BITI 3533 ARTIFICIAL INTELLIGENCE PROJECT MANAGEMENT

TITLE:

Elevator Predictive Maintenance

LECTURER:

Profesor Ts. Dr. Burhanuddin Bin Mohd Aboobaider

PROJECT MEMBER:

Name	Matric No.	Project Role
Darrick Tang Cheng Ong	B032110246	Project Manager
Quek Yu Xuan	B032110216	Member
Lai Chun Wai	B032110178	Member
Tan Zi Bin	B032110156	Member

TABLE OF CONTENTS

Introduction

In our modern urban landscape, the towering structures that define city skylines rely heavily on a crucial but often overlooked element: elevators. These vertical conduits are the lifelines of high-rise buildings, facilitating seamless movement and access across multiple floors. Yet, while they streamline our daily routines, elevators themselves are intricate mechanical systems susceptible to wear, tear, and unexpected glitches.

The reliability of these systems is paramount, ensuring not just convenience but also safety for occupants while minimizing disruptions to building operations. However, disruptions are inevitable due to the inherent complexities of these mechanisms, leading to inconveniences for users and operational challenges for building managers. Moreover, these interruptions entail increased maintenance costs and, in worst-case scenarios, potential safety hazards.

To counter these challenges, a groundbreaking initiative has emerged: the Elevator Failure Prediction Project. Its mission is to transform maintenance strategies by leveraging the wealth of data generated by a myriad of IoT sensors installed within elevator systems. These sensors capture an array of parameters, from the minute details recorded by electromechanical sensors like the Door Ball Bearing Sensor to environmental factors such as Humidity, as well as crucial physical metrics like Vibration.

The essence of this project lies in predictive maintenance, a revolutionary approach that employs AI-based analytics to foresee and report potential issues before they materialize. By amalgamating real-time sensor data with field information, this system alerts users and maintenance teams about impending breakdowns, ensuring that maintenance remains scheduled and predictable. Essentially, predictive maintenance is a proactive, data-driven method that evaluates equipment condition, foreseeing when maintenance should be undertaken.

Artificial Intelligence (AI) plays a pivotal role in this paradigm shift, enhancing elevator maintenance through predictive capabilities, bolstering system monitoring, and optimizing maintenance schedules. This fusion of technology not only ensures smoother operations but also sets a new standard for elevators, prioritizing safety, efficiency, and reliability in vertical transportation.

Background

In the realm of elevators, the Elevator Failure Prediction project emerges as a pioneering endeavor, driven by a rich dataset extracted from a network of IoT sensors. These sensors, ranging from electromechanical marvels like the Door Ball Bearing Sensor to environmental influencers like Humidity, and even tangible metrics like Vibration, form the backbone of this ambitious initiative.

Central to this project is a laser-sharp focus on predicting the absolute value of Vibration—a critical parameter in the world of elevator functionality. Vibration, when decoded correctly, holds the key to unraveling potential malfunctions before they surface, serving as a vital precursor to looming elevator failures.

By dissecting and understanding this multifaceted dataset, the goal is to craft predictive models capable of preempting these issues. The overarching aim is to revolutionize elevator maintenance by forecasting and preventing failures, thereby elevating safety, reliability, and operational efficiency to unprecedented heights.

Case Study

This project delves into an in-depth analysis of elevator operational data collected at a frequency of 4Hz during high-peak and evening usage periods within a building, spanning from 16:30 to 23:30. The time series data provides a rich source of information, capturing the nuances of elevator performance during crucial usage hours.

The predictive maintenance aspect targets the elevator car door system, aiming to preemptively detect and address issues that might lead to failure. By identifying patterns in the data and creating predictive models, this study seeks to optimize maintenance schedules, mitigate unplanned stops, and extend the longevity of elevator systems.

Utilizing machine learning and predictive analytics techniques, the project endeavors to establish a model capable of accurately predicting vibration levels. The insights derived from this analysis are poised to significantly impact the elevator industry by enabling timely interventions, ensuring smoother operations, and enhancing user experience within buildings.

This report encapsulates the methodology, findings, and implications of the Elevator Failure Prediction project, underscoring the importance of data-driven approaches in preemptive maintenance strategies for critical infrastructure like elevators.

Detail of software used

In the course of conducting the Elevator Failure Prediction project, a suite of software tools played pivotal roles across various stages of data analysis, model development, and documentation.

1. **Visual Studio Code**
   Python emerged as a cornerstone, leveraging libraries such as Pandas and NumPy for intricate data manipulation, while Matplotlib facilitated comprehensive data visualization, allowing for the creation of insightful plots and graphical representations. The Scikit-learn and TensorFlow frameworks were instrumental in crafting and refining machine learning models for predictive analysis. Complementing Python, Visual Studio Code provided an interactive environment conducive to data exploration, model prototyping, and facilitating a seamless workflow.

2. **Microsoft Excel**
   A portion of the Elevator Failure Prediction project dataset was initially stored in Microsoft Excel spreadsheets, containing crucial information on sensors (e.g., Door Ball Bearing Sensor), environmental factors (e.g., Humidity), and key physical metrics like Vibration measurements. Excel facilitated basic data organization and initial insights through its tabular structure and simple visualization tools.

3. **Microsoft Word**
   Microsoft Word served as the central platform for comprehensive documentation in the Elevator Failure Prediction project, housing detailed reports that structured methodologies, findings, and conclusions. This software facilitated a well-organized layout for presenting diverse project aspects such as project background, data preprocessing techniques, exploratory data analysis (EDA), model development, results, and conclusions. Its formatting tools enabled the creation of a coherent document structure with headings, subheadings, and bullet points. Additionally, collaborative features like track changes and comments facilitated iterative improvements by team members, ensuring accuracy and refinement. The final report, polished using Word's review and formatting capabilities, presented a comprehensive synthesis of insights ready for stakeholders' review and understanding.

