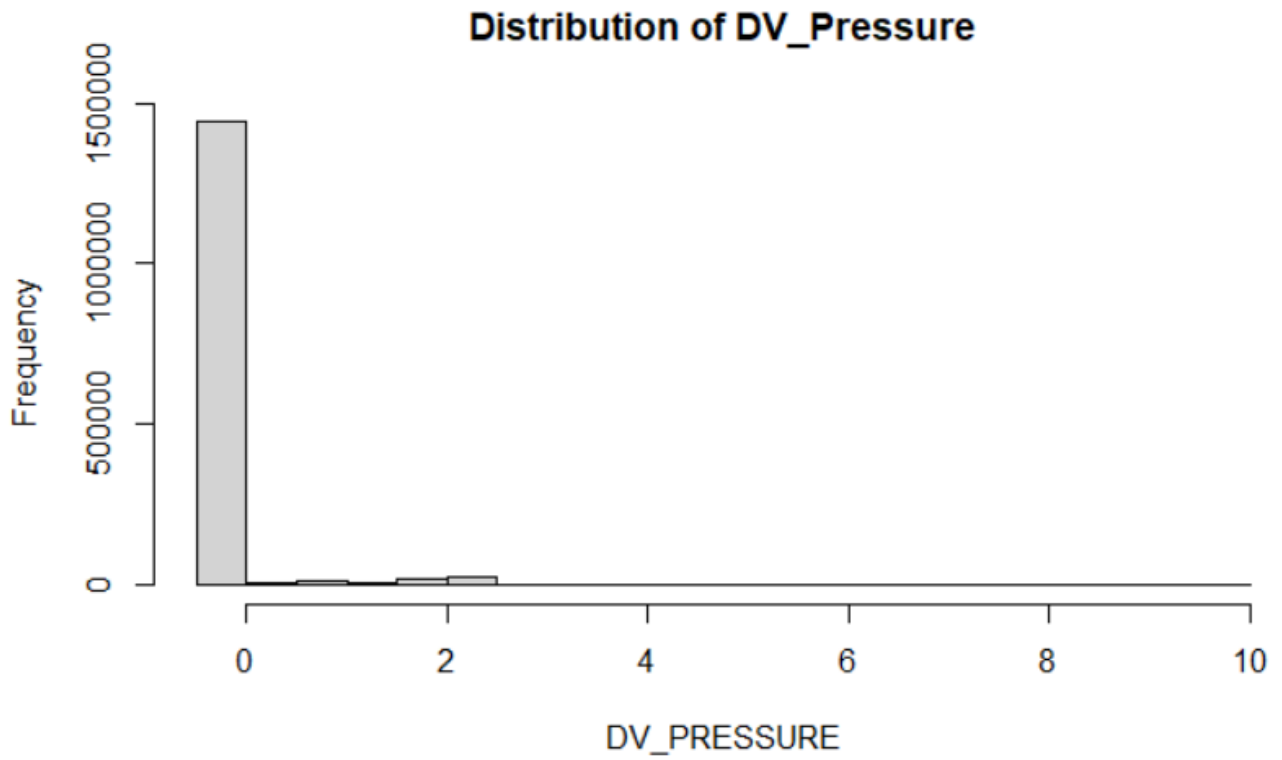


Figure 16: DV_Pressure Distribution

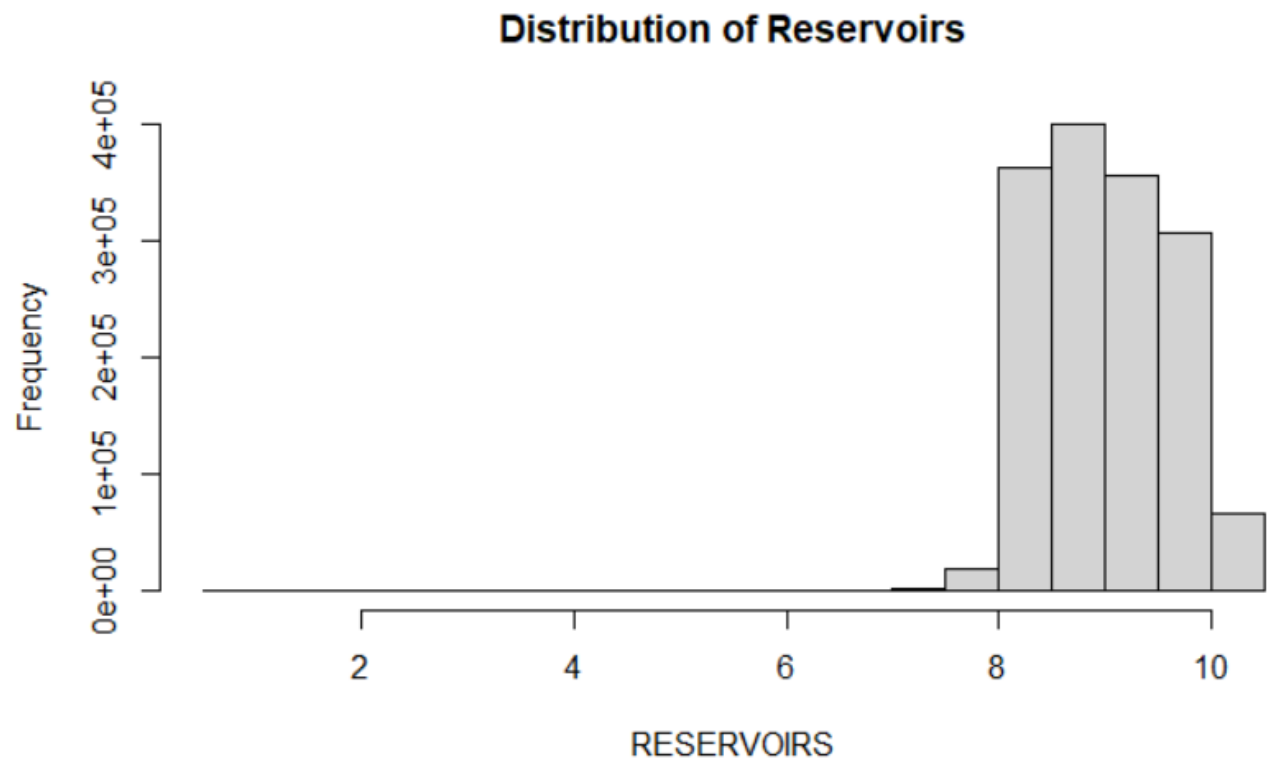


Figure 17: Reservoirs Distribution

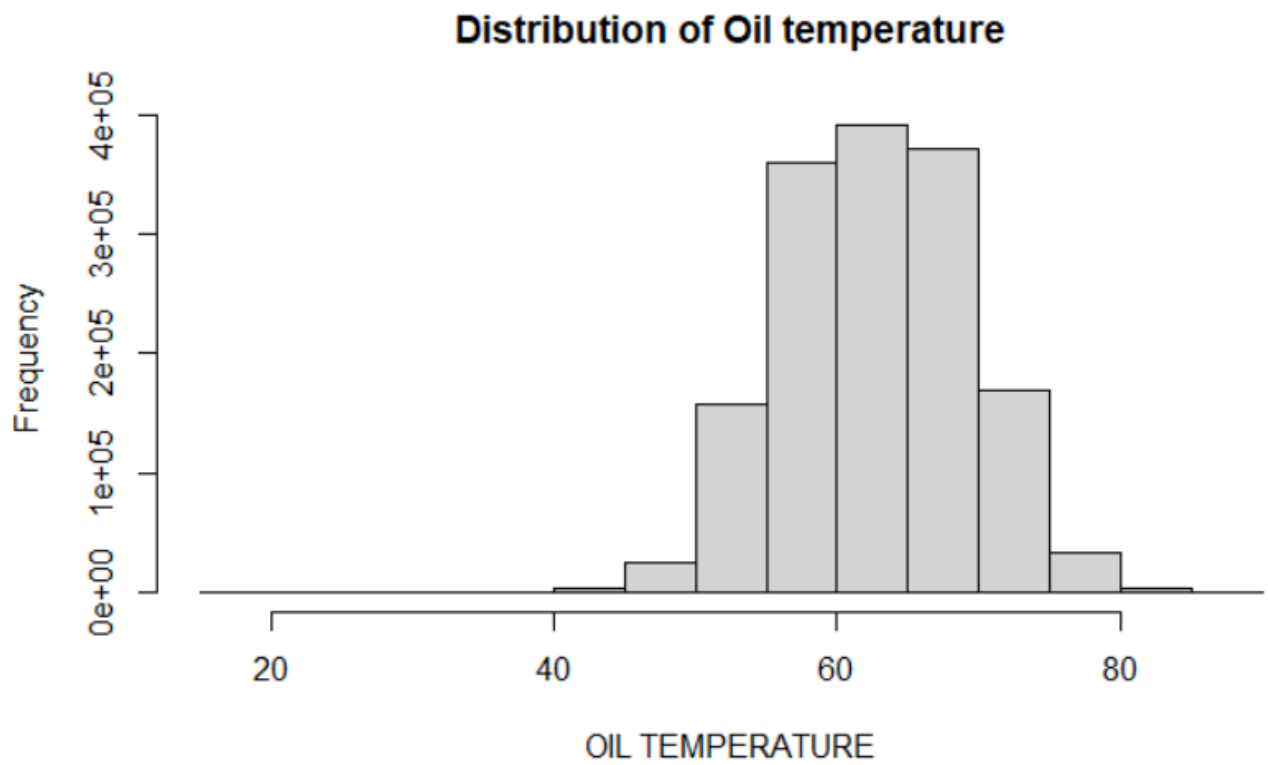


Figure 18: Oil Temperature Distribution

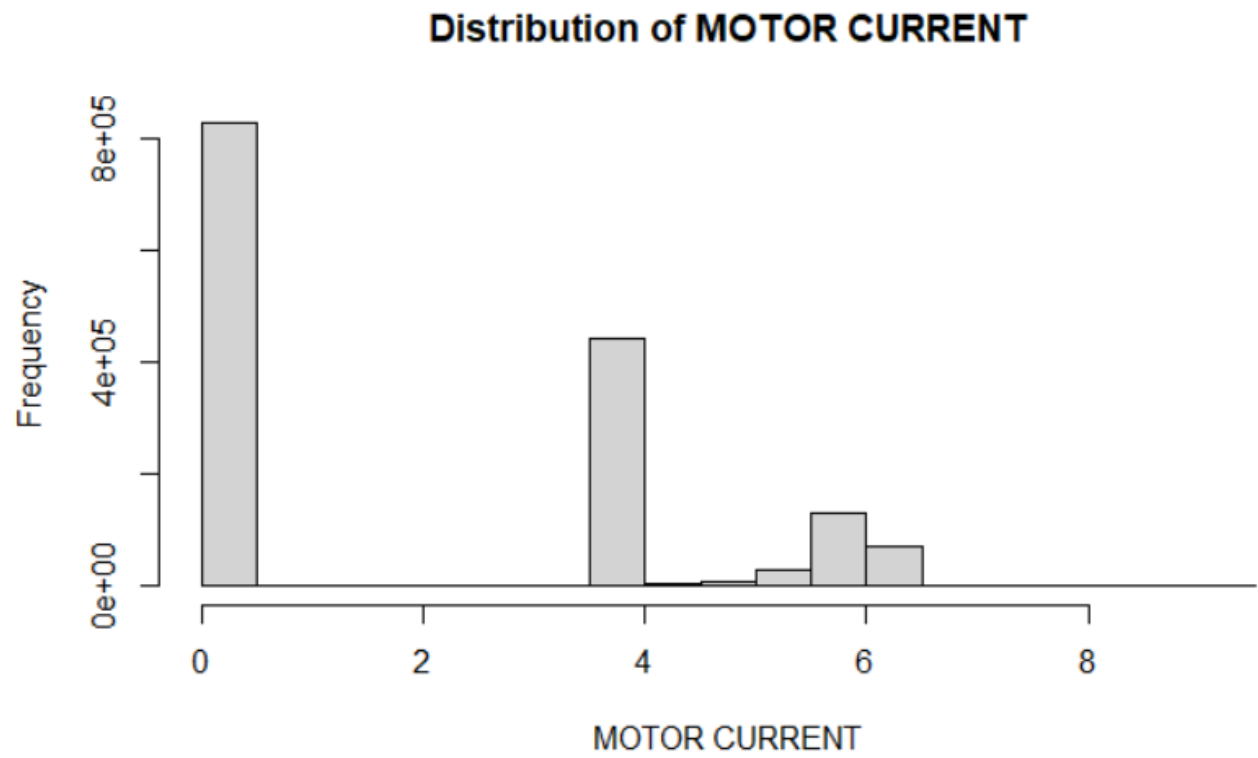


Figure 19: Motor Current Distribution

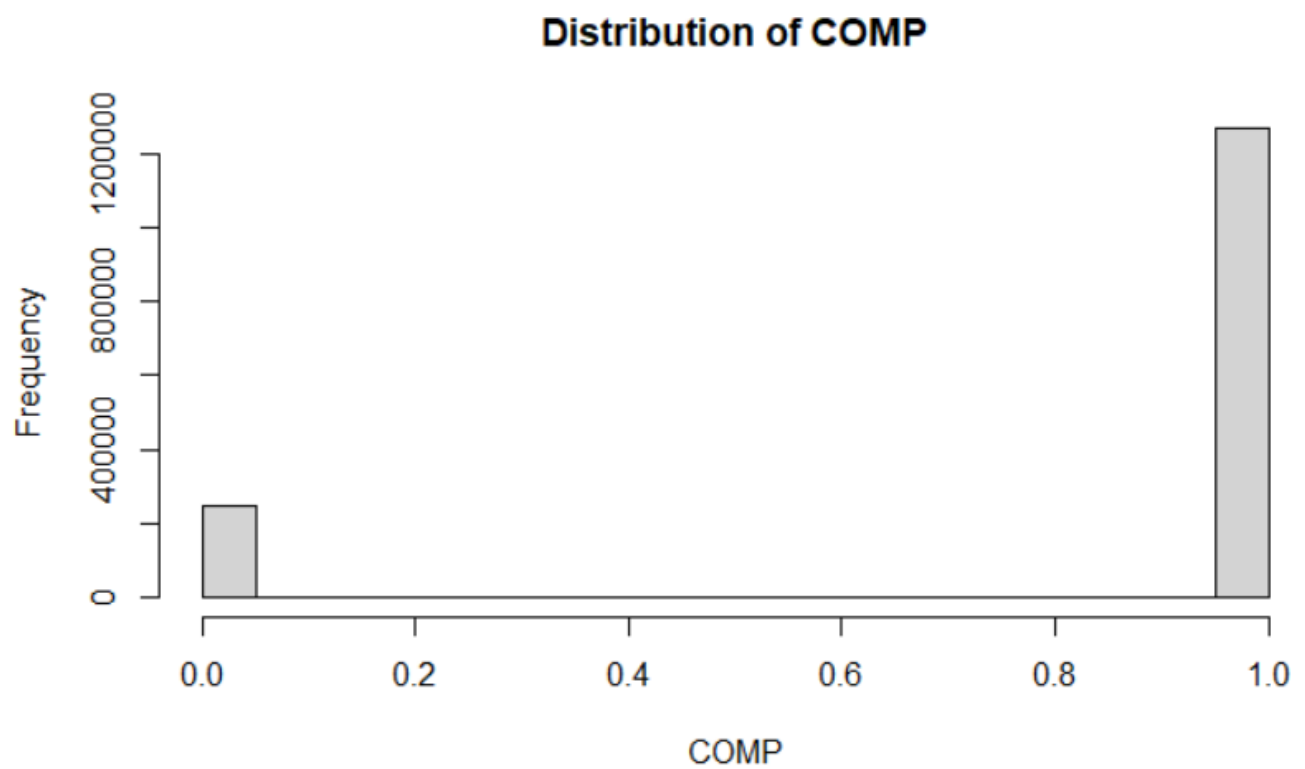


Figure 20: COMP Distribution

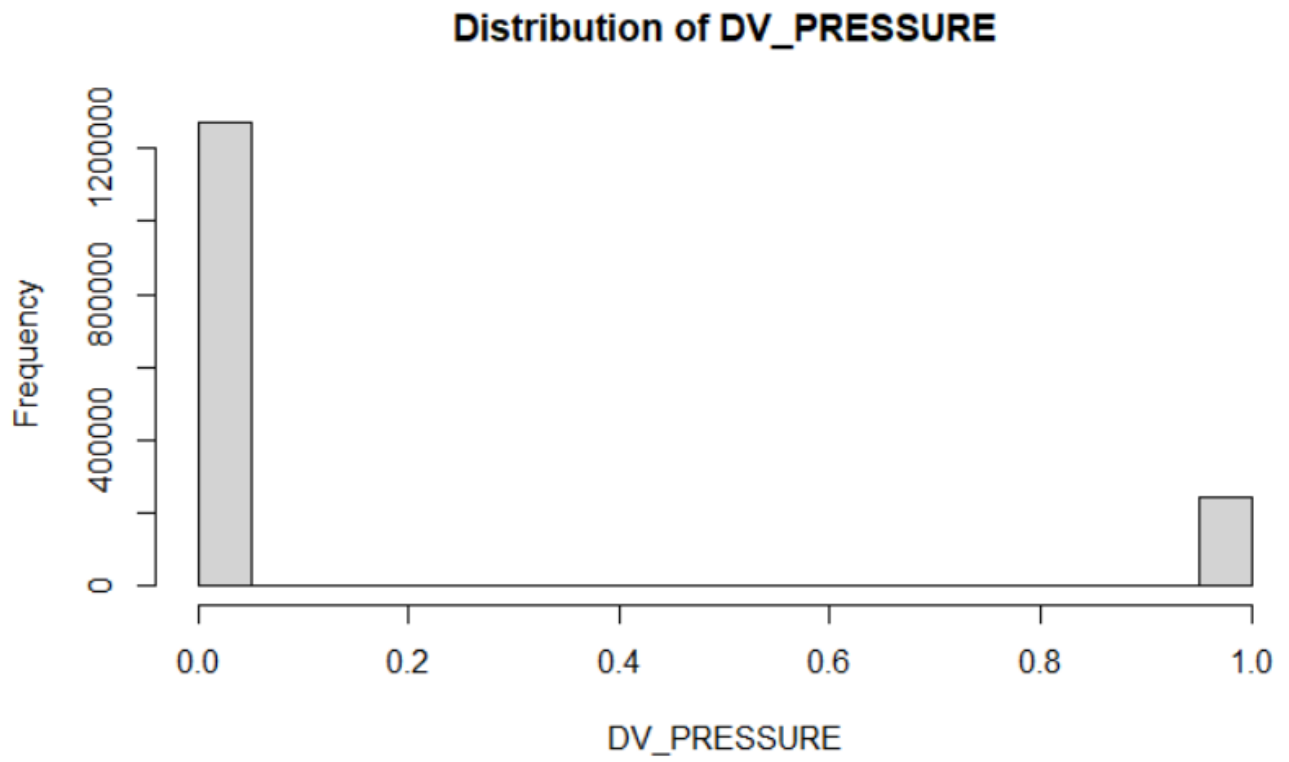


Figure 21: DV_Pressure Distribution

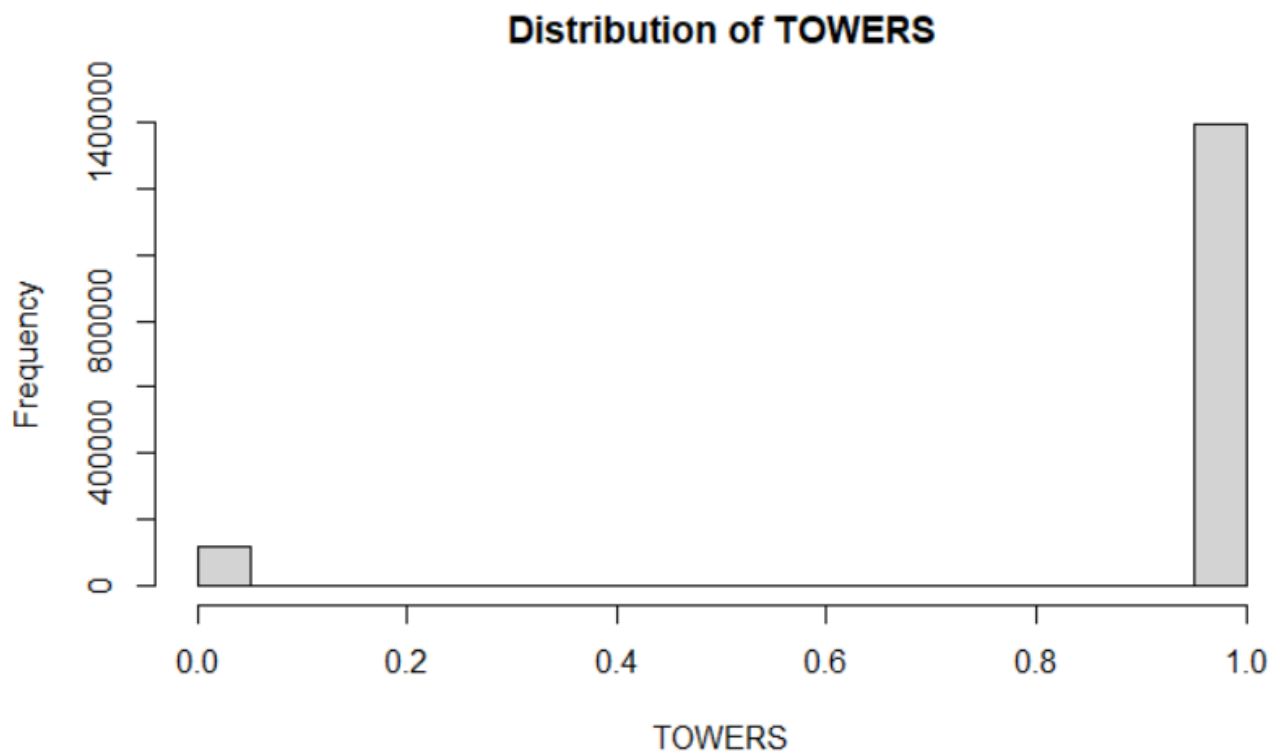


Figure 22: TOWERS Distribution

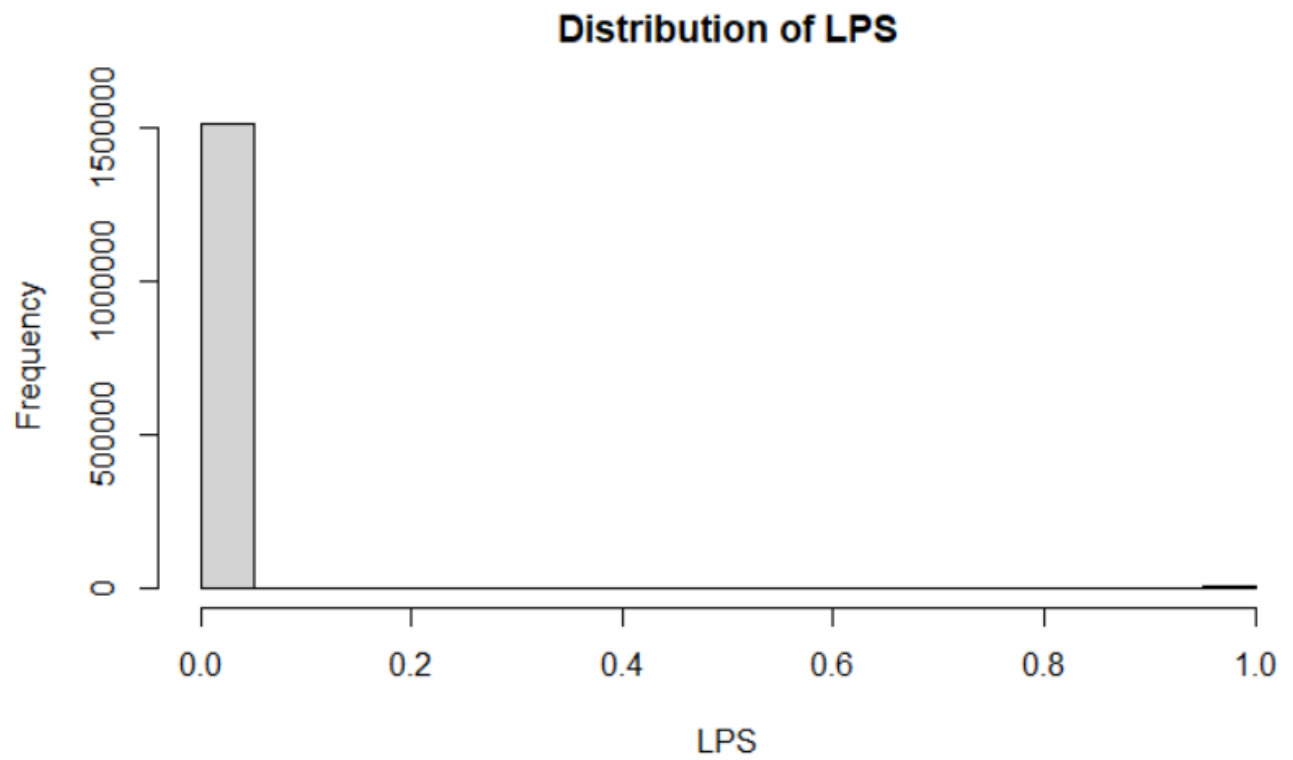


Figure 23: LPS Distribution

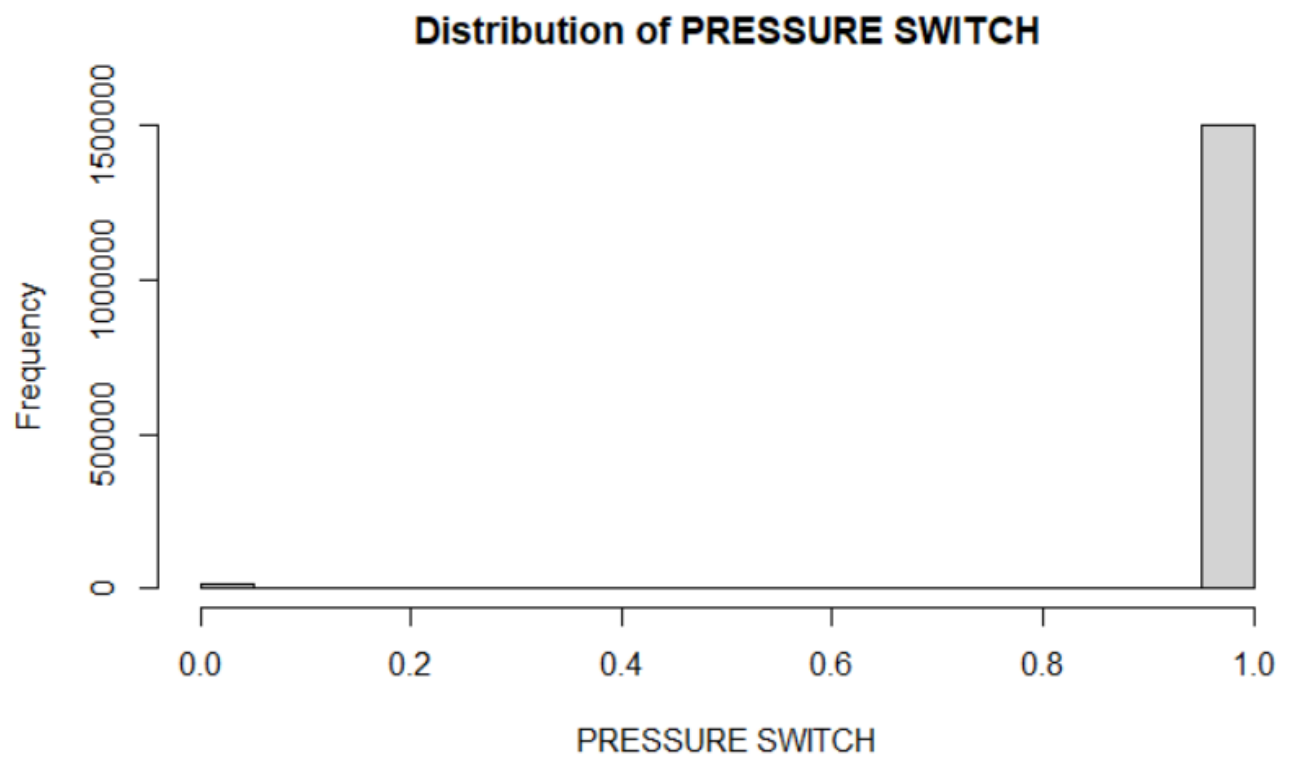


Figure 24: PRESSURE SWITCH Distribution

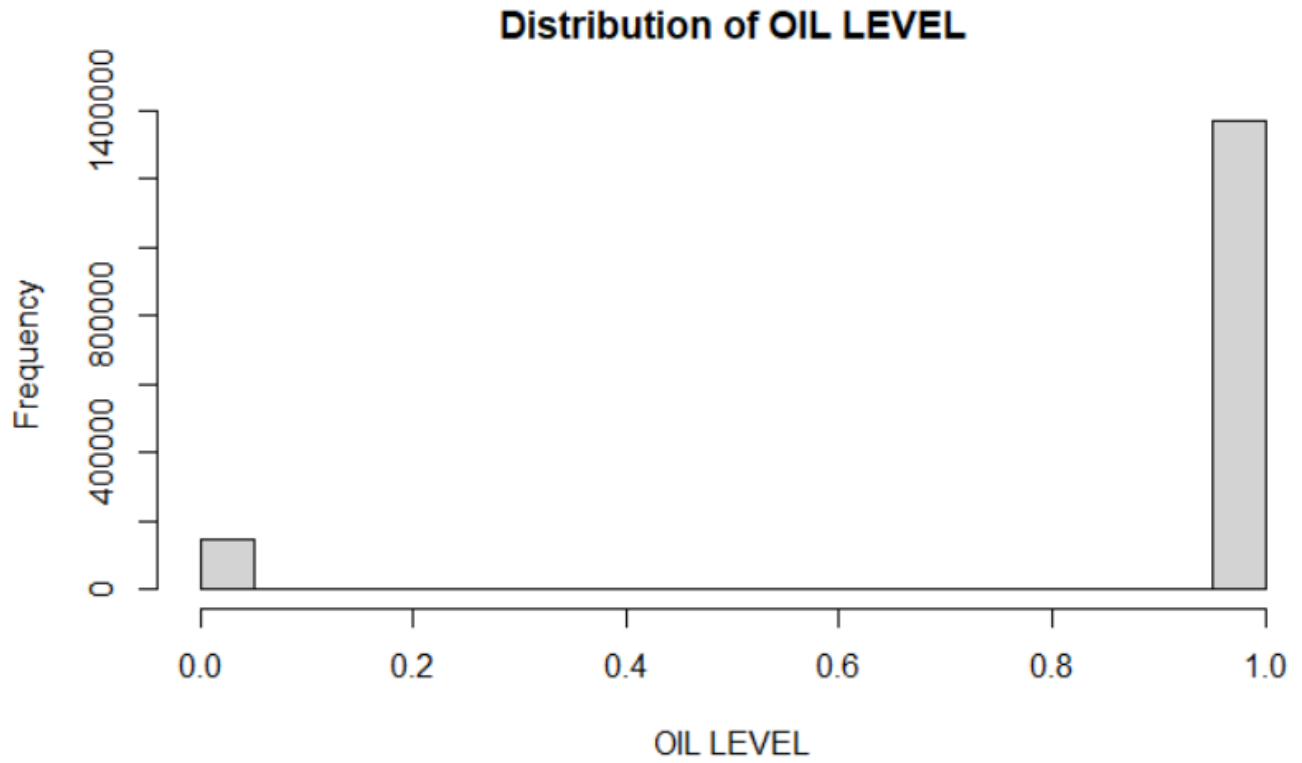


Figure 25: OIL LEVEL Distribution

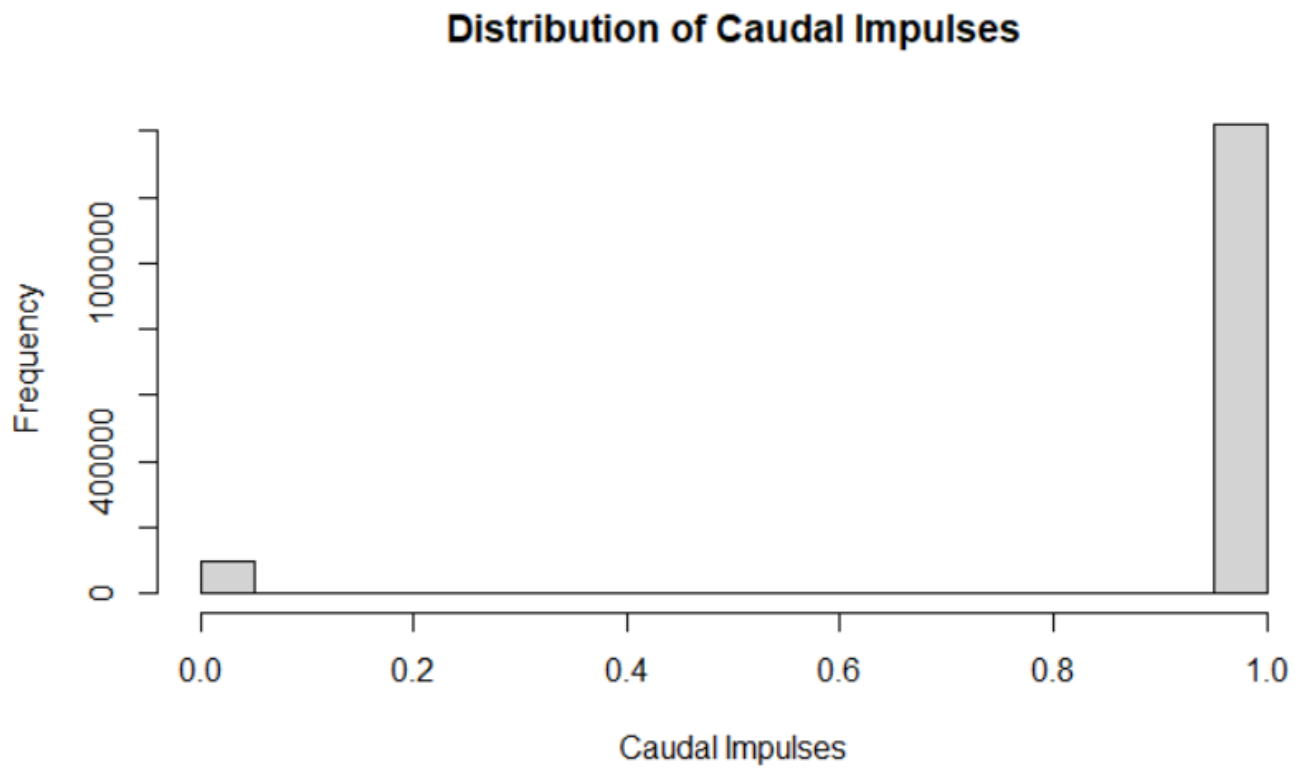


Figure 26: Caudal Impulses Distribution

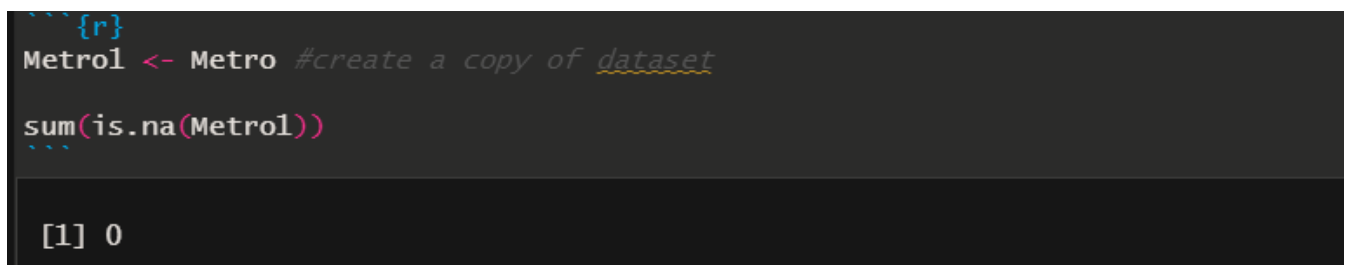
3.3.3 Data Preparation

Data preparation refers to the processes carried out to ensure the dataset is ready for modelling purposes (Brownlee, 2020). This is one of the most crucial parts of the project as it contains the data pre-processing steps which include data engineering for the Aircompfail column, checking and handling missing values, checking for outliers and determine whether they are to be handled or not.

Data pre-processing: Data pre-processing are the steps taken to ensure data quality and data reliability. These are steps carried out to ensure a dataset is ready before analysis is carried out. This section of the report explores the pre-processing steps carried out before statistical analysis. Steps such as data cleaning, checking for missing values and handling them, checking for outliers using boxplot, printing out the list of outliers and crosschecking the printed list with the dataset if they are true outliers or not before handling them (Luengo et al., 2020).

Some of the steps involved in data pre-processing include:

1. **Checking for Missing Values:** Figure below shows there are Zero (0) missing values



```
{r}
Metro1 <- Metro #create a copy of dataset
sum(is.na(Metro1))

[1] 0
```

Figure 27: Missing Values

2. **Checking for Outliers:** the figure below shows some of the variables have outliers, however after cross checking with the dataset and considering that the data was collected in real time, the outliers were deemed not to be true outliers and therefore not handled.

```
{r}  
#Timestamp  
boxplot(Metro1$timestamp, main = "Timestamp Boxplot")
```

Figure 28: Outlier Code

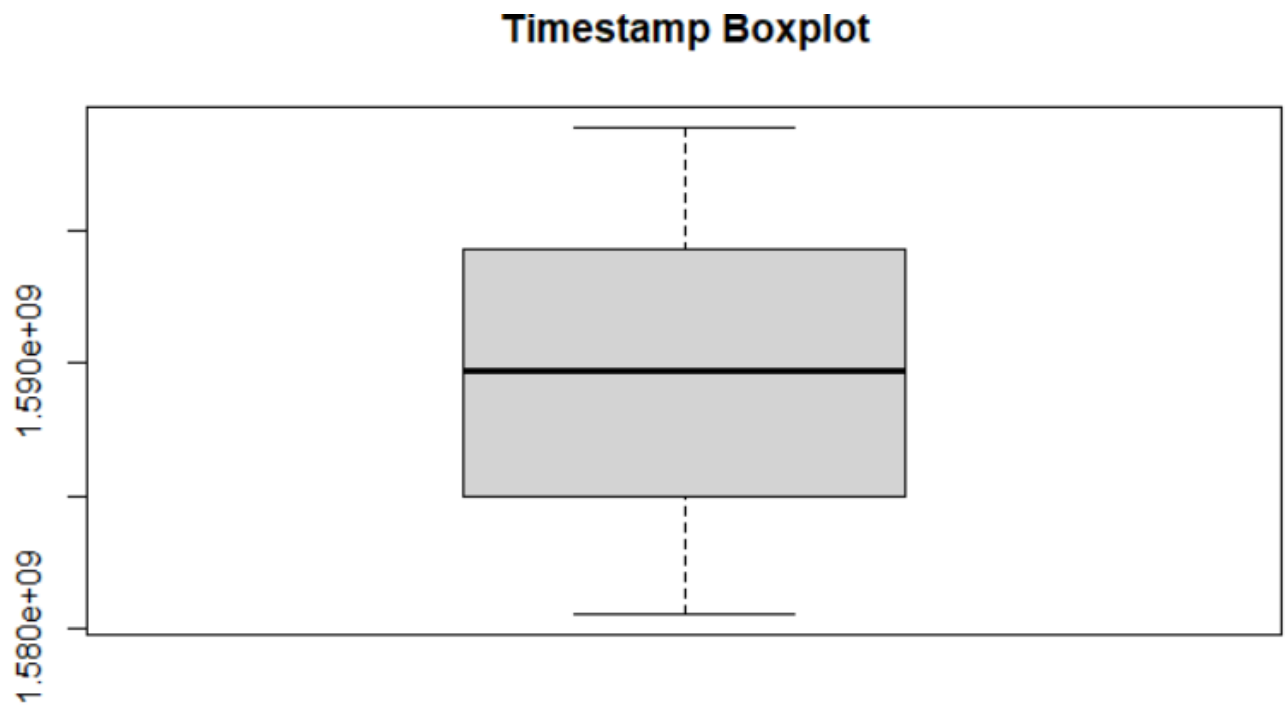


Figure 29: Timestamp Boxplot

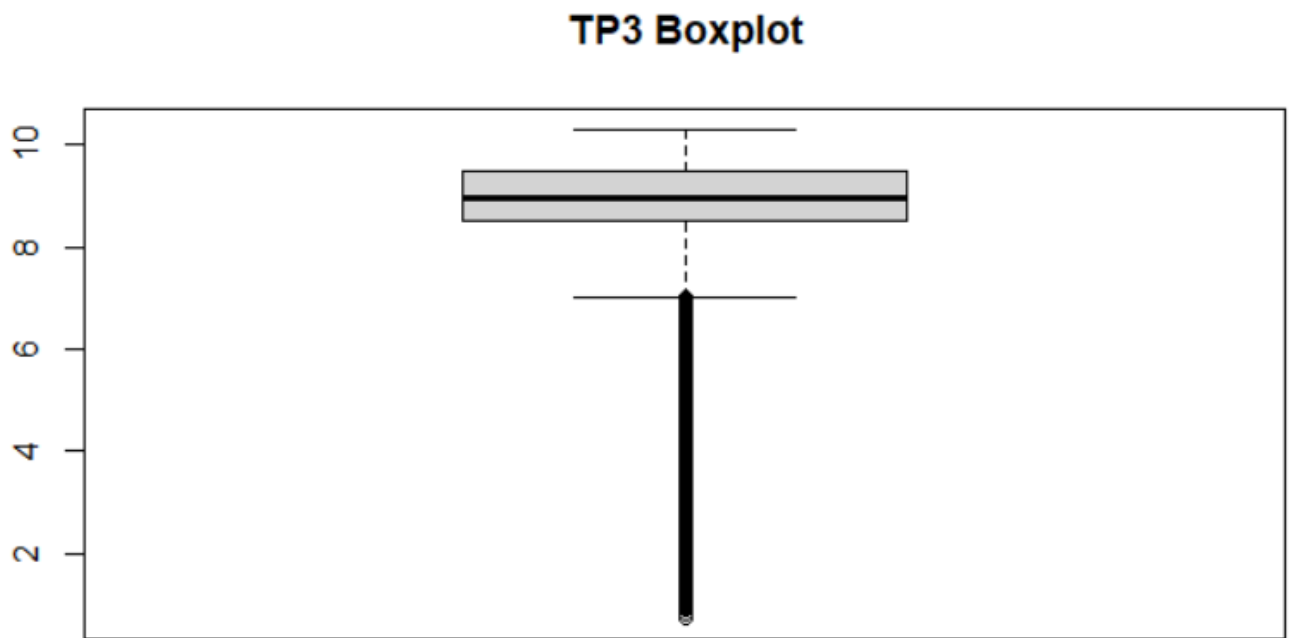


Figure 30: TP3 Boxplot

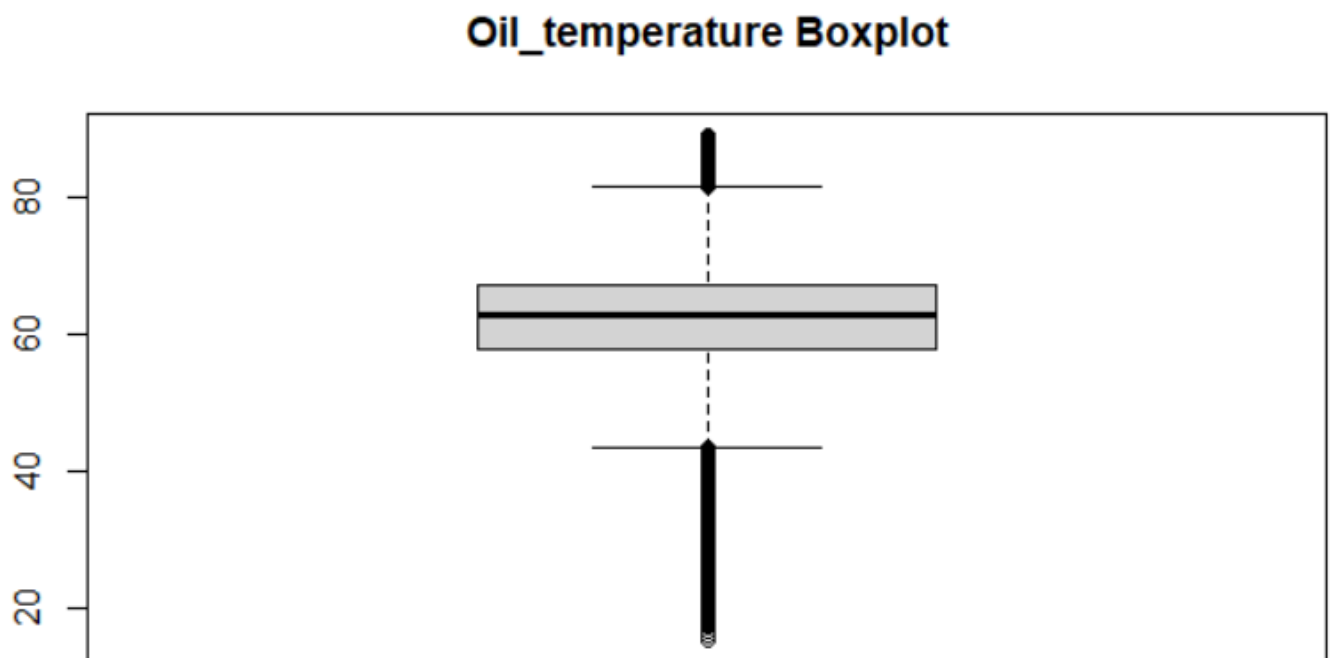


Figure 31: Oil Temperature Boxplot

Feature Engineering: This involves the manipulation of the dataset by adding a new column to contain the failures registered on the APU. The dataset information provides an understanding that some failures

were recording showing the start time and end time of the failures. This made it easy to manipulate the dataset into creating a new “Aircompfail” column to accommodate the registered failures and success dates as shown in the figure below (Kuhn and Johnson, 2020).

| Nr. | Start Time | End Time | Failure | Severity | Report |
|-----|-----------------|-----------------|----------|-------------|-------------------------------|
| #1 | 4/18/2020 0:00 | 4/18/2020 23:59 | Air leak | High stress | |
| #1 | 5/29/2020 23:30 | 5/30/2020 6:00 | Air Leak | High stress | Maintenance on 30Apr at 12:00 |
| #3 | 6/5/2020 10:00 | 6/7/2020 14:30 | Air Leak | High stress | Maintenance on 8Jun at 16:00 |
| #4 | 7/15/2020 14:30 | 7/15/2020 19:00 | Air Leak | High stress | Maintenance on 16Jul at 00:00 |

Figure 32: Failure Information (Davari et al., 2021)

```

# Define the failure ranges of indices
range1 <- 562565:571227 # First range (4/18/2020 0:00 4/18/2020 23:59)
range2 <- 840741:843105 # Second range (5/29/2020 23:30 5/30/2020 6:00)
range3 <- 887240:908125 # Third range (6/5/2020 10:00 6/7/2020 14:30)
range4 <- 1171094:1172715 # Fourth range (7/15/2020 14:30 7/15/2020 19:00)

# Check if the dataset has enough rows to include the specified ranges
if (max(c(range1, range2, range3, range4)) > nrow(Metro1)) {
  stop("The specified index ranges exceed the number of rows in the dataset.")
}

# Add a new column that assigns 1 to the specified ranges and 0 to others
Metro1$Aircompfail <- ifelse(
  seq_len(nrow(Metro1)) %in% c(range1, range2, range3, range4), 1, 0)

```

Figure 33: Feature Engineering Code

| | DV_electric | Towers | MPG | LPS | Pressure_switch | Oil_level | Caudal_impulses | Aircompfail |
|---|-------------|--------|-----|-----|-----------------|-----------|-----------------|-------------|
| 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 |
| 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 1 |
