Development of a Smart Autonomous Bilge Management System Using Synthetic Data and Machine Learning Algorithms

Shishir Dutt[1], Dr. Sanjeet Kanungo[2]

[1] *PhD Research Scholar, School of Marine Engineering, Indian Maritime University, East Coast Road, Chennai, India*

[2] *Principal, Tolani Maritime Institute, Professor, Marine Engineering, Talegaon-Chakan Road, Induri, Pune, India*

Corresponding Author: Shishir Dutt, Email: shishirdutt@gmail.com

Abstract

Artificial Intelligence (AI) and Machine Learning (ML) are revolutionizing industrial operations by enhancing efficiency, ensuring regulatory compliance, and promoting sustainability. This paper presents the development of a Smart Autonomous Bilge Management System (SABIMS) for ships, focusing on ML integration through its AI-Driven Decision-Making Module (AIDDM) in conjunction with the highly automated SABIMS Logic Operations Module (SLOM). A synthetic dataset, carefully designed to reflect realistic maritime operational conditions, was utilized to train and evaluate Multi-Class Logistic Regression (MCLR) and Decision Tree (DT) models. The results demonstrated that DT outperformed MCLR across key performance metrics, including precision, recall, F1 score, accuracy, and fit, while maintaining good class balance. By addressing the challenges of bilge water management through predictive and autonomous decision-making, this paper outlines a practical roadmap for achieving enhanced MARPOL compliance. The findings highlight the critical role of synthetic data and robust ML models in advancing sustainable and efficient shipboard systems.

Keywords

SABIMS, SLOM, AIDDM, AI/ML, Artificial Intelligence, Machine Learning, Synthetic Data, Bilge Management, Smart, Autonomous, Marine, MARPOL, Multi-Class Logistic Regression (MCLR), Decision Tree (DT)

1. Introduction

The maritime industry faces increasing scrutiny to comply with international environmental regulations, particularly those outlined by the International Convention for the Prevention of Pollution from Ships (MARPOL), a global treaty aimed at preventing pollution from ships through stringent discharge standards and operational requirements. Despite stringent regulations and penalties for violations, enforcing compliance remains challenging, especially on the high seas[1].Additionally, some shipping companies are motivated by financial gains to circumvent rules, even after repeated convictions, hefty fines, and reputational damages[2].

While advancements in Oily Water Systems (OWS) and Oil Alarm Monitors (OAM) have been made, and the International Maritime Organization (IMO) has implemented regulatory changes, illegal discharges persist. Current OAM devices primarily use automated deterministic controls, which perform well within set parameters but struggle to adapt to changing operational conditions and complex regulatory requirements [3].

Artificial Intelligence (AI) and Machine Learning (ML) are revolutionizing industries by automating processes, enhancing efficiency, and ensuring regulatory compliance. In the maritime sector, these technologies improve fuel efficiency, reduce maintenance costs, optimize voyage planning, and enhance safety and pollution prevention. For example, AI-powered systems are used to predict optimal sailing routes by analyzing weather patterns and ocean currents, reducing fuel consumption and emissions. Additionally, AI-based collision avoidance systems and fully autonomous ships demonstrate the transformative potential of AI/ML-driven innovations [4, 5].

Section 2 of this paper explores the motivations behind illegal bilge water discharges, highlighting economic and operational factors that drive non-compliance with MARPOL Annex I regulations. It presents findings from Port State Control (PSC) reports, showing how deficiencies in compliance persist despite regulatory oversight. Additionally, it discusses the lack of AI/ML applications in bilge water management, contrasting it with AI-driven advancements in water level prediction and environmental monitoring.

Section 3 reviews current bilge water management technologies, including intelligent Oily Water Separator (OWS) systems, and highlights their limitations due to deterministic logic and static threshold-based automation. These existing systems lack the ability to adapt dynamically to operational variations, making them susceptible to non-compliance and inefficiencies in bilge water discharge management.

To address these challenges, this section introduces the Smart Autonomous Bilge Management System (SABIMS)—a transformative AI-driven solution designed to enhance regulatory compliance, operational efficiency, and environmental sustainability. SABIMS integrates the deterministic framework of the SABIMS Logic Operations Module (SLOM) with the predictive and adaptive capabilities of the AI-Driven Decision-Making Module (AIDDM), enabling a transition from rule-based automation to a fully autonomous bilge water management system.

Section 4 discusses the methodology of generation of a synthetic dataset, designed to emulate real-world shipboard conditions, enabling meaningful ML training. It outlines the selection of MCLR as the baseline model, following literature recommendations, and its limitations due to class imbalance and feature dependency. The section further explains the decision to implement Decision Trees, which do not require feature independence and are better suited to handling non-linear relationships in bilge water discharge prediction.

Section 5 evaluates the performance of MCLR and Decision Trees on the synthetic dataset. MCLR struggled with class imbalance despite class weighting and SMOTE, failing to generalize effectively. In contrast, Decision Trees significantly outperformed MCLR, achieving higher accuracy, minimal overfitting, and better class-wise balance, though challenges remained in minority class detection. The findings highlight the need for ensemble learning or balancing techniques as future enhancements.

By addressing the challenges of regulatory compliance and environmental sustainability, this paper presents a practical roadmap for implementing SABIMS, emphasizing a phased integration strategy. The results demonstrate how AI/ML can revolutionize bilge water management, enabling predictive and autonomous decision-making while setting a benchmark for broader AI adoption in shipboard systems. The study further underscores the transformative potential of AI-driven automation in maritime operations, paving the way for future research on enhanced learning models, real-world validation, and full-scale deployment of autonomous bilge management systems.

2. Regulatory Compliance: Motivations Behind Illegal Discharges

2.1. Port State Control Compliance Reports

Port State Control (PSC) involves the inspection of foreign ships in ports to verify compliance with international maritime regulations, ensuring ship safety, crew well-being, and marine environmental protection. Recently published information and reports from various entities and various PSC authorities were reviewed to understand compliance with respect to Annex I.

Authorities are adopting a stricter stance on MARPOL violations, with shipowners, operators, and crew facing severe penalties, including hefty fines and imprisonment for deliberate infractions. The most frequently reported malpractice remains the use of a 'magic pipe,' often accompanied by falsified Oil Record Book (ORB) entries, allowing authorities—particularly in the U.S.—to prosecute shipowners even for violations committed beyond their jurisdiction. Instances of falsified crew statements, concealed bypass equipment, and record manipulation contribute significantly to regulatory actions and financial penalties[1].

A longitudinal study by Mantoju (2021) [6] analyzed PSC reports from various Memoranda of Understanding (MOUs) between 2009 and 2019, revealing that MARPOL Annex I deficiencies accounted for approximately 42% of total reported MARPOL deficiencies, indicating persistent compliance gaps. Figure 1 presents the distribution of deficiencies across various MOUs.

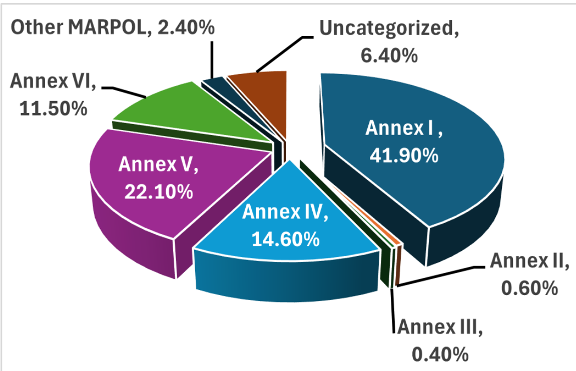


Figure 1: MARPOL Annex-wise PSC Deficiencies from 2009−2019.
Source: Mantoju (2021) [6]

Multiple PSC reports further highlight recurring deficiencies related to Oily Water Separators (OWS), improper ORB entries, and unauthorized discharges. The Tokyo MOU PSC observed an increase in MARPOL Annex I-related deficiencies and detentions due to poor OWS maintenance and fraudulent ORB documentation [7]. Similarly, the Paris MOU 45th Amendment (2023) placed greater emphasis on oil pollution prevention systems, focusing on proper maintenance, operational checks, and enforcement actions against non-compliant vessels [8]. The Bahamas Maritime Authority PSC Report (2023) identified additional issues, including crew unfamiliarity with OWS operations, unauthorized modifications, and incomplete ORB records [9].

In contrast, the USCG PSC Annual Report 2023 recorded a 50% reduction in MARPOL Annex I-related deficiencies compared to 2022. However, persistent violations included unauthorized piping modifications, ineffective maintenance practices, and poorly maintained OWS units, underscoring the need for stricter compliance measures and enhanced oversight [10].

Despite longstanding guidance from Maritime International Secretariat Services, which stresses zero tolerance for MARPOL violations through regular OWS maintenance, tamper-proof devices, audits, crew training, and ORB accuracy [11], deficiencies remain prevalent. The systemic and operational failures highlighted across multiple reports suggest that mere inspections are insufficient. A more sustainable compliance strategy, emphasizing crew training, rigorous equipment maintenance, and fostering a culture of accountability, is essential for long-term regulatory adherence.

2.2. Illegal Discharges: Motivation and Incentives

Illegal discharges continue to be a challenge in the maritime industry, driven largely by economic and operational pressures. As highlighted in our earlier study[3], cost-related incentives, operational inefficiencies, and inadequate facilities play a pivotal role in motivating ship operators to bypass MARPOL Annex I requirements. For example, data from OECD report[2] highlighted how non-compliance offered substantial cost advantages for operators, especially in regions with weak enforcement and limited reception facilities.In addition, the high costs associated with maintenance of related equipment,compliant disposal of oily bilge water and crew training further incentivize these practices.

