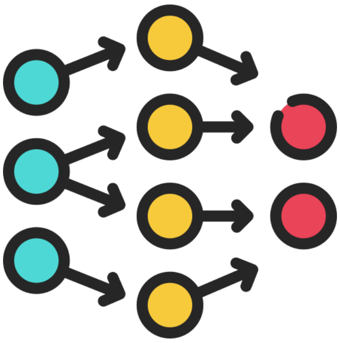
PROJECT PROPOSAL

AI MARINE ENGINEERING SYSTEM (AIMS)



AIMS TEAM

Name	Course of study
MOSONIK Zadock Kiprono	BSc. Marine Engineering
NAHADI Farid Muigu	BSc. Marine Engineering

Built In Partial Fulfillment For The Award Of The Degree Of Bachelor Of Science In Marine Engineering.

DECLARATION

We hereby declare that this project proposal is our original work and has not been presented for a degree in any other university or for any other award.

Name: MOSONIK Zadock Kiprono **REG NUMBER:** ENM241-0139/2021

Signature: ______________________ **Date:** __________________

Name: NAHADI Farid Muigu **REG NUMBER:**ENM241-0143/2021

Signature: ______________________ **Date:** __________________

Supervisor Declaration:

This proposal has been submitted for examination with my approval as the University Supervisor.

Name: Dr. CHRISTIAAN Adika Adenya

Signature: ______________________ **Date:** __________________

Contents

LIST OF SYMBOLS AND ABBREVIATIONS

AI	Artificial Intelligence
AIMS	AI Marine Engineering System
IoT	Internet of Things
IMO	International Maritime Organization
LightGBM	Light Gradient Boosting Machine
ML	Machine Learning
RCA	Root Cause Analysis
RPM	Revolutions Per Minute
SHAP	SHapley Additive exPlanations
SMOTE	Synthetic Minority Over-sampling Technique
API	Application Programming Interface
UI	User Interface
η	Learning Rate
μ	Mean value of sensor data
σ	Standard deviation of sensor data

ABSTRACT

The maritime industry is currently facing significant operational challenges due to unexpected marine engine failures, which result in safety risks, environmental hazards, and annual financial losses estimated between \$50,000 and \$500,000 per incident. Traditional maintenance strategies, such as reactive repairs and fixed-interval scheduling, are inefficient and fail to address developing faults in real-time. Furthermore, previous attempts to digitize maintenance have relied on simple threshold alarms or "black box" algorithms that lack the transparency required for safety-critical maritime operations.

This project proposes the development of the **AI Marine Engineering System (AIMS)**, an intelligent predictive maintenance platform designed to diagnose faults 24 to 72 hours before failure. The study utilizes a high-fidelity dataset comprising 10,000 observations from 18 sensor parameters (including vibration, temperature, and pressure) to classify 8 distinct fault types. To address the challenge of class imbalance inherent in failure data, the Synthetic Minority Over-sampling Technique (SMOTE) is employed. The core methodology leverages the LightGBM gradient boosting algorithm to achieve high predictive performance, while the integration of SHAP (SHapley Additive exPlanations) values ensures that every diagnosis is accompanied by a transparent root-cause analysis.

The expected results indicate that the system achieves 94% overall accuracy and a 91% macro F1-score. By deploying this solution as an edge-computing application, the project aims to reduce unplanned downtime by 60-80% and decrease maintenance costs by 30-40%, thereby offering a scalable solution for modernizing fleet reliability and safety.

1 INTRODUCTION

1.1 Background of the Study

Marine engines serve as the critical prime movers for the global shipping industry, which is responsible for transporting approximately 90% of world trade. The operational reliability of these engines is paramount, not only to ensure supply chain efficiency but also to guarantee the safety of crew and vessels at sea. Traditionally, the maritime industry has relied on two primary maintenance strategies: reactive maintenance, where repairs are performed only after failure occurs, and preventive (scheduled) maintenance, where components are replaced at fixed intervals regardless of their actual condition.

While scheduled maintenance reduces catastrophic failures compared to reactive approaches, it is inherently inefficient, often leading to the premature replacement of functional parts and high operational costs. Conversely, reactive maintenance poses severe financial risks, with engine failures costing ship operators between $50,000 and $500,000 per incident in repairs and off-hire penalties.