| 1 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 1 |
| 1 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 1 |

Figure 34: New Column “Aircompfail”

Data Sampling: A class imbalance was observed during the data count process and the hybrid sampling approach was adopted to ensure a more comprehensive and informative dataset sample (Lohr, 2022), (Lu and Lohr, 2021).

The hybrid approach combines the oversampling and under-sampling techniques where one increases the representation of the minority class and the other reduces the dominance of the majority class.

```
Check the class distribution
```{r}
Display the original class distribution
print("Original class distribution:")
print(table(Metro2$Aircompfail))
```

[1] "Original class distribution:"

      0      1
1483412 33536
```

Figure 35: Class Distribution

```
```{r}
library(ROSE)

Create a balanced dataset
Metro3 <- ovun.sample(Aircompfail ~ .,
 data = Metro2,
 method = "both", # Use both over-sampling and under-
 p = 0.5, # Aim for a 50/50 class distributio
 seed = 1)$data # Set seed for reproducibility
```

Figure 36: Hybrid Data Sampling Code

```
Loaded ROSE 0.0-4

[1] "Class distribution after balancing:"

 0 1
758744 758204
```

Figure 37: New Class Distribution

### 3.3.4 Data Modelling

Another critical step in the study is data modelling as it involves the use of statistical analysis tools, ML and DL techniques to analyse trends and patterns observed in the dataset. The observations are then used to make predictions or forecasts.

Data modelling involves selecting the preferred modelling techniques, splitting the dataset, training the model using the split data and eventually deploying the model trained. In machine learning and deep learning, the restructured and pre-processed Sheffield dataset is split using the train test split, the modelling techniques are trained, and forecast is then afterwards. This involves initiating different ML and DL techniques to explore the connection between different independent variables and the dependent variable. The dependent variable in this study is the “Aircompfail” column, and the independent variables are the analogue and digital sensor data. Considering it is a classification related project, analysis is done by launching different classification models such as the Logistic regression, Decision tree, Random Forest, Gradient Boosting Machine, Neural Network and picking the best fit for hybrid modelling if need be. The predictive maintenance model development process involves using Machine learning regression techniques and deep learning techniques. Upon initiating the models, the performance metric used to test the models are accuracy, precision, recall and F1-score.

Below are some of the data modelling steps:

1. **Feature Selection:** Due to the volume of the dataset and the number of sensors attached to the dataset, there is a need to choose the best set of features that will be suitable for the analysis. There are 15 sensors present in the dataset, of which 7 were analogue sensor data and the last 7 were digital sensor data. This resulted in a need to select a subset of the dataset that can be used which will improve the model performance, reduce overfitting, and therefore ensure proper interpretations.

```
Find highly correlated features (correlation > 0.75)
Metro4highlyCorrelated <- findCorrelation(Metro4corrmatrix, cutoff = 0.75)
Remove highly correlated features
Metro4 <- Metro4[, -Metro4highlyCorrelated]
```

Figure 38: Feature Selection Code

	TP3	DV_pressure	Oil_temperature	Towers	LPS	Pressure_switch	Oil_level	Caudal_impulses	Aircompfail
1	9.068	-0.022	55.325	1	0	1	1	1	0
2	8.506	-0.014	65.150	0	0	0	0	0	0

Figure 39: Features Selected

2. **Train\_Test\_Split:** MetroPT-3 dataset is split into training and testing dataset using a specific splitting ratio using the train\_test\_split function. In this document, the splitting ratio used is 0.8 specifying 80/20 for training and testing respectively. Upon successful splitting the dataset, the training dataset contains 1179849 rows and testing dataset contains 337099 as shown below.

```
Split the dataframe using 80/20 split
```{r}
Metro4split <- sample.split(Metro4, SplitRatio = 0.8)

Initiate training set and testing set
```{r}
m4train <- Metro4[Metro4split,]
m4test <- Metro4[!Metro4split,]
```

Figure 40: Train\_Test\_Split Code

```
check for number of rows in each
```{r}
nrow(m4train)

[1] 1179849
```

Figure 41: Train Dataset

```
```{r}
nrow(m4test)

[1] 337099
```

Figure 42: Test Dataset

3. **Machine learning:** Machine learning classification models are utilized in this section of the report to predict the failures that might occur. This project considers things like robustness, complexity, flexibility and accuracy of model in choosing the ML techniques to be used. Another thing put into consideration is the fact that a hybrid sampling has been done on the dataset.