The economic incentives documented in 2003 remain relevant today, with operators continuing to prioritize cost savings over environmental compliance, particularly in regions with limited monitoring capabilities.Table 1 shows monetary incentives for illegal discharges.

Table 1: Economic Incentives of Non-Compliance[2]

Factor	**Compliance Practice**	**Non-Compliance Advantage**	**Savings in 2003 (USD)**
Waste Management Costs	Proper disposal at port reception facilities	Avoidance of disposal costs	\$50,000–\$400,000 annually
Maintenance Costs	Regular maintenance of bilge water systems	Skipping maintenance saves expenses	\$3,000–\$5,000 annually
Equipment and Training	Regular crew training and equipment upgrades	Avoidance of investment in training	\$10,000–\$100,000 (one-time)
Equipment Replacement	Timely replacement of damaged or outdated equipment	Delaying replacement saves costs	\$10,000 per set (one-time)

3. Existing Systems and Proposed Innovations

3.1. Existing Smart Systems

Modern intelligent Oily Water Separator (OWS) systems, such as CBM-LINK (RWO-VEOLIA) and BlueBox SA (Alfa Laval), incorporate GPS integration, tamper-proof logging, predictive maintenance, and Alarm Monitoring System (AMS) integration for enhanced control and compliance [12, 13]. However, they rely on static thresholds and deterministic logic, making them unable to adapt to dynamic maritime conditions. Despite in-built safeguards, they remain vulnerable to bypass due to the absence of AI-driven proactive anomaly detection.

A prior study by the authors [3] identified these limitations, particularly the lack of AI/ML integration for predictive analytics and real-time decision-making. To address these gaps, the Smart Autonomous Bilge Management System (SABIMS) is proposed, leveraging AI to optimize operations, enhance compliance, and minimize human intervention.

3.2. SABIMS: Advancing Beyond Automation

Overcoming the limitations of current "intelligent" OWS systems requires moving beyond fixed, rule-based operations to adaptive, real-time decision-making. The Smart Autonomous Bilge Management System (SABIMS) bridges this gap by integrating AI and machine learning, enabling risk prediction, dynamic adjustments, and reduced human intervention. This approach enhances bilge water management in evolving maritime conditions.

SABIMS is an advanced autonomous framework integrating real-time monitoring, automated decision-making, and regulatory compliance for operational efficiency and environmental safety. To enhance modularity, it comprises two key modules: the highly automated SABIMS Logic Operations Module (SLOM) and the AI-Driven Decision Module (AIDDM) for intelligent control.

3.2.1 SLOM: SABIMS Logic Operations Module

The Logic Operations flow diagram of SLOM is given in Figure 2

3.2.1.1 Core Functional Components Following is a brief description of the core components of SLOM:

- **Sensors and Monitoring Devices:** Bilge Well Sensors (BWA, BWFP, BWFS), Tank Level Sensors (BTL, STL, APTL), Trim and List Monitors (TM, LM), Oil Alarm Monitor (OAM), Geo-Fencing GPS Monitor (GPSM), En-Route Monitor (ERM)

- **Pumps and Valves:** Engine Room Bilge Pump (ERBP), Fire & G.S. Pump (FGSP), Bilge & G.S. Pump (BGSP), Oily Water Separator (OWS), Valves- Key valves include BWASV (aft bilge well suction), BTSV (bilge tank suction), and FGSPODV (FGSP overboard discharge)

- **Alerts and Notifications:** Real-Time Alerts indicate abnormal conditions, suction failures, or critical overflow scenarios while Emergency Notifications Trigger actions when all bilge wells and tanks reach critical levels.

  For more description of SABIMS components refer to Appendix A.

3.2.1.2 Operational Logic Flow

- **Monitoring and Initial Calculations:** Monitors Elapsed time (t1-t0 $>$ 0), Bilge well level changes (BWAL1 - BWAL0 $>$ 0), calculates corrected volumes (BWAV) using trim (TM) and list monitors (LM).
- **Decision Nodes and Conditions:** Rate of Rise (ROR) is categorized as 0,1,2,3 and $>$3, depicting Normal, slightly abnormal, Abnormal, Alarming and Emergency con-

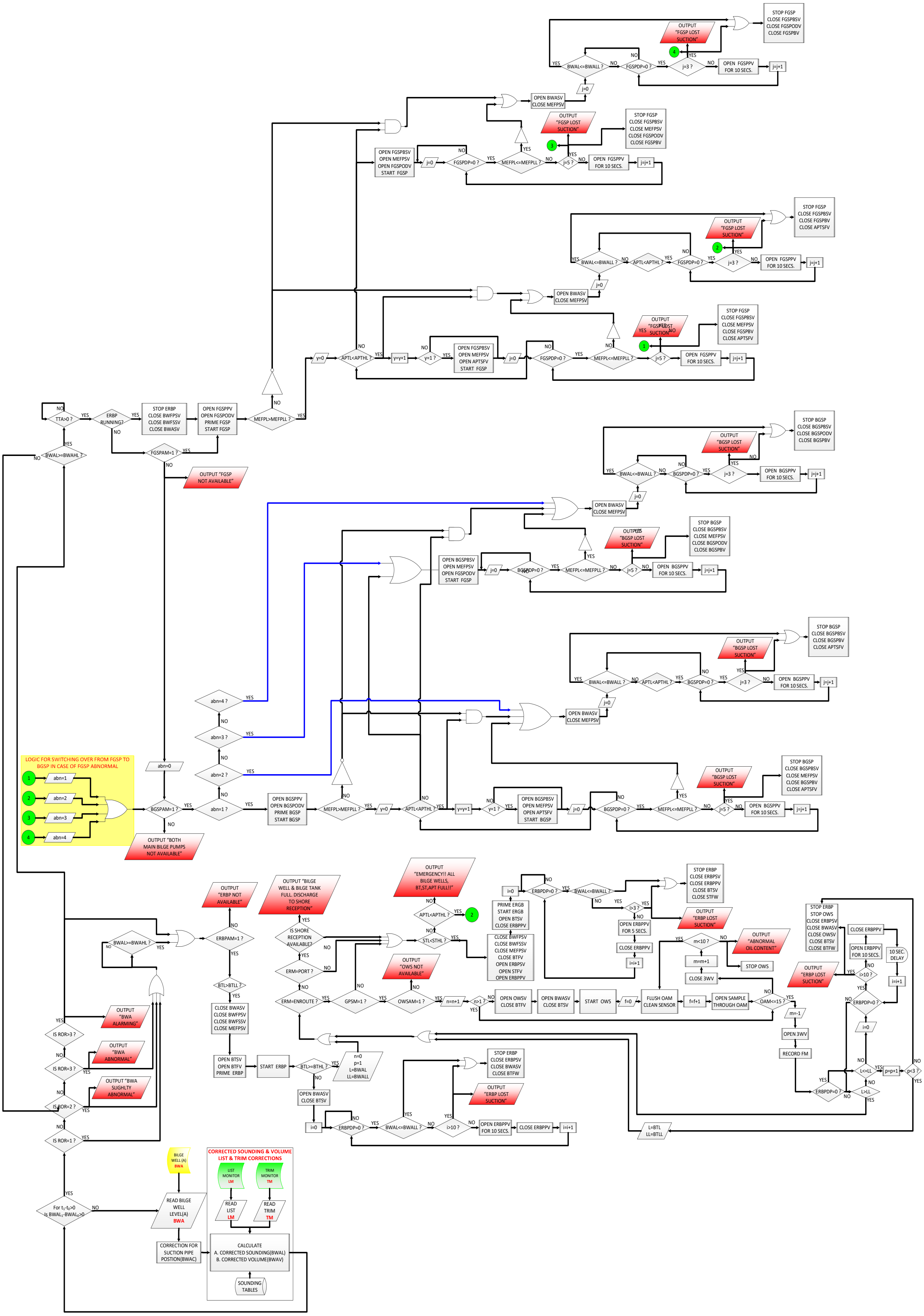


Figure 2: SABIMS Logic Diagram

ditions respectively. High level thresholds (BWAHL,BTHL) start pumps and trigger alarms. Low Level threshold(BWALL, BTLL) stops pumps.

- **Pump Operation:** The Engine Room Bilge Pump (ERBP) activates at the threshold, retries suction up to 10 times, and stops with fail-safes. If ERBP fails, the Fire & G.S. Pump (FGSP) and Bilge & G.S. Pump (BGSP) activate sequentially, with retry mechanisms handling suction loss (up to 5 retries), and bilge water is redirected to alternate tanks in emergencies.
- **Oily Water Separator (OWS) Operations:** OWS activates when oil content ≤ 15 PPM (OAM), the ship is > 12 NM from shore (GPSM), and status is EN-ROUTE (ERM); system directs bilge water to OWS for separation, flushes sensors on abnormal readings, and shuts down with alerts if oil >15 PPM.
- **Emergency and Overflow Management:** Redirects excess bilge water to sludge (STL) or aft peak tanks (APTL) and activates emergency bilge suction valves (EBSV) for bilge well overflows; triggers alerts when tanks are full and verifies shore reception availability before action.
- **System safety Mechanism:** Pumps retry suction for a preset time (ERBP: 10 retries, FGSP/BGSP: 5 retries) before triggering "Suction Lost" alerts; redirects bilge water to alternative tanks during failures and, while in port, discharges to shore reception if the bilge tank is full.