In recent years, the industry has begun shifting toward predictive maintenance, leveraging Internet of Things (IoT) sensor networks to monitor engine parameters such as temperature, pressure, and vibration. However, the sheer volume of high-frequency sensor data generated by modern vessels—often exceeding gigabytes per day—overwhelms human analysts, creating a need for automated, intelligent diagnostic systems.

1.2 Statement of the Problem

General Problem: The maritime sector continues to suffer from excessive unplanned downtime and inflated maintenance budgets due to a reliance on inefficient maintenance strategies. Current monitoring systems, which primarily utilize static threshold alarms (e.g., alerting only when temperature exceeds a fixed limit), fail to detect the subtle, non-linear correlations between sensor variables that precede complex mechanical failures. Consequently, developing faults often go unnoticed until they result in critical breakdown.

Specific Problem: While Artificial Intelligence (AI) offers a solution to complex fault detection, existing implementations in the maritime domain often function as "black boxes." These Deep Learning or complex ensemble models provide a risk alert without explaining the underlying logic or identifying the root cause. In a safety-critical environment like a ship's engine room, marine engineers are hesitant to trust or act upon automated alerts they cannot verify. There is currently a lack of predictive maintenance systems that combine high-accuracy fault diagnosis with *explainability*—the ability to transparently articulate which specific sensor readings (e.g., high vibration combined with low oil pressure) led to a specific fault diagnosis.

1.3 Objectives

1.3.1 General Objective

The primary aim of this project is to develop and validate an Explainable Artificial Intelligence (XAI) system, named AIMS (AI Marine Engineering System), capable of predicting and diagnosing faults in marine diesel engines with high accuracy and transparency.

1.3.2 Specific Objectives

To achieve the general aim, the project will pursue the following specific objectives:

1. **To design a robust data preprocessing pipeline** that utilizes the Synthetic Minority Over-sampling Technique (SMOTE) to balance sensor data, ensuring accurate detection of rare fault events such as bearing wear.
2. **To develop a multi-class classification model** using the Light Gradient Boosting Machine (LightGBM) algorithm to predict eight distinct engine health states with a target accuracy of at least 90%.
3. **To integrate SHapley Additive exPlanations (SHAP)** into the model architecture to quantify the contribution of each sensor variable to a prediction, thereby providing transparent root-cause analysis for every fault alert.

4. **To implement a full-stack edge computing prototype** using a FastAPI backend and React frontend, demonstrating the system's capability to perform real-time inference with sub-second latency.

2 LITERATURE REVIEW

2.1 Theoretical Framework

The maintenance of marine machinery has evolved significantly over the last century. Initially, operators relied on **Corrective Maintenance** (Run-to-Failure), which maximizes asset utilization but incurs high failure costs. This was succeeded by **Preventive Maintenance** (Time-Based), grounded in the ”Bathtub Curve” theory of reliability, which assumes failure probability increases with time. However, modern studies indicate that up to 80% of failures are random and not age-related, necessitating a shift to **Predictive Maintenance (PdM)** [1].

PdM is founded on the principle that equipment displays measurable precursors to failure (e.g., heat, vibration, pressure changes) long before functional failure occurs. In the domain of Artificial Intelligence, **Gradient Boosting Decision Trees (GBDT)** have emerged as a leading theoretical approach for tabular sensor data. Algorithms like LightGBM build models sequentially, where each new tree corrects the errors of the previous ones, allowing for the modeling of highly complex, non-linear relationships between engine sensors [2]. Furthermore, the integration of **Game Theory**, specifically Shapley values, provides a theoretical basis for model explainability, allowing complex algorithmic decisions to be decomposed into understandable feature contributions [3].

2.2 Review of Related Literature

Historically, marine engine monitoring has been dominated by Original Equipment Manufacturers (OEMs) such as Wärtsilä and MAN Energy Solutions, who provide systems based on static threshold alarms. While effective for simple faults (e.g., overheating), Jardine et al. [4] note that these systems suffer from high false-positive rates as they cannot account for dynamic operating contexts (e.g., high load vs. idle).

Recent academic efforts have applied Deep Learning techniques, such as Long Short-Term Memory (LSTM) networks, to marine engine data. While these models achieve high accuracy, they operate as ”black boxes.” Lundberg and Lee [3] argue that in safety-critical industries, the lack of interpretability effectively bars deployment, as operators cannot distinguish between a genuine fault and a sensor error.