4. **Microsoft Project**

Microsoft Project was instrumental in managing the Elevator Failure Prediction project, providing a comprehensive platform for planning, organizing, and tracking project progress. Its Gantt chart feature facilitated the creation of detailed schedules, depicting task dependencies and resource allocation, offering a clear roadmap for project phases and deadlines. The software allowed for efficient resource management by allocating human and material resources effectively across different project stages, ensuring optimal utilization. Collaboration features enabled real-time communication among team members, aiding in addressing potential delays and maintaining alignment with project objectives. Additionally, the software's reporting capabilities generated detailed progress reports, empowering stakeholders to monitor advancements and make informed decisions, while its flexibility allowed for seamless adjustments in response to changing project requirements, ensuring project agility.

5. **Git/GitHub**

Git, a distributed version control system, played a pivotal role in managing code repositories and tracking changes throughout the Elevator Failure Prediction project. It facilitated collaborative development by enabling team members to work on code simultaneously while preserving version history. GitHub, built around Git, served as a centralized platform for hosting repositories, fostering seamless collaboration through features like pull requests, code reviews, and issue tracking. Pull requests facilitated code review and discussion before integration, ensuring code quality. GitHub's issue tracking system organized tasks and bug fixes, while its repository stored not only code but also project documentation, offering a centralized location for project assets. Together, Git and GitHub streamlined development, ensuring transparency, version control, and effective collaboration among team members.

Development Plan

Project Objective

a. To improve the safety and reliability of elevator systems by utilizing data from IoT sensors and implementing predictive maintenance techniques.
b. To optimize maintenance schedules through the analysis of real-time sensor data, particularly focusing on critical parameters like Vibration, the project aims to develop predictive models.
c. To set a new standard for elevator operations by integrating artificial intelligence into maintenance processes.

SWOT Analysis

a. Strengths

i. Advanced Data Utilization

The project taps into real-time data streams critical for predictive maintenance strategies by utilizing a sophisticated network of Internet of Things (IoT) sensors. The wealth of data gathered allows for a thorough understanding of lift functionality.

ii. Incorporation of AI Integration

The project takes maintenance strategies to an unprecedented level by incorporating cutting-edge artificial intelligence (AI). AI-powered predictive analytics not only improves the accuracy of maintenance predictions, but it also optimizes scheduling for preventive maintenance and repairs.

iii. Comprehensive Planning and Strategy

A meticulously crafted project plan covers a wide range of topics, from risk management strategies to stringent quality standards. Budget estimations that are detailed ensure transparency and accountability in financial allocations.

iv. **Highly Competent and Specialized Team**

A skilled team boasting diverse expertise in software development, data analytics, IoT, and maintenance ensures comprehensive coverage across all project facets.

v. **Structured Milestones and Clear Objectives**

The project's roadmap delineates clear milestones and well-defined objectives, ensuring systematic progress tracking and alignment with project goals.

vi. **Thorough Budget Estimation**

Detailed cost breakdowns covering personnel, materials, equipment, and budgetary projections demonstrate an acute awareness of financial planning and resource allocation.

b. Weakness

i. **Dependency on Data Quality**

The availability and accuracy of data obtained from IoT sensors are critical to the project's success. Predictive models' efficacy may be jeopardized by inaccurate or insufficient data.

ii. **Integration Challenges**

Integration complexities with existing lift infrastructure pose potential hurdles. Aligning predictive maintenance solutions seamlessly within current systems might require extensive resources and time.

iii. **Financial Constraints**

The high cost of implementing AI algorithms, high-tech sensors, and predictive analytics tools may put a strain on the project's budget.

iv. **Navigating Regulatory Compliance**

Adherence to stringent industry regulations and safety standards may influence technology choices and alter strategic approaches, potentially affecting the project's trajectory.

c. Opportunities

i. **Enhanced Safety and Reliability**

Elevator systems with enhanced predictive maintenance capabilities can significantly improve safety standards, attracting safety-conscious customers and enhancing market reputation.

ii. **Market Leadership Potential**

Pioneering predictive maintenance within the elevator systems industry positions the project as a potential frontrunner, setting new standards and garnering industry recognition.

iii. **Potential for Technological Advancements**

The project opens doors for innovation within elevator systems, potentially leading to breakthroughs and advancements in the field.

iv. **Collaborative Partnerships**

There are numerous opportunities for strategic partnerships with elevate manufacturers or data analytics firms, fostering mutual growth and innovation.

d. Threads

i. Data Quality Concerns

Inaccurate or inadequate data collected from IoT sensors might compromise the precision and effectiveness of predictive models, impacting maintenance strategies.

ii. Integration Complexity

Aligning predictive maintenance with existing elevate systems presents a formidable challenge that may cause project milestones to be delayed or obstructed.

iii. Financial Uncertainties

Unexpected expenses and financial limitations might strain the allocated budget, potentially limiting the implementation of essential components of the project.

iv. Regulatory Compliance Challenges

Adapting to evolving safety standards and industry regulations might necessitate adjustments, potentially impacting project timelines and strategies.

This in-depth SWOT analysis highlights internal strengths and weaknesses while shedding light on external opportunities and threats that could influence the Elevator Failure Prediction Project's trajectory and success. Addressing these factors strategically will be imperative in navigating challenges and maximizing opportunities for project advancement.

Project Constraint

a. **Data Quality and Availability**

The project's success is largely dependent on the availability and quality of data from Internet of Things sensors. Limitations resulting from missing or erroneous data could reduce the predictive models' efficacy and impede the project's overall goals.

b. **Integration Challenges**

There may be integration issues when implementing predictive maintenance solutions with the current infrastructure and lift systems. The timeline and resources for the project may be limited by compatibility problems and the requirement for a seamless integration.

c. **Budgetary Constraints**

The project's budgetary constraints may limit its potential for success. Predictive analytics tools, AI algorithms, and sophisticated sensors can be expensive to implement. A crucial factor is making sure the project is effective while adhering to financial restrictions.

d. **Regulatory Compliance**

The project needs to follow the safety guidelines and industry rules that govern lift systems. Compliance requirements could complicate the project by influencing the technology selection, data processing protocols, and predictive maintenance strategy as a whole.