3.2.1.3 Advantages

- Enhanced Environmental Compliance
- Enhanced Operational Efficiency
- Safety and Redundancy
- Proactive Alerts and Fallbacks

The SABIMS Logic Operations Module (SLOM) automates bilge water management through advanced monitoring, decision-making, and fallback mechanisms, ensuring compliance and reliability. Integrating the AI-Driven Decision Module (AIDDM) enhances SLOM's autonomy, enabling proactive decision-making, adaptive self-learning, and enhanced compliance in dynamic operational conditions.

3.2.2 Artificial Intelligence Driven Decision Module (AIDDM)

The AI-Driven Decision Module (AIDDM) is the transformative core of the Smart Autonomous Bilge Management System (SABIMS), transforming it from rule-based automation to full autonomy. By integrating machine learning (ML), real-time data, and feedback loops, AIDDM enables dynamic adaptability, predictive capabilities, and operational autonomy.

3.2.2.1 Dynamic Decision-Making AIDDM adapts SABIMS to operational conditions by analyzing live sensor inputs like bilge levels, tank capacities, pump statuses, and environmental data. It predicts issues, ensures regulatory compliance, and dynamically adjusts pump thresholds and discharge schedules to optimize performance while continuously retraining with synthetic and historical data. Figure 3 shows the operational flow process of SABIMS including AIDDM training using synthetic/historical data, while the live operational data is utilized for its continuous retraining and improvement.

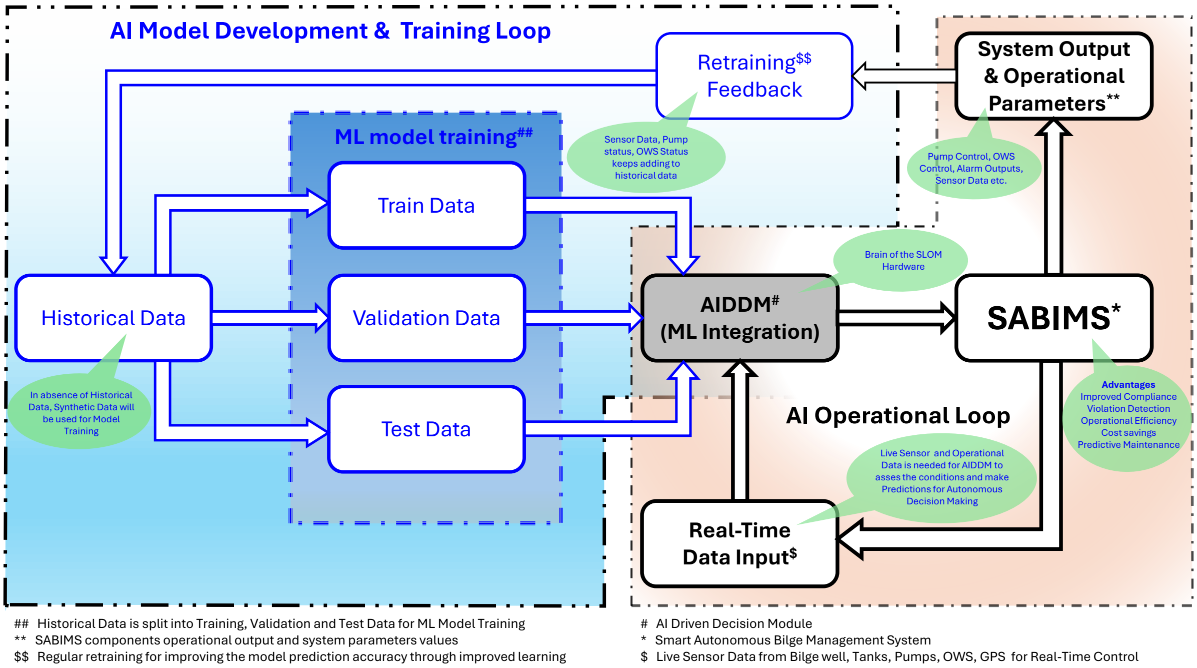


Figure 3: SABIMS Training and Operations
Source: Dutt et al.[3]

3.2.2.2 Prediction and Pattern Recognition AIDDM employs ML models to predict system behavior and recognize patterns in real-time and historical data:

- **Predictive Maintenance:** Anticipates failures in pumps, OWS, and other components using historical trends and anomaly detection, Enables proactive interventions, reducing downtime and emergency repairs.
- **Anomaly Detection:** Flags irregular patterns, such as spikes in oil content or unexpected inflows, and triggers corrective actions.
- **Operational Optimization:** Recommends resource-efficient pumping schedules, discharge timings, and operational adjustments to enhance system efficiency.

3.2.2.3 Autonomous Adjustments AIDDM autonomously refines SABIMS operations, enhancing adaptability and efficiency:

- **Threshold Calibration:** Adjusts bilge tank and well high/low-level thresholds based on dynamic conditions, such as rolling and pitching intensities.
- **Pump Optimization:** Balances suction rates and activation intervals to manage bilge water effectively and minimize equipment wear.
- **OWS Tuning:** Monitors OWS performance, triggering sensor calibration and cleaning cycles as needed to maintain separation efficiency.

3.2.2.4 Learning Capabilities AIDDM's self-learning features enable continuous improvement through feedback integration:

- **Model Refinement:** Updates predictive models using operational data, identifying new patterns and improving accuracy.
- **Adaptive Feedback:** Incorporates crew inputs and recurring scenarios into decision-making algorithms for enhanced autonomy.
- **Feedback Loop:** Real-time operational data enriches the historical database, enabling regular retraining and more precise predictions.

3.2.2.5 Key Benefits of AIDDM

- **Enhanced Compliance**
- **Improved Efficiency**
- **Enhanced Operational Safety**
- **Scalability**

- **Cost Savings**

The successful implementation of AIDDM within SABIMS depends on a robust training process for the ML models. Leveraging historical and synthetic data is critical to ensure AIDDM can accurately analyze real-time conditions, predict outcomes, and make autonomous decisions. As the system learns and improves over time, it transforms bilge water management into a self-sustaining, fully autonomous process, meeting both operational and regulatory challenges.

4. Methodology

This section details the approach adopted for developing a Smart Autonomous Bilge Management System. It encompasses the creation of synthetic datasets and the application of machine learning algorithms, presenting the processes and techniques employed to ensure effective implementation and evaluation of the proposed system.

4.1. Scope of Study

This research focuses on a subset of SABIMS, specifically predicting pump operations (ERBP, FGSP, BGSP) based on bilge water inflow, water levels, and rate of rise (ROR). While complete autonomy requires extensive synthetic data, simulation, and broader integration (e.g., ship's list, trim, OWS operations), these aspects are beyond scope of the study. The study provides a proof-of-concept for AI-driven bilge management, establishing a scalable foundation for future advancements in maritime environmental compliance.

4.2. Synthetic Data Generation

4.2.1 Introduction and Importance in Maritime Context

The implementation of Machine Learning (ML) for SABIMS requires diverse datasets, but real-time ship data is scarce due to limited sensors and lack of time-series records. Synthetic data provides a vital solution, bridging the gap between unavailable real-world datasets and the need for operationally relevant data. Widely used in healthcare, environmental monitoring, and maritime research, it addresses data scarcity, privacy concerns, and operational constraints [14, 15]. High-quality synthetic datasets are crucial for training AI/ML models while ensuring operational and statistical integrity.

4.2.2 Parameter Selection and Realism

The synthetic data generation process began by defining three critical input features and one output variable relevant to bilge water management:

4.2.2.1 Input Features

- Bilge water inflow rate (xr) (cc/min)
- Bilge well level (y) (cc)
- Bilge tank level (btl) (cc)

These input features represent key operational variables that influence bilge water management, aligning with real-world conditions and shipboard scenarios.

4.2.2.2 Output Variable

The output variable, pump status (pp), is categorized into four distinct classes based on operational scenarios. The meaning of each class is summarized in Table2.

Table 2: Meaning of Output Classes in Pump Status (pp)

Class (pp)	Description	Pump Status
Class 0	No pump running	All Pumps Off
Class 1	Engine Room Bilge Pump (ERBP) running	ERBP Run
Class 2	ERBP running but unable to lower water level; FGSP also starts (ERBP + FGSP)	(ERBP + FGSP) Run
Class 3	ERBP + FGSP running; higher water inflow; water reaches 15 cm above tank top (MSWP starts)	(ERBP + FGSP + MSWP) Run

The parameters and class definitions were tailored using capacity plans and the General Arrangement of a 53,000 DWT Handymax bulk carrier, along with Class rules and MARPOL regulations. This ensures that the dataset is operationally valid and aligned with real-world maritime conditions. The importance of domain-specific parameterization, as emphasized by Flores-Alsina et al. (2014) [16], was integral in creating realistic and reliable synthetic data.

4.2.3 Rate of Rise (ROR) Classification and Data Variability

4.2.3.1 Introduction to Rate of Rise (ROR)

The Rate of Rise (ROR), defined as bilge water inflow over time, is crucial for monitoring bilge well and tank levels. For the case ship, ROR classification considers operational scenarios, pump capacities, bilge well volume, and a 218.3 m^2 tank top area, ensuring synthetic data reflects real constraints and flooding scenarios. It enables modeling of normal, abnormal, and emergency conditions, essential for training robust ML models.