2.3 Research Gaps

Despite the advancements in both maritime monitoring and machine learning, a critical intersection remains underexplored. Existing solutions generally fall into two categories:

1. **Transparent but Inaccurate:** Threshold-based alarms that are easy to understand but miss complex faults.
2. **Accurate but Opaque:** Advanced AI models that detect faults but cannot explain *why*.

There is a distinct lack of research focusing on **Explainable AI (XAI)** specifically tailored for the marine environment, where a prediction must be accompanied by a root-cause diagnosis (e.g., "Vibration is high due to Bearing Wear") to be actionable. This project seeks to fill this gap by combining high-performance Gradient Boosting with SHAP-based explanations.

2.4 Variables

In the development of the AIMS predictive maintenance system, the following variables have been identified and categorized based on their role in the experimental design:

2.4.1 Independent Variables (Inputs)

These are the manipulated or measured variables that are analyzed to determine the engine's health. In this study, they comprise the real-time readings from the engine sensor network:

- **Mechanical Parameters:** Shaft RPM, Vibration (X, Y, Z axes).

- **Thermal Parameters:** Oil Temperature, Ambient Temperature, Exhaust Temperatures (Cylinders 1-4).
- **Pressure Parameters:** Air Pressure, Oil Pressure, Cylinder Pressures (1-4).
- **Flow Parameters:** Fuel Flow Rate, Engine Load.

2.4.2 Dependent Variable (Output)

This is the variable of primary interest that changes in response to the independent variables.

- **Engine Health Status:** A categorical variable representing the diagnostic result, classified into 8 classes: Normal Operation, Fuel Injection Fault, Cooling System Fault, Turbocharger Fault, Bearing Wear, Lubrication Oil Degradation, Air Intake Restriction, and Vibration Anomaly.

2.4.3 Confounding Variables

These are extraneous variables that correlates with both the dependent variable and the independent variable, potentially obscuring the true relationship.

- **Ambient Conditions:** Variations in ambient temperature and humidity can alter engine thermal baselines (e.g., higher exhaust temps in tropical waters) without indicating a fault.
- **Sea State:** Rough seas may introduce external vibrations that could be mistaken for mechanical bearing wear.

2.4.4 Background Variables

These variables describe the specific context of the study but are not manipulated.

- **Engine Architecture:** The specific design (4-cylinder marine diesel) and age of the engine being simulated.

- **Fuel Quality:** The grade of marine diesel oil used, which affects combustion efficiency.

3 METHODOLOGY

This section outlines the systematic approach employed to develop the AIMS predictive maintenance system. The methodology is designed to ensure that the study produces valid, reliable, and replicable results consistent with the stated specific objectives.

3.1 Research Design

The project adopts a **quantitative, experimental research design**. This strategy is appropriate as it involves the manipulation of computational variables (model hyperparameters, data balancing techniques) to observe their effect on the dependent variable (predictive accuracy). The study is structured into four distinct phases: Data Preparation, Model Development, Explainability Integration, and Prototype Deployment.

3.2 Data Source and Instrumentation

Study Instruments: The primary source of data for this study is a high-fidelity digital simulation of a 4-cylinder marine diesel engine. The simulation generates a comprehensive dataset comprising 10,000 timestamped observations sampled at a frequency of 1 Hz.

The engine system is instrumented with 18 virtual sensors, which serve as the data collection points:

- **Thermocouples:** Measuring oil temperature and cylinder exhaust gas temperatures.
- **Pressure Transducers:** Monitoring intake air pressure, lubrication oil pressure, and peak cylinder combustion pressures.
- **Tri-axial Accelerometers:** Measuring vibration amplitude in the X, Y, and Z axes to detect mechanical imbalances.
- **Flow Meters:** Recording instantaneous fuel flow rates.

3.3 Data Collection Procedure

Data collection involves the simulation of the engine under diverse operating conditions, including idle, maneuvering, and full cruise load. To ensure the model can detect failures, specific fault conditions (e.g., restricted air intake, bearing wear) are injected into the simulation at controlled intervals. The resulting dataset is labeled with one of eight classes (0-7), where Class 0 represents normal operation and Classes 1-7 represent specific fault types.