Project Risk Management

a. Event Risk: Predictive models may not work as well if IoT sensors provide inaccurate or insufficient data.
    i. Risk Probability: Moderate
    ii. Risk Impact: High
    iii. Risk Mitigation:
        1. Put in place strict procedures for validating data and create guidelines for ongoing evaluation and enhancement of data quality.
    iv. Contingency Plan:
        1. In the event of data discrepancies, have a backup data source or a fallback mechanism to ensure continuity in predictive maintenance.

b. Event Risk: Challenges in combining predictive maintenance programmes with the infrastructure and systems currently in place for lifts.
    i. Risk Probability: High
    ii. Risk Impact: Moderate
    iii. Risk Mitigation:
        1. Perform in-depth analyses of the system, work closely with lift manufacturers, and allot enough funds to ensure a smooth integration.
    iv. Contingency Plan:
        1. Develop a phased integration approach, allowing for gradual implementation and identifying alternative integration paths if challenges arise.

c. Event Risk: Unforeseen expenses related to the use of AI algorithms, sophisticated sensors, and predictive analytics tools.
    i. Risk Probability: Moderate
    ii. Risk Impact: High
    iii. Risk Mitigation:
        1. Develop a comprehensive budget, conduct regular cost assessments, and prioritize cost-effective solutions without compromising project objectives.
    iv. Contingency Plan:

        1. Establish a contingency fund for unexpected expenses and regularly revisit the budget to adjust allocations based on project needs.

d. Event Risk: Failure to comply with industry regulations and safety standards governing elevator systems.
    i. Risk Probability: Low
    ii. Risk Impact: High
    iii. Risk Mitigation:
        1. Establish a dedicated regulatory compliance team, stay informed about evolving standards, and conduct regular audits to ensure adherence.
    iv. Contingency Plan:
        1. Collaborate with regulatory bodies, maintain open communication, and have a rapid response plan in place to address compliance issues promptly.

Proposed Solution

Work Break Down (WBS)

Elevator Failure Prediction Project

1. Project Initiation
    1.1. Define Project Title and Team Members
    1.2. Develop Project Charter
        1.2.1. Outline Project Scope and Deliverables
        1.2.2. Define Project Roles and Responsibilities
2. Planning and Documentation
    2.1. Project Background
    2.2. Case Study
    2.3. Detail Software Used
    2.4. AI Project Management
        2.4.1. Planning
            2.4.1.1. Develop Project Plan
        2.4.2. Risk Management
            2.4.2.1. Identify Project Risks
            2.4.2.2. Develop Risk Mitigation Strategies
        2.4.3. Work Breakdown Structure (WBS)
            2.4.3.1. Break Down Major Project Phases
            2.4.3.2. Assign Tasks to Team Members
        2.4.4. Cost Management
            2.4.4.1. Estimate Project Costs
            2.4.4.2. Create Budget
        2.4.5. Time Management
            2.4.5.1. Schedule Milestones and Deliverables
        2.4.6. Quality Management
            2.4.6.1. Define Quality Standards
            2.4.6.2. Establish Quality Assurance Processes

3. Algorithm Development
    3.1. Research and Select AI Algorithms
    3.2. Develop Predictive Models
    3.3. Define Vibration Parameter Analysis
    3.4. Problem Solving Approach
4. Project Implementation
    4.1. Set Up IoT Sensors
    4.2. Data Collection and Preprocessing
    4.3. Model Training and Testing
    4.4. Integration with Elevator Systems
    4.5. User and Stakeholder Training
    4.6. Continuous Improvement Mechanism
5. Documentation and Reporting
    5.1. Compile Project Report
    5.2. Create Flowcharts and Diagrams
    5.3. Document Implementation Process
    5.4. Capture Project Outputs and Results
6. Project Review and Closure
    6.1. Evaluate Project Success
    6.2. Gather Feedback from Users
    6.3. Finalize Project Documentation
    6.4. Project Closure Report

WBS Gantt chart please refer to Appendix I (WBS Gantt Chart)

Cost Management

1. Personnel Cost

Team Member	Role	Description	Hour	Hourly rate
Darrick	Project Manager	Overseeing the project, coordinating teams	40 hours/ week	RM120
Tan	Data Scientists/Analysts	Developing predictive models, analyzing data	40 hours/ week	RM100
Quek	Software Engineer/Developer	Building and maintaining software infrastructure	40 hours/ week	RM80
Lai	IoT Specialists	Handling IoT devices, connectivity	40 hours/ week	RM90
Lai	Maintenance Technicians/Engineers	Installing and maintaining sensors/equipment	40 hours/ week	RM70

2. Material Costs

Materials	Description	Quantity	Unit Cost (RM)	Total Cost (RM)
Data analytics software	This can give the insights of the data from the sensors and can quickly locate the problems.	1	200	200

Remote monitoring systems	This can monitor and control all sensors to do some configurations to predict the maintenance.	1	400	400
Repairing Materials	There are some materials for repairing the hardwares, for example, sensors.	1	1700	1700
Repairing Tools	To do the repairing, the tools are needed to apply the materials correctly and precisely.	2	300	600
Total Cost Required (RM)	2900			

3. Equipment Cost

Sensor Type	Description	Quantity	Unit Cost (RM)	Total Cost (RM)
Electromechanical sensors	Used to monitor the condition of door ball bearings.	5	200	1000
Ambiance (Humidity)	Used to monitor humidity levels, which can affect elevator performance.	1	300	300
Physics (Vibration)	Used to monitor vibration levels, which can indicate potential problems with the elevator.	1	400	400
Total	1700			

4. Budget Estimation

Budget	Total Cost (RM)
Personnel Cost	73600 (1 month)
Material	2900
Equipment	1700
Total (RM)	78200

Time Management

Project Milestone	W 1	W 2	W 3	W 4	W 5	W 6	W 7	W 8	W 9	W 10	W 11	W 12
Project Initiation Phase												
Planning and Documentation Phase												
Software Details and Algorithm Development Phase												
Project Implementation Phase												
Documentation and Reporting Phase												
Project Review and Closure Phase												

1. Project Initiation Phase

Milestone 1: Project Kick-off

- Date: Week 1 - Week 2
- Objective: Official commencement of the project, including team introductions and distribution of roles.

- Deliverables:
    - Signed project charter.
    - Team roles and responsibilities document.

Milestone 2: Project Charter Approval

- Date: Week 1 - Week 2
- Objective: Formal approval of the project charter.
- Deliverables:
    - Approved project charter.