4.2.3.2 Factors Influencing ROR Classification: ROR is determined by pump capacities (ERBP: 5 m^3/hr, FGSP/BGSP: 200 m^3/hr, MSWP: 640 m^3/hr), bilge well volume (2 m^3), tank top area (218.3 m^2, flooding threshold 32,745 liters), and emergency timing (3,275 L/min over 10 min). This classification follows Pezoulas et al. (2024) [**?**], who emphasize realistic scenario modeling in synthetic datasets to enhance ML model robustness and generalizability.

4.2.3.3 ROR Classification

ROR classifications are based on the case ship's operational constraints, reflecting pump capacities and flooding thresholds are given in Table 3

Table 3: Rate-of-Rise (ROR) Classification with Pump Activation Scenarios

ROR Scale	Category	Max ROR (Liters/Day)	Pump(s) Required	Triggering Condition
—	Normal	0–600	ERBP	Routine operations
1	Normal	1,200	ERBP	Routine operations
2	Slightly Abnormal	2,400	ERBP	Slight increase in inflow
3	Abnormal	12,000	ERBP	High-level alarm triggered
4	Highly Abnormal	120,000	ERBP + FGSP/BGSP	Overflow of bilge well
5	Critical	6,000,000	ERBP + FGSP/BGSP	Continuous inflow exceeds pump capacity
6	Extreme Emergency	6,000,000	ERBP +FGSP/BGSP + MSWP	Water reaches 15 cm above tank top (32,745 liters)

4.2.3.4 Sub-Range Division for Data Generation

To ensure diverse and realistic synthetic data, ROR ranges were divided into sub-ranges with controlled increments as shown in Table4

Table 4: ROR Sub-Range Division for Data Generation

ROR Class	Range (Liters/Day)	Sub-Range Increment (Liters/Day)	Number of Sub-Ranges	Data Points per Sub-Range
Low	0–1,200	100	12	200
Moderate	1,201–12,000	500	22	200
High	12,001–120,000	5,000	22	200
Very High	120,001–600,000	25,000	19	200
Critical	600,001–6,000,000	500,000	11	200
Extreme Emergency	6,000,001–6,000,000	500,000	11	200

The structured variability approach aligns with techniques discussed by Seongbin An et al. (2023)[17], ensuring operationally diverse yet statistically balanced datasets for robust ML model training.

4.2.3.5 Insights on ROR and Synthetic Data Design:

Pump activation follows a hierarchy where ERBP manages routine inflows, FGSP/BGSP handles overflow, and MSWP activates in extreme emergencies (>15 cm flooding). Controlled variability ensures finer resolution in normal ROR levels while maintaining realistic representation of rare emergencies. This approach aligns with best practices in synthetic data generation, balancing variability and realism (McDuff et al., 2021) and incorporating statistical validation techniques like Kullback-Leibler divergence (Goncalves et al., 2020).

4.2.3.6 Time-Series Structuring for Realism

Synthetic data was structured in 30-second intervals, capturing gradual variations for normal operations and sudden spikes for rare events like pipe bursts. While temporally structured, time values were excluded as input features to ensure compatibility with static ML models like MCLR and DT, aligning with Choi et al. (2021) on balancing realism and model applicability.

4.2.3.7 Data Generation and Validation:
Data points were generated using the RANDBETWEEN function in Microsoft Excel, a resource-efficient method aligning with Appenzeller et al. (2022)[18]. Validation involved ORB data comparison, expert reviews, and statistical analysis to ensure realism, following best practices from Goncalves et al. (2020) [19] and Rudenko et al. (2023) [20] on aligning synthetic data with domain-specific realities.

4.2.3.8 Train-Test Splitting for Balanced Representation:
The dataset was manually split into 80% training and 20% testing to ensure balanced representation across all ROR sub-classes, mitigating distribution imbalances common in automated methods. Singh et al. (2024) [15] emphasize that such balance is crucial for unbiased and reliable ML model evaluation.

4.2.4 Advantages of Synthetic Data Generation

- Privacy Protection [18, 21]
- Addressing Data Scarcity [20, 22]
- Cost and Time Efficiency [16, 23]
- Controlled Variability [15, 24]
- Flexibility and Customization [22, 25]
- Improved Model Training and Robustness [14, 26]
- Bias Mitigation [15, 24]
- Enables Scenario Simulations [16, 27]
- Enhanced Accessibility [18, 28]
- Enables Testing of Rare or Extreme Conditions [20, 23]
- Scalability [14, 25]

4.2.5 Summary

This methodology offers a practical, domain-specific approach to synthetic data generation for SABIMS. By integrating structured variability and expert validation, it ensures reliable ML training while addressing data scarcity and diversity in specialized applications.

4.3. Data Pre-processing and Validation

To ensure the synthetic dataset was suitable for ML training in bilge water management, key pre-processing and validation steps were applied.

- **Feature Selection** was based on statistical methods (ANOVA, p-value analysis) and domain expertise to ensure relevance.
- **Outlier Management** was handled at the data generation stage, eliminating post-processing corrections.
- **Class Balance** was maintained by ensuring higher densities in normal conditions and progressively sparser data in abnormal and emergency scenarios. Class 0, representing super normal conditions, has the highest frequency, followed by Class 1 and Class 2, which represent higher bilge inflow rates. Class 3, an emergency condition where bilge water exceeds 15 cm above the tank top, has the lowest frequency, ensuring a realistic dataset distribution (Table 5).

Table 5: Class Distribution in the Synthetic Dataset

Class	Training Samples	Testing Samples	Test/Train Ratio
Class 0	18,654	4,675	0.457
Class 1	11,291	2,828	0.277
Class 2	8,093	2,035	0.198
Class 3	2,756	702	0.068

- **Scaling and Regularization** were applied for feature standardization and to mitigate overfitting.
- **Statistical Validation** used feature importance analysis and ANOVA to confirm meaningful contributions across ML models like MCLR and Decision Trees.
- **Iterative Validation** was carried out, incorporating feature refinement based on model feedback, adjusting distributions to represent real-world scenarios, and ensuring rare events (Class 3) were adequately represented. Class 3 occurrences, extremely rare in actual ship operations, were enhanced to 6-7% for effective model learning. The dataset was optimized to 50k samples for computational feasibility, balancing realism and practical constraints. Overlapping data ranges and scarcity in feature space were systematically addressed to maintain diversity and prevent artifacts that could impair ML training.

4.4. Algorithm Selection: ML Models Overview and Rationale

4.4.1 Key Considerations in ML Algorithm Selection

Selecting an appropriate ML model is crucial for reliable classification. The choice is guided by dataset characteristics, interpretability, computational efficiency, and handling of class imbalance.

- **Dataset Characteristics**: A structured, multi-class classification problem with three numeric features and four output classes.
- **Interpretability**: Decision Trees offer transparency, while ANNs lack explainability, making them less suitable.
- **Scalability**: The dataset (50k samples) requires efficient models like MCLR and DT over computationally intensive SVMs and ANNs.
- **Class Imbalance**: Decision Trees inherently handle imbalance, whereas MCLR requires balancing techniques.

4.4.2 Overview of Machine Learning Models for Multi-Class Classification

Potential models include:

- **Ensemble Methods**: Robust but require extensive tuning [29].
- **SVMs**: Effective but computationally expensive [30].
- **KNN, Naïve Bayes**: Simple but inefficient for large datasets [29].
- **ANNs**: High flexibility but lacks interpretability and requires large datasets [29].

This study initially focuses on **MCLR and DT** as baseline models. Their performance will guide future exploration of advanced methods.

4.4.3 Suitability of Multinomial Logistic Regression (MCLR)

- **Advantages**: Probabilistic classification, efficient training, interpretability, and compatibility with numeric datasets [29–32].
- **Limitations**: Assumes linearity, requires class balancing, and is sensitive to multicollinearity and feature scaling [33–35].
- **Rationale**: Chosen for computational efficiency, interpretability, and as a baseline for

probabilistic classification.

4.4.4 Suitability of Decision Trees (DT)

- **Advantages**: Captures non-linear patterns, ranks feature importance, robust to class imbalance, and requires no feature scaling [29, 30].
- **Limitations**: Risk of overfitting, computational complexity for deep trees, sensitivity to data variations, and weaker performance than ensembles [29].
- **Rationale**: Chosen for its complementary approach to MCLR and ability to capture non-linearity while serving as a benchmark for ensemble methods.

Decision Trees and MCLR establish a foundation for evaluating classification trends. Future work will explore ensembles like Random Forest and Gradient Boosting to assess performance gains.

4.5. Theoretical Foundations

4.5.1 Detailed Overview of Multinomial Logistic Regression (MCLR)

4.5.1.1 Introduction to Logistic Regression

Logistic regression is a supervised learning algorithm commonly used for binary classification problems. It models the relationship between input features (x) and the probability of a binary outcome ($y \in \{0, 1\}$) [**?**].

The **sigmoid function** forms the mathematical basis of logistic regression:

$$\text{Sigmoid}(z) = \frac{1}{1 + e^{-z}}$$

where $z = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n$.

Logistic regression uses a **decision boundary (threshold)**, typically 0.5, for classification. As shown in Figure 4, if the sigmoid output is greater than or equal to 0.5, the prediction is $y = 1$; otherwise, $y = 0$ [29, 36]. The threshold allows the model to distinguish between classes effectively and aligns with the probabilistic nature of the sigmoid function.