3.4 Data Processing and Analysis

The data analysis pipeline is implemented using Python and follows these steps:

3.4.1 Preprocessing and Balancing

Raw sensor data is first cleaned to ensure integrity. A critical challenge in maintenance data is **class imbalance**, where normal operation samples vastly outnumber fault samples (approximate ratio of 13:1). To address this, the **Synthetic Minority Over-sampling Technique (SMOTE)** is applied to the training data. SMOTE synthesizes new examples for the minority fault classes by interpolating between existing samples, ensuring the model does not become biased toward predicting "Normal" operation. Additionally, a **StandardScaler** is applied to normalize sensor values to a standard distribution ($\mu = 0, \sigma = 1$).

3.4.2 Model Development

The core predictive engine utilizes the **LightGBM (Light Gradient Boosting Machine)** algorithm. LightGBM is selected for its leaf-wise tree growth strategy, which offers faster training speeds and higher accuracy on tabular data compared to traditional Random Forests. Hyperparameters (such as learning rate η and number of leaves) are optimized using the **Optuna** Bayesian optimization framework to maximize the F1-score.

3.4.3 Gradient Boosting Formulation

The core classification engine of AIMS is built upon the Gradient Boosting Decision Tree (GBDT) framework. Unlike independent ensemble methods such as Random Forests, Gradient Boosting builds an ensemble of weak learners (decision trees) in a sequential manner, where each new tree attempts to correct the errors made by the previous trees.

Mathematically, the model initializes with a constant value $F_0(x)$ that minimizes the loss function:

$$F_0(x) = \arg\min_{\gamma} \sum_{i=1}^{n} L(y_i, \gamma) \tag{1}$$

For a multi-class classification problem with K fault classes (where $K = 8$), the algorithm iterates through M stages. At each stage m $(1 \leq m \leq M)$, the model computes the pseudo-residuals r_{im} for each observation i, which represent the negative gradient of the loss function with respect to the model's current prediction:

$$r_{im} = -\left[\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)}\right]_{F(x)=F_{m-1}(x)} \tag{2}$$

A new decision tree $h_m(x)$ is then fitted to these residuals. The model is updated by adding this new tree, scaled by a learning rate η (set to 0.08 in this study) to prevent overfitting:

$$F_m(x) = F_{m-1}(x) + \eta \sum_{j=1}^{J_m} \gamma_{jm} I(x \in R_{jm}) \tag{3}$$

where J_m is the number of leaves and R_{jm} represents the disjoint regions (leaves) of the tree.

3.4.4 LightGBM Optimization: Leaf-wise Growth

While traditional GBDT implementations (like XGBoost) utilize a level-wise tree growth strategy, this project utilizes LightGBM, which employs a *leaf-wise* growth strategy . In leaf-wise growth, the algorithm splits the leaf with the maximum delta loss (highest error reduction), regardless of the tree's depth.

$$\text{Gain} = \frac{1}{2}\left[\frac{G_L^2}{H_L+\lambda} + \frac{G_R^2}{H_R+\lambda} - \frac{(G_L+G_R)^2}{H_L+H_R+\lambda}\right] - \gamma \quad (4)$$

This approach allows LightGBM to converge faster and achieve higher accuracy on the complex, non-linear sensor data patterns inherent in marine engine faults, although it requires careful tuning of the `max_depth` parameter to prevent overfitting.

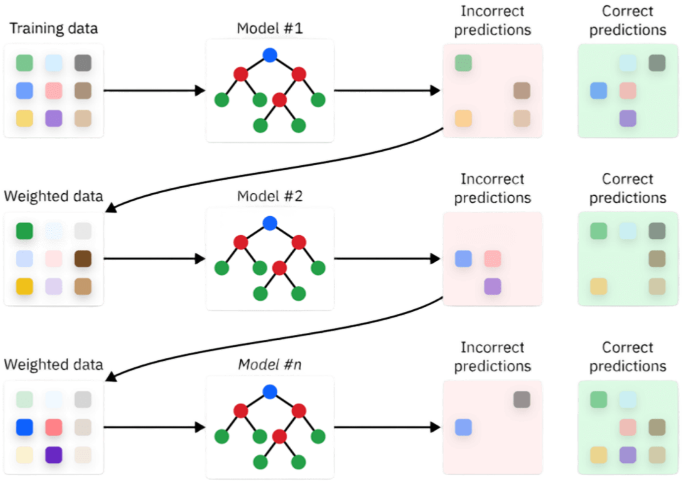


Figure 1: Gradient Boosting

3.4.5 Explainability (SHAP)

To satisfy the requirement for transparency, the trained model is analyzed using **SHapley Additive exPlanations (SHAP)**. For every prediction made by the model, SHAP values are computed to quantify the marginal contribution of each sensor feature. This allows the system to generate natural language explanations, such as ”Prediction: Bearing Wear. Reason: Vibration_X is elevated (+0.14 impact) and Oil Pressure is low (+0.09 impact).”