2. Planning and Documentation Phase

Milestone 3: Gantt Chart Finalization

- Date: Week 3
- Objective: Completion of the Gantt chart outlining project phases, tasks, and timelines.
- Deliverables:
    - Finalized Gantt chart.

Milestone 4: Software Selection

- Date: Week 3 - Week 7
- Objective: Identification and selection of the required software for the project.
- Deliverables:
    - Software requirements document.
    - Chosen software documentation.

Milestone 5: AI Project Management Plan

- Date: Week 3 - Week 7
- Objective: Completion of the AI project management plan, including WBS, cost estimates, and risk management strategies.
- Deliverables:
    - AI project management plan document.

3. Software Details Phase

Milestone 6: Software Installation and Configuration

- Date: Week 4 - Week 9
- Objective: Successful installation and configuration of the selected software.
- Deliverables:
    - Installed and configured software.

Milestone 7: Integration Protocols Defined

- Date: Week 4 - Week 9
- Objective: Establishment of protocols for integrating the software with elevator systems.
- Deliverables: Documented integration protocols.

4. Algorithm Development Phase

Milestone 8: AI Algorithm Selection

- Date: Week 4 - Week 9
- Objective: Finalization of AI algorithms to be used in predictive maintenance.
- Deliverables:
    - Documented AI algorithm selection.

Milestone 9: Vibration Parameter Analysis

- Date: Week 4 - Week 9
- Objective: Identification of key vibration parameters for analysis.
- Deliverables:
    - Documented vibration parameter analysis.

5. Project Implementation Phase

Milestone 10: IoT Sensor Setup

- Date: Week 5 - Week 11
- Objective: Successful installation and setup of IoT sensors.
- Deliverables:
    - Installed and configured IoT sensors.

Milestone 11: Model Training and Testing

- Date: Week 6 - Week 11
- Objective: Training and testing of predictive models using collected data.
- Deliverables:
    - Trained and tested predictive models.

Milestone 12: Integration with Elevator Systems

- Date: Week 6 - Week 11
- Objective: Successful integration of predictive maintenance systems with existing elevator systems.
- Deliverables:
    - Integrated elevator and AI system.

Milestone 13: User and Stakeholder Training

- Date: Week 7 - Week 11
- Objective: Training sessions conducted for end-users and stakeholders.
- Deliverables:
    - User training materials.
    - Stakeholder training documentation.

Milestone 14: Continuous Improvement Mechanism Established

- Date: Week5 - Week11
- Objective: Implementation of continuous monitoring processes and feedback loops.
- Deliverables:
    - Documented continuous improvement mechanism.

6. Documentation and Reporting Phase

Milestone 15: Project Report Compilation

- Date: Week 8 - Week 12
- Objective: Compilation of the comprehensive project report.
- Deliverables:
  - Final project report.

Milestone 16: Flowcharts and Diagrams

- Date: Week 8 - Week 12
- Objective: Creation of flowcharts and diagrams to visualize system processes.
- Deliverables:
  - Flowcharts and system architecture diagrams.

Milestone 17: Implementation Process Documentation

- Date: Week 8 - Week 12
- Objective: Detailed documentation of the step-by-step implementation process.
- Deliverables:
  - Documented implementation process.

Milestone 18: Project Outputs and Results

- Date: Week 8 - Week 12
- Objective: Capture and analysis of project outputs and results.
- Deliverables:
  - Recorded AI model outputs.
  - Analysis and interpretation of results.

7. Project Review and Closure Phase

Milestone 19: Project Evaluation

- Date: Week 9 - Week 12
- Objective: Evaluation of project success against initial objectives.
- Deliverables:
    - Project success evaluation report.

Milestone 20: Feedback Collection

- Date: Week 9 - Week 12
- Objective: Collection of feedback from end-users and stakeholders.
- Deliverables:
    - Compiled user and stakeholder feedback.

Milestone 21: Finalization of Project Documentation

- Date: Week 9 - Week 12
- Objective: Review and finalization of all project documentation.
- Deliverables:
    - Finalized project documentation.

Milestone 22: Project Closure Report

- Date: Week 9 - Week 12
- Objective: Summarization of project achievements and identification of areas for future improvement.
- Deliverables:
    - Project closure report.

Quality Management

1. Introduction

Enforcing strict quality standards is essential to the Elevator Failure Prediction Project's success. This section describes the specific quality standards that will control different project aspects, making sure that the final product satisfies or surpasses predetermined standards.

2. Software Development Standards

2.1. Code Quality

2.1.1. Objective: Ensure readability, maintainability, and efficiency of the codebase.

2.1.2. Criteria:

- Adherence to coding standards (e.g., naming conventions, indentation).
- Use of comments for complex sections or logic.
- Minimization of code duplication.

2.2. Testing Standards

2.2.1. Objective: Validate the correctness and reliability of the developed algorithms and software.

2.2.2. Criteria:

- Comprehensive test coverage for all critical functionalities.
- Execution of unit tests, integration tests, and system tests.
- Documentation of test cases and results.

2.3. Documentation Standards

2.3.1. Objective: Facilitate understanding, future maintenance, and knowledge transfer.

2.3.2. Criteria:

- Clear and comprehensive documentation for code, algorithms, and software architecture.
- User manuals and guides for end-users and maintenance teams.
- Regular updates to documentation as the project progresses.

3. Data Management Standards

3.1. Data Quality

3.1.1. Objective: Ensure accuracy, completeness, and reliability of the dataset.

3.1.2. Criteria:

- Regular data validation processes to identify and rectify anomalies.
- Implementation of data integrity checks.
- Documentation of data collection methodologies.

3.2. Privacy and Security

3.2.1. Objective: Protect sensitive information and ensure compliance with data protection regulations.

3.2.2. Criteria:

- Encryption of sensitive data during transmission and storage.
- Access controls and authentication mechanisms.
- Regular security audits and vulnerability assessments.

4. Project Management Standards

4.1. Planning and Monitoring

4.1.1. Objective: Efficiently manage project timelines and resources.

4.1.2. Criteria:

- Adherence to the project schedule outlined in the Gantt chart.
- Regular progress monitoring and adjustments as needed.
- Resource utilization within defined limits.