4.5.1.2 Extension to Multiclass Problems

- MCLR extends binary logistic regression to handle multiclass classification tasks. It

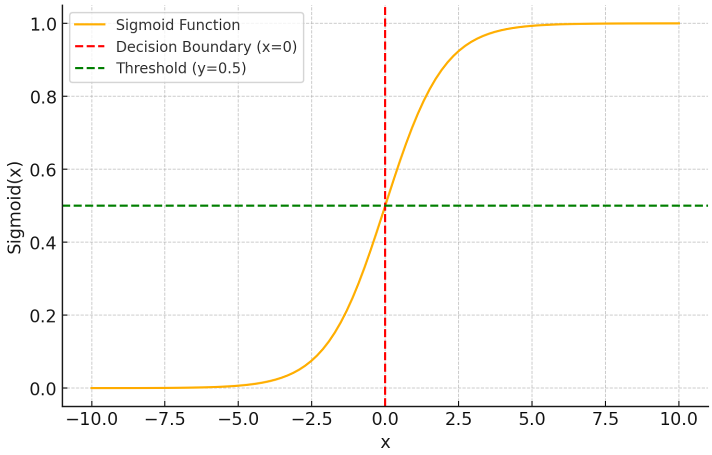


Figure 4: Sigmoid Function (Threshold: 0.5).

predicts one of C possible classes ($y \in \{1, 2, \ldots, C\}$) using the **softmax function**. A common approach to solving the multiclass problem is the **One-vs-Rest (OvR)** strategy, where separate binary classifiers are trained for each class against all other classes, allowing for efficient multiclass classification [29, 34, 37–39].

$$P(y = c \mid x) = \frac{e^{\beta_c^\top x}}{\sum_{j=1}^{C} e^{\beta_j^\top x}}, \quad c = 1, 2, \ldots, C$$

- The predicted class is the one with the highest probability:

$$\hat{y} = \text{argmax}_c \, P(y = c \mid x)$$

- MCLR optimizes the **cross-entropy loss function** to estimate coefficients [33]:

$$L = -\sum_{i=1}^{N} \sum_{c=1}^{C} y_{ic} \log P(y = c \mid x_i)$$

4.5.2 Detailed Overview of Decision Trees (DT)

4.5.2.1 Introduction to Decision Trees

Decision Trees are supervised learning models for classification and regression, recursively splitting data based on feature thresholds to form a hierarchical structure [29]. The algorithm selects features that maximize information gain or minimize impurity (e.g., Gini Index, Entropy, or Mean Squared Error). Splitting continues until a stopping criterion (e.g., maximum depth, minimum samples per split) is met. This approach enables Decision Trees to model complex decision boundaries, capture non-linear relationships, and provide interpretable decision-making.

4.5.2.2 Structure:
A Decision Tree consists of a **Root Node** (entire dataset), **Internal Nodes** (feature-based splits), and **Leaf Nodes** (predicted class labels or output values).

4.5.2.3 Splitting Criteria

- Decision Trees split data using measures such as:
  - **Gini Impurity** [?]:
  $$G = 1 - \sum_{i=1}^{C} p_i^2$$
  - **Entropy** [?]:
  $$H = -\sum_{i=1}^{C} p_i \log_2(p_i)$$
  - **Information Gain (IG)** [?]:
  $$IG = H(\text{parent}) - \sum_j \frac{N_j}{N} H(\text{child}_j)$$

4.6. Training Methodology

4.6.1 Multinomial Logistic Regression (MCLR)

4.6.1.1 Feature Standardization:
To ensure consistent scaling, features were standardized using the `StandardScaler` function:

```
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
```

4.6.1.2 Handling Class Imbalance:
Class imbalance was addressed using the Synthetic Minority Oversampling Technique (SMOTE):

```
smote = SMOTE(sampling_strategy={1:14000,2:10000},random_state=0)
X_train_resampled,y_train_resampled = smote.fit_resample(X_train_scaled
    , y_train)
```

4.6.1.3 Class Weights:
Class weights were assigned to adjust for class imbalance:

```
class_weight_dict = {0: 1.0, 1: 2.5, 2: 3.0, 3: 1.5}
```

4.6.1.4 Regularization:
Regularization was incorporated to prevent overfitting by tuning the penalty term:

```
model = LogisticRegression(penalty='l2', class_weight=class_weight_dict
    , solver='liblinear', max_iter=200)
```

4.6.1.5 Hyperparameter Tuning and Cross-Validation:
Optimal hyperparameters were determined using cross-validation with `RandomizedSearchCV`:

```
search = RandomizedSearchCV(model, param_grid, cv=5)
search.fit(X_train_resampled, y_train_resampled)
```

4.6.1.6 Dataset Usage:
The training and testing datasets were used as separate files, ensuring no data leakage. The dataset was loaded as follows:

```
train_data = pd.read_csv('train_data.csv')
test_data = pd.read_csv('test_data.csv')
```

4.6.2 Decision Trees (DT)

4.6.2.1 Model Initialization:
The Decision Tree model was initialized with impurity metrics and depth constraints:

```
model = DecisionTreeClassifier(criterion='gini', max_depth=10,
    min_samples_split=10, min_samples_leaf=2, random_state=0)
```

4.6.2.2 Hyperparameter Tuning and Cross-Validation:
Hyperparameters such as impurity criteria, tree depth, and minimum sample requirements were optimized using cross-validation:

```
param_grid = {'criterion':['gini','entropy'],'max_depth':[2,5,10],
              'min_samples_split':[2,5,10],'min_samples_leaf':[1,2,4]}
cv = StratifiedKFold(n_splits=5, shuffle=True, random_state=0)
search = GridSearchCV(model, param_grid, cv=cv).fit(X_train, y_train)
```

4.6.2.3 Handling Class Imbalance:
Class imbalance was addressed inherently by the Decision Tree's splitting strategy, which optimizes class separation. Additionally, weight balancing was considered to enhance fairness:

```
model = DecisionTreeClassifier(class_weight='balanced',criterion='gini
    ', max_depth=10,min_samples_split=10,min_samples_leaf=2,
    random_state=0)
```

4.6.2.4 Dataset Usage: The training and testing datasets were used as separate files, ensuring no data leakage. The dataset was loaded as follows:

```
train_data = pd.read_csv('train_data.csv')
test_data = pd.read_csv('test_data.csv')
```

4.6.2.5 Best Model Selection: The best model was selected based on cross-validation performance:

```
best_model = search.best_estimator_
```

5. Results and Discussion

5.1. Evaluation Metrics

5.1.1 Confusion Matrix

The confusion matrix provides an overview of the model's predictions compared to the actual labels, detailing true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN). It serves as the foundation for calculating other evaluation metrics.

Table 6: Confusion Matrix Overview

Actual \Predicted	Predicted Positive	Predicted Negative	Explanation
Actual Positive	True Positive (TP)	False Negative (FN)	Correct or missed positives detected
Actual Negative	False Positive (FP)	True Negative (TN)	Correct or false negatives detected

5.1.2 Derived Metrics

Metrics derived from the confusion matrix include precision, recall, F1-score, and accuracy. These metrics are defined in the Table7

5.1.3 Classification Reports

The classification report provides detailed performance metrics for each class, including precision, recall, F1-score, Macro Averaged F1-Score, weighted F1-Score, Overall accuracy and support. Some of the additional metrics used in the classification report are described here:

Table 7: Description of Performance Metrics

Metric	Description
$\text{Precision} = \dfrac{TP}{TP + FP}$	Precision should be ideally 1 (high) for a good classifier. Precision can become 1 when FP is zero. As FP increases, the Precision value decreases.
$\text{Recall} = \dfrac{TP}{TP + FN}$	Recall should be ideally 1 (high) for a good classifier. Recall can become 1 when FN is zero. As FN increases, recall value decreases.
$\text{Accuracy} = \dfrac{TP + TN}{TP + TN + FP + FN}$	Accuracy represents the number of correctly classified data instances over the total number of data instances.
$\text{F1-Score} = 2 \times \dfrac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$	F1 Score becomes 1 when both Precision and Recall are high. F1 Score is the Harmonic Mean of Precision and Recall and is a better measure than Accuracy.

5.1.3.1 Support: The total number of actual instances present in the dataset for each class.

5.1.3.2 Macro Averaged F1-Score: It is the unweighted average of the F1 scores across all classes. It treats each class equally, regardless of how many instances exist for that class.It can be misleading when class imbalance exists.

5.1.3.3 Weighted F1-Score: It is the average F1-score across all classes, but it assigns weights based on the number of actual instances (support) in each class. It accounts for class imbalance by adjusting weights according to the support of each class.

5.1.4 Train-Test Accuracy: Assessing Model Fit and Generalization

Model fit analysis examines the difference between training and testing accuracy to evaluate the model's generalization ability. A smaller difference indicates better generalization, while a large gap suggests poor generalization. A model that performs exceptionally well on training data (high training accuracy) but poorly on test data suffers from poor generalization. In general, cross-validation helps improve generalization by optimizing model fitting.

5.2. Model Performance Evaluation

5.2.1 MCLR Model Training and Testing Results

Figure 5 illustrates the performance of the MCLR model on the train and test datasets. Figure 5(a) and Figure 5(b) show the confusion matrices for train and test datasets, respectively.