```

Import Libraries
{r}
Load necessary libraries
library(caret) #
library(e1071) #
library(randomForest)
library(gbm)
library(xgboost)

```

Figure 43: ML Libraries

Below are some of the ML models used and the reasons behind this:

- **Logistic regression:** This is a simple interpretable linear model and best used for binary classification problems like the one in the project. It provides coefficients of each feature that sometimes affect the expected outcome.

```

{r}
#Train the model
Logmodel <- glm(Aircompfail~., data = m4train, family = "binomial")

#Predict the model
Logpredict <- predict(Logmodel, m4test, type = "response")
Logpredict

```

2	8	11	17	20	26
0.0003645296	0.0608213748	0.0053563480	0.0356725958	0.0459829741	0.0153790028
35	38	44	47	53	56

Figure 44: Logistic Regression Code

- **Decision Tree:** This model splits data into subsets based on the values of the features and they do not require standardization or normalization. It makes use of a set of standardized rules that allows for easy interpretation.

```
library(rpart.plot)
Fit the decision tree model using the training data
Dtmodel <- rpart(Aircompfail ~ ., data = m4train, method = "class")
```

Figure 45: Decision Tree Code

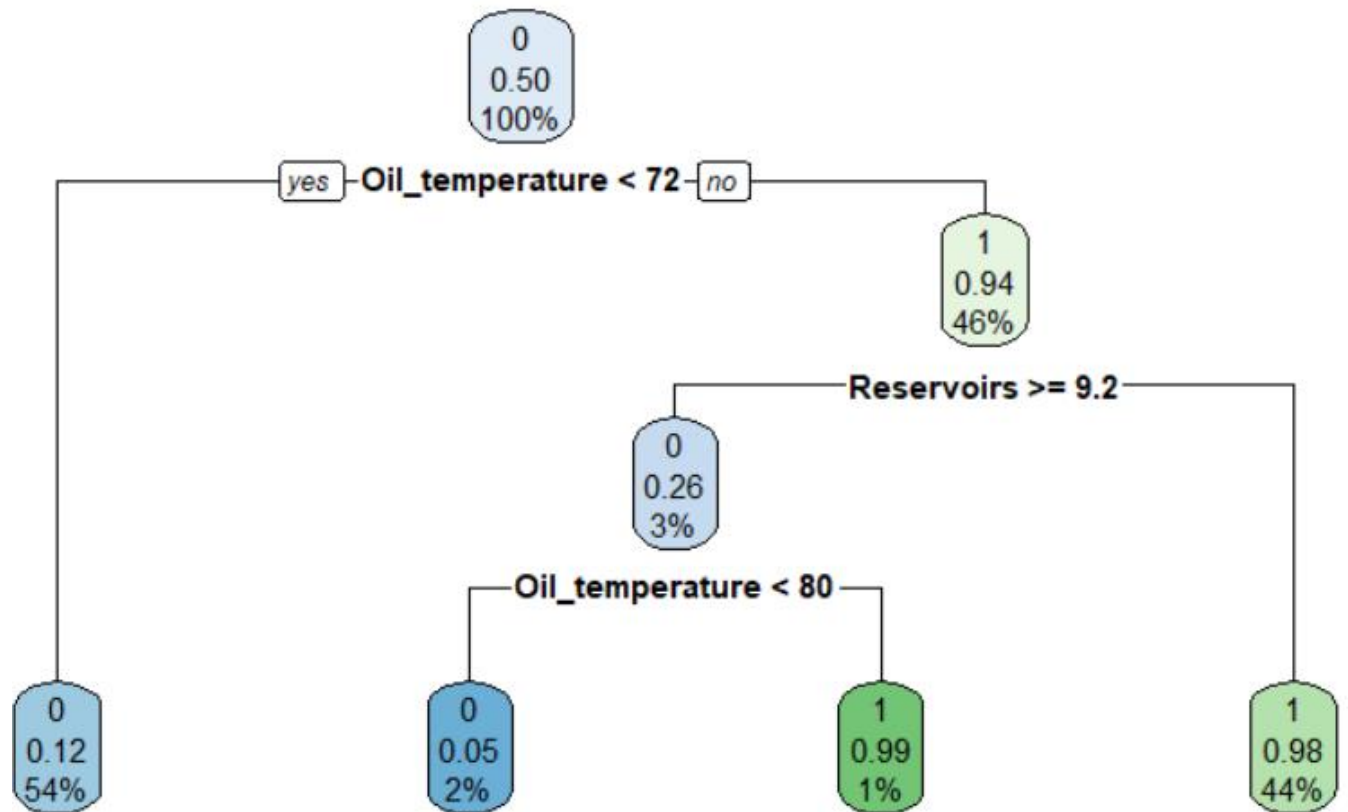


Figure 46: Decision Tree

- **Random Forest:** This uses the decision tree concept as it uses an ensemble method that builds multiple decision trees and gets the average of their predictions. It is known for reducing overfitting and helps increase robustness which is something that results from using sampling techniques.

```
Train the Random Forest model
Rfmodel <- ranger(Aircompfail ~ ., data = m4train, probability = TRUE)

#predict on the test set
Rfpredict <- predict(Rfmodel, m4test)$predictions
summary(Rfpredict)
...
```

Attaching package: 'ranger'

The following object is masked from 'package:randomForest':

importance

```
Growing trees.. Progress: 22%. Estimated remaining time: 1 minute, 47 seconds.
Growing trees.. Progress: 41%. Estimated remaining time: 1 minute, 30 seconds.
Growing trees.. Progress: 61%. Estimated remaining time: 59 seconds.
Growing trees.. Progress: 83%. Estimated remaining time: 25 seconds.
 v1 v2
```

Figure 47: Random Forest Code

- **Gradient Boosting Machine (GBM):** This makes use of a sequential model building process that improves and corrects the errors encountered in previous models. GBM has a high level of accuracy and knows how to handle complex non-linear relationships.

```
Train the GBM model
Gbmmodel <- gbm(Aircompfail ~ ., data = m4train,
 distribution = "bernoulli", # For binary classification
 n.trees = 100, # Number of trees
 interaction.depth = 3, # Depth of each tree
 shrinkage = 0.01, # Learning rate
 cv.folds = 5, # Number of cross-validation folds
 verbose = TRUE) # Print progress
```

Figure 48: GBM Code

4. **Deep learning:** This is a subset of machine learning which is an attempt to simulate the processing power of the brain. It makes use of multiple layered neural networks to solve complex problems and an example of this is the neural network.



```

Import Libraries
```{r}
# Load necessary libraries
library(keras)
library(tensorflow)
library(nnet) #function for neural network
library(caret) #streamlines training and evaluation

```

Figure 49: DL Libraries

- **Neural Network:** This model is good for large datasets and consists of multiple interconnected nodes or neurons which makes it easy to learn and understand complex patterns or relationships (Gurney, 2018).

```

#FIT NN
NNmodel <- nnet(Aircompfail~., data = dl4train, size = 5, decay = 0.1, maxit = 2
FALSE)

#Neural Network Predict
NNpred <- predict(NNmodel, dl4test, type = "raw")
NNpred
summary(NNpred)
```

weights: 51
initial value 389937.713416
iter 10 value 128067.298323

```

Figure 50: NN Code

### 3.3.5 Evaluation

Evaluation involves checking the accuracy and reliability of the models deployed. This is used to confirm if the models deployed got a better result or not. A low score would suggest that the model deployed isn't the best model to use and vice versa.

Evaluation metrics such as accuracy, precision, f1-score are adopted in this project to evaluate the deployed machine learning and deep learning models. The model with the highest score is used to determine the best model for time series forecasting.

1. **Accuracy:** This measures the overall correctness of the model predictions.

$$\text{Accuracy} = \frac{(\text{True Positives} + \text{True Negatives})}{(\text{Total Number of Instances})}$$

2. **Precision:** This measures the proportion of true positives to the number of total positives predicted.

$$\text{Precision} = \frac{(\text{True Positives})}{(\text{True Positives} + \text{False Positives})}$$

3. **Recall:** This measures the number of total positives predicted.

$$\text{Recall} = \frac{(\text{True Positives})}{(\text{True Positives} + \text{False Positives})}$$

4. **F1-score:** This is a balanced approach between Precision and Recall, which is preferred over accuracy for classification-based models.

$$\text{F1-score} = 2 \cdot \frac{(\text{Precision} * \text{Recall})}{(\text{Precision} + \text{Recall})}$$

```

Convert probabilities to class labels
NNpredclass <- ifelse(NNpred > 0.5, "1", "0")
NNpredclass <- as.factor(NNpredclass)

print(paste("Length of predicted classes: ", length(NNpredclass)))
print(paste("Length of actual classes: ", length(d14test$Aircompfail)))

Create a confusion matrix
NNconfmatrix <- confusionMatrix(NNpredclass, factor(d14test$Aircompfail))

Extract recall, precision, and F1 score
NNaccuracy <- NNconfmatrix$overall['Accuracy']
NNrecall <- NNconfmatrix$byClass["Recall"]
NNprecision <- NNconfmatrix$byClass["Precision"]
NNf1score <- NNconfmatrix$byClass["F1"]

Print the metrics
print(NNconfmatrix)

cat("Accuracy:", NNaccuracy, "\n")
cat("Recall:", NNrecall, "\n")
cat("Precision:", NNprecision, "\n")
cat("F1 Score:", NNf1score, "\n")

```

Figure 51: Evaluation Code

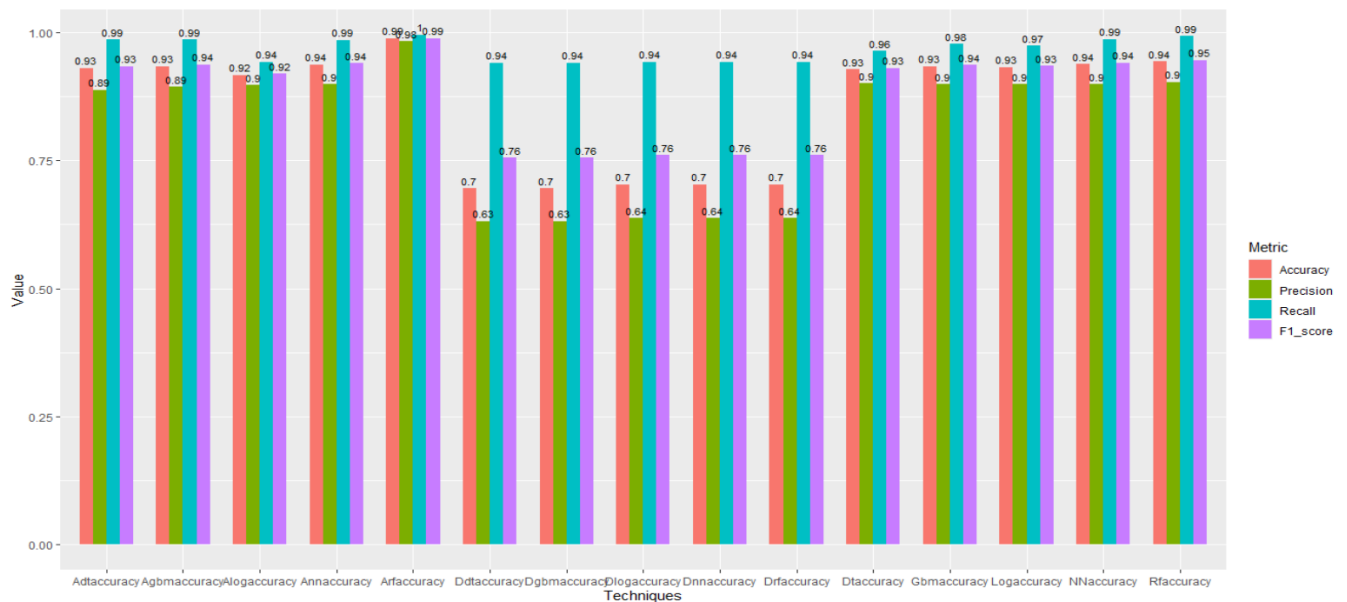


Figure 52: Model Evaluation

### **3.3.6 Deployment**

Deployment phase involves taking the results of the models generated and integrating them into the stakeholders identified in metro rail and other transportation stations

Deployment as the name says talks about the predictive maintenance models being implemented into the metro railway station system and being used for operational purposes. The integration of the system in a live metro railway station production environment will help in forecasting failures and making pre-emptive decisions to avoid or reduce potential downtime. This will reduce disruptions in services due to faulty trains, and inadvertently increase public perception of the railway service.

## 4.0 IMPLEMENTATION ANALYSIS

### 4.1 Data Analysis Tools

Some of the data analysis tools used during this report are:

Some of the data analysis techniques adopted include:

1. **Excel**

The dataset was downloaded in comma-separated values or otherwise known as csv format from the Metro PT-3 website which gave a general preview of all the layout and structure of the dataset (Nelson, 2002).

2. **R programming:**

R programming is a statistical analysis and data analysis tool used in performing experiments during the project. During the data manipulation process, numerous libraries such as lubridate, Hmisc, DescTools, dplyr, caTools, etc were imported for their statistical framework and computing capabilities (McNulty, 2021).

3. **Tableau Visualization**

Due to its ease of use and accessibility, Tableau will be adopted by Metro Railway Station for its dashboard and report functionalities. It will show a dashboard contain various information regarding the metro rail components, detailing its lifespan, maintenance schedules, predicted downtime and failure rates amongst other things. This interactive reporting dashboard will ultimately the decision-making process (Zhou, 2023), (Milligan and Guillevin, 2018).

## 4.2 Data Analysis Technique

Data analysis techniques refer to the processes and techniques adopted in transforming the raw data collected from MetroPT-3, the steps taken in processing the dataset, and interpreting them into information.

Some of the data analysis techniques adopted include:

1. **Descriptive Analysis:** This is done to better understand the trends and patterns observed in the dataset.
2. **Data Pre-processing:** Data pre-processing are the steps taken to ensure data quality and data reliability. Dataset is cleaned and checked for outliers.
3. **Exploratory Data Analysis (EDA):** This involves exploring the dataset further to understand the patterns, trends, and relationships between the variables.
4. **Feature Engineering:** This involves the manipulation of the dataset by adding a new column to contain the failures registered on the APU. The relevant features are extracted.
5. **Data Sampling:** A class imbalance was observed during the data count process and the hybrid sampling approach was adopted to ensure a more comprehensive and informative dataset sample.
6. **Feature selection:** A subset of the dataset is extracted based on the most prevalent features that will yield the highest levels of accuracy.
7. **Model Development:** MetroPT-3 dataset is split into using the `train_test_split` function after feature selection is done, ML and DL techniques are applied to train and make predictions. The techniques used are: Logistic Regression, Decision Tree, Random Forest, Gradient Boosting Machine, and Neural Networks.
8. **Model Evaluation:** ML and DL model evaluation is done using Accuracy, Precision, Recall and F1-score and results of the models for the varying datasets are compared against one another.
9. **Deployment:** Predictive maintenance system is deployed and integrated into the metro rail real time monitoring system to check the APU system and other components and accurately predict potential failures.
10. **Conclusion and Reporting:** It confirms whether the proposed hypothesis is correct or incorrect and answers the research questions. Reporting involves communicating the results of the

experiments conducted in an official written format either in terms of publications or presentations.

The application of predictive maintenance techniques would lead to significant improvements in both reliability and efficiency of the metro train operations as it ensures minimal downtimes and reduced maintenance costs while also increasing overall operations and safety.

## 4.3 Statistical Analysis

Statistical analysis involves performing experiments and tests to confirm the hypothesis and provide answers to the research questions (Wang et al., 2022).

In this section, the findings are interpreted from the analysis. It points out the patterns, trends and relationships between the variables therefore answering our research questions and insights into the hypothesis. The bivariate relationship between the variables is checked in Question1 to 3 while question 4 checks the multivariate relationship between the variables.

### Bivariate Analysis

Bivariate analysis works with two variables. It analyses the relationships and dependencies between the pairs of variables. The methods of analysing bivariate relationships are dependent on the type of dataset and variables we are working on. In this dataset, the variables are numerical. Histogram is used to check the normal distribution of the variables to determine which method will be used as done when doing the data normalization check while we use correlation to check the relationship between the said variables to find if there's any significant relationship.

1. Is there a relationship between TP2 and DV\_Pressure.

```
Q1. Is there a relationship between TP2 and DV_Pressure ?
```{r}
# Bivariate Analysis -Correlation coefficient
cor(Metro1$TP2, Metro1$DV_pressure, method = 'spearman')
[1] 0.5503808
```

Figure 53: Correlation Coefficient 1

The figure above shows a result of 0.5 which signifies an average relationship between TP2 and DV pressure.

2. Is there a relationship between MPG and Pressure Switch?

```
Q2. Is there a relationship between MPG and Pressure Switch?
```{r}
Correlation coefficient
cor(Metro1$MPG, Metro1$Pressure_switch, method = 'spearman')
```

[1] 0.1802498
```

Figure 54: Correlation Coefficient 2

The figure above shows a result of 0.18 which signifies a weak relationship between MPG and Pressure Switch.

3. Is there a relationship between H1 and TP2

```
Q3. Is there a relationship between H1 and TP2
```{r}
Correlation coefficient
cor(Metro1$TP2, Metro1$H1, method = 'spearman')
```

[1] -0.3906992
```

Figure 55: Correlation Coefficient 3

The figure above shows a result of -0.39 which suggests a negative relationship between H1 and TP2.

Multivariate Analysis:

Multivariate analysis involves analysis variables that are more than two simultaneously to understand the complexity, patterns, and interactions between the variables.

4. is there a relationship between TP2, DV_Pressure and Aircompfail?

```

Multiple corellation
{r}
#selected_data <- Sheffieldd[, c("TSK", "RAINC", "SMOIS")]
Mulcorrmatrix <- cor(Metro1[, c("TP2", "DV_pressure", "Aircompfail")])
Mulcorrmatrix

```

| | TP2 | DV_pressure | Aircompfail |
|-------------|-----------|-------------|-------------|
| TP2 | 1.0000000 | 0.4150254 | 0.2775826 |
| DV_pressure | 0.4150254 | 1.0000000 | 0.6323020 |
| Aircompfail | 0.2775826 | 0.6323020 | 1.0000000 |

```

{r}
Mulcorrelation <- sqrt(det(Mulcorrmatrix))
Mulcorrelation

```

```

[1] 0.7046866

```

Figure 56: Correlation Coefficient 4

Mulcorrmatrix shows a breakdown of how each variable correlates with one another individually while Mulcorrelation shows the collective relationship between the variables.

The figure above shows a result of 0.7 which suggests a positive and significant relationship between the variables.

4.3 Discussion of Results and Analysis.

4.3.1 Hypothesis Testing Discussion:

1. Accept Null Hypothesis 1: There is a significant correlation between TP2 and DV_Pressure.

The analysis accepts the null hypothesis, indicating a significant moderate correlation between TP2 and DV_Pressure. This suggests that changes in TP2 will affect DV_Pressure.

2. Accept Null Hypothesis 1: There is a significant correlation between MPG and Pressure Switch.
The analysis accepts the null hypothesis, indicating a significant moderate correlation between MPG and Pressure Switch.

3. Reject Null Hypothesis 1: There is no significant correlation between TP2 and H1.
The result -0.3906992 suggesting there is a negative correlation between TP2 and H1.

4. Accept Null Hypothesis 1: There is a significant correlation between TP2, DV_Pressure and Aircompfail.
The result 0.7046866 of the analysis suggests a significant and positive relationship between the variables and therefore accepts the null hypothesis, indicating a significant moderate correlation between TP2, DV_Pressure and Aircompfail.

4.3.2 Analysis Discussion

The machine learning and deep learning techniques initiated are adopted and used in making predictions on the main MetroPT-3 (Metro4) dataset, and the analogue (Analog) and digital (Digital) dataset which are both subsets of the main dataset.

The predictive maintenance techniques specific to classification related experiments used are Logistic Regression, Random Forest, Decision Tree, Gradient Boosting, and Neural Networks. For each of the datasets namely Metro4, Analog and Digital, feature selection was performed, splitting

the model into training and testing dataset, followed by initiating and running both ML and DL techniques mentioned above. Upon running the models, the models are therefore evaluated using Accuracy, Precision, Recall and F1- score which have been detailed in the table below.

The table below shows a breakdown of the models and their evaluation metrics on different datasets observed in the project.

4.3.2.1 MetroPT-3 Dataset.

By comparing the values of the algorithms on theMetroPT-3 dataset, it is observed that Random Forest has the highest level of accuracy compared to the other models. Random Forest has an accuracy of 96% as shown in the figure below.

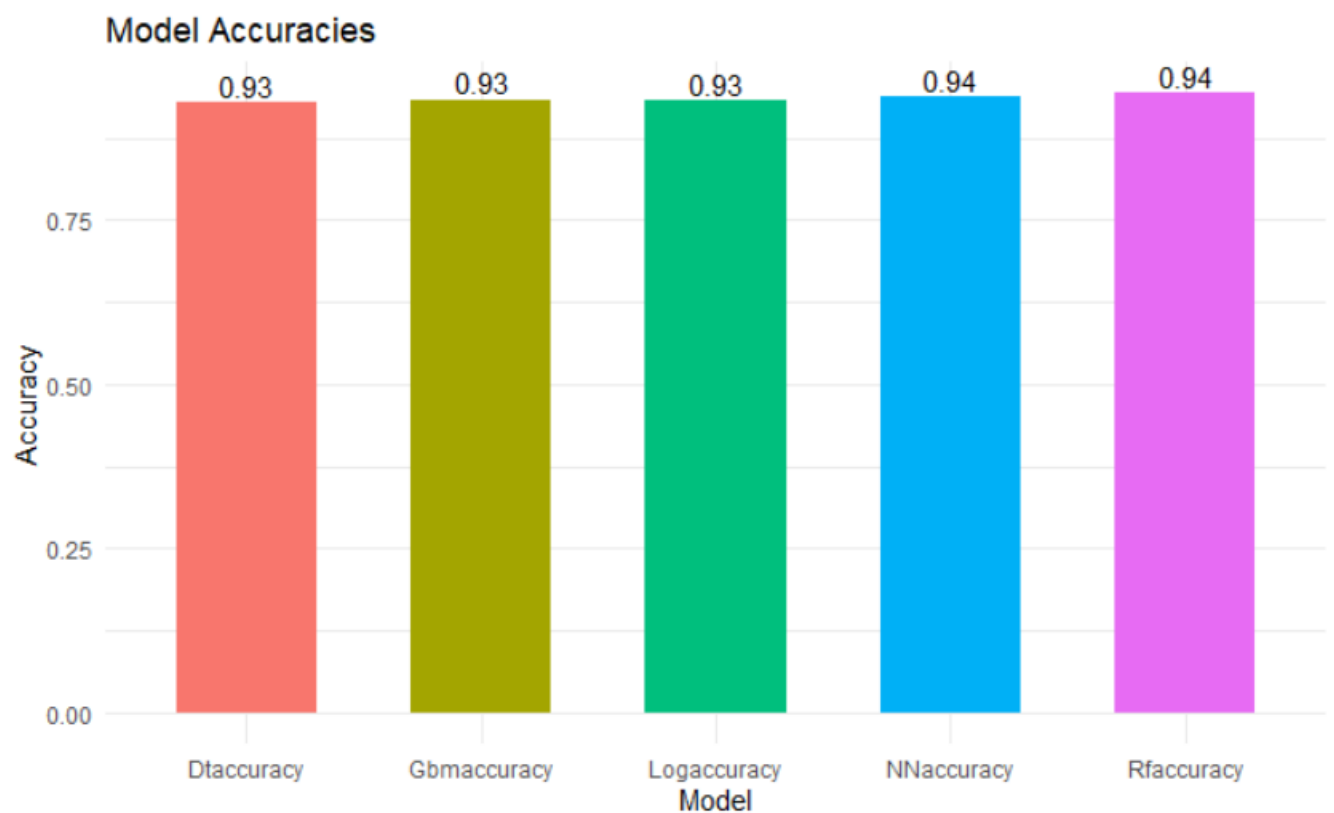


Figure 57: Metro4 Evaluation

The code below is written to print the value of the model with the highest accuracy given the figure above doesn't give conclusive evidence on which model is best.

```

{r}
# Print the highest accuracy and the corresponding model
cat("The highest accuracy is", max_accuracy, "achieved by", best_model, "\n")

```

The highest accuracy is 0.9435982 achieved by Rfaccuracy

Figure 58: Metro4 RF Accuracy

4.3.2.2 Analogue Dataset

The analogue dataset is a subset of the Metro4 dataset. It comprises of continuous data found within the dataset. Features 1-7 of the MetroPt-3 dataset are continuous variables as shown in the figure below.

| TP2 | TP3 | H1 | DV_pressure | Reservoirs | Oil_temperature | Motor_current |
|--------|-------|-------|-------------|------------|------------------|---------------|
| -0.012 | 9.358 | 9.34 | -0.024 | 9.358 | 53.6000000000000 | 0.04 |
| -0.014 | 9.348 | 9.332 | -0.022 | 9.348 | 53.6750000000000 | 0.04 |
| -0.012 | 9.338 | 9.322 | -0.022 | 9.338 | 53.6000000000000 | 0.0425 |
| -0.012 | 9.328 | 9.312 | -0.022 | 9.328 | 53.4249999999999 | 0.04 |

Figure 59: Analogue Sensor Data

```

{r}
# Print the highest accuracy and the corresponding model
cat("The highest accuracy is", Amax_accuracy, "achieved by", Abest_model, "\n")