4.2. Risk Management

4.2.1. Objective: Identify, assess, and mitigate project risks..

4.2.2. Criteria:

- Regular risk assessments and updates to the risk register.
- Implementation of risk mitigation strategies.
- Proactive identification of emerging risks.

5. Reporting Standards

5.1. Progress Reports

5.1.1. Objective: Provide transparent and informative updates on project progress.

5.1.2. Criteria:

- Regular generation of progress reports according to the predefined schedule.
- Inclusion of key performance indicators (KPIs) and milestone achievements.
- Clear communication of challenges and proposed solutions.

5.2. Quality Assurance Audits

5.2.1. Objective: Ensure compliance with established quality standards.

5.2.2. Criteria:

- Regular internal quality assurance audits.
- External audits by independent assessors, if applicable.
- Timely implementation of corrective actions.

6. Conclusion

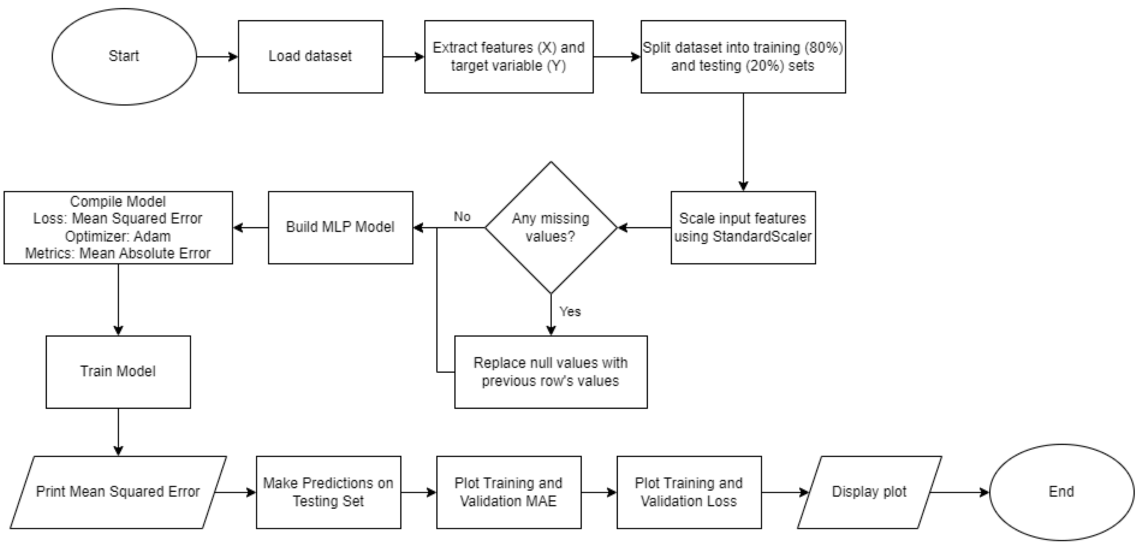
For the Elevator Failure Prediction Project to succeed, strong quality standards in software development, data management, project management, and reporting must be established. Following these guidelines will guarantee the project's successful conclusion and enhance the developed solution's long-term dependability and maintainability. Throughout the project lifecycle, continuous adherence to these quality standards will be ensured through periodic reviews and ongoing monitoring.

Flow, Algorithm and Problem-Solving

In this project, the algorithm used is a type of neural network used in this research called a Multi-Layer Perceptron (MLP). A particular kind of feedforward artificial neural network called an MLP comprises several layers of nodes, each of which is connected to the nodes in the layers above and below it. An input layer, two hidden layers with 64 neurons each, and an output layer with a single neuron make up this implementation's architecture.

The Rectified Linear Unit (ReLU), a popular option for adding non-linearity to neural networks, is the activation function utilized in the hidden layers. Since the job involves predicting a continuous output (vibration levels), it appears that the output layer is employed for regression as it lacks an explicit activation function.

The Adam optimizer is used to update the weights during the training process, and the Mean Squared Error (MSE) is used as the loss function in the model's training. A statistic used to track the model's performance during training and assessment is the Mean Absolute Error (MAE).



Flowchart of the Algorithm

The method of creating a predictive maintenance model with a Multi-Layer Perceptron (MLP) neural network is illustrated in the flowchart. The loading and preparation of the dataset, along with the feature and target variable extraction, come first. After that, input features are scaled and the dataset is divided into training and testing sets. To accommodate missing values, the flow has a decision point that may be checked and replaced as needed. The architecture, activation functions, and loss measures used in the construction of the neural network model are particular. The model is trained using training and validation data for a predetermined number of epochs following compilation. The Mean Squared Error (MSE) is printed during the assessment phase, which evaluates the model's performance on the testing set. Metrics for every epoch are saved in text files, and predictions are made and stored. Images of plots showing the Mean Absolute Error (MAE) and loss for training and validation are created.

The provided code addresses the problem of predictive maintenance in the elevator industry, specifically focusing on the early detection of potential failures in elevator car doors. The dataset, which was gathered from a range of Internet of Things (IoT) sensors, includes vital operating data such as vibration, ambient conditions like humidity, and electromechanical sensor data (Door Ball Bearing Sensor).

Predicting the absolute value of vibration, a crucial metric suggestive of possible problems with lift doors is the main objective. The Multi-Layer Perceptron (MLP) neural network, which is the implementation of the predictive maintenance model, is intended to analyze time series data taken at a high frequency (4Hz) during high-peak and evening lift usage periods.

With this model, the elevator industry may reduce unplanned stops by using this approach to proactively identify trends and abnormalities in vibration levels, enabling prompt repair and intervention. The model seeks to enhance the total equipment life cycle, minimize downtime, and improve reliability. The model's training and assessment are comprehensively understood thanks to the visualization tools and metrics included in the code, which also contribute to the model's efficacy for predictive maintenance applications in the lift sector.

Project Implementation and Output

Methodology

In this project of the Elevator Predictive Maintenance, we had used the Artificial Intelligence (AI) technique to achieve our goal. We chose the Neural Network which is one of the AI fields. Based on the Neural Network, we decided to use the Multilayer Perceptron (MLP) to build the prediction model. MLP is the feedforward neural network that has fully connected neurons and an activation function for the whole neural network. We use Rectified Linear Unit (ReLu) as the activation function.