5.2.1.1 Train Metrics by Class (Figure 5(c)):

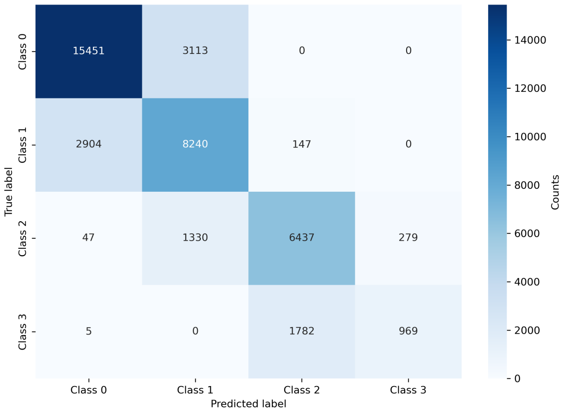
- **Class 0:** Precision = **0.84**, Recall = **0.83**, F1-Score = **0.84** – Model performed consistently well.
- **Class 1:** Precision = **0.65**, Recall = **0.73**, F1-Score = **0.69** – Moderate performance; recall higher than precision.
- **Class 2:** Precision = **0.77**, Recall = **0.80**, F1-Score = **0.78** – Model performed reasonably well with slight imbalance.
- **Class 3:** Precision = **0.35**, Recall = **0.48**, F1-Score = **0.41** – Poor performance due to significant misclassification.

Inference: Model trained well for Classes 0 and 2 but struggled with Class 3 and moderately with Class 1. Despite scaling, regularization, and hyperparameter tuning, training performance was suboptimal for some classes.

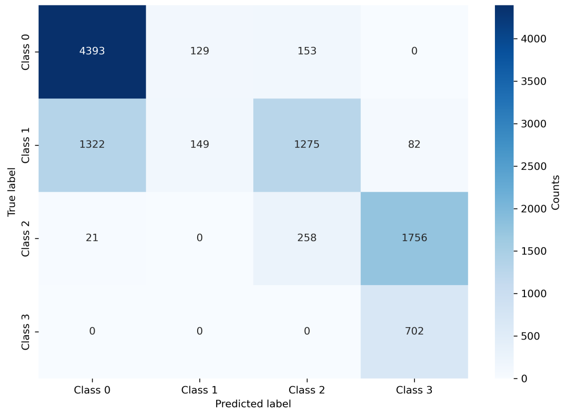
5.2.1.2 Test Metrics by Class (Figure 5(d)):

- **Class 0:** Precision= **0.77**, Recall= **0.94**, F1-Score= **0.84** – Good performance but recall significantly higher than precision, indicating the number of FPs are high.
- **Class 1:** Precision= **0.05**, Recall= **0.10**, F1-Score= **0.07** – Extremely poor performance, indicating model failure for this class.
- **Class 2:** Precision= **0.15**, Recall= **0.13**, F1-Score= **0.14** – Extremely poor performance, indicating model failure for this class.
- **Class 3:** Precision= **0.28**, Recall= **1.00**, F1-Score= **0.43** – While TPs are all high, FNs are zero, but at the same time FPs are very high.

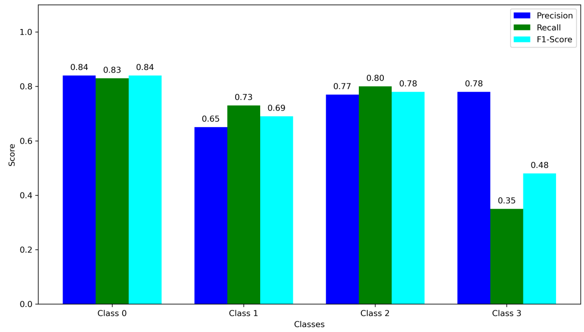
Inference: Model performed well for Class 0 but failed for Classes 1, 2, and 3, with severe under-performance and imbalances in predictions.



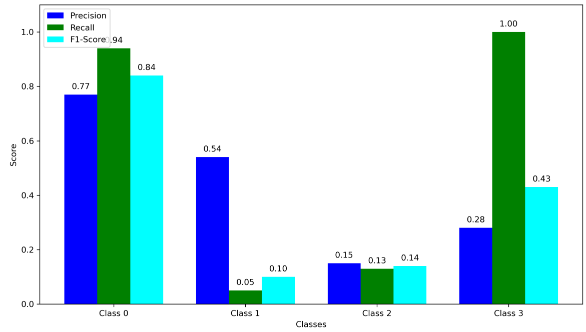
(a) Train Confusion Matrix



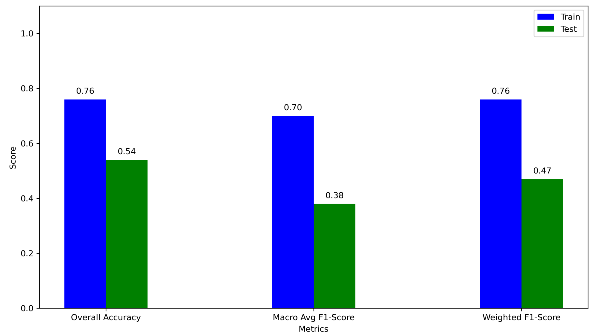
(b) Test Confusion Matrix



(c) Train Metrics by Class



(d) Test Metrics by Class



(e) Macro F1 and Overall Accuracy

Figure 5: Class-Wise & Overall Performance Evaluation for MCLR

5.2.1.3 Overall Metrics (Figure 5(e)):

- **Train Data:** Overall Acc.= **0.80**, Macro F1-Score= **0.70**, Weighted F1-Score= **0.76**.
- **Test Data:** Overall Acc.= **0.54**, Macro F1-Score= **0.38**, Weighted F1-Score= **0.47**.

Inference: Overall metrics indicate that the model performed above average on training data, the test data results were poor, indicating overfitting and inability to generalize effectively.

5.2.1.4 Insights and Way Forward:
The MCLR model demonstrates severe performance inconsistencies across different classes. While it performs reasonably well for Class 0 and Class 3, it fails catastrophically for Class 1 and Class 2, as evident from the various metrics. Given these limitations, an improved approach was attempted by adjusting class weights and applying SMOTE to mitigate class imbalance and enhance model generalization.

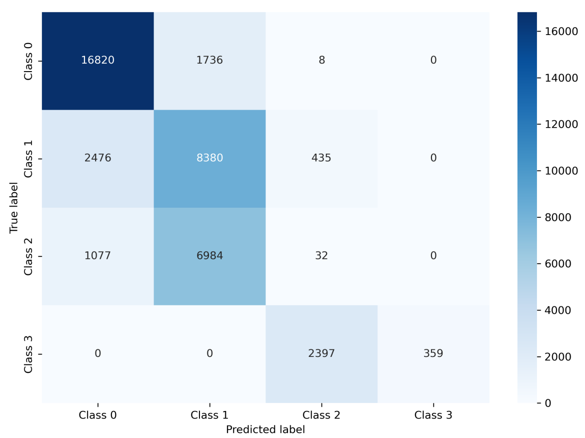
5.2.2 MCLR with Weight adjustments and SMOTE

In alignment with standard practices from the reviewed literature, class weighting and SMOTE (Synthetic Minority Oversampling Technique) were applied to the MCLR model to address class imbalance.

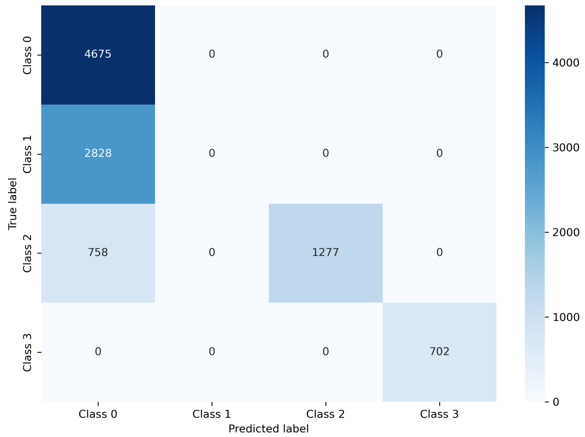
5.2.2.1 Model Performance:
Modified MCLR model performance results are shown in Figure6. Applied methods failed to improve performance and instead worsened generalization. The model completely failed to predict Class 1 in the test data and performed poorly for Class 3. This degradation may have been caused by the inherent class imbalance (Table 5) and dataset complexities, which amplified noise and further degraded results.

5.2.2.2 Probable Reasons of MCLR failure:
The combined challenges of class imbalance and feature interdependence rendered MCLR unsuitable for this dataset. Despite applying all recommended techniques, including scaling, regularization, hyperparameter tuning, cross-validation, weight adjustments, and SMOTE, the model failed to generalize effectively and performed inconsistently for minority classes (Class 3 being the smallest, Class 2 slightly larger, and even Class 1 performing poorly despite having more samples). This underscores the limitations of MCLR for datasets with overlapping features and severe imbalances.

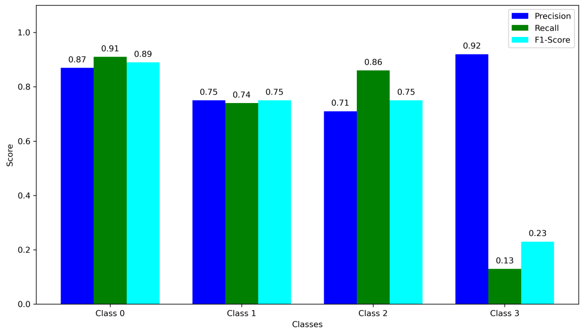
5.2.2.3 Way Forward:
Given these limitations, it became evident that alternative modeling approaches, such as tree-based methods, might better address the dataset's inherent challenges. Decision Tree models, in particular, offer the flexibility to handle class imbalance and feature inter-dependencies, making them a logical next step for evaluation.