```

The highest accuracy is 0.98879 achieved by Arfaccuracy

Figure 60: Analogue Sensor Accuracy

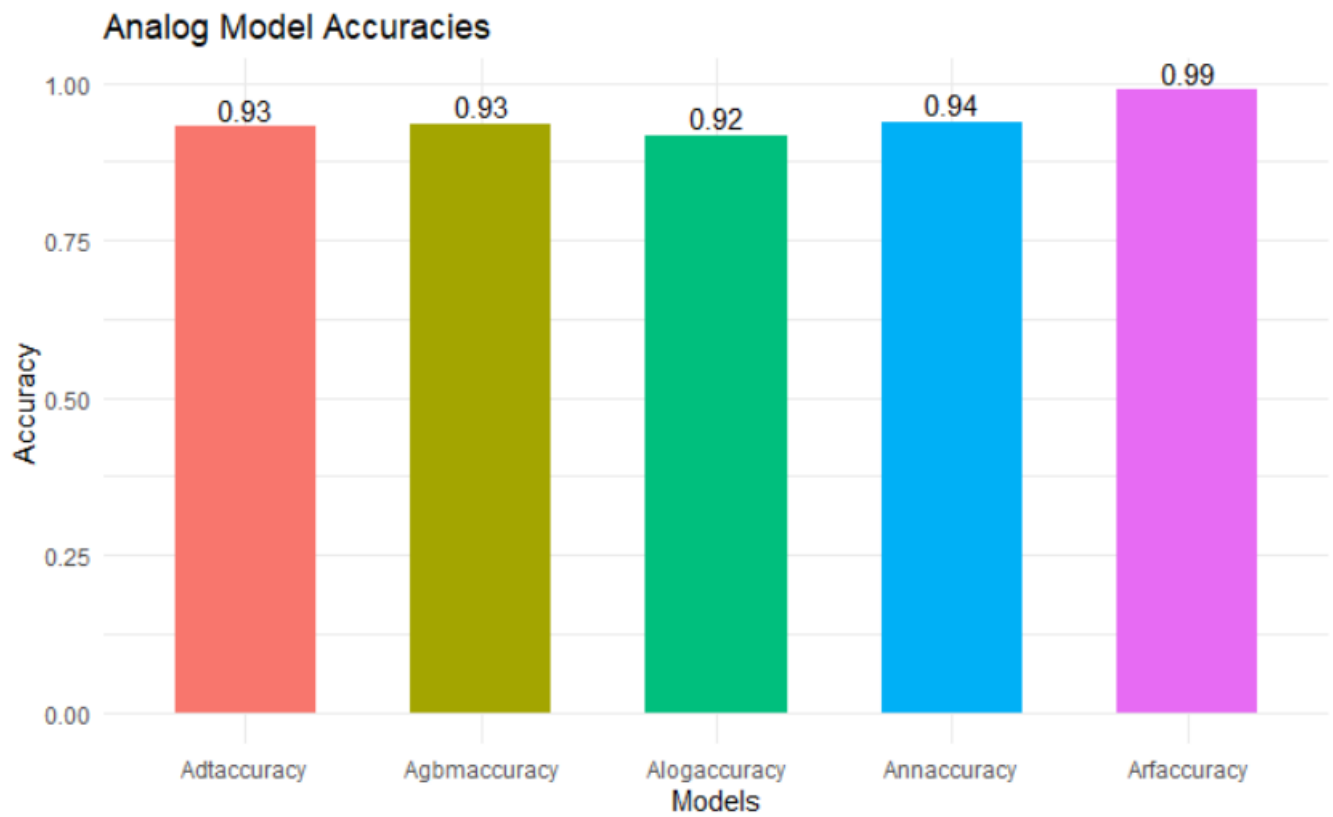


Figure 61: Analogue Model Accuracy

4.3.2.3 Digital Dataset

The Digital dataset is a subset of the MetroPT-3 Dataset. It contains discrete datapoints that signal the state of components. Features 8-15 of the MetroPt-3 dataset are discrete variables as shown in the figure below.

| COMP | DV_eletric | Towers | MPG | LPS | Pressure_switch | Oil_level | Caudal_impulses |
|------|------------|--------|-----|-----|-----------------|-----------|-----------------|
| 1 | 0 | 1 | 1 | 0 | 1 | 1 | 1 |
| 1 | 0 | 1 | 1 | 0 | 1 | 1 | 1 |
| 1 | 0 | 1 | 1 | 0 | 1 | 1 | 1 |

Figure 62: Digital Sensor Data

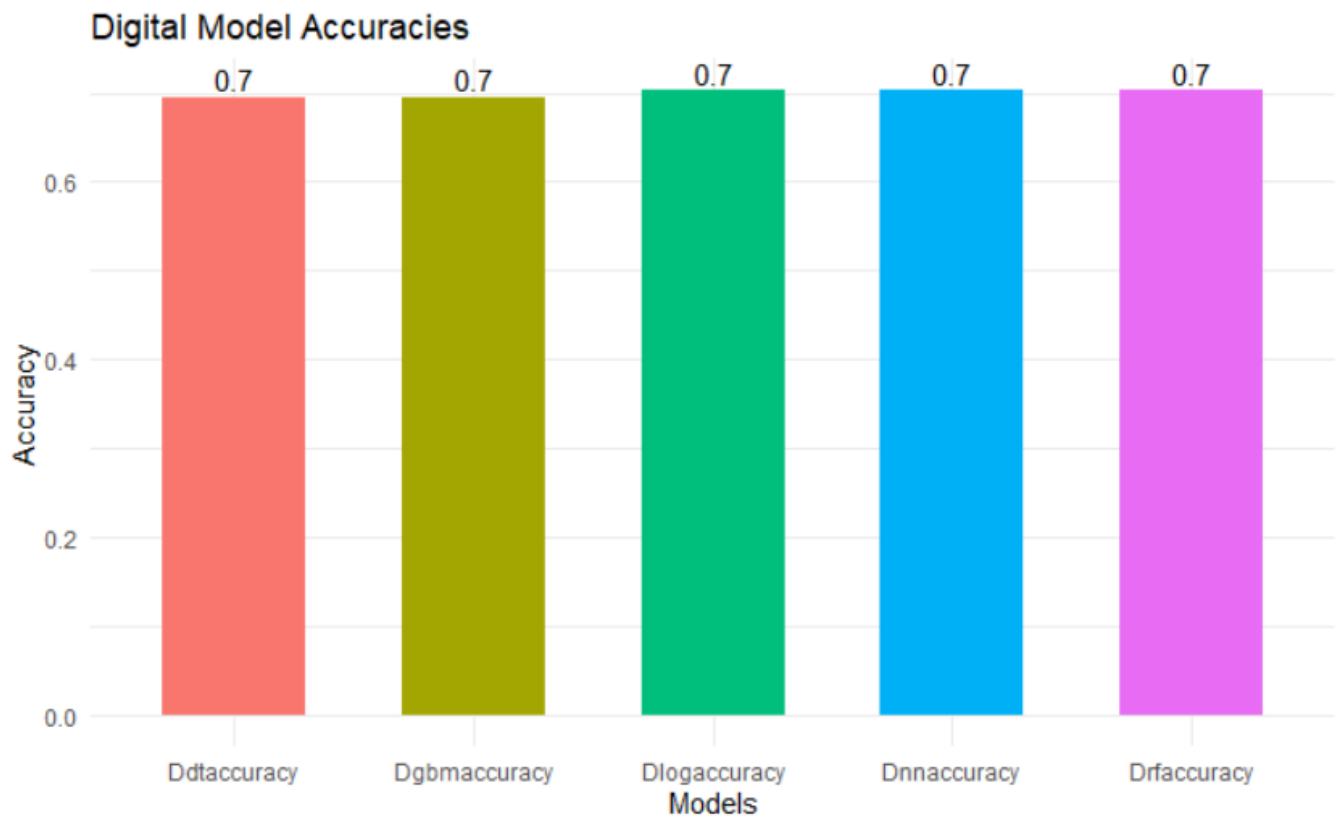


Figure 63: Digital Model Accuracy

```
{r}
# Print the highest accuracy and the corresponding model
cat("The highest accuracy is", Dmax_accuracy, "achieved by", Dbest_model, "\n")
```

The highest accuracy is 0.7028913 achieved by Dlogaccuracy

Figure 64: Digital LOG Accuracy

4.3.2.4 Model Comparison by Accuracy, Precision, Recall and F1-score.

In this section of the documentation, there is a comparison of the evaluation values gotten, this would help us identify which model is best for predictive maintenance purposes and which dataset is best in predicting potential failures.

Metro 4 and Analogue Dataset show that the Random Forest Model has the highest level of accuracy while the Digital Dataset shows Logistic Regression has the highest level of accuracy.

```
# Create a data frame with the accuracies
Evaluationdf <- data.frame(
  Techniques = c("Logaccuracy", "Dtaccuracy", "Rfaccuracy", "Gbmaccuracy", "NNaccuracy",
    "Alogaccuracy", "Adtaccuracy", "Arfaccuracy", "Agbmaccuracy", "Annaccuracy",
    "Dlogaccuracy", "Ddtaccuracy", "Drfaccuracy", "Dgbmaccuracy", "Dnnaccuracy"),
  Accuracy = c(Logaccuracy, Dtaccuracy, Rfaccuracy, Gbmaccuracy, NNaccuracy,
    Alogaccuracy, Adtaccuracy, Arfaccuracy, Agbmaccuracy, Annaccuracy,
    Dlogaccuracy, Ddtaccuracy, Drfaccuracy, Dgbmaccuracy, Dnnaccuracy),
  Precision = c(Logprecision, Dtprecision, Rfprediction, Gbmprecision, NNprecision,
    Alogprecision, Adtprecision, Arfprediction, Agbmprecision, Annprecision,
    Dlogprecision, Ddtprecision, Drfprediction, Dgbmprecision, Dnnprecision),
  Recall = c(Logrecall, Dtrecall, Rfrecall, Gbmrecall, NNrecall,
    Alogrecall, Adtrecall, Arfrecall, Agbmrecall, Annrecall,
    Dlogrecall, Ddtrecall, Drfrecall, Dgbmrecall, Dnnrecall),
  F1_score = c(Logf1score, Dtf1score, Rff1score, Gbmf1score, NNf1score,
    Alogf1score, Adtf1score, Arff1score, Agbmf1score, Annf1score,
    Dlogf1score, Ddtf1score, Drff1score, Dgbmf1score, Dnnf1score)
)
```

Figure 65: Evaluation Data frame

The code above shows a data frame created to accept the values of each evaluation metric used and is therefore used to plot a graph which compares the results against one another as shown in figure 62 below.

Table 2: Model Comparison

| S/N | DATASET | MODEL | ACCURACY | PRECISION | RECALL | F1-SCORE |
|----------------------------|----------------|---------------------|-----------|-----------|-----------|-----------|
| MetroPT-3 (Metro4) Dataset | | | | | | |
| 1 | Metro4 dataset | Logistic Regression | 0.9321386 | 0.8985375 | 0.9743491 | 0.9349089 |
| 2 | Metro4 dataset | Decision Tree | 0.9283949 | 0.9003325 | 0.9635016 | 0.9308466 |
| 3 | Metro4 dataset | Random Forest | 0.9435982 | 0.9035375 | 0.9932804 | 0.9462860 |
| 4 | Metro4 dataset | GBM | 0.9338117 | 0.8989365 | 0.9775755 | 0.9366082 |
| 5 | Metro4 dataset | Neural Network | 0.9380123 | 0.8989984 | 0.9869521 | 0.9409244 |
| Analogue Sensor Dataset | | | | | | |
| 6 | Analog Dataset | Logistic Regression | 0.9170735 | 0.8969832 | 0.9424444 | 0.9191520 |
| 7 | Analog Dataset | Decision Tree | 0.9304853 | 0.8874385 | 0.9860955 | 0.9341695 |
| 8 | Analog dataset | Random Forest | 0.9887900 | 0.9825141 | 0.9953015 | 0.9888664 |
| 9 | Analog Dataset | GBM | 0.9343483 | 0.8937816 | 0.9859109 | 0.9375885 |
| 10 | Analog Dataset | Neural Network | 0.9372720 | 0.8988208 | 0.9855287 | 0.9401799 |

| | | | | | | |
|------------------------|------------------|---------------------|-----------|-----------|-----------|-----------|
| 11 | Analogue Dataset | Hybrid Model | 0.9393089 | 0.9896012 | 0.8991952 | 0.9422346 |
| Digital Sensor Dataset | | | | | | |
| 11 | Digital Dataset | Logistic Regression | 0.7028913 | 0.6372372 | 0.9425815 | 0.7604012 |
| 12 | Digital Dataset | Decision Tree | 0.6955602 | 0.6313136 | 0.9407113 | 0.7555654 |
| 13 | Digital Dataset | Random Forest | 0.7028913 | 0.6372372 | 0.9425815 | 0.7604012 |
| 14 | Digital Dataset | GBM | 0.6955602 | 0.6313136 | 0.9407113 | 0.7555654 |
| 15 | Digital Dataset | Neural Network | 0.7028913 | 0.6372372 | 0.9425815 | 0.7604012 |

From the analysis and the comparison of values in the dataset, it is deduced that Random Forest has the highest and most consistent evaluation metric amongst the other models. It also shows that in this study, the analogue dataset being used gives the best results compared to the digital and entire MetroPT-3 dataset.

After concluding that Analogue dataset is the best dataset for predicting failures in this study, a hybrid model approach is then initiated using the two best models which are Random Forest and Neural Network as shown in the figure below. The hybrid model sums the values of the random forest and neural network and finds an average. The average is then used to calculate its probability index, and the confusion matrix is extracted giving an Accuracy result of 0.9393089 which now makes it the second-best model after Random Forest.

```

# 8. Analogue Hybrid Model
...{r}
length(Annpredclass)
length(Arfpredclass)

Arfpredict_pos_class <- Arfpredict[, 2]
length(Arfpredict_pos_class)

class(Arfpredict_pos_class)
class(Annpredclass)

# Combined model using Random Forest and Neural Network by averaging of the probabilities
combined_predictions <- (Arfpredict_pos_class + (as.numeric(as.character(Annpredclass)))) / 2

# Step 5: Convert Combined Predictions to Class Labels
combined_predictions_class <- ifelse(combined_predictions > 0.5, 1, 0)
length(combined_predictions_class)
...

```

Figure 66: Analogue Hybrid Model

```

Ahyaccuracy
Ahyrecall
Ahyprecision
Ahyf1score
...

  Accuracy
0.9393089
  Recall
0.9896012
Precision
0.8991952
      F1
0.9422346

```

Figure 67: Hybrid Model Performance

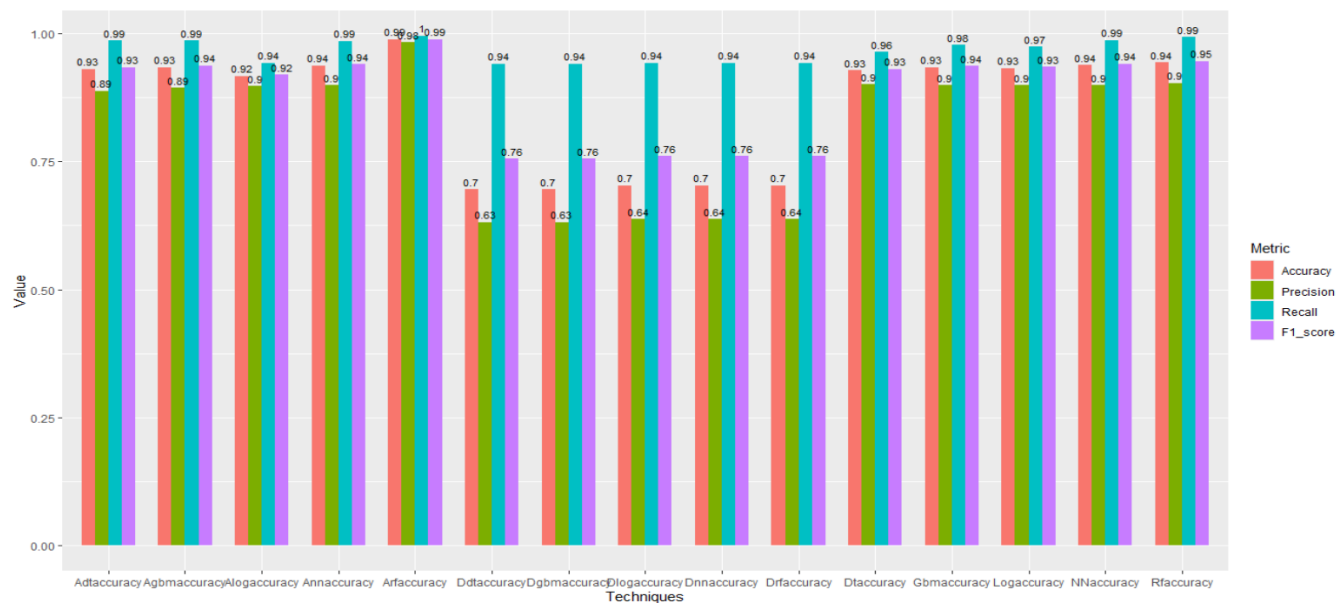


Figure 68: Model Comparison

4.4 Potential Challenges and Mitigation Strategies

This report shows several significant factors to consider such as improved operational efficiency, optimized equipment lifecycle management, advanced machine learning models, etc.

Table 3: Challenges and Mitigation

| S/n | Challenge | Mitigation |
|-----|-----------------------------------|--|
| 1 | Data quality and accessibility | Handle Null values, check for true outliers and handle them. |
| 2 | Fairness and Bias | Data collection is fair as it is real time data from a working machinery. Models use is also crosschecked and validated. |
| 3 | Dataset Imbalance | Perform hybrid sampling to address class imbalance |
| 4 | Model Development and Validation | Initiate ML and DL algorithms to perform predictive analysis and make use of Accuracy, Recall, Precision and F1-score to validating the performance results of the models. |
| 5 | Integration with Existing Systems | Engage Metro Railway Stations staffs, Stakeholders and IT Department on ease of integration into the existing systems. |
| 6 | Data Privacy and Security | Implement data security protocols and controls.

Perform regular system audits and checks. |
| 7 | Transparency and Accountability | Ensure full transparency of model and decision process. Assign stakeholder to be accountable to verify the decision-making process |

Some of the challenges and mitigation being considered during the project are:

Challenge 1. Data Quality and Accessibility: Predictive maintenance in assets has a reliance on high-quality and reliable data. Different sectors must adhere to strict processes to ensure proper data collection is done and limit access to those who have access to manipulate the data (Lee et al., 2020).

Mitigation: Perform pre-processing techniques to address the null values, outliers.

Challenge 2. Model Development and Validation: There is a level of expertise required in predictive maintenance. Organizations should ensure they invest in quality and well-versed personnel to develop and ensure quality (Ersöz et al., 2022).

Mitigation: Perform regression analysis to pick the best regression models for hybrid modelling and perform accuracy checks.

Challenge 3. Integration with Existing Systems: The integration of the ML model into existing infrastructure poses a challenge to the interoperability and compatibility of solutions.

Mitigation: Perform data harmonization checks and compatibility assessments while engaging stakeholders involved in the project and the project IT department to ensure proper requirement gathering and system integration. Also, provide proper detailed documentation for knowledge transfer.

Challenge 4. Data Privacy and Security: Predictive maintenance relies on access to sensitive data, including equipment performance metrics and maintenance logs. Organizations must ensure the privacy of customers and ensure proper security while maintaining regulatory requirements, such as GDPR or CCPA (Baig et al., 2020) , (FLICHE and YANG, 2018).

Mitigation: Implement security protocols that align with the GDPR, HIPAA, etc. regulations and perform regular system audits while monitoring system access and usage.

Challenge 5. Imbalanced Dataset: Due to the class imbalance in the dependent variables i.e. failure, it poses a significant challenge as this can result in biased models and therefore affect the decision-making process.

Mitigation: A hybrid sampling technique that involves overfitting the minority class and underfitting the majority class to create a balanced data is used to address the observed class imbalance in the dataset leading to improved performance levels.

Challenge 6. Fairness and Bias Mitigation: Machine learning algorithms to be used must be bias-free. It should ensure there is adequate data collection from different sectors and different fields. (Dwork et al., 2012), (Selbst et al., 2019).

Mitigation: Ensure the data collected is not tampered with and collected directly from the data source and there is no class imbalance.

Challenge 7. Transparency and Accountability: Predictive maintenance decisions can have significant implications for equipment maintenance, resource allocation, and operational efficiency. (Lepri et al., 2016), (IOSCO, 2020).

Mitigation: Ensure there is full transparency in the decision-making process and proper accountability of roles.

4.5 Research Considerations

4.5.1 Practical Implications

A practical implication of this approach includes the integration of the hybrid model techniques used in other systems and sectors. It is possible to scale the use of this research into other systems as the model learns from the input variables provided irrespective of the different systems and societies. The practical usage of this approach would aid in machine or equipment failure predicting and suggest solutions and alternatives and actively provide into industrial applications or equipment in need of maintenance.

4.5.2 Ethical Implications

Some ethical implications of this approach to be considered have been listed in the solution considerations. Although the model is designed to aid in predictive maintenance and decision support, it is also important to consider some ethical considerations such as data privacy, possible algorithmic bias etc. Another implication of this research model is the possibility of jobs being displaced as a technological solution has been designed and automated to perform the jobs of some specified people. The subject of fairness, security, transparency, and levels of accountability remains an ongoing conversation according to the GDPR and OECD frameworks. (Shemtob, 2021), (OECD, 2021) , (OECD.AI, n.d.), (Myers and Nejkov, 2020), (Hamidovic et al., 2019).

The OECD Council adopted the Recommendation on AI at Ministerial level on 22-23 May 2019. The OECD AI Principles focus on how governments can shape a human-centric approach to trustworthy AI, and aim to promote the use of AI that is innovative and trustworthy, respecting human rights and democratic values.

The Recommendation identifies five complementary values-based principles for the responsible stewardship of trustworthy AI:

- AI should benefit people and the planet by driving inclusive growth, sustainable development and well-being.
- AI systems should be designed in a way that respects the rule of law, human rights, democratic values and diversity, and they should include appropriate safeguards – for example, enabling human intervention where necessary – to ensure a fair and just society.
- There should be transparency and responsible disclosure around AI systems to ensure that people understand AI-based outcomes and can challenge them.
- AI systems must function in a robust, secure and safe way throughout their life cycles and potential risks should be continually assessed and managed.
- Organisations and individuals developing, deploying or operating AI systems should be held accountable for their proper functioning in line with the above principles.

The OECD also provides five recommendations to governments:

- Facilitate public and private investment in research & development to spur innovation in trustworthy AI.
- Foster accessible AI ecosystems with digital infrastructure and technologies and mechanisms to share data and knowledge.
- Ensure a policy environment that will open the way to deployment of trustworthy AI systems.
- Empower people with the skills for AI and support workers for a fair transition.
- Co-operate across borders and sectors to progress on responsible stewardship of trustworthy AI.

Figure 69: OECD Principles

Source: (OECD, 2021)

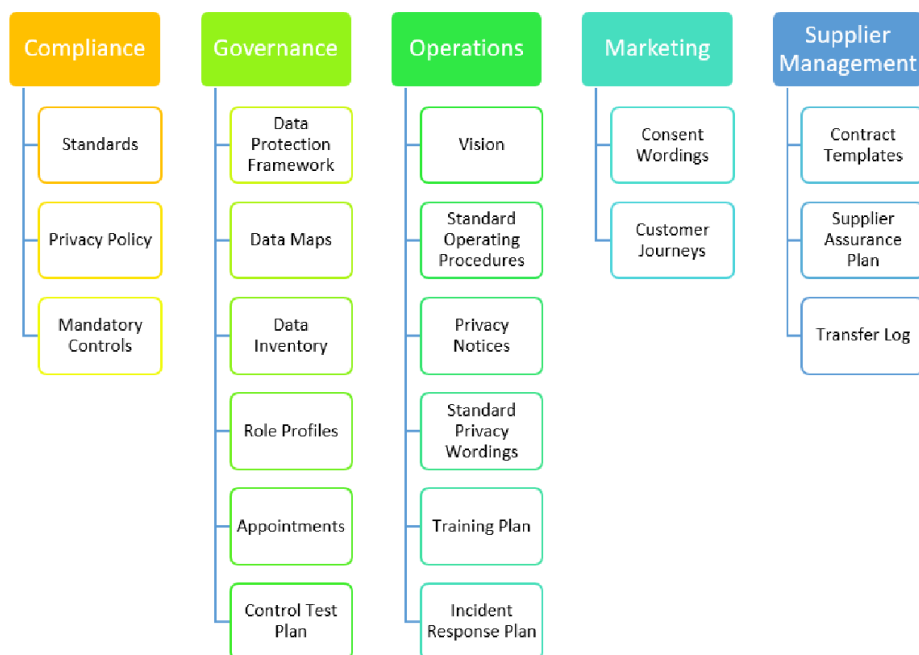


Figure 70: GDPR Framework

Source: (Alweis, 2018)

5.0 CONCLUSION

In conclusion, this research document shows a prevalent increase in the deployment and integration of big data solutions in the enhancement and optimization of Asset and Equipment Lifecycle Management with a keen interest in industrial application uses such as the metro railway stations system.

Industrialization has brought about an increase in the integration of big data and machine learning technologies in applications and equipment management through the continuous advancements being made in optimizing equipment Lifecycle management to aid decision support therefore making it possible to improve operational efficiency, effectiveness and mitigate against downtime(s) and reaction times needed. By adopting a proactive approach using advanced predictive maintenance techniques such as Logistic Regression, Random Forest, Decision Tree, Gradient Boosting Machine, Neural Networks in detecting equipment failures, industries can make proactive quick decisions that help mitigate against downtimes, improve operational effectiveness and efficiency.

The innovate approach of using advanced predictive models on different iterations of the dataset such as the MetroPT-3, Analogue and Digital Datasets gives an insight into how feature engineering, data sampling and feature selection can significantly improve the accuracy of models used with the Analogue Random Forest model achieving an Accuracy and F1-score of 98.8%.

Notwithstanding, the impact of this model on Asset and Equipment Lifecycle Management, it is essential for us to consider the practical and ethical use and implication of this research on data privacy and security. Another aspect to consider is the role of transparency, fairness, model understanding, and explanation as this play's vital roles in the accuracy and effectiveness of the model used.

It is imperative for further research and contributions to be made, with a focus on the integration of diverse data sources and features to enhance the performance of the ML and DL models. Another area of focus should be the initiation of more DL techniques such as Recurrent Neural Network (RNN), Convolutional Neural Networks (CNNs), Long Short-Term Memory (LSTM) networks, and Autoencoders as they are widely regarded for being able to handle complexities in datasets.