Tools and Technologies

Apart from the softwares that we used, we used Python as the programming language to build the coding and model. Since Python has many needed libraries for the programmers. To implement the MLP into our model, TensorFlow is the open source in machine learning for everyone. In TensorFlow, Keras is the high level API to build the deep learning model. By using Keras, we can use the layer structures and model structures in Keras to build the MLP. In Keras, we used the Sequential Model, this model must have one input tensor and one output tensor only for each layer.

Data Collection and Preprocessing

The data that we used in this project is predictive-maintenance-dataset.csv, this dataset is gathered by Axenie A. and Bortoli S. by using the IoT sensors, which are Electromechanical sensors (Door Ball Bearing Sensor), Ambiance (Humidity), and Physics (Vibration). We noticed that there are null values inside the dataset. To clean the empty data, we used the replacement method. We replaced the null values by the values of the previous row. Then, we updated the data to a new dataset, called updated_file.csv. By using the new dataset, 80% of data is splitted into training data and 20% as the testing data.

Model Development

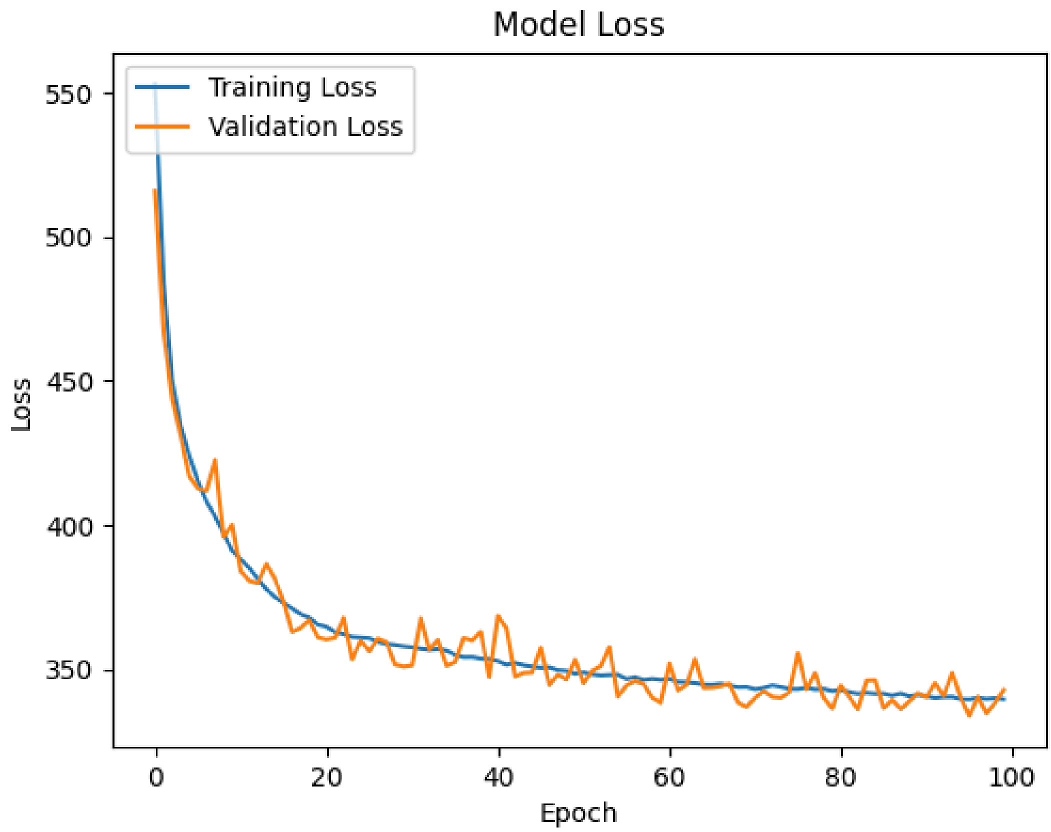
The first step is to read the dataset. From the dataset, we extracted input feature (X) and target variable (y). The input feature (X) is the arrays that are combined by the columns which are 'revolutions', 'humidity', 'x1', 'x2', 'x3', 'x4', 'x5' (x1 to x5 represents 5 sensors). For the 'vibration' column, we used it as the target variable. After that, we scaled the input feature by standardizing the training data and testing data separately. During standardizing, the means and the standard deviation are calculated for each feature, then do the transformation based on these statistics. Next, build the MLP model that has 2 hidden layers and each layer contains 64 neurons. The ReLu activation function is used. The output layer consists of 1 neuron only, since we wanted to do the regression tasks for prediction.

During the model compilation, we used Mean Squared Error (MSE) as the loss function, since it measures the average squared difference between the predicted and actual values. For metrics, Mean Absolute Error (MAE) is chosen to measure the average absolute differences and the average magnitude of errors. Then, we can evaluate the performance of our model. We trained the model with 100 epochs and 15 of batch size. After that, we wrote 2 text files for MSE and MAE to keep the errors as record. Based on MSE and MAE, we plotted the data into regression lines as the visualization of the results.