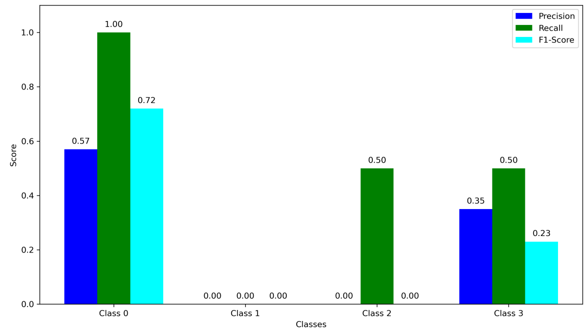
(a) Train Confusion Matrix



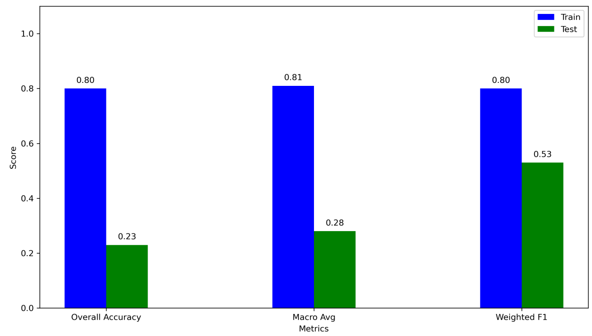
(b) Test Confusion Matrix



(c) Train Metrics by Class



(d) Test Metrics by Class



(e) Macro F1 and Overall Accuracy

Figure 6: Class-Wise & Overall Performance evaluation for MCLR with weights & SMOTE

5.2.3 Decision Tree Performance Evaluation

The performance of the Decision Tree (DT) model on the train and test datasets is presented in Figure 7. The confusion matrices for the train and test datasets are shown in Figures 7(a) and 7(b), respectively.

5.2.3.1 Train Metrics by Class (Figure 7(c)):

- **Class 0:** Precision= 0.92, Recall= 0.91, F1-Score= 0.92 – Model performed consistently well on this class.
- **Class 1:** Precision= 0.80, Recall= 0.84, F1-Score= 0.82 – Moderate performance with a good balance between precision and recall.
- **Class 2:** Precision= 0.85, Recall= 0.84, F1-Score= 0.85 – Model handled this class effectively, better than Class 1.
- **Class 3:** Precision= 0.76, Recall= 0.66, F1-Score= 0.71 – Performance declined due to lower recall. Lower precision shows that number of FPs were high, while lower recalls show that FNs were even higher. The performance was average with a room for improvement

Inference: The model performed well for Classes 0 and 2, moderately for Class 1, and faced challenges with Class 3 due to lower recall.

5.2.3.2 Test Metrics by Class (Figure 7(d)):

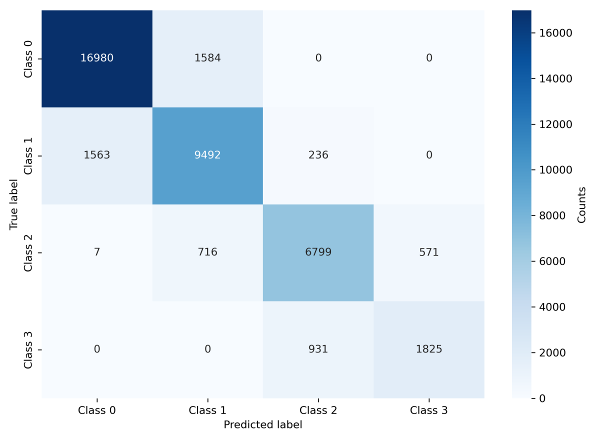
- **Class 0:** Precision= 0.88, Recall= 0.85, F1-Score= 0.84 – Good performance with slight decline compared to training.
- **Class 1:** Precision= 0.71, Recall= 0.78, F1-Score= 0.77 – Moderate performance with balanced metrics.
- **Class 2:** Precision= 0.75, Recall= 0.64, F1-Score= 0.69 – Reduced performance due to lower recall.
- **Class 3:** Precision= 0.73, Recall= 0.69, F1-Score= 0.66 – Reduced performance due to lower recall.

Inference: While the model performed well for Class 0 and moderately for Class 1, performance dropped moderately for Classes 2 and 3, indicating some challenges in generalization of these classes.

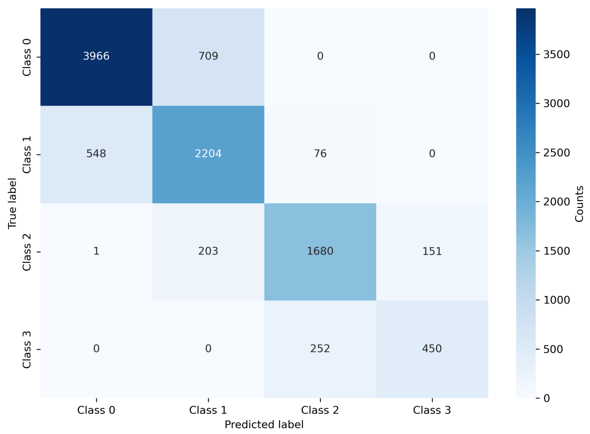
5.2.3.3 Overall Metrics (Figure 7(e)):

- **Train Data:** Overall Acc.= 0.86, Macro F1-Score= 0.82, Weighted F1-Score= 0.86.
- **Test Data:** Overall Acc.= 0.81, Macro F1-Score= 0.78, Weighted F1-Score= 0.81

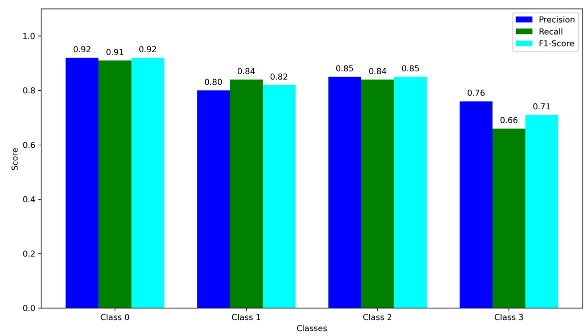
5.2.3.4 Key Takeaways:



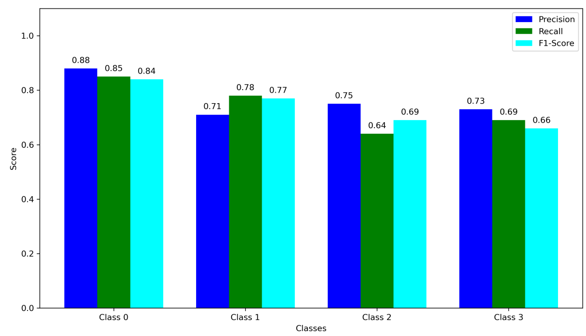
(a) Train Confusion Matrix



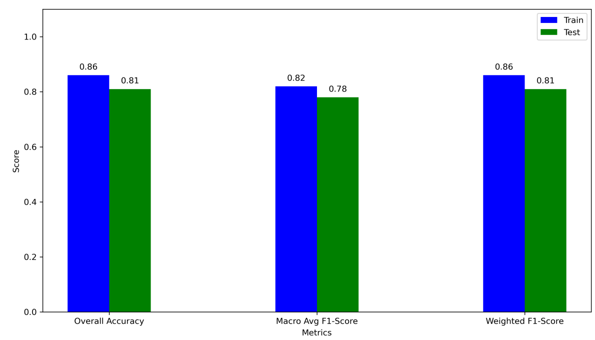
(b) Test Confusion Matrix



(c) Train Metrics by Class



(d) Test Metrics by Class



(e) Macro F1 and Overall Accuracy

Figure 7: Class-Wise & Overall Performance evaluation for Decision Tree

- **Minimal Overfitting:** The small difference between training accuracy (0.86) and test accuracy (0.81) reflects an excellent fit. The model generalizes well to unseen data, with the difference (0.05) indicating minimal overfitting.
- **Good Performance Across Metrics:** The model performs consistently across overall metrics, including accuracy, macro-averaged F1-scores, and weighted F1-scores, for both training and test datasets. This highlights its robustness and reliability.
- **Accuracy in Context:** Accuracy should not be evaluated in isolation, especially in

imbalanced datasets. In this case, the Decision Tree demonstrates robust performance across all three overall metrics, making it a dependable model.

- **Struggles with Minority Classes:** Despite overall robustness, the model struggles with minority class detection due to the inherent dataset imbalance. This indicates potential areas for improvement, such as applying rebalancing techniques or exploring ensemble methods.

6. Conclusion

This study presents the development and evaluation of the **Smart Autonomous Bilge Management System** , an AI/ML-integrated solution designed to enhance regulatory compliance, operational efficiency, and environmental sustainability in maritime bilge water management. While ML has been explored in maritime domains, its application to bilge management remains largely unexplored. By leveraging synthetic data and ML models, this research demonstrates the feasibility and effectiveness of autonomous decision-making in shipboard systems.

A major challenge in implementing ML to bilge water management was the lack of real-world training data, which was addressed by generating a synthetic dataset emulating realistic shipboard conditions for a case vessel. A phased approach, guided by literature, used MCLR as a baseline before transitioning to Decision Trees, which better handle class imbalance and feature dependence.