5.1 Answering Research Questions

Table 4: Research Question and Answers

| s/n | Research Question | Answers |
|-----|--|--|
| 1 | What contemporary ML and DL techniques are prevalent in addressing Asset and Equipment Lifecycle Management via Predictive Maintenance Systems? | This was answered in the Literature review section. Chapter 2 of the Literature review shows a breakdown of different literatures, its relevance and their impact in the field. |
| 2 | What equipment features and attributes of equipment are the most indicative for forecasting potential failures in Metro Railways | <p>The features most indicative of potential failures in the MetroPT-3 dataset are:</p> <p>The features most indicative of potential failures in the Analogue dataset are: TP2, Reservoirs, Oil temperature and Motor Current.</p> <p>The features most indicative of potential failures in the digital dataset are:</p> |
| 3 | How can predictive maintenance models be optimized using feature engineering, data sampling and feature selection to accurately forecast equipment failures? | <p>Some of the strategies adopted during the study are ensuring data quality by performing pre-processing related tasks, feature engineering, use of advanced machine learning techniques such as Random Forest, Gradient Boosting Machine, Neural Networks etc.</p> <p>Feature engineering is used to extract new variables from information provided for the failure events.</p> |

| | | |
|---|--|---|
| | | <p>Hybrid Data Sampling is used to create a balanced dataset between the majority and minority class of the failure events.</p> <p>Feature selection is used to select the best variables with the most predictive values for predicting failure events.</p> <p>These strategies help to optimize predictive maintenance techniques for more accurate and timely forecasts.</p> |
| 4 | Which ML or DL modelling technique is most effective for predicting equipment failures and what performance metric should be prioritized in evaluating models? | <p>Random Forest gives is the preferred technique due to its robustness and ability to handle different complexities accounted in the dataset.</p> <p>The results for each model was evaluated using Accuracy, Precision, Recall and F1-score and the Analogue Random Forest Model gives the highest performance of 98.8%.</p> |
| 5 | How will the implementation and integration of a predictive maintenance system benefit Metro Railways and other industries and what impact would it have on their decision-making process? | <p>The direct beneficiary of this study is the Metro Railway System. Integration would lead to less downtime, a reduction in maintenance cost and improved downtimes. Industries can also now switch from their reactive approach to a proactive approach.</p> <p>It also leads to resource allocation optimization, enhancement in operational efficiency, and safety improvement by ensuring asset and equipment reliability.</p> |

5.2 Limitations/ Future Works

Some of the things to consider in terms of further research and future works include the integration of more sensor data, environmental data, and temperature for more accurate predictions. This is to be done to consider the other conditions that might influence the failures of the equipment. Another field to consider is the application of the models in other sectors and other use cases.

Although the research is primarily focused on Metro Railway System, further study can be conducted considering other forms of data and other sectors. The approach used in this research is scalable to other sectors and systems that require maintenance.

Part of the future work considered should address the identified gaps by introducing a broader range of data sources, and explore advanced deep learning techniques like Recurrent Neural Networks (RNN), Convolutional Neural Networks (CNN), Long-Short Term Memory (LSTM), and Autoencoders

5.3 Research schedule

Table 5: Research Schedule

| Research phase | Objectives | Deadline |
|----------------|---|------------|
| Prepare Phase | Choose a research topic,

Approve topic,

Data collection | 06/06/2024 |
| Explore Phase | Data understanding,

Formulate hypothesis and
research questions,

Extensive literature review | 14/06/2024 |
| Design Phase | Data preparation,

Handle missing values and
outliers,

Data sampling,

Feature selection,

Machine learning,

Deep learning techniques,
Evaluating and deploying the
model. | 18/07/2024 |

| | | |
|--------------|---|------------|
| Report Phase | Documenting experiment results,

Answering hypotheses and research questions from experimentation result. | 01/08/2024 |
|--------------|---|------------|

REFERENCE

- Abhari, B. (2024) 'Intrusion Detection Using Heterogeneous Data Sources.' *University of Calgary*, January, p. 118.
- Abusitta, A., Silva De Carvalho, G. H., Abdel Wahab, O., Halabi, T., Fung, B. C. M. and Al Mamoori, S. (2022) 'Deep Learning-Enabled Anomaly Detection for IoT Systems.' *SSRN Electronic Journal*.
- Achouch, M., Dimitrova, M., Ziane, K., Sattarpanah Karganroudi, S., Dhouib, R., Ibrahim, H. and Adda, M. (2022) 'On Predictive Maintenance in Industry 4.0: Overview, Models, and Challenges.' *Applied Sciences*. Multidisciplinary Digital Publishing Institute, 12(16) p. 8081.
- Alamr, A. and Artoli, A. (2023) 'Unsupervised Transformer-Based Anomaly Detection in ECG Signals.' *Algorithms*, 16(3) p. 152.
- Alweis, L. (2018) *Top 10 GDPR Frameworks. The EU's General Data Protection...* | by Lauren Alweis | *Alpin.io* | *Medium*. Medium. [Online] Available from: <https://medium.com/alpin-io/top-10-gdpr-frameworks-ec5ad4bfdeab> [Accessed on 12th December 2023].
- Amram, M., Dunn, J., Toledano, J. J. and Zhuo, Y. D. (2021a) 'Interpretable predictive maintenance for hard drives.' *Machine Learning with Applications*, 5, September, p. 100042.
- Amram, M., Dunn, J., Toledano, J. J. and Zhuo, Y. D. (2021b) 'Interpretable predictive maintenance for hard drives.' *Machine Learning with Applications*, 5, September, p. 100042.
- Baig, H., Yogesh, K. S. and Syed, Z. A. (2020) 'Privacy-Preserving in Big Data Analytics: State of the Art.' *In*. Amity University Dubai, United Arab Emirates: Elsevier, p. 6.
- Barros, M., Veloso, B., Pereira, P. M., Ribeiro, R. P. and Gama, João (2020) 'Failure Detection of an Air Production Unit in Operational Context.' *In* Gama, Joao, Pashami, S., Bifet, A., Sayed-Mouchawe, M., Fröning, H., Pernkopf, F., Schiele, G., and Blott, M. (eds) *IoT Streams for Data-Driven Predictive Maintenance and IoT, Edge, and Mobile for Embedded Machine Learning*. Cham: Springer International Publishing (Communications in Computer and Information Science), pp. 61–74.
- Beinschroth, J. (2022) 'Implementing an effective qualitative risk analysis.' *In* 2022 IEEE 10th Jubilee International Conference on Computational Cybernetics and Cyber-Medical Systems (ICCC). Reykjavík, Iceland: IEEE, pp. 000143–000148.
- Bergmann, J. (2024) 'Research Philosophy, Methodological Implications, and Research Design.' *In* *At Risk of Deprivation*. Wiesbaden: Springer Fachmedien Wiesbaden (Studien zur Migrations- und Integrationspolitik), pp. 57–89.

- Berroukham, A., Housni, K., Lahraichi, M. and Boulfrifi, I. (2023) 'Deep learning-based methods for anomaly detection in video surveillance: a review.' *Bulletin of Electrical Engineering and Informatics*, 12(1) pp. 314–327.
- Brownlee, J. (2020) *Data Preparation for Machine Learning: Data Cleaning, Feature Selection, and Data Transforms in Python*. Machine Learning Mastery.
- Budai, G., Huisman, D. and Dekker, R. (2006) 'Scheduling preventive railway maintenance activities.' *Journal of the Operational Research Society*, 57(9) pp. 1035–1044.
- Campbell, J. D., Jardine, A. K. S. and McGlynn, J. (2016) *Asset Management Excellence: Optimizing Equipment Life-Cycle Decisions*. Second Edition, Boca Raton, Fla: CRC Press.
- Carvalho, T. P., Soares, F. A. A. M. N., Vita, R., Francisco, R. da P., Basto, J. P. and Alcalá, S. G. S. (2019) 'A systematic literature review of machine learning methods applied to predictive maintenance.' *Computers & Industrial Engineering*, 137, November, p. 106024.
- Cazacu, M. and Titan, E. (2020) 'Adapting CRISP-DM for Social Sciences.' *BRAIN. Broad Research in Artificial Intelligence and Neuroscience*, 11(2sup1) pp. 99–106.
- Chauhan, S. and Vig, L. (2015) 'Anomaly detection in ECG time signals via deep long short-term memory networks.' In *2015 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*. Campus des Cordeliers, Paris, France: IEEE, pp. 1–7.
- Chen, C., Liu, Y., Wang, S., Sun, X., Di Cairano-Gilfedder, C., Titmus, S. and Syntetos, A. A. (2020) 'Predictive maintenance using cox proportional hazard deep learning.' *Advanced Engineering Informatics*, 44, April, p. 101054.
- Chen, K., Pashami, S., Fan, Y. and Nowaczyk, S. (2019) 'Predicting Air Compressor Failures Using Long Short Term Memory Networks.' In Moura Oliveira, P., Novais, P., and Reis, L. P. (eds) *Progress in Artificial Intelligence*. Cham: Springer International Publishing (Lecture Notes in Computer Science), pp. 596–609.
- Cheng, X., Chaw, J. K., Goh, K. M., Ting, T. T., Sahrani, S., Ahmad, M. N., Abdul Kadir, R. and Ang, M. C. (2022) 'Systematic Literature Review on Visual Analytics of Predictive Maintenance in the Manufacturing Industry.' *Sensors*. Multidisciplinary Digital Publishing Institute, 22(17) p. 6321.
- Coandă, P., Avram, M. and Constantin, V. (2020) 'A state of the art of predictive maintenance techniques.' *IOP Conference Series: Materials Science and Engineering*, 997(1) p. 012039.
- Copeland, M., Soh, J., Puca, A., Manning, M. and Gollob, D. (2015) *Microsoft Azure: planning, deploying, and managing your data center in the cloud*. New York, NY: Apress (The expert's voice in Microsoft Azure).
- Corrales, D. C., Ledezma, A. and Corrales, J. C. (2015) 'A Conceptual Framework for Data Quality in Knowledge Discovery Tasks (FDQ-KDT): A Proposal.' *Journal of Computers*, 10(6) pp. 396–405.
- Daniel, E. (2016) 'The Usefulness of Qualitative and Quantitative Approaches and Methods in Researching Problem-Solving Ability in Science Education Curriculum.' *Journal of Education and Practice*.

Davari, N., Veloso, B., Ribeiro, R. and Gama, J. (2021) 'MetroPT-3 Dataset.' UCI Machine Learning Repository.

Davari, N., Veloso, B., Ribeiro, R. P., Pereira, P. M. and Gama, J. (2021) 'Predictive maintenance based on anomaly detection using deep learning for air production unit in the railway industry.' *In 2021 IEEE 8th International Conference on Data Science and Advanced Analytics (DSAA)*. Porto, Portugal: IEEE, pp. 1–10.

Dawson, C. (2019) *Introduction to research methods: a practical guide for anyone undertaking a research project*. 5th edition, London: Robinson.

Dayo-Olupona, O., Genc, B., Celik, T. and Bada, S. (2023a) 'Adoptable approaches to predictive maintenance in mining industry: An overview.' *Resources Policy*, 86, October, p. 104291.

Dayo-Olupona, O., Genc, B., Celik, T. and Bada, S. (2023b) 'Adoptable approaches to predictive maintenance in mining industry: An overview.' *Resources Policy*, 86, October, p. 104291.

Dwork, C., Hardt, M., Pitassi, T., Reingold, O. and Zemel, R. (2012) 'Fairness through awareness.' *In Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*. Cambridge Massachusetts: ACM, pp. 214–226.

Ersöz, O. Ö., İnal, A. F., Aktepe, A., Türker, A. K. and Ersöz, S. (2022) 'A Systematic Literature Review of the Predictive Maintenance from Transportation Systems Aspect.' *Sustainability*, 14(21) p. 14536.

FLICHE, O. and YANG, S. (2018) 'Artificial intelligence: challenges for the financial sector.'

Gonzalez-Prida, V., Parra, C. and Crespo Márquez, A. (eds) (2022) *Cases on optimizing the asset management process*. Hershey, PA: IGI Global, Business Science Reference (Advances in logistics, operations, and management science (ALOMS) book series).

Goodfellow, I., Bengio, Y. and Courville, A. (2023) *Deep Learning*. Erscheinungsort nicht ermittelbar: Alanna Maldonado.

Gupta, V. A. (2023) 'Metro Rail — Predictive Maintenance Based On Anomaly Detection.' Medium. 11th December. [Online] Available from: <https://medium.com/@vgupta701/metro-rail-predictive-maintenance-based-on-anomaly-detection-0008ffa7a5b7> [Accessed on 6th September 2024].

Gurney, K. (2018) *An introduction to neural networks*. 2nd ed., Place of publication not identified: CRC Press.

Hamidovic, H., Kabil, J. and Šehić, E. (2019) 'EU General data protection regulation (GDPR) - Anonymisation and pseudonymisation in function of data protection.' *In*.

Hasri, H., Mohd Aris, S. A. and Ahmad, R. (2023) 'Comparison of Auto ARIMA and Auto SARIMA Performance in COVID-19 Prediction.' *In 2023 IEEE 2nd National Biomedical Engineering Conference (NBEC)*. Melaka, Malaysia: IEEE, pp. 106–110.

Hotz, N. (2018) 'What is CRISP DM?' Data Science Process Alliance. 10th September. [Online] Available from: <https://www.datascience-pm.com/crisp-dm-2/> [Accessed on 10th March 2024].

Huang, N. E. and Wu, Z. (2008) 'A review on Hilbert-Huang transform: Method and its applications to geophysical studies.' *Reviews of Geophysics*, 46(2) p. 2007RG000228.

IOSCO (2020) *The use of artificial intelligence and machine learning by market intermediaries and asset managers*. <https://www.iosco.org/>. [Online] Available from: <https://www.iosco.org/> [Accessed on 10th December 2023].

Islam, M. R., Begum, S. and Ahmed, M. U. (2024) 'Artificial Intelligence in Predictive Maintenance: A Systematic Literature Review on Review Papers.' In Kumar, U., Karim, R., Galar, D., and Kour, R. (eds) *International Congress and Workshop on Industrial AI and eMaintenance 2023*. Cham: Springer Nature Switzerland (Lecture Notes in Mechanical Engineering), pp. 251–261.

Kanawaday, A. and Sane, A. (2017) 'Machine learning for predictive maintenance of industrial machines using IoT sensor data.' In *2017 8th IEEE International Conference on Software Engineering and Service Science (ICSESS)*. Beijing, China: IEEE, pp. 87–90.

Kandel, B. (2020) 'Qualitative Versus Quantitative Research,' 1, September, pp. 1–5.

Kaparthi, S. and Bumblauskas, D. (2020) 'Designing predictive maintenance systems using decision tree-based machine learning techniques.' *International Journal of Quality & Reliability Management*. Emerald Publishing Limited, 37(4) pp. 659–686.

Keet, C. M. (2018) *An introduction to ontology engineering*. London: College Publications.

Khanday, S. and Khanam, D. (2023) 'THE RESEARCH DESIGN,' 06, February, p. 376.

Kuhn, M. and Johnson, K. (2020) *Feature engineering and selection: a practical approach for predictive models*. Boca Raton London New York: CRC Press, Taylor & Francis Group (Chapman & Hall/CRC data science series).

Lee, H., Kim, C. F., Kim, M.-S., Kim, Y.-H., Park, H.-K. and Lee, J.-S. (2024) 'Fault Detection of Air Defense Radar Systems Using Machine Learning.' In *2024 18th International Conference on Ubiquitous Information Management and Communication (IMCOM)*, pp. 1–7.

Lee, J., Ni, J., Singh, J., Jiang, B., Azamfar, M. and Feng, J. (2020) 'Intelligent Maintenance Systems and Predictive Manufacturing.' *Journal of Manufacturing Science and Engineering*, 142(11) p. 110805.

Lee, W.-J. (2020) 'Anomaly Detection and Severity Prediction of Air Leakage in Train Braking Pipes.' *International Journal of Prognostics and Health Management*, 8(3).

Lepri, B., Staiano, J., Sangokoya, D., Letouzé, E. and Oliver, N. (2016) 'The Tyranny of Data? The Bright and Dark Sides of Data-Driven Decision-Making for Social Good,' December.

Li, G. and Jung, J. J. (2023) 'Deep learning for anomaly detection in multivariate time series: Approaches, applications, and challenges.' *Information Fusion*, 91, March, pp. 93–102.

- Lohr, S. L. (2022) *Sampling: design and analysis*. Third edition, Boca Raton London New York: CRC Press, Taylor & Francis Group (Chapman & Hall/CRC texts in statistical science).
- Löwen, U., Maier, U., Zhao, H. and Chen, J. (2019) *Use Case 'Equipment Lifecycle Management.'*
- Löwen, U., Maier, U., Zhao, H. T. and Chen, J. N. (2019) 'Use Case "Equipment Lifecycle Management."' *Federal Ministry of Economic Affairs and Energy*. Unpublished, March, p. 21.
- Lu, Y. and Lohr, S. L. (2021) *R Companion for Sampling: Design and Analysis, Third Edition*. CRC Press.
- Luengo, J., García-Gil, D., Ramírez-Gallego, S., García, S. and Herrera, F. (2020) *Big data preprocessing: enabling smart data*. Cham: Springer.
- Marczyk, G. R., DeMatteo, D. and Festinger, D. (2005) *Essentials of research design and methodology*. Hoboken, N.J: John Wiley & Sons (Essentials of behavioral science series).
- Mbanaso, U. M., Abrahams, L. and Okafor, K. C. (2023) 'Research Philosophy, Design and Methodology.' *In Research Techniques for Computer Science, Information Systems and Cybersecurity*. Cham: Springer Nature Switzerland, pp. 81–113.
- McNulty, K. (2021) *Handbook of Regression Modeling in People Analytics: With Examples in R and Python*. CRC Press.
- Milligan, J. N. and Guillevin, T. (2018) *Tableau 10 Complete Reference: Transform your business with rich data visualizations and interactive dashboards with Tableau 10*. Packt Publishing Ltd.
- Musa, U., Adebisi, M. O., Adebisi, Abayomi A. and Adebisi, Avodele A. (2023) 'Development of a Machine Learning Model For Big Data Analytics.' *In 2023 International Conference on Science, Engineering and Business for Sustainable Development Goals (SEB-SDG)*, pp. 1–6.
- Myers, G. and Nejkov, K. (2020) *Developing Artificial Intelligence Sustainably: Toward a Practical Code of Conduct for Disruptive Technologies*. International Finance Corporation, Washington, DC.
- Nandhakumar, S. (2023) 'Asset Lifecycle Management: A Complete Guide for 2024 | Infraon.' *Asset Lifecycle Management*. 5th May. [Online] Available from: <https://infraon.io/blog/asset-lifecycle-management/> [Accessed on 13th March 2024].
- Nelson, S. L. (2002) *Excel data analysis for dummies*. New York: John Wiley & Sons.
- OECD (2021) *Artificial Intelligence, Machine Learning and Big Data in Finance*.
- OECD.AI (n.d.) *AI-Principles Overview - OECD.AI*. OECD.AI Policy Observatory. [Online] Available from: <https://oecd.ai/en/principles> [Accessed on 12th December 2023].
- Ogu, E., J., O. and O., A. (2016) 'Beginning Research: A First-Principle Guide for Even the Computer Science Researcher.' *International Journal of Advanced Research in Computer Science*, 7, April, pp. 79–92.

- Pan, Q., Bao, Y. and Li, H. (2023) 'Transfer learning-based data anomaly detection for structural health monitoring.' *Structural Health Monitoring*, 22(5) pp. 3077–3091.
- Paolanti, M., Romeo, L., Felicetti, A., Mancini, A., Frontoni, E. and Loncarski, J. (2018) 'Machine Learning approach for Predictive Maintenance in Industry 4.0.' *In*, pp. 1–6.
- R. Patil, C., K. Jadhav, S., L. Bardiya, A., P. Davande, A. and P. Raverkar, M. (2023) 'Machine Learning-Based Predictive Maintenance of Industrial Machines.' *International Journal of Computer Trends and Technology*, 71(3) pp. 50–56.
- Schröer, C., Kruse, F. and Gómez, J. M. (2021) 'A Systematic Literature Review on Applying CRISP-DM Process Model.' *Procedia Computer Science*, 181 pp. 526–534.
- Selbst, A. D., Boyd, D., Friedler, S. A., Venkatasubramanian, S. and Vertesi, J. (2019) 'Fairness and Abstraction in Sociotechnical Systems.' *In Proceedings of the Conference on Fairness, Accountability, and Transparency*. Atlanta GA USA: ACM, pp. 59–68.
- Shanthamallu, U. S. and Spanias, A. (2022) *Machine and Deep Learning Algorithms and Applications*. Springer Nature.
- Sharma, M. and Gupta, R. (2023) 'The Significance of using Data Extraction Methods for an Effective Big Data Mining Process.' *In 2023 2nd International Conference for Innovation in Technology (INOCON)*, pp. 1–4.
- Shemtob, L. (2021) 'GDPR versus GDPR.' *British Journal of General Practice*, 71(710) pp. 419–419.
- Sol, K. and Heng, K. (2022) 'Understanding epistemology and its key approaches in research.' *Cambodian Journal of Educational Research*, 2(2) pp. 80–99.
- Sujjaviriyasup, T. and Pitiruek, K. (2017) 'A comparison between MODWT-SVM-DE hybrid model and ARIMA model in forecasting primary energy consumptions.' *In 2017 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*. Singapore: IEEE, pp. 799–802.
- Tatineni, S. (2020) 'AN INTEGRATED APPROACH TO PREDICTIVE MAINTENANCE USING IOT AND MACHINE LEARNING IN MANUFACTURING.' *INTERNATIONAL JOURNAL OF ELECTRICAL ENGINEERING & TECHNOLOGY*, 11, October, pp. 251–265.
- Tran, Trung, Truong, T., Tran, Tung, Hải, N. and Đào, Q. (2021) 'An overview of the application of machine learning in predictive maintenance.' *Petrovietnam Journal*, 10, November, pp. 47–61.
- Urbani, M., Brunelli, M. and Punkka, A. (2023) 'An approach for bi-objective maintenance scheduling on a networked system with limited resources.' *European Journal of Operational Research*, 305(1) pp. 101–113.
- Veloso, B., Ribeiro, R. P., Gama, J. and Pereira, P. M. (2022) 'The MetroPT dataset for predictive maintenance.' *Scientific Data*, 9(1) p. 764.

- Waheeb, W., Ghazali, R. and Shah, H. (2019) 'Nonlinear Autoregressive Moving-average (NARMA) Time Series Forecasting Using Neural Networks.' *In 2019 International Conference on Computer and Information Sciences (ICIS)*, pp. 1–5.
- Wang, J. (2021) 'Reliability analysis and data driven modelling of railway component failure.' Nanyang Technological University.
- Wang, Y., Chai, H. and Ravishankar, J. (2022) 'Statistical Analysis Methods in Engineering Education Research: A state-of-the-art Review.' *In 2022 IEEE Global Engineering Education Conference (EDUCON)*. Tunis, Tunisia: IEEE, pp. 438–444.
- West, S. and Pascual, A. (2015) *The use of equipment life-cycle analysis to identify new service opportunities*.
- Yan, J., Meng, Y., Lu, L. and Li, L. (2017) 'Industrial Big Data in an Industry 4.0 Environment: Challenges, Schemes, and Applications for Predictive Maintenance.' *IEEE Access*, 5 pp. 23484–23491.
- Zhao, Y., Ma, Z., Yang, Y., Jiang, W. and Jiang, X. (2020) 'Short-Term Passenger Flow Prediction With Decomposition in Urban Railway Systems.' *IEEE Access*, 8 pp. 107876–107886.
- Zhenhui, L. (2020) 'Data Processing Strategy in Internet of Things Application System.' *In 2020 International Conference on Big Data & Artificial Intelligence & Software Engineering (ICBASE)*, pp. 86–89.
- Zhou, L. (2023) 'An Introduction to Data Visualization.' *Highlights in Science, Engineering and Technology*, 31, February, pp. 60–63.
- Zhu, Z., Lei, Y., Qi, G., Chai, Y., Mazur, N., An, Y. and Huang, X. (2023) 'A review of the application of deep learning in intelligent fault diagnosis of rotating machinery.' *Measurement*, 206, January, p. 112346.

BIBLIOGRAPHY

Alhamed, M. and Rahman, M.H., 2023. A Systematic Literature Review on Penetration Testing in Networks: Future Research Directions. *Applied Sciences*, 13(12), p.6986.

Carrillo-Mondéjar, J., Turtiainen, H., Costin, A., Martínez, J.L. and Suarez-Tangil, G. (2022) HALE-IoT: Hardening Legacy Internet-of-Things devices by retrofitting defensive firmware modifications and implants. *IEEE Internet of Things Journal*, Volume:10, Issue:10.

Da Silva, S. F. O. (2021) Isolated environments for threat detection and mitigation (Doctoral dissertation, Universidade do Porto (Portugal)).

Firmansyah, A. and Sasongko, A.T., 2023. Shielding the Digital Realm with K-Nearest Neighbors in Network Security. *Proceeding International Pelita Bangsa*, 1(01), pp.201-208.

Keskin, S. and Okatan, E. (2023) Machine Learning Methods for Intrusion Detection in Computer Networks: A Comparative Analysis. *International Journal of Engineering and Innovative Research*, 5(3), pp.268-279.

Menzies, T. and Yedida, R. (2021) On the value of oversampling for deep learning in software defect prediction. *IEEE Transactions on Software Engineering*, [Online] Available from: <https://ieeexplore.ieee.org/document/9429914>. [Accessed 14 February 2022].

Kidwai, A., Arya, C., Singh, P., Diwakar, M., Singh, S., Sharma, K. and Kumar, N. (2021) A comparative study on shells in Linux: A review. *Materials Today: Proceedings*, 37, pp.2612-2616.

Kiruthika, M., Charivukalayil, J.J., Chavan, S., Mathew, J.J. and Cardoza, C., 2023. Enhancement of detection mechanisms for HTTP based DoS/DDoS attacks. *ADBU J Eng Technol*, 12(1).

Raikar, M. M. and Meena, S. M. (2021) SSH brute force attack mitigation in Internet of Things (IoT) network: An edge device security measure. In *2021 2nd international conference on secure cyber computing and communications (ICSCCC)* (pp. 72-77). IEEE.