Performance Metrics

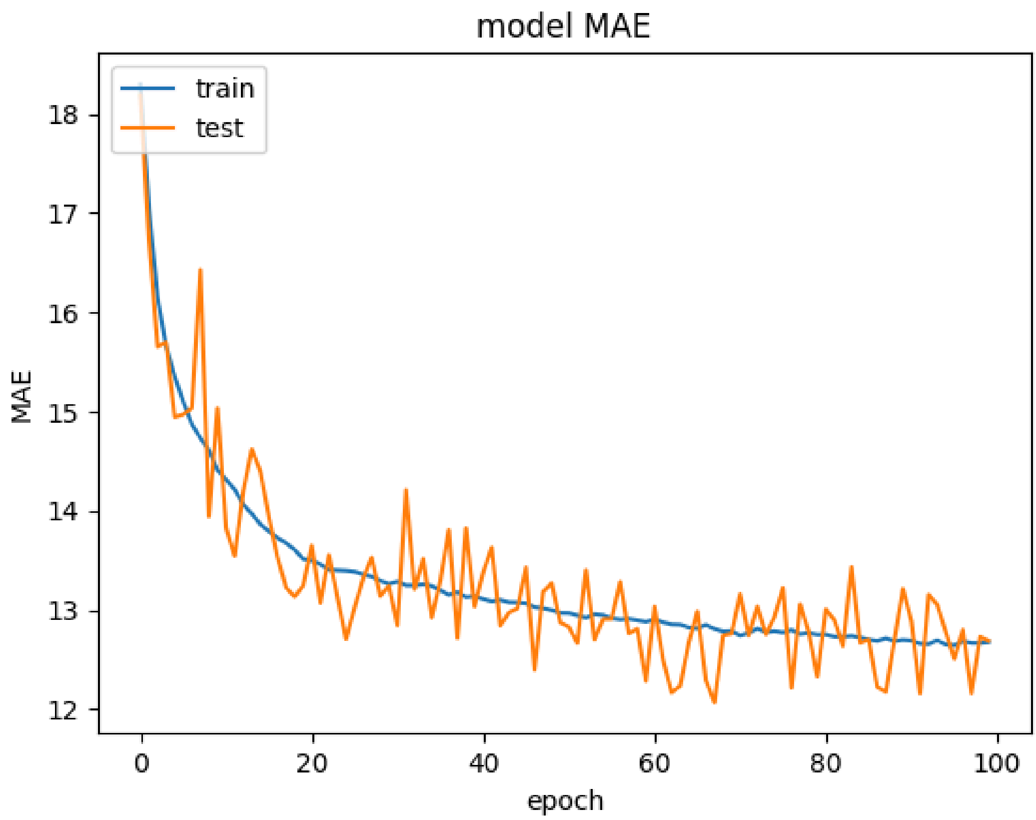
In the project, we used the Mean Squared Error (MSE) and Mean Absolute Error (MAE) as the evaluation metrics. After testing on the model that we trained, we have concluded that we have 342.76517 for MSE and 12.68732 for MAE. After more epochs, the errors become smaller. Then, when we do the comparison between the training data and testing data, we noticed that the performance on the testing data followed the performance on the training data, that means the model can be more accurate with more epochs, since the model can reduce the errors in the testing data also.

Visualization or Demonstration



Graph of Mean Squared Error (MSE)

Based on the graph, we can observe that the validation loss (MSE of the testing data) followed the trend and shape of the training loss (MSE of the training data). That means the model can also effectively reduce the errors after more epochs. That means the accuracy of the model improved over the epochs.



Graph of Mean Absolute Error (MAE)

The graph showed that the MAE of the training data reduces after more epochs. Then, the MAE of the testing data also reduces when more epochs are done. Although the MAE of the testing data was not as stable as the training data, it still follows the shape of the line. We can conclude that the accuracy will increase when more epochs are done.

Challenges

The challenges of this project is the quality of hardware. Since we used the laptop as the hardware to do the computing and train the model, there is the limitation. The quality of hardware is important since it can decide how many epochs that can run in the laptop. Then, the training process is time consuming. Although we used 2 hidden layers only but we have 64 neurons in each layer, this made the processes longer because more calculations are done.

Lessons Learned

During the training process, we noticed that the hardware is very important for neural network training. It may limit the performance and the potential of the model due to low computing power. We learnt that the devices that have good computation power are needed to get the desired results from the model. We confirmed that more model training is required to get and build a strong predictive model.

Conclusion

In the project of Elevator Predictive Maintenance, our goal is to build a predictive model to predict the maintenance needed before the elevator failure. This predictive model is to prevent the horrific elevator accident from happening and ensure the safety of the passengers. To achieve this goal, several sensors for detecting the humidity, revolutions (rotations), and vibration are used to collect the data for predicting the maintenance needed. The product of this project achieved the goal to predict the maintenance needed by using the MLP in the field of the neural network. The results show that the accuracy improves when more training is done.

To know the performance of the model, we had done the evaluation by using MSE and MAE metrics. By using this approach, we found out the difference of result between the training data and testing data. We identified that the errors decrease when more epochs are done. We can conclude that to improve the accuracy of the model, more training should be done. That means the predictive model can predict the maintenance more accurately when more training is done. However, the shortcoming of this model is the number of training is not enough to minimize the error since the limitation of the devices.

By following the processes to manage the project, we clearly understand the importance of the process. The processes give us a clearer view to ensure which is the current stage or process of the project. There are 5 processes for projects which are Initiating Process, Planning Process, Executing Process, Monitoring and Controlling Process, and Closing Process. For each process, we have different tasks to do and different purposes to achieve. During the Initiating Process and Planning Process, we discussed using the neural network approach, python programming language and their libraries to build the model we want.

During the Executing Process, we do the tasks to achieve the goals that we set in the Planning Process. To ensure our project always focuses on the goal, we have to do the tasks for Monitoring and Controlling Process. For the Closing Process, we have to complete the report and the project should end properly in this process. We have different outputs for different processes, for example, the reports for the WBS, Risk Management, Cost Management, Time Management, Quality Management, Project Objectives, SWOT Analysis, and Project Constraints.

For future recommendations, we recommend that more training should be done to build a more accurate predictive model. However, we should find the balance to prevent the overfitting problem. Then, strong computing devices are needed to achieve the goals and get the desired result.. Then, the devices can handle more training processes without going to crash and reduce the process time. We have learned that the hardware to build a neural network model is very important. Then, we should choose the appropriate libraries to build the MLP in the model. This is because some libraries need more computation power to use, it should depend on the quality of the device used in the project.

This Elevator Predictive Maintenance project is important for the safety of the passengers. This model can help us to schedule the maintenance needed for the elevator. So, we can depend on the result of the predictive model, to predict when the maintenance is needed to prevent the accidents from happening. Every building that has the elevator should always do the maintenance to prevent the tragedy happening and get more loss in financial and reputation. This project achieved the balance between the maintenance fees and the safety of the passengers. This project also can extend the life cycle of the elevator for a long time, since it can predict the maintenance, then we can prevent the further damage to the elevator.