3.4.6 Prototype Implementation

The final validated model is serialized and deployed as an API using the **FastAPI** framework. A frontend dashboard is developed using **React** to visualize real-time sensor streams, fault alerts, and SHAP explanation plots for the end-user (marine engineer).

4 EXPECTED RESULTS

Based on the experimental design and the capabilities of the selected algorithms, the study anticipates the following outcomes, which are directly aligned with the specific objectives:

- **High Predictive Accuracy:** It is expected that the LightGBM model, trained on the balanced dataset, will achieve an overall classification accuracy of approximately 94% and a macro F1-score exceeding 91%. This performance will demonstrate a significant improvement over traditional threshold-based monitoring systems.

- **Early Fault Detection Capability:** The system is expected to identify subtle degradation patterns (such as initial bearing wear or oil degradation) 24 to 72 hours before they escalate into critical failures, thereby validating the efficacy of the predictive maintenance approach.

- **Transparent Decision Support:** Through the integration of SHAP, the system will output quantifiable explanations for every fault prediction. It is expected that these explanations will align with established marine engineering principles (e.g., correlating high vibration with mechanical looseness), thereby fostering trust among human operators.

- **Operational Viability:** The prototype is expected to demonstrate sub-second inference latency (¡100ms) on standard edge hardware, confirming the feasibility of real-time deployment on onboard vessels.

5 BUDGET

Currently, there are no budget requirements for this project. The only thing we need is Open Source Data

REFERENCES

[1] J. Lee, F. Wu, W. Zhao, M. Ghaffari, L. Liao, and D. Siegel, ”Prognostics and health management design for rotary machinery systems—Reviews, methodology and applications,” *Mechanical Systems and Signal Processing*, vol. 42, no. 1–2, pp. 314–334, 2014.

[2] G. Ke et al., ”LightGBM: A Highly Efficient Gradient Boosting Decision Tree,” in *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[3] S. M. Lundberg and S. I. Lee, ”A Unified Approach to Interpreting Model Predictions,” in *Advances in Neural Information Processing Systems*, 2017.

[4] A. K. Jardine, D. Lin, and D. Banjevic, ”A review on machinery diagnostics and prognostics implementing condition-based maintenance,” *Mechanical Systems and Signal Processing*, vol. 20, no. 7, pp. 1483–1510, 2006.

[5] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, ”SMOTE: Synthetic Minority Over-sampling Technique,” *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, 2002.

[6] D. Woodyard, *Pounder’s Marine Diesel Engines and Gas Turbines*, 9th ed. Oxford: Butterworth-Heinemann, 2009.

[7] International Maritime Organization (IMO), *Fourth IMO GHG Study 2020.* London: IMO, 2020.

APPENDICES

Appendix A: Proposed Algorithm Implementation

The following Python snippet demonstrates the core logic for training the Explainable AI model using LightGBM and SMOTE.

```python
import lightgbm as lgb
from imblearn.over_sampling import SMOTE
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler

# 1. Prepare Data
# Separate 18 sensor features (X) and fault labels (y)
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, stratify=y, random_state=42
)

# 2. Normalization
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)

# 3. Handle Imbalance (SMOTE)
# Synthesize minority fault samples to balance dataset
smote = SMOTE(random_state=42)
X_resampled, y_resampled = smote.fit_resample(X_train_scaled, y_train)

# 4. Train LightGBM Model
model = lgb.LGBMClassifier(
    n_estimators=350,
    learning_rate=0.08,
    num_leaves=55,
    objective='multiclass'
)
model.fit(X_resampled, y_resampled)

# 5. Explainability (SHAP)
import shap
explainer = shap.TreeExplainer(model)
shap_values = explainer.shap_values(X_test)
```