Šimon, M., Huraj, L. and Čerňanský, M. (2015) Performance evaluations of IPTables firewall solutions under DDoS attacks. *Journal of Applied Mathematics, Statistics, and Informatics*, 11(2), pp.35-45.

Sirait, A. and Siddik, M. (2021) Implementation of Dynamic IP Blocking Techniques on the Tanungbalai City Library Website. The IJICS (International Journal of Informatics and Computer Science), 5(3), pp.344-352.

Visutsak, P. and Wongpheng, K. (2020) Software defect prediction using convolutional neural network. In: The 35th International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC). (2020) Nagoya, Japan, 3-6 July 2020. [Online] Available from: <https://ieeexplore.ieee.org/document/9182919>. [Accessed 04 March 2022].

Sarker, I. H. (2021) Deep learning: a comprehensive overview on techniques, taxonomy, applications and research directions. SN Computer Science, [Online] Available from: <https://link.springer.com/article/10.1007/s42979-021-00815-1>. [Accessed 14 February 2022].

Makopa, J., Christopher, A., Shah, R. and Mandela, N. (2023) Internet of Things (IoT) Network Forensic Analysis Using the Raspberry Pi 4 Model B and Open-Source Tools. In 2023 International Conference on Quantum Technologies, Communications, Computing, Hardware, and Embedded Systems Security (iQ-CCHES) (pp. 1-7). IEEE.

APPENDIX A – GANTT CHART

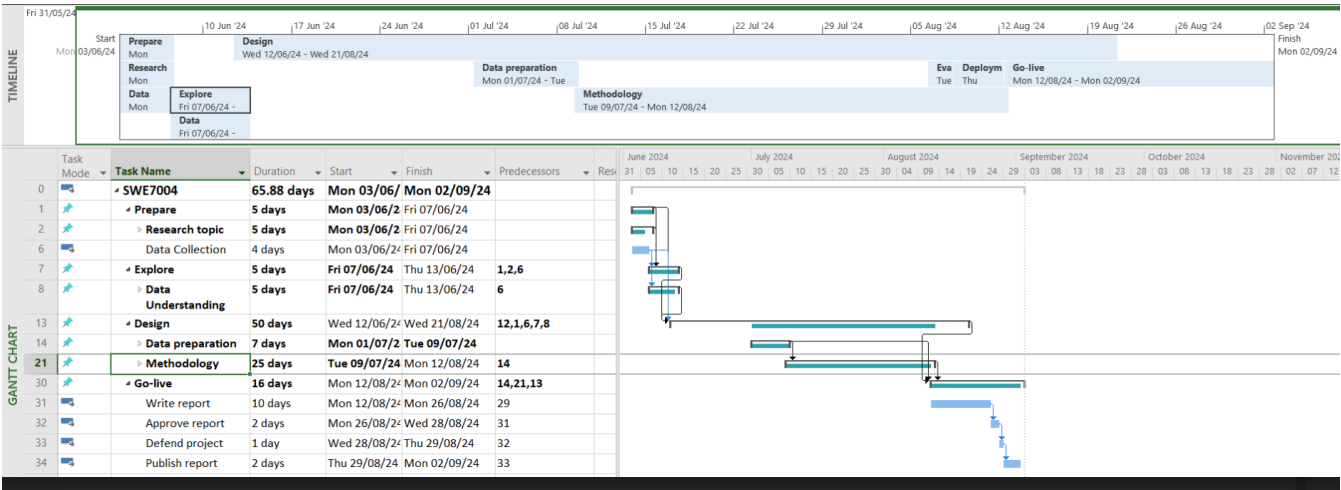


Figure 71: Gantt Chart Header

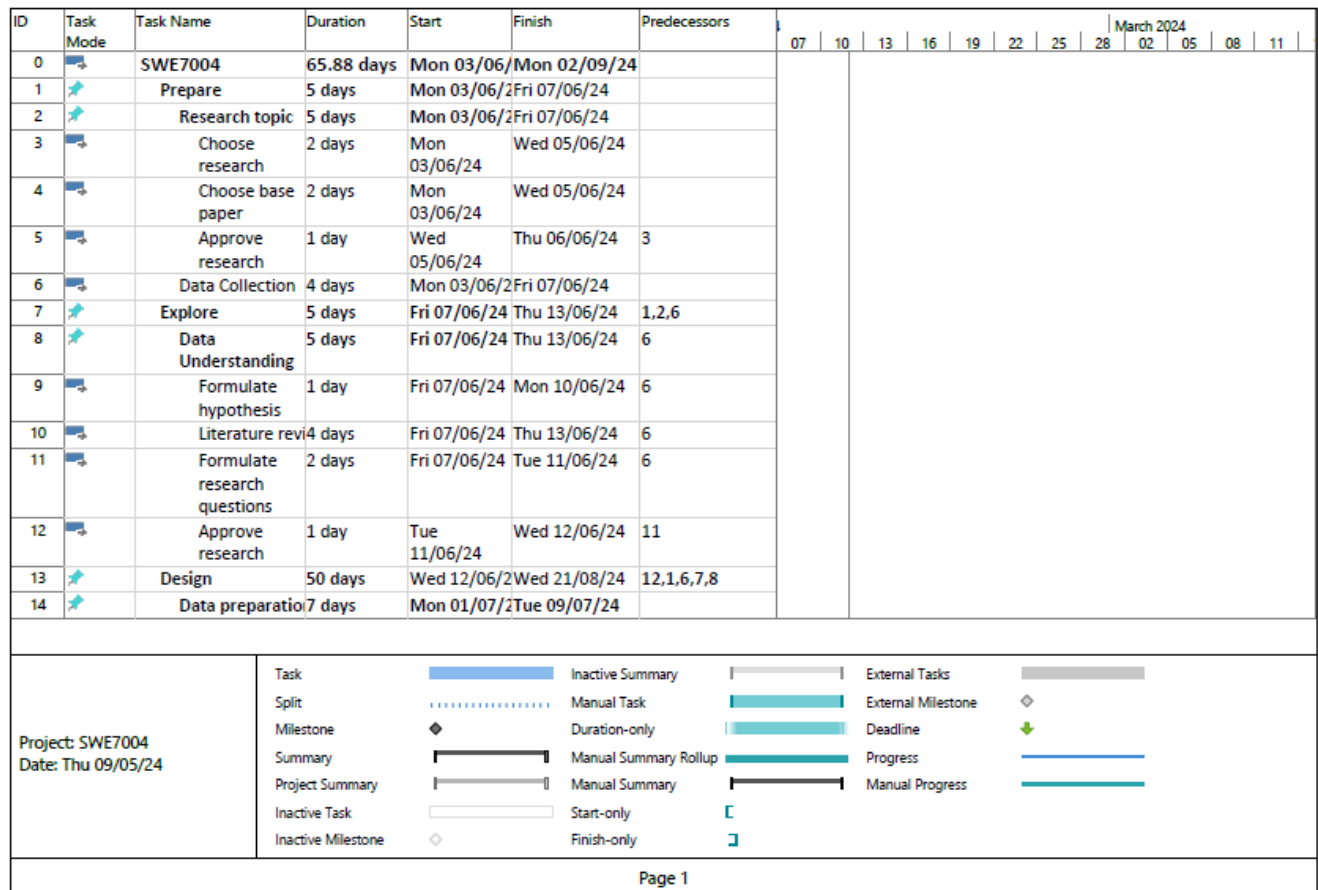


Figure 72: Gantt Chart Page 1

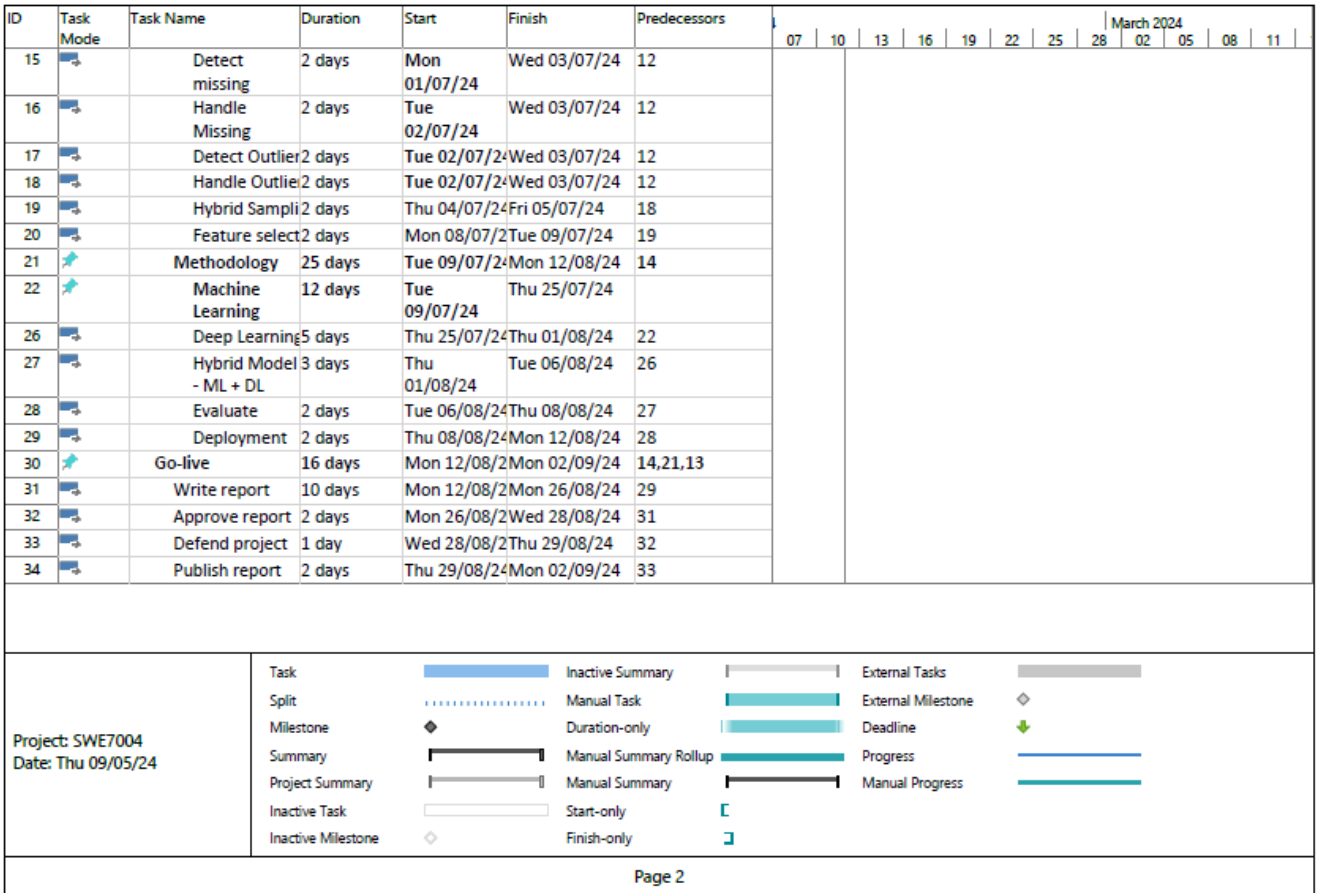


Figure 73: Gantt Chart Page 2

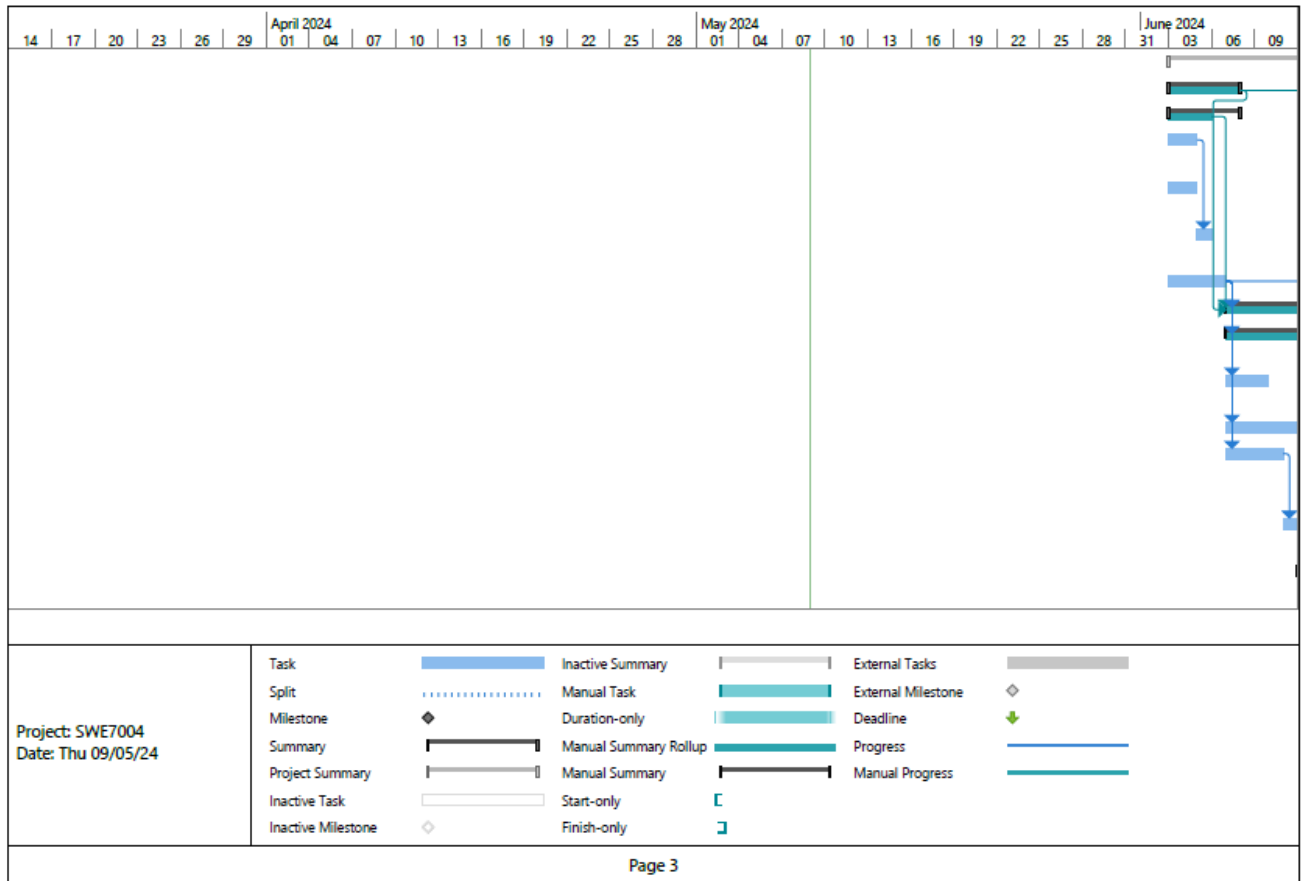


Figure 74: Gantt Chart Page 3

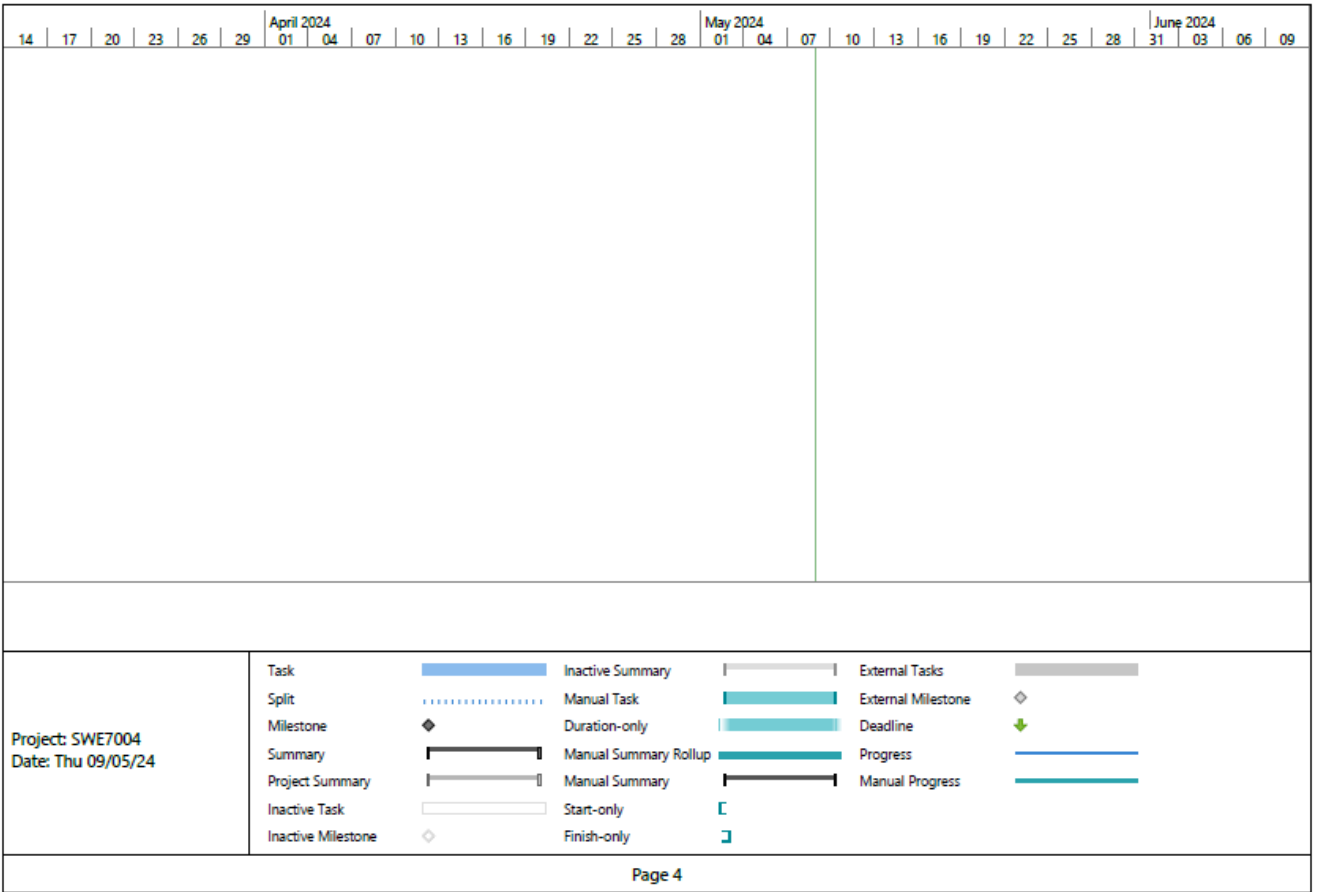


Figure 75: Gantt Chart Page 4

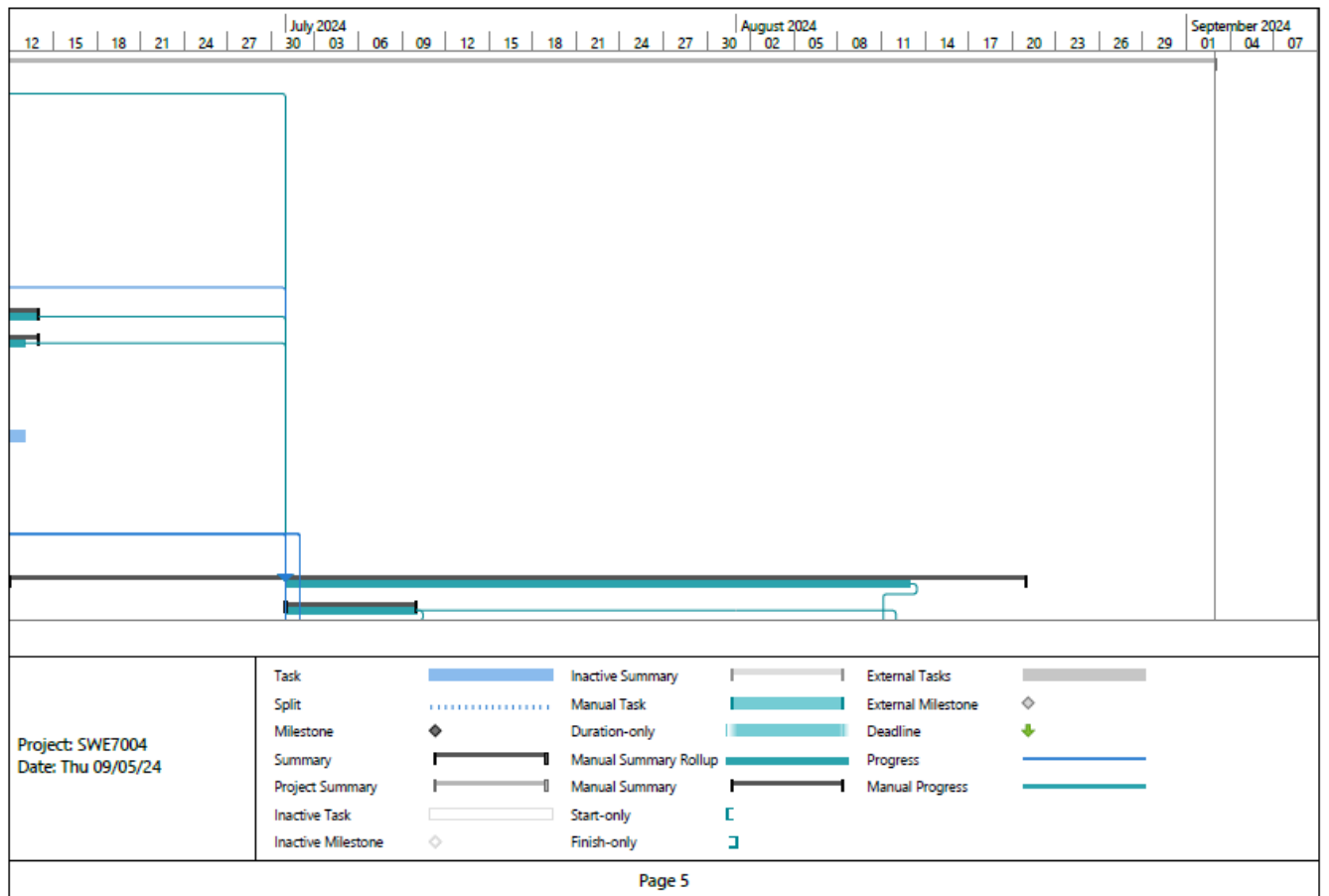


Figure 76: Gantt Chart Page 5

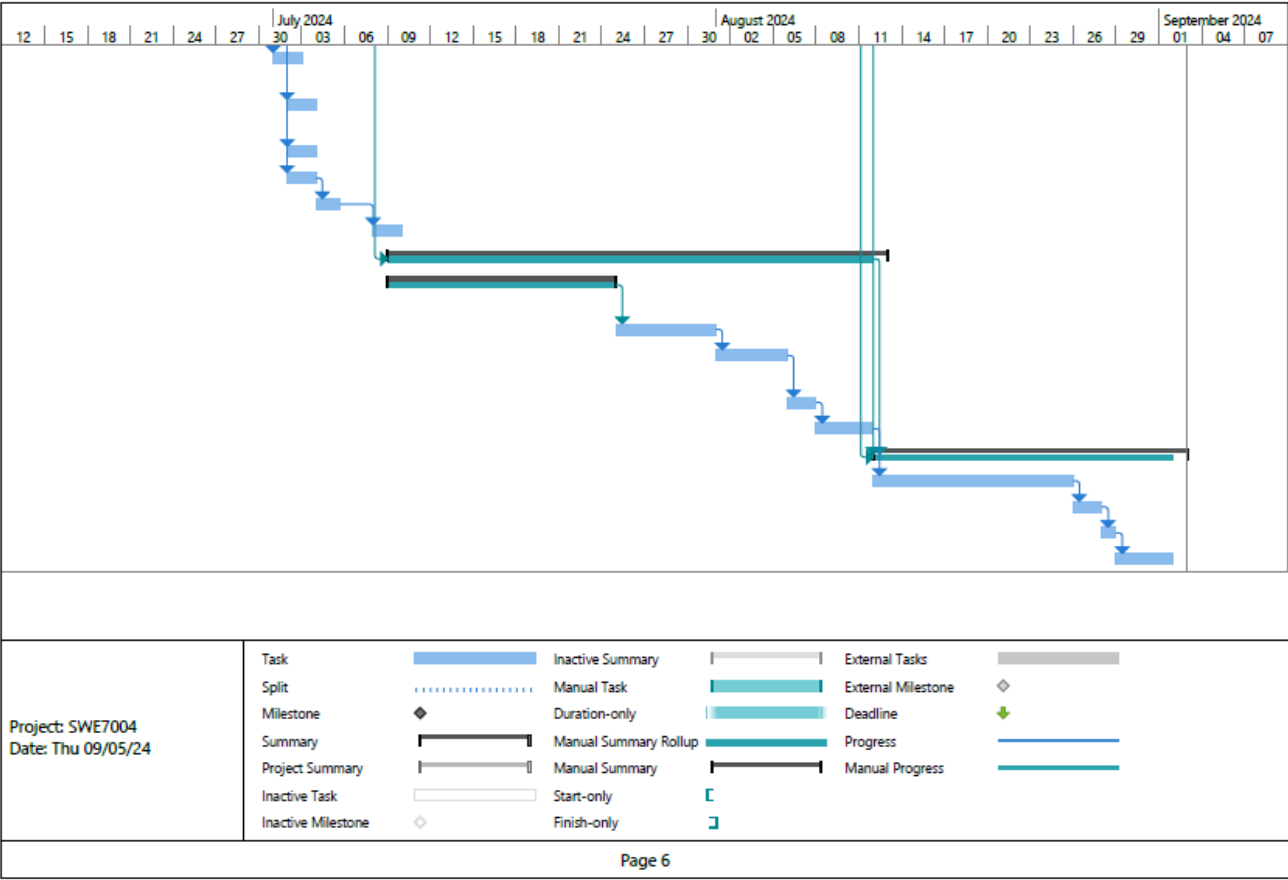


Figure 77: Gantt Chart Page 6

APPENDIX B – R SCRIPT

title: "SEC7001"

author: '2312602'

date: "2024-08-19"

output:

word_document:

toc: true

fig_caption: true

fig_height: 4

pdf_document: default

html_document: default

```
``{r setup, include=FALSE}
```

```
knitr::opts_chunk$set(echo = TRUE, fig.cap = "Figure: ")
```

```
``
```

```
## DAT7006 ASSESSMENT 2
```

```
# 1. Business Understanding
```



```
## Research Hypothesis
```

```
Pacman library package loads pacman::p_load(pacman, dplyr, GGally, ggplot2, ggthemes, ggvis, httr,  
lubridate, plotly, rio, rmarkdown, shiny, stringr, tidyr)
```

```
Load library
```

```
``{r}
```

```
library(pacman) # No message
```

```
library(dplyr)
```

```
``
```

```
# 2. Data Understanding
```

```
Load the dataset
```

```
``{r}
```

```
Metro <- read.csv("C:/MetroPT3(AirCompressor).csv", header = TRUE)
```

```
#DROP THE INDEX COLUMN X
```

```
Metro <- Metro[, !names(Metro) %in% "X"]
```

```
View(Metro)
```

```
'''
```