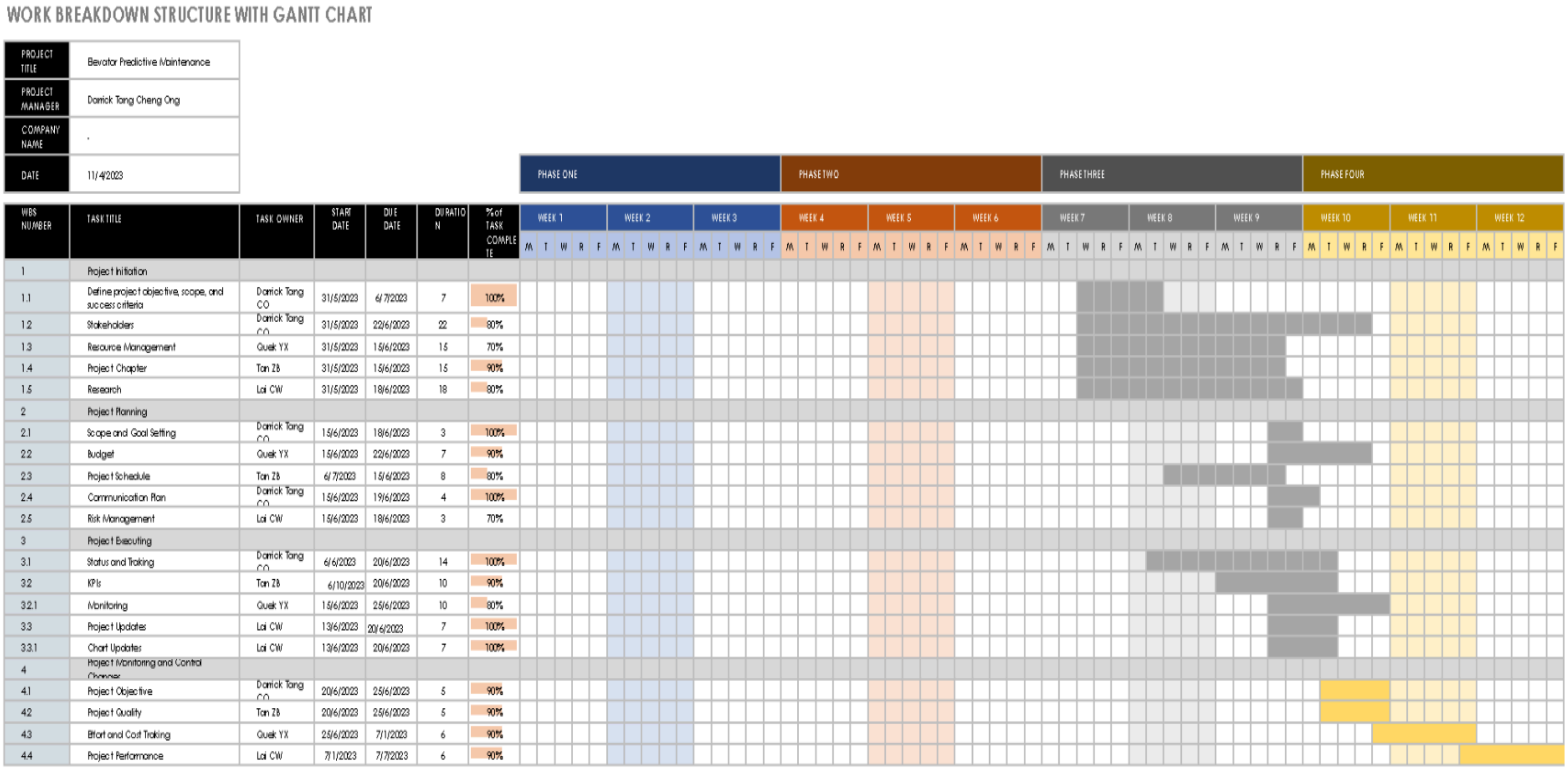
References

Axenie; A., Bortoli, S..(2020, February 7). Predictive maintenance dataset. Zenodo. https://zenodo.org/records/3653909

TensorFlow.(2023, June 8). Keras: The high-level API for TensorFlow. TensorFlow. https://www.tensorflow.org/guide/keras

Appendix

Appendix I (WBS Gantt Chart)



WORK BREAKDOWN STRUCTURE WITH GANTT CHART

PROJECT TITLE	Elevator Predictive Maintenance
PROJECT MANAGER	Darrick Tang Cheng Ong
COMPANY NAME	.
DATE	11/4/2023

WBS NUMBER	TASK TITLE	TASK OWNER	START DATE	DUE DATE	DURATION	% of TASK COMPLETE
1	Project Initiation					
1.1	Define project objective, scope, and success criteria	Darrick Tang CO	31/5/2023	6/7/2023	7	100%
1.2	Stakeholders	Darrick Tang CO	31/5/2023	22/6/2023	22	80%
1.3	Resource Management	Quek YX	31/5/2023	15/6/2023	15	70%
1.4	Project Chapter	Tan ZB	31/5/2023	15/6/2023	15	90%
1.5	Research	Lai CW	31/5/2023	18/6/2023	18	80%
2	Project Planning					
2.1	Scope and Goal Setting	Darrick Tang CO	15/6/2023	18/6/2023	3	100%
2.2	Budget	Quek YX	15/6/2023	22/6/2023	7	90%
2.3	Project Schedule	Tan ZB	6/7/2023	15/6/2023	8	80%
2.4	Communication Plan	Darrick Tang CO	15/6/2023	19/6/2023	4	100%
2.5	Risk Management	Lai CW	15/6/2023	18/6/2023	3	70%
3	Project Executing					
3.1	Status and Traking	Darrick Tang CO	6/6/2023	20/6/2023	14	100%
3.2	KPIs	Tan ZB	6/10/2023	20/6/2023	10	90%
3.2.1	Monitoring	Quek YX	15/6/2023	25/6/2023	10	80%
3.3	Project Updates	Lai CW	13/6/2023	20/6/2023	7	100%
3.3.1	Chart Updates	Lai CW	13/6/2023	20/6/2023	7	100%
4	Project Monitoring and Control Changes					
4.1	Project Objective	Darrick Tang CO	20/6/2023	25/6/2023	5	90%
4.2	Project Quality	Tan ZB	20/6/2023	25/6/2023	5	90%
4.3	Effort and Cost Traking	Quek YX	25/6/2023	7/1/2023	6	90%
4.4	Project Performance	Lai CW	7/1/2023	7/7/2023	6	90%