2.1 Descriptive Analysis

2.2.1 Data Info

Check number of columns.

```
'''{r}
```

```
ncol(Metro)
```

```
'''
```

Check number of rows.

```
'''{r}
```

```
nrow(Metro)
```

```
'''
```

Check the characters.

```
'''{r}
```

```
str(Metro)
```

```
'''
```

To change the timestamp column to its actual datetime character format, `type.convert` is used as shown below.

```
``{r}  
  
library(lubridate)  
  
# Convert the character column to POSIXct (datetime format)  
  
Metro$timestamp <- ymd_hms(as.character(Metro$timestamp))  
  
``
```

check the new character

```
``{r}  
  
str(Metro)  
  
``
```

2.2.2 Data summary

```
``{r}  
  
summary(Metro)  
  
``
```

2.2 EDA

2.2.3 Metro Correlation Matrix

```
``{r}  
  
# Correlation analysis
```

```
correlation_matrix <- cor(Metro[, -c(1, 12)], use = "complete.obs")
```

```
# Visualization of correlation matrix
```

```
require(corrplot)
```

```
# Set the size of the plot window
```

```
corrplot(correlation_matrix, method = "color", type = "lower",
```

```
        addCoef.col = "black", tl.col = "black", tl.srt = 5)
```

```
...
```

```
### 2.2.4 Normalization check
```

```
``{r}
```

```
hist(Metro$TP2, main = "Distribution of TP2", xlab = "TP2")
```

```
...
```

```
``{r}
```

```
hist(Metro$TP3, main = "Distribution of TP3", xlab = "TP3")
```

```
...
```

```
``{r}
```

```
hist(Metro$H1, main = "Distribution of H1", xlab = "H1")
```

```
...
```

```
``{r}
```

```
hist(Metro$DV_pressure, main = "Distribution of DV_Pressure", xlab = "DV_PRESSURE")
```

```
...
```

```
``{r}
```

```
hist(Metro$Reservoirs, main = "Distribution of Reservoirs", xlab = "RESERVOIRS")
```

```
...
```

```
``{r}
```

```
hist(Metro$Oil_temperature, main = "Distribution of Oil temperature", xlab = "OIL TEMPERATURE")
```

```
...
```

```
``{r}
```

```
hist(Metro$Motor_current, main = "Distribution of MOTOR CURRENT", xlab = "MOTOR CURRENT")
```

```
...
```

```
``{r}
```

```
hist(Metro$COMP, main = "Distribution of COMP", xlab = "COMP")
```

```
...
```

```
``{r}
```

```
hist(Metro$DV_eletric, main = "Distribution of DV_PRESSURE", xlab = "DV_PRESSURE")
```

```
...
```

```
``{r}
```

```
hist(Metro$Towers, main = "Distribution of TOWERS", xlab = "TOWERS")
```

```
...
```

```
``{r}
```

```
hist(Metro$LPS, main = "Distribution of LPS", xlab = "LPS")
```

```
...
```

```
``{r}
```

```
hist(Metro$Pressure_switch, main = "Distribution of PRESSURE SWITCH", xlab = "PRESSURE SWITCH")
```

```
...
```

```
``{r}
```

```
hist(Metro$Oil_level, main = "Distribution of OIL LEVEL", xlab = "OIL LEVEL")
```

```
...
```

```
``{r}
```

```
hist(Metro$Caudal_impulses, main = "Distribution of Caudal Impulses", xlab = "Caudal Impulses")
```

```
...
```

3. Data Preparation

Data Pre-processing

3.1 Missing Values

code below shows prints number of missing values in the dataset.

```
``{r}
```

```
Metro1 <- Metro #create a copy of dataset
```

```
sum(is.na(Metro1))
```

```
``
```

code below displays the total number of missing values in each column as shown below.

```
``{r}
```

```
colSums(is.na(Metro1))
```

```
``
```

```
``{r}
```

```
is.na(Metro1)
```

```
``
```

3.2 Outliers

```
``{r}
```

```
#Timestamp
```

```

boxplot(Metro1$timestamp, main = "Timestamp Boxplot")
...

```{r}

#TP2

boxplot(Metro1$TP2, main = "TP2 Boxplot")
...

```{r}

#TP3

boxplot(Metro1$TP3, main = "TP3 Boxplot")
...

```{r}

#DV_pressure

boxplot(Metro1$DV_pressure, main = "DV_pressure Boxplot")
...

```{r}

#Reservoirs

boxplot(Metro1$Reservoirs, main = "Reservoirs Boxplot")
...

```



```
``{r}
```

```
#Oil_temperature
```

```
boxplot(Metro1$Oil_temperature, main = "Oil_temperature Boxplot")
```

```
``
```

```
``{r}
```

```
#Motor_current
```

```
boxplot(Metro1$Motor_current, main = "Motor_current Boxplot")
```

```
``
```

```
``{r}
```

```
#COMP
```

```
boxplot(Metro1$COMP, main = "COMP Boxplot")
```

```
``
```

```
``{r}
```

```
#DV_eletric
```

```
boxplot(Metro1$DV_eletric, main = "DV_eletric Boxplot")
```

```
``
```

```
``{r}
```

```
#Towers
```

```
boxplot(Metro1$Towers, main = "Towers Boxplot")
```

```
``
```

```
``{r}
```

```
#MPG
```

```
boxplot(Metro1$MPG, main = "MPG Boxplot")
```

```
``
```

```
``{r}
```

```
#LPS
```

```
boxplot(Metro1$LPS, main = "LPS Boxplot")
```

```
``
```

```
``{r}
```

```
#Pressure_switch
```

```
boxplot(Metro1$Pressure_switch, main = "Pressure_switch Boxplot")
```

```
``
```

```
``{r}
```

```
#Oil_level
```

```
boxplot(Metro1$Oil_level, main = "Oil_level Boxplot")
```

```
```
```

```
```{r}
```

```
#Caudal_impulses
```

```
boxplot(Metro1$Caudal_impulses, main = "Caudal_impulses Boxplot")
```

```
```
```

```
```{r}
```

```
library(Hmisc)
```

```
```
```

### ## 3.3 Feature Engineering

```
```{r}
```

```
# Define the failure ranges of indices
```

```
range1 <- 562565:571227 # First range (4/18/2020 0:00 4/18/2020 23:59)
```

```
range2 <- 840741:843105 # Second range (5/29/2020 23:30 5/30/2020 6:00)
```

```
range3 <- 887240:908125 # Third range (6/5/2020 10:00 6/7/2020 14:30)
```

```
range4 <- 1171094:1172715 # Fourth range (7/15/2020 14:30 7/15/2020 19:00)
```

```
# Check if the dataset has enough rows to include the specified ranges
```

```

if (max(c(range1, range2, range3, range4)) > nrow(Metro1)) {

  stop("The specified index ranges exceed the number of rows in the dataset.")

}

```

```

# Add a new column that assigns 1 to the specified ranges and 0 to others

```

```

Metro1$Aircompfail <- ifelse(

  seq_len(nrow(Metro1)) %in% c(range1, range2, range3, range4), 1, 0)

```

```

# View the rows where Aircompfail is 1 (for demonstration purposes)

```

```

View(Metro1)

```

```

head(Metro1)

```

```

...

```

```

Datacount for Aircompfail.

```

```

``{r}

```

```

table(Metro1$Aircompfail)

```

```

...

```

```

## 3.5 Hypothesis Testing:

```

```

Q1. Is there a relationship between TP2 and DV_Pressure ?

```

```

``{r}

```

Bivariate Analysis -Correlation coefficient

```
cor(Metro1$TP2, Metro1$DV_pressure, method = 'spearman')
```

```
```
```

Q2. Is there a relationship between MPG and Pressure Switch?

```
```{r}
```

Correlation coefficient

```
cor(Metro1$MPG, Metro1$Pressure_switch, method = 'spearman')
```

```
```
```

Q3. Is there a relationship between H1 and TP2

```
```{r}
```

Correlation coefficient

```
cor(Metro1$TP2, Metro1$H1, method = 'spearman')
```

```
```
```

Multivariate Analysis

Q4. is there a relationship between TP2, DV\_Pressure and Aircompfail?

Multiple corellation

```
```{r}
```

```
#selected_data <- Sheffield[, c("TSK", "RAINC", "SMOIS")]
```

```
Mulcorrmatrix <- cor(Metro1[, c("TP2", "DV_pressure", "Aircompfail")])
```

```
Mulcorrmatrix
```

```
```
```

```
```{r}
```

```
Mulcorrelation <- sqrt(det(Mulcorrmatrix))
```

```
Mulcorrelation
```

```
```
```

### ## 3.6 Data Sampling

Create a copy of dataset

```
```{r}
```

```
library(caTools)
```

```
Metro2 <- Metro1 #create a copy of dataset for modelling
```

```
str(Metro2)
```

```
```
```

Check the class distribution

```
```{r}
```

```
# Display the original class distribution
```

```
print("Original class distribution:")
```

```
print(table(Metro2$Aircompfail))
```

```
...
```

Hybrid-Sampling technique

```
``{r}
```

```
library(ROSE)
```

```
# Create a balanced dataset
```

```
Metro3 <- ovun.sample(Aircompfail ~ .,
```

```
  data = Metro2,
```

```
  method = "both", # Use both over-sampling and under-sampling
```

```
  p = 0.5,      # Aim for a 50/50 class distribution
```

```
  seed = 1)$data # Set seed for reproducibility
```

```
# Display the class distribution after balancing
```

```
print("Class distribution after balancing:")
```

```
print(table(Metro3$Aircompfail))
```

```
...
```

```
Check column headers
```

```
``{r}
```

```
names(Metro3)
```

```
...
```

```
# Modelling - Dataframes
```

```
``{r}
```

```
# Create a copy of Metro3 and drop a timestamp column
```

```
Metro4 <- Metro3[ , !(names(Metro3) %in% c("timestamp"))] #used on the entire dataset
```

```
Metro5 <- Metro3[ , !(names(Metro3) %in% c("timestamp"))] #used to separate the dataset into two
```

```
#Analog Sensor Data
```

```
Analog <- Metro5[,c("Aircompfail", "TP2", "TP3", "H1", "DV_pressure", "Reservoirs", "Oil_temperature",  
"Motor_current")]
```

```
#Digital Sensor Data
```

```
Digital <- Metro5[,c("Aircompfail", "COMP", "DV_eletric", "Towers", "MPG", "LPS", "Pressure_switch",  
"Oil_level", "Caudal_impulses")]
```

```
...
```

```
# 4. MetroPT-3 Dataset
```

```
## 4.1 Feature Selection
```

```
``{r}
```



```

library(corrplot)

# Calculate the correlation matrix

Metro4corrmatrix <- cor(Metro4[, -ncol(Metro4)])


# Plot the correlation matrix

corrplot(Metro4corrmatrix, method = "circle")

...


``{r}

library(caret)

# Find highly correlated features (correlation > 0.75)

Metro4highlyCorrelated <- findCorrelation(Metro4corrmatrix, cutoff = 0.75)


# Print the indices of highly correlated attributes

print(Metro4highlyCorrelated)

...


``{r}

# Remove highly correlated features

Metro4 <- Metro4[, -Metro4highlyCorrelated]


View(Metro4)

```

```
'''
```

4.2 Train_Test_Split

Split the dataframe using 80/20 split

```
'''{r}
```

```
Metro4split <- sample.split(Metro4, SplitRatio = 0.8)
```

```
'''
```

Initiate training set and testing set

```
'''{r}
```

```
m4train <- Metro4[Metro4split,]
```

```
m4test <- Metro4[!Metro4split,]
```

```
'''
```

the data has been split using 80/20 split. 80% of the data goes to the training model and the 20% goes to test model.

```
'''{r}
```

```
names(Metro4)
```

```
'''
```

check for number of rows in each

```
``{r}
```

```
nrow(m4train)
```

```
``
```

```
``{r}
```

```
nrow(m4test)
```

```
``
```

4.3 Modelling Techniques

Import Deep Learning Libraries

```
``{r}
```

```
# Load necessary libraries
```

```
library(caret) #
```

```
library(e1071) #
```

```
library(randomForest)
```

```
library(gbm)
```

```
library(xgboost)
```

```
``
```

4.3.1. Logistic Regression

```
``{r}
```

```

#Train the model

Logmodel <- glm(Aircompfail~., data = m4train, family = "binomial")


#Predict the model

Logpredict <- predict(Logmodel, m4test, type = "response")

Logpredict

```

Logistic Regression Evaluation

```{r}

# Convert probabilities to binary outcomes (assuming 0.5 as threshold)

Logpredclass <- ifelse(Logpredict > 0.5, 1, 0)

Logpredclass <- as.factor(Logpredclass)


summary(Logpredclass)

```

```{r}

# Ensure that the lengths of predclass and testdata match

print(paste("Length of predicted classes: ", length(Logpredclass)))

print(paste("Length of actual classes: ", length(m4test$Aircompfail)))

```

```
M4Ircmatrix <- confusionMatrix(Logpredclass, (factor(m4test$Aircompfail)))
```

```
M4Ircmatrix
```

```
Logaccuracy <- M4Ircmatrix$overall['Accuracy']
```

```
Logrecall <- M4Ircmatrix$byClass["Recall"]
```

```
Logprecision <- M4Ircmatrix$byClass["Precision"]
```

```
Logf1score <- M4Ircmatrix$byClass["F1"]
```

```
Logaccuracy
```

```
Logrecall
```

```
Logprecision
```

```
Logf1score
```

```
...
```

```
### 4.3.2. Decision Tree
```

```
```{r}
```

```
#import libraries
```

```
library(rpart)
```

```
library(rpart.plot)
```

```
Fit the decision tree model using the training data
```

```
Dtmodel <- rpart(Aircompfail ~ ., data = m4train, method = "class")
```

```
Plot the decision tree
```

```
rpart.plot(Dtmodel)
```

```
...
```

```
Predict the model
```

```
``{r}
```

```
Predict class labels
```

```
Dtpredict <- predict(Dtmodel, m4test, type = "class") # Get class labels
```

```
summary(Dtpredict)
```

```
...
```

```
Decision Tree Evaluation
```

```
``{r}
```

```
Ensure that the lengths of predclass and testdata match
```

```
print(paste("Length of predicted classes: ", length(Dtpredict)))
```

```
print(paste("Length of actual classes: ", length(m4test$Aircompfail)))
```

```
...
```

```
Decision Tree Confusion matrix
```

```
``{r}
```

```
#Evaluate model performance
```

```
Dtconfmatrix <- confusionMatrix(factor(Dtpredict), factor(m4test$Aircompfail))
```

```
Dtaccuracy <- Dtconfmatrix$overall['Accuracy']
```

```
Dtrecall <- Dtconfmatrix$byClass["Recall"]
```

```
Dtprecision <- Dtconfmatrix$byClass["Precision"]
```

```
Dtf1score <- Dtconfmatrix$byClass["F1"]
```

```
Dtconfmatrix
```

```
Dtaccuracy
```

```
Dtrecall
```

```
Dtprecision
```

```
Dtf1score
```

```
...
```

```
4.3.3. Random forest
```

```
``{r}
```

```
library(ranger) #for faster implementation
```

```
Train the Random Forest model
```

```
Rfmodel <- ranger(Aircompfail ~ ., data = m4train, probability = TRUE)
```

```
#predict on the test set
```

```
Rfpredict <- predict(Rfmodel, m4test)$predictions
```

```

summary(Rfpredict)

'''

Random Forest Evaluation

```{r}

# Convert predictions and actual values to factors with the same levels

Rfpredclass <- ifelse(Rfpredict[,2] > 0.5, "1", "0")


print(paste("Length of predicted classes: ", length(Rfpredclass)))

print(paste("Length of actual classes: ", length(m4test$Aircompfail)))


# Generate the confusion matrix and calculate accuracy

Rfconfmatrix <- confusionMatrix(as.factor(Rfpredclass), factor(m4test$Aircompfail))


# Extract recall, precision, and F1 score

Rfaccuracy <- Rfconfmatrix$overall['Accuracy']

Rfrecall <- Rfconfmatrix$byClass["Recall"]

Rfprecision <- Rfconfmatrix$byClass["Precision"]

Rff1score <- Rfconfmatrix$byClass["F1"]


# Print the metrics

```



```
print(Rfconfmatrix)
```

```
cat("Accuracy:", Rfaccuracy, "\n")
```

```
cat("Recall:", Rfrecall, "\n")
```

```
cat("Precision:", Rfprecision, "\n")
```

```
cat("F1 Score:", Rff1score, "\n")
```

```
``
```

4.3.4. Gradient Boosting Machine

```
``{r}
```

```
# Load the gbm package
```

```
library(gbm)
```

```
# Train the GBM model
```

```
Gbmmodel <- gbm(Aircompfail ~ ., data = m4train,
```

```
  distribution = "bernoulli", # For binary classification
```

```
  n.trees = 100,           # Number of trees
```

```
  interaction.depth = 3,   # Depth of each tree
```

```
  shrinkage = 0.01,       # Learning rate
```

```
  cv.folds = 5,           # Number of cross-validation folds
```

```
  verbose = TRUE)        # Print progress
```

```

# Print the summary of the model to get information on the performance

summary(Gbmmodel)

str(Gbmmodel)


# Determine the best number of trees using cross-validation results

bestntrees <- gbm.perf(Gbmmodel, method = "cv")

print(paste("Best number of trees:", bestntrees))

...


``{r}

# Make predictions on the test set

Gbmpredict <- predict(Gbmmodel, newdata = m4test, n.trees = bestntrees, type = "response")

summary(Gbmpredict)

...


#### GBM Evaluate

``{r}

# Convert probabilities to class labels

Gbmpredclass <- ifelse(Gbmpredict > 0.5, "1", "0")

Gbmpredclass <- as.factor(Gbmpredclass)

```

```

print(paste("Length of predicted classes: ", length(Gbmpredclass)))

print(paste("Length of actual classes: ", length(m4test$Aircompfail)))


# Create a confusion matrix

Gbmconfmatrix <- confusionMatrix(Gbmpredclass, factor(m4test$Aircompfail))


# Extract recall, precision, and F1 score

Gbmaccuracy <- Gbmconfmatrix$overall['Accuracy']

Gbmrecall <- Gbmconfmatrix$byClass["Recall"]

Gbmprecision <- Gbmconfmatrix$byClass["Precision"]

Gbmf1score <- Gbmconfmatrix$byClass["F1"]


# Print the metrics

print(Gbmconfmatrix)


cat("Accuracy:", Gbmaccuracy, "\n")

cat("Recall:", Gbmrecall, "\n")

cat("Precision:", Gbmprecision, "\n")

cat("F1 Score:", Gbmf1score, "\n")

...


### 4.3.5. Neural Network

```

Import Libraries

```
``{r}
```

```
# Load necessary libraries
```

```
library(keras)
```

```
library(tensorflow)
```

```
library(nnet) #function for neural network
```

```
library(caret) #streamlines training and evaluation
```

```
# Scale the data
```

```
dl4train <- m4train
```

```
dl4test <- m4test
```

```
``
```

```
``{r}
```

```
dl4train[-which(names(dl4train) == "Aircompfail")] <- scale(dl4train[-which(names(dl4train) ==  
"Aircompfail")])
```

```
dl4test[-which(names(dl4test) == "Aircompfail")] <- scale(dl4test[-which(names(dl4test) ==  
"Aircompfail")])
```

```
#FIT NN
```

```
NNmodel <- nnet(Aircompfail~., data = dl4train, size = 5, decay = 0.1, maxit = 200, linout = FALSE)
```

```
#Neural Network Predict
```

```
NNpred <- predict(NNmodel, dl4test, type = "raw")
```

```
NNpred
```

```
summary(NNpred)
```

```
...
```

```
#### NN Evaluate
```

```
``{r}
```

```
# Convert probabilities to class labels
```

```
NNpredclass <- ifelse(NNpred > 0.5, "1", "0")
```

```
NNpredclass <- as.factor(NNpredclass)
```

```
print(paste("Length of predicted classes: ", length(NNpredclass)))
```

```
print(paste("Length of actual classes: ", length(dl4test$Aircompfail)))
```

```
# Create a confusion matrix
```

```
NNconfmatrix <- confusionMatrix(NNpredclass, factor(dl4test$Aircompfail))
```

```
# Extract recall, precision, and F1 score
```

```
NNaccuracy <- NNconfmatrix$overall['Accuracy']
```

```

NNrecall <- NNconfmatrix$byClass["Recall"]

NNprecision <- NNconfmatrix$byClass["Precision"]

NNf1score <- NNconfmatrix$byClass["F1"]

```

```

# Print the metrics

```

```

print(NNconfmatrix)

```

```

cat("Accuracy:", NNaccuracy, "\n")

```

```

cat("Recall:", NNrecall, "\n")

```

```

cat("Precision:", NNprecision, "\n")

```

```

cat("F1 Score:", NNf1score, "\n")

```

```

...

```

```

## 4.4 Model Comparison

```

```

``{r}

```

```

#print accuracy for each model

```

```

Logaccuracy

```

```

Dtaccuracy

```

```

Rfaccuracy

```

```

Gbmaccuracy

```

```

NNaccuracy

```

```

...

```

```
``{r}
```

```
# Store all accuracies in a named vector
```

```
accuracies <- c(Logaccuracy, Dtaccuracy, Rfaccuracy, Gbmaccuracy, NNaccuracy)
```

```
names <- c("Logaccuracy", "Dtaccuracy", "Rfaccuracy", "Gbmaccuracy", "NNaccuracy")
```

```
# Create a data frame for ggplot
```

```
accuracy_df <- data.frame(Model = names, Accuracy = accuracies)
```

```
# Find the highest accuracy
```

```
max_accuracy <- max(accuracies)
```

```
# Find which model achieved the highest accuracy
```

```
best_model <- names[which.max(accuracies)]
```

```
# Visualize the accuracies using ggplot2
```

```
ggplot(accuracy_df, aes(x = Model, y = Accuracy, fill = Model)) +
```

```
  geom_bar(stat = "identity", width = 0.6) +
```

```
  geom_text(aes(label = round(Accuracy, 2)), vjust = -0.3) +
```

```
  labs(title = "Model Accuracies", y = "Accuracy", x = "Model") +
```

```
  theme_minimal() +
```

```
  theme(legend.position = "none")
```

```
'''
```

```
```{r}
```

```
Print the highest accuracy and the corresponding model
```

```
cat("The highest accuracy is", max_accuracy, "achieved by", best_model, "\n")
```

```
'''
```

```
5. Analogue Sensor
```

```
```{r}
```

```
View(Analog)
```

```
str(Analog)
```

```
'''
```

```
## 5.1 Feature Selection
```

```
```{r}
```

```
Calculate the correlation matrix
```

```
Analogcorrmatrix <- cor(Analog[, -ncol(Analog)])
```

```
Plot the correlation matrix
```

```
corrplot(Analogcorrmatrix, method = "circle")
```



```
'''
```

```
'''{r}
```

```
Find highly correlated features (correlation > 0.75)
```

```
AnaloghighlyCorrelated <- findCorrelation(Analogcorrmatrix, cutoff = 0.75)
```

```
Print the indices of highly correlated attributes
```

```
print(AnaloghighlyCorrelated)
```

```
'''
```

```
'''{r}
```

```
Remove highly correlated features
```

```
Analog <- Analog[, -AnaloghighlyCorrelated]
```

```
str(Analog)
```

```
'''
```

```
5.2 Train_Test_Split
```

```
Split the dataframe using 80/20 split
```

```
'''{r}
```

```
Analogsplit <- sample.split(Analog, SplitRatio = 0.8)
```

```
'''
```

Initiate training set and testing set

```
'''{r}
```

```
Atrain <- Analog[Analogsplit,]
```

```
Atest <- Analog[!Analogsplit,]
```

```
'''
```

80% to training and 20% to testing dataset

```
'''{r}
```

```
names(Analog)
```

```
'''
```

check for number of rows in each

```
'''{r}
```

```
nrow(Atrain)
```

```
'''
```

```
'''{r}
```

```
nrow(Atest)
```

```
'''
```

## ## 5.3 Modelling Techniques

### ### 5.3.1. Logistic Regression

```
``{r}
```

```
#Train the model
```

```
ALogmodel <- glm(Aircompfail~., data = Atrain, family = "binomial")
```

```
#Predict the model
```

```
ALogpredict <- predict(ALogmodel, Atest, type = "response")
```

```
ALogpredict
```

```
...
```

### #### Logistic Regression Evaluation

```
``{r}
```

```
Convert probabilities to binary outcomes (assuming 0.5 as threshold)
```

```
ALogpredclass <- ifelse(ALogpredict > 0.5, 1, 0)
```

```
ALogpredclass <- as.factor(ALogpredclass)
```

```
summary(ALogpredclass)
```

```
...
```

```
``{r}
```

```
Ensure that the lengths of predclass and testdata match
```

```
print(paste("Length of predicted classes: ", length(ALogpredclass)))
```

```
print(paste("Length of actual classes: ", length(Atest$Aircompfail)))
```

```
Alogmatrix <- confusionMatrix(ALogpredclass, (factor(Atest$Aircompfail)))
```

```
Alogmatrix
```

```
Alogaccuracy <- Alogmatrix$overall['Accuracy']
```

```
Alogrecall <- Alogmatrix$byClass["Recall"]
```

```
Alogprecision <- Alogmatrix$byClass["Precision"]
```

```
Alogf1score <- Alogmatrix$byClass["F1"]
```

```
Alogaccuracy
```

```
Alogrecall
```

```
Alogprecision
```

```
Alogf1score
```

```
``
```

### ### 5.3.2. Decision Tree

```
``{r}
```

```
Fit the decision tree model using the training data
```

```
Adtmodel <- rpart(Aircompfail ~ ., data = Atrain, method = "class")
```

```
Plot the decision tree
```

```
rpart.plot(Adtmodel)
```

```
...
```

```
Predict the model
```

```
``{r}
```

```
Predict class labels
```

```
Adtpredict <- predict(Adtmodel, Atest, type = "class") # Get class labels
```

```
summary(Adtpredict)
```

```
...
```

```
Decision Tree Evaluation
```

```
``{r}
```

```
Ensure that the lengths of predclass and testdata match
```

```
print(paste("Length of predicted classes: ", length(Adtpredict)))
```

```
print(paste("Length of actual classes: ", length(Atest$Aircompfail)))
```

```
...
```

```
Decision Tree Confusion matrix
```

```
``{r}
```

```

#Evaluate model performance

Adtconfmatrix <- confusionMatrix(factor(Adtpredict), factor(Atest$Aircompfail))

Adtaccuracy <- Adtconfmatrix$overall['Accuracy']

Adtrecall <- Adtconfmatrix$byClass["Recall"]

Adtprecision <- Adtconfmatrix$byClass["Precision"]

Adtf1score <- Adtconfmatrix$byClass["F1"]

```

```
Adtconfmatrix
```

```
Adtaccuracy
```

```
Adtrecall
```

```
Adtprecision
```

```
Adtf1score
```

```
```
```

```
### 5.3.3. Random forest
```

```
```{r}
```

```
Train the Random Forest model
```

```
Arfmodel <- ranger(Aircompfail ~ ., data = Atrain, probability = TRUE)
```

```
#predict on the test set
```

```
Arfpredict <- predict(Arfmodel, Atest)$predictions
```

```
summary(Arfpredict)
```

```
'''
```

```
Random Forest Evaluation
```

```
```{r}
```

```
# Convert predictions and actual values to factors with the same levels
```

```
Arfpredclass <- ifelse(Arfpredict[,2] > 0.5, "1", "0")
```

```
print(paste("Length of predicted classes: ", length(Arfpredclass)))
```

```
print(paste("Length of actual classes: ", length(Atest$Aircompfail)))
```

```
# Generate the confusion matrix and calculate accuracy
```

```
Arfconfmatrix <- confusionMatrix(as.factor(Arfpredclass), factor(Atest$Aircompfail))
```

```
# Extract recall, precision, and F1 score
```

```
Arfaccuracy <- Arfconfmatrix$overall['Accuracy']
```

```
Arfrecall <- Arfconfmatrix$byClass["Recall"]
```

```
Arfprecision <- Arfconfmatrix$byClass["Precision"]
```

```
Arff1score <- Arfconfmatrix$byClass["F1"]
```

```
# Print the metrics
```

```
print(Arfconfmatrix)
```

```
cat("Accuracy:", Arfaccuracy, "\n")
```

```
cat("Recall:", Arfrecall, "\n")
```

```
cat("Precision:", Arfprecision, "\n")
```

```
cat("F1 Score:", Arff1score, "\n")
```

```
``
```

5.3.4. Gradient Boosting Machine

```
``{r}
```

```
# Train the GBM model
```

```
Agbmmodel <- gbm(Aircompfail ~ ., data = Atrain,
```

```
  distribution = "bernoulli", # For binary classification
```

```
  n.trees = 100,           # Number of trees
```

```
  interaction.depth = 3,   # Depth of each tree
```

```
  shrinkage = 0.01,       # Learning rate
```

```
  cv.folds = 5,           # Number of cross-validation folds
```

```
  verbose = TRUE)        # Print progress
```

```
# Print the summary of the model to get information on the performance
```

```
summary(Agbmmodel)
```

```
str(Agbmmodel)
```



```

# Determine the best number of trees using cross-validation results

Abestntrees <- gbm.perf(Agbmmodel, method = "cv")

print(paste("Best number of trees:", Abestntrees))

...

``{r}

# Make predictions on the test set

Agbmpredict <- predict(Agbmmodel, newdata = Atest, n.trees = Abestntrees, type = "response")

summary(Agbmpredict)

...

##### GBM Evaluate

``{r}

# Convert probabilities to class labels

Agbmpredclass <- ifelse(Agbmpredict > 0.5, "1", "0")

Agbmpredclass <- as.factor(Agbmpredclass)

print(paste("Length of predicted classes: ", length(Agbmpredclass)))

print(paste("Length of actual classes: ", length(Atest$Aircompfail)))

# Create a confusion matrix

```

```
Agbmconfmatrix <- confusionMatrix(Agbmpredclass, factor(Atest$Aircompfail))
```

```
# Extract recall, precision, and F1 score
```

```
Agbmaccuracy <- Agbmconfmatrix$overall['Accuracy']
```

```
Agbmrecall <- Agbmconfmatrix$byClass["Recall"]
```

```
Agbmprecision <- Agbmconfmatrix$byClass["Precision"]
```

```
Agbmf1score <- Agbmconfmatrix$byClass["F1"]
```

```
# Print the metrics
```

```
print(Agbmconfmatrix)
```

```
cat("Accuracy:", Agbmaccuracy, "\n")
```

```
cat("Recall:", Agbmrecall, "\n")
```

```
cat("Precision:", Agbmprecision, "\n")
```

```
cat("F1 Score:", Agbmf1score, "\n")
```

```
...
```

```
### 5.3.5. Neural Network
```

```
``{r}
```

```
# Create a copy of the data
```

```
dlAtrain <- Atrain
```

```
dlAtest <- Atest
```

```
#scale the data
```

```
dlAtrain[-which(names(dlAtrain) == "Aircompfail")] <- scale(dlAtrain[-which(names(dlAtrain) ==  
"Aircompfail")])
```

```
dlAtest[-which(names(dlAtest) == "Aircompfail")] <- scale(dlAtest[-which(names(dlAtest) ==  
"Aircompfail")])
```

```
#FIT NN
```

```
Annmodel <- nnet(Aircompfail~., data = dlAtrain, size = 5, decay = 0.1, maxit = 200, linout = FALSE)
```

```
#Neural Network Predict
```

```
Annpred <- predict(Annmodel, dlAtest, type = "raw")
```

```
Annpred
```

```
summary(Annpred)
```

```
...
```

```
#### NN Evaluate
```

```
``{r}
```

```
# Convert probabilities to class labels
```

```
Annpredclass <- ifelse(Annpred > 0.5, "1", "0")
```

```
Annpredclass <- as.factor(Annpredclass)
```

```

print(paste("Length of predicted classes: ", length(Annpredclass)))

print(paste("Length of actual classes: ", length(dlAtest$Aircompfail)))


# Create a confusion matrix

Annconfmatrix <- confusionMatrix(Annpredclass, factor(dlAtest$Aircompfail))


# Extract recall, precision, and F1 score

Annaccuracy <- Annconfmatrix$overall['Accuracy']

Annrecall <- Annconfmatrix$byClass["Recall"]

Annprecision <- Annconfmatrix$byClass["Precision"]

Annf1score <- Annconfmatrix$byClass["F1"]


# Print the metrics

print(Annconfmatrix)


cat("Accuracy:", Annaccuracy, "\n")

cat("Recall:", Annrecall, "\n")

cat("Precision:", Annprecision, "\n")

cat("F1 Score:", Annf1score, "\n")

...


## 5.4 Model Comparison

```

```

``{r}

# print accuracy for each model

Alogaccuracy

Adtaccuracy

Arfaccuracy

Agbmaccuracy

Annaccuracy

...

``{r}

# Store all accuracies in a named vector

Analog_accuracies <- c(Alogaccuracy, Adtaccuracy, Arfaccuracy, Agbmaccuracy, Annaccuracy)

Anames <- c("Alogaccuracy", "Adtaccuracy", "Arfaccuracy", "Agbmaccuracy", "Annaccuracy")

# Create a data frame for ggplot

Analog_df <- data.frame(Model = Anames, Accuracy = Analog_accuracies)

# Find the highest accuracy

Amax_accuracy <- max(Analog_accuracies)

# Find which model achieved the highest accuracy

Abest_model <- Anames[which.max(Analog_accuracies)]

```

```

# Visualize the accuracies using ggplot2

ggplot(Analog_df, aes(x = Model, y = Accuracy, fill = Model)) +

  geom_bar(stat = "identity", width = 0.6) +

  geom_text(aes(label = round(Accuracy, 2)), vjust = -0.3) +

  labs(title = "Analog Model Accuracies", y = "Accuracy", x = "Models") +

  theme_minimal() +

  theme(legend.position = "none")

...


```{r}

Print the highest accuracy and the corresponding model

cat("The highest accuracy is", Amax_accuracy, "achieved by", Abest_model, "\n")

...

6. Digital Sensors - Modelling

```{r}

View(Digital)

str(Digital)

...


## 6.1 Feature Selection

```

```
``{r}
```

```
# Calculate the correlation matrix
```

```
Digitalcorrmatrix <- cor(Digital[, -ncol(Digital)])
```

```
# Plot the correlation matrix
```

```
corrplot(Digitalcorrmatrix, method = "circle")
```

```
``
```

```
``{r}
```

```
# Find highly correlated features (correlation > 0.75)
```

```
DigitalhighlyCorrelated <- findCorrelation(Digitalcorrmatrix, cutoff = 0.75)
```

```
# Print the indices of highly correlated attributes
```

```
print(DigitalhighlyCorrelated)
```

```
``
```

```
``{r}
```

```
# Remove highly correlated features
```

```
Digital <- Digital[, -DigitalhighlyCorrelated]
```

```
str(Digital)
```

```
``
```

6.2 Train_Test_Split

Split the dataframe using 80/20 split

```
``{r}
```

```
Digitalsplit <- sample.split(Digital, SplitRatio = 0.8)
```

```
``
```

Initiate training set and testing set

```
``{r}
```

```
Dtrain <- Digital[Digitalsplit,]
```

```
Dtest <- Digital[!Digitalsplit,]
```

```
``
```

80% to training and 20% to testing dataset

```
``{r}
```

```
names(Digital)
```

```
``
```

check for number of rows in each

```
``{r}
```



```
nrow(Dtrain)
```

```
```
```

```
```{r}
```

```
nrow(Dtest)
```

```
```
```

## ## 6.3 Digital Sensor

### ### 6.3.1. Logistic Regression

```
```{r}
```

```
#Train the model
```

```
DLogmodel <- glm(Aircompfail~., data = Dtrain, family = "binomial")
```

```
#Predict the model
```

```
DLogpredict <- predict(DLogmodel, Dtest, type = "response")
```

```
DLogpredict
```

```
```
```

### #### Logistic Regression Evaluation

```
```{r}
```

```
# Convert probabilities to binary outcomes (assuming 0.5 as threshold)
```

```
DLogpredclass <- ifelse(DLogpredict > 0.5, 1, 0)
```

```
DLogpredclass <- as.factor(DLogpredclass)
```

```
summary(DLogpredclass)
```

```
```
```

```
```{r}
```

```
# Ensure that the lengths of predclass and testdata match
```

```
print(paste("Length of predicted classes: ", length(DLogpredclass)))
```

```
print(paste("Length of actual classes: ", length(Dtest$Aircompfail)))
```

```
Dlogmatrix <- confusionMatrix(DLogpredclass, (factor(Dtest$Aircompfail)))
```

```
Dlogmatrix
```

```
Dlogaccuracy <- Dlogmatrix$overall['Accuracy']
```

```
Dlogrecall <- Dlogmatrix$byClass["Recall"]
```

```
Dlogprecision <- Dlogmatrix$byClass["Precision"]
```

```
Dlogf1score <- Dlogmatrix$byClass["F1"]
```

```
Dlogaccuracy
```

```
Dlogrecall
```

```
Dlogprecision
```

```
Dlogf1score
```

```
```
```

```
6.3.2. Decision Tree
```

```
``{r}
```

```
Fit the decision tree model using the training data
```

```
Ddtmodel <- rpart(Aircompfail ~ ., data = Dtrain, method = "class")
```

```
Plot the decision tree
```

```
rpart.plot(Ddtmodel)
```

```
Predict class labels
```

```
Ddtpredict <- predict(Ddtmodel, Dtest, type = "class") # Get class labels
```

```
summary(Ddtpredict)
```

```
```
```

```
#### Decision Tree Evaluation
```

```
``{r}
```

```
# Ensure that the lengths of predclass and testdata match
```

```
print(paste("Length of predicted classes: ", length(Ddtpredict)))
```

```
print(paste("Length of actual classes: ", length(Dtest$Aircompfail)))
```

```
```
```

Decision Tree Confusion matrix

```
``{r}
```

```
#Evaluate model performance
```

```
Ddtconfmatrix <- confusionMatrix(factor(Ddtpredict), factor(Dtest$Aircompfail))
```

```
Ddtaccuracy <- Ddtconfmatrix$overall['Accuracy']
```

```
Ddtrecall <- Ddtconfmatrix$byClass["Recall"]
```

```
Ddtprecision <- Ddtconfmatrix$byClass["Precision"]
```

```
Ddtf1score <- Ddtconfmatrix$byClass["F1"]
```

Ddtconfmatrix

Ddtaccuracy

Ddtrecall

Ddtprecision

Ddtf1score

```
``
```

### 6.3.3. Random forest

```
``{r}
```

```
Train the Random Forest model
```

```
Drfmodel <- ranger(Aircompfail ~ ., data = Dtrain, probability = TRUE)
```

```

#predict on the test set

Drfpredict <- predict(Drfmodel, Dtest)$predictions

summary(Drfpredict)

...

Random Forest Evaluation

``{r}

Convert predictions and actual values to factors with the same levels

Drfpredclass <- ifelse(Drfpredict[,2] > 0.5, "1", "0")

print(paste("Length of predicted classes: ", length(Drfpredclass)))

print(paste("Length of actual classes: ", length(Dtest$Aircompfail)))

Generate the confusion matrix and calculate accuracy

Drfconfmatrix <- confusionMatrix(as.factor(Drfpredclass), factor(Dtest$Aircompfail))

Extract recall, precision, and F1 score

Drfaccuracy <- Drfconfmatrix$overall['Accuracy']

Drfrecall <- Drfconfmatrix$byClass["Recall"]

Drfprecision <- Drfconfmatrix$byClass["Precision"]

Drff1score <- Drfconfmatrix$byClass["F1"]

```

```
Print the metrics
```

```
print(Drfconfmatrix)
```

```
cat("Accuracy:", Drfaccuracy, "\n")
```

```
cat("Recall:", Drfrecall, "\n")
```

```
cat("Precision:", Drfprediction, "\n")
```

```
cat("F1 Score:", Drff1score, "\n")
```

```
...
```

#### ### 6.3.4. Gradient Boosting Machine

```
``{r}
```

```
Train the GBM model
```

```
Dgbmmodel <- gbm(Aircompfail ~ ., data = Dtrain,
```

```
 distribution = "bernoulli", # For binary classification
```

```
 n.trees = 100, # Number of trees
```

```
 interaction.depth = 3, # Depth of each tree
```

```
 shrinkage = 0.01, # Learning rate
```

```
 cv.folds = 5, # Number of cross-validation folds
```

```
 verbose = TRUE) # Print progress
```

```

Print the summary of the model to get information on the performance

summary(Dgbmmodel)

str(Dgbmmodel)

Determine the best number of trees using cross-validation results

Dbestntrees <- gbm.perf(Dgbmmodel, method = "cv")

print(paste("Best number of trees:", Dbestntrees))

...


```{r}

# Make predictions on the test set

Dgbmpredict <- predict(Dgbmmodel, newdata = Dtest, n.trees = Dbestntrees, type = "response")

summary(Dgbmpredict)

...


##### GBM Evaluate

```{r}

Convert probabilities to class labels

Dgbmpredclass <- ifelse(Dgbmpredict > 0.5, "1", "0")

Dgbmpredclass <- as.factor(Dgbmpredclass)

print(paste("Length of predicted classes: ", length(Dgbmpredclass)))

```

```

print(paste("Length of actual classes: ", length(Dtest$Aircompfail)))

Create a confusion matrix

Dgbmconfmatrix <- confusionMatrix(Dgbmpredclass, factor(Dtest$Aircompfail))

Extract recall, precision, and F1 score

Dgbmaccuracy <- Dgbmconfmatrix$overall['Accuracy']

Dgbmrecall <- Dgbmconfmatrix$byClass["Recall"]

Dgbmprecision <- Dgbmconfmatrix$byClass["Precision"]

Dgbmf1score <- Dgbmconfmatrix$byClass["F1"]

Print the metrics

print(Dgbmconfmatrix)

cat("Accuracy:", Dgbmaccuracy, "\n")

cat("Recall:", Dgbmrecall, "\n")

cat("Precision:", Dgbmprecision, "\n")

cat("F1 Score:", Dgbmf1score, "\n")

...

6.3.5. Neural Network

```{r}

# Copy the data

```



```
dIDtrain <- Dtrain
```

```
dIDtest <- Dtest
```

```
#scale the data
```

```
dIDtrain[-which(names(dIDtrain) == "Aircompfail")] <- scale(dIDtrain[-which(names(dIDtrain) ==  
"Aircompfail")])
```

```
dIDtest[-which(names(dIDtest) == "Aircompfail")] <- scale(dIDtest[-which(names(dIDtest) ==  
"Aircompfail")])
```

```
#FIT NN
```

```
Dnnmodel <- nnet(Aircompfail~., data = dIDtrain, size = 5, decay = 0.1, maxit = 200, linout = FALSE)
```

```
#Neural Network Predict
```

```
Dnnpred <- predict(Dnnmodel, dIDtest, type = "raw")
```

```
Dnnpred
```

```
summary(Dnnpred)
```

```
...
```

```
#### NN Evaluate
```

```
`{r}
```

```
# Convert probabilities to class labels
```

```
Dnnpredclass <- ifelse(Dnnpred > 0.5, "1", "0")
```

```

Dnnpredclass <- as.factor(Dnnpredclass)

print(paste("Length of predicted classes: ", length(Dnnpredclass)))

print(paste("Length of actual classes: ", length(dIDtest$Aircompfail)))

# Create a confusion matrix

Dnnconfmatrix <- confusionMatrix(Dnnpredclass, factor(dIDtest$Aircompfail))

# Extract recall, precision, and F1 score

Dnnaccuracy <- Dnnconfmatrix$overall['Accuracy']

Dnnrecall <- Dnnconfmatrix$byClass["Recall"]

Dnnprecision <- Dnnconfmatrix$byClass["Precision"]

Dnnf1score <- Dnnconfmatrix$byClass["F1"]

# Print the metrics

print(Dnnconfmatrix)

cat("Accuracy:", Dnnaccuracy, "\n")

cat("Recall:", Dnnrecall, "\n")

cat("Precision:", Dnnprecision, "\n")

cat("F1 Score:", Dnnf1score, "\n")

...

```

6.4 Model Comparison

```
``{r}
```

```
#print accuracy for each model
```

```
Dlogaccuracy
```

```
Ddtaccuracy
```

```
Drfaccuracy
```

```
Dgbmaccuracy
```

```
Dnnaccuracy
```

```
``
```

```
``{r}
```

```
# Store all accuracies in a named vector
```

```
Digital_accuracies <- c(Dlogaccuracy, Ddtaccuracy, Drfaccuracy, Dgbmaccuracy, Dnnaccuracy)
```

```
Dnames <- c("Dlogaccuracy", "Ddtaccuracy", "Drfaccuracy", "Dgbmaccuracy", "Dnnaccuracy")
```

```
# Create a data frame for ggplot
```

```
Digital_df <- data.frame(Model = Dnames, Accuracy = Digital_accuracies)
```

```
# Find the highest accuracy
```

```
Dmax_accuracy <- max(Digital_accuracies)
```

```

# Find which model achieved the highest accuracy

Dbest_model <- Dnames[which.max(Digital_accuracies)]


# Visualize the accuracies using ggplot2

ggplot(Digital_df, aes(x = Model, y = Accuracy, fill = Model)) +

  geom_bar(stat = "identity", width = 0.6) +

  geom_text(aes(label = round(Accuracy, 2)), vjust = -0.3) +

  labs(title = "Digital Model Accuracies", y = "Accuracy", x = "Models") +

  theme_minimal() +

  theme(legend.position = "none")

```

```{r}

# Print the highest accuracy and the corresponding model

cat("The highest accuracy is", Dmax_accuracy, "achieved by", Dbest_model, "\n")

```

7. Evaluation

```{r}

```

```

```
``{r}
```

```
Create a data frame with the accuracies
```

```
Evaluationdf <- data.frame(
 Techniques = c("Logaccuracy", "Dtaccuracy", "Rfaccuracy", "Gbmaccuracy", "NNaccuracy",
 "Alogaccuracy", "Adtaccuracy", "Arfaccuracy", "Agbmaccuracy", "Annaccuracy",
 "Dlogaccuracy", "Ddtaccuracy", "Drfaccuracy", "Dgbmaccuracy", "Dnnaccuracy"),
 Accuracy = c(Logaccuracy, Dtaccuracy, Rfaccuracy, Gbmaccuracy, NNaccuracy,
 Alogaccuracy, Adtaccuracy, Arfaccuracy, Agbmaccuracy, Annaccuracy,
 Dlogaccuracy, Ddtaccuracy, Drfaccuracy, Dgbmaccuracy, Dnnaccuracy),
 Precision = c(Logprecision, Dtprecision, Rfprecision, Gbmprecision, NNprecision,
 Alogprecision, Adtprecision, Arfpprecision, Agbmprecision, Annprecision,
 Dlogprecision, Ddtprecision, Drfpprecision, Dgbmprecision, Dnnprecision),
 Recall = c(Logrecall, Dtrecall, Rfrecall, Gbmrecall, NNrecall,
 Alogrecall, Adtre recall, Arfrecall, Agbmrecall, Annrecall,
 Dlogrecall, Ddtrecall, Drfrecall, Dgbmrecall, Dnnrecall),
 F1_score = c(Logf1score, Dtf1score, Rff1score, Gbmf1score, NNf1score,
 Alogf1score, Adtf1score, Arff1score, Agbmf1score, Annf1score,
 Dlogf1score, Ddtf1score, Drff1score, Dgbmf1score, Dnnf1score)
)
```

```
Print the accuracy table
```

```
Evaluationdf
```

```
...
```

```
``{r}
```

```
library(reshape2)
```

```
Find the highest accuracy and the corresponding model
```

```
max_eval <- max(Evaluationdf$Accuracy)
```

```
evalbestmodel <- Evaluationdf$Techniques[which.max(Evaluationdf$Accuracy)]
```

```
Convert the data from wide to long format for easier plotting
```

```
Evaluationdf_long <- melt(Evaluationdf, id.vars = "Techniques",
```

```
 variable.name = "Metric", value.name = "Value")
```

```
Plotting the metrics for each technique
```

```
ggplot(Evaluationdf_long, aes(x = Techniques, y = Value, fill = Metric)) +
```

```
 geom_bar(stat = "identity", position = "dodge", width = 0.7) +
```

```
 geom_text(aes(label = round(Value, 2)),
```

```
 position = position_dodge(width = 0.7),
```

```
 vjust = -0.5, size = 3)
```

```
labs(title = "Comparison of Metrics Across Techniques",
```

```
 x = "Techniques",
```

```
 y = "Value") +
```

```

theme_minimal() +

theme(axis.text.x = element_text(angle = 45, hjust = 1)) +

scale_fill_brewer(palette = "Set3") # Use a color palette for differentiation
...

```{r}

# Visualize the accuracies using ggplot2

ggplot(Evaluationdf, aes(y = Techniques, x = Accuracy, fill = Techniques)) +

  geom_bar(stat = "identity", width = 0.6) +

  geom_text(aes(label = round(Accuracy, 2)), vjust = -0.3) +

  labs(title = "Model Accuracies", y = "Accuracy", x = "Techniques") +

  theme_minimal() +

  theme(legend.position = "none") +

  coord_flip() # Flip the coordinates for better readability if there are many models
...

```{r}

Print the highest accuracy and the corresponding model

cat("The highest accuracy is", max_eval, "achieved by", evalbestmodel, "\n")
...

8. Analogue Hybrid Model

```{r}

length(Annpredclass)

```

```
length(Arfpredclass)
```

```
Arfpredict_pos_class <- Arfpredict[, 2]
```

```
length(Arfpredict_pos_class)
```

```
class(Arfpredict_pos_class)
```

```
class(Annpredclass)
```

```
#Combined model using Random Forest and Neural Network by averaging of the probabilities
```

```
combined_predictions <- (Arfpredict_pos_class + (as.numeric(as.character(Annpredclass)))) / 2
```

```
# Step 5: Convert Combined Predictions to Class Labels
```

```
combined_predictions_class <- ifelse(combined_predictions > 0.5, 1, 0)
```

```
length(combined_predictions_class)
```

```
```
```

```
Analogue Hybrid Evaluate
```

```
```{r}
```

```
length(combined_predictions_class)
```

```
length(dlAtest$Aircompfail)
```

```
# Ensure that both vectors are factors
```



```
combined_predictions_factor <- factor(combined_predictions_class,
levels(as.factor(Atest$Aircompfail)))
```

```
actual_factor <- factor(Atest$Aircompfail)
```

```
# Calculate performance metrics
```

```
Ahybridconf_matrix <- confusionMatrix(combined_predictions_factor, actual_factor)
```

```
print(Ahybridconf_matrix)
```

```
...
```

```
``{r}
```

```
# Extract recall, precision, and F1 score
```

```
Ahyaccuracy <- Ahybridconf_matrix$overall['Accuracy']
```

```
Ahyrecall <- Ahybridconf_matrix$byClass["Recall"]
```

```
Ahyprecision <- Ahybridconf_matrix$byClass["Precision"]
```

```
Ahyf1score <- Ahybridconf_matrix$byClass["F1"]
```

```
Ahyaccuracy
```

```
Ahyrecall
```

```
Ahyprecision
```

```
Ahyf1score
```

```
...
```

```
``{r}
```

```
# Save Metro5 as a CSV file  
  
write.csv(Metro5, "Metro5.csv", row.names = FALSE)  
  
'''
```