Our findings highlight the superiority of the Decision Tree (DT) model over Multi-Class Logistic Regression (MCLR) in accurately classifying bilge water management scenarios, emphasizing the importance of non-linear classification techniques in complex maritime operations. While MCLR struggled with class imbalance and failed to generalize effectively, even after weight adjustments and SMOTE, Decision Trees significantly outperformed MCLR, demonstrating better generalization, improved accuracy, and robustness. However, minority class detection remained a challenge, suggesting that balancing techniques or ensemble learning methods could further improve model performance.

The integration of the AI-Driven Decision-Making Module (AIDDM) with the SABIMS Logic Operations Module (SLOM) enables a transition from rule-based automation to fully autonomous predictive control, significantly reducing human intervention while ensuring MARPOL compliance. Beyond the immediate implications for bilge water management, this study underscores the transformative potential of AI/ML in advancing smart maritime systems. The synthetic dataset methodology adopted herein not only addresses the critical issue of data scarcity but also sets a precedent for future research in applying AI-driven solutions to regulatory and safety-critical maritime challenges.

7. Scope for Further Research

While the results validate the feasibility of ML-based bilge management system, several areas remain open for further research and development:

- Full-Scale Implementation of SABIMS: This study focused on a partial subset of SABIMS functionalities. Future research should extend ML integration to a fully autonomous system with real-time shipboard data collection.
- Exploring Advanced ML Models: Ensemble methods (e.g., Random Forest, XGBoost) or Deep Learning approaches could enhance classification accuracy, particularly for minority classes.
- Operational Validation with Real-World Data: While synthetic data enabled model development, future research should validate the models using actual shipboard data to assess real-world performance.

Future research should explore real-world deployment of AI-driven shipboard systems, reinforcing the shift towards intelligent, self-regulating maritime operations in an era of increasing digitalization and automation

References

[1] Standard Club. MARPOL fines for oil pollution and operational best practice. 2020.

[2] Organisation for Economic Co-operation and Development (OECD). Cost Savings Stemming from Non-Compliance with International Environmental Regulations in the Maritime Sector, 2003. Available at: `https://rosap.ntl.bts.gov/view/dot/34106/dot_34106_DS1.pdf`.

[3] Shishir Dutt and Sanjeet Kanungo. Need for smart autonomous bilge management system: A review. Preprint on engrXiv, 2025.

[4] T. Statheros, G. Howells, and K. M. Maier. Autonomous ship collision avoidance navigation concepts, technologies and techniques. *Journal of Navigation*, 61(1):129–142, 2008.

[5] Dragos Simion, Florin Postolache, Bogdan Fleacă, and Elena Fleacă. Ai-driven predictive maintenance in modern maritime transport—enhancing operational efficiency and reliability. *Applied Sciences*, 14(20), 2024.

[6] Capt. Deepak Mantoju. Analysis of marpol implementation based on port state control statistics. *Journal of International Maritime Safety, Environmental Affairs, and Shipping*, 5(3):132–145, 2021.

[7] Tokyo MoU Secretariat. *Annual Report on Port State Control in the Asia-Pacific Region 2023*. Memorandum of Understanding on Port State Control in the Asia-Pacific Region, Tokyo, Japan, 2023.

[8] Paris MoU Secretariat. *Paris Memorandum of Understanding on Port State Control - Including 45th Amendment*. Paris MoU Member States, Paris, France, 2023.

[9] Bahamas Maritime Authority. *Port State Control - Recurring Deficiencies*. Bahamas Maritime Authority, Nassau, The Bahamas, 2023.

[10] United States Coast Guard. *2023 Port State Control Annual Report*. Department of Homeland Security, Washington, D.C., USA, 2023.

[11] Maritime International Secretariat Services Limited. *Shipping Industry Guidance on the Use of Oily Water Separators*. BIMCO, Intercargo, International Chamber of Shipping, Intertanko, Oil Companies International Marine Forum, London, UK, 2006.

[12] Alfa Laval. Removing oily water discharge risks with the alfa laval bluebox sa. `https://www.alfalaval.com/products/separation/separation-control-equpment/bluebox-sa/`, 2018. Accessed: 2025-01-09.

[13] Veolia Water Technologies Deutschland GmbH. Cbm-link: Intelligent clean bilge monitoring system. `https://www.rwo.de/products/link/`, 2019. Accessed: 2025-01-09.

[14] Ranjith Gopalan and Ganesh Kumar Sivasubramanian. Crafting artificial insights: Mastering synthetic data for model training. *International Journal of Scientific Research in Engineering and Management (IJSREM)*, 7(4):1–14, 2023.

[15] Krishnakant Singh, Thanush Navaratnam, Jannik Holmer, Simone Schaub-Meyer, and Stefan Roth. Is synthetic data all we need? benchmarking the robustness of models trained with synthetic images. *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2024.

[16] Xavier Flores, C. M. Ort, Cristoph Ort, Cristina Martin, Lorenzo Benedetti, Evangelina Belia, L. J. P. Snip, Ramesh Saagi, Mansour Talebizadeh, Peter A. Vanrolleghem, Ulf Jeppsson, and Krist V. Gernay. Generation of (synthetic) influent data for performing wastewater treatment modelling studies. *Researchgate*, 2014.

[17] Seongbin An, Trang Do, Juhee Lee, Jiwoo Kim, Yong Jae Kim, Yunji Kim, Changwon Yoon, Sungkyu Jung, Dongha Kim, Sunghoon Kwon, Hang J Kim, Jeongyoun Ahn, and Cheolwoo Park. A comparison of synthetic data approaches using utility and disclosure risk measures. *The Korean Journal of Applied Statistics*, 36(2):141–166, 2023.

[18] Arno Appenzeller, Moritz Leitner, Patrick Philipp, Erik Krempel, and Jürgen Beyerer. Privacy and utility of private synthetic data for medical data analyses. *Applied Sciences*, 12(23):12320, 2022.

[19] Andre Goncalves, Priyadip Ray, Braden Soper, Jennifer Stevens, Linda Coyle, and Ana Paula Sales. Generation and evaluation of synthetic patient data. *BMC Medical Research Methodology*, 20:108, 2020.

[20] O. Rudenko, O. Bezsonov, Kateryna Vashchenko, and Sofiia Rutska. Synthetic dataset generation for efficient neural network training. *International Conference on Computational Linguistics and Intelligent Systems*, 2023.

[21] Yaoxuan Liu, Hao Xu, and Ziling Tang. Synthetic data generation for medical by generative adversarial network. In *2022 International Conference on Electronics and Devices, Computational Science (ICEDCS)*, pages 1–6.

[22] Chris M. Ward, Josh Harguess, and Cameron Beau Hilton. Ship classification from overhead imagery using synthetic data and domain adaptation. *OCEANS 2018 MTS/IEEE Charleston*, 2018.

[23] Woochang Choi and Sukjoon Pyun. Synthetic training data generation for fault detection based on deep learning. *Geophysics and Geophysical Exploration*, 24(3):89–97, 2021.

[24] Vasileios C. Pezoulas, Dimitrios I. Zaridis, Eugenia Mylona, Christos Androutsos, Kosmas Apostolidis, Nikolaos S. Tachos, and Dimitrios I. Fotiadis. Synthetic data generation methods in healthcare: A review on open-source tools and methods. *Computational and Structural Biotechnology Journal*, 23:2892–2910, 2024.

[25] Daniel J. McDuff, Xin Liu, Javier Hernández, Erroll Wood, and T. Baltrušaitis. Synthetic data for multi-parameter camera-based physiological sensing. *Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 2021.

[26] Theodora Kokosi and Katie Harron. Synthetic data in medical research. *BMJ Medicine*, 2022.

[27] Ian Chen, Lucy Jen Huang, Jack Qiao, Dan E. Tamir, and Naphtali Rishe. Combining perception considerations with artificial intelligence in maritime threat detection systems. *International Symposium on Service Oriented Software Engineering*, 2022.

[28] Bill Howe, Julia Stoyanovich, Haoyue Ping, Bernease Herman, and Matt Gee. Synthetic data for social good. *arXiv: Computers and Society*, 2017.

[29] Pang-Ning Tan, Michael Steinbach, and Vipin Kumar. *Introduction to Data Mining*. Pearson, 2006.

[30] Kamal Omari. Comparative study of machine learning algorithms for phishing website detection. *International Journal of Advanced Computer Science and Applications*, 14(9), 2023.

[31] Kyriakos Paraschopoulos. A comparative study of various machine learning classification algorithms. Master's thesis, University of Macedonia, Thessaloniki, Greece, 2020. Supervisor: Konstantinos Psannis.

[32] David W. Hosmer, Stanley Lemeshow, and Rodney X. Sturdivant. *Applied Logistic Regression*. John Wiley & Sons, Hoboken, NJ, third edition, 2013.

[33] Max Kuhn and Kjell Johnson. *Applied Predictive Modeling*. Springer, 2013.

[34] Valentijn M T de Jong, Marinus J C Eijkemans, Ben van Calster, Dirk Timmerman, Karel G M Moons, Ewout W Steyerberg, and Maarten van Smeden. Sample size considerations and predictive performance of multinomial logistic prediction models. *Statistics in Medicine*, 38(9):1601–1619, 2019.

[35] Anass Bayaga. Multinomial logistic regression: Usage and application in risk analysis. *Journal of Applied Quantitative Methods*, 5(2):288–297, 2010.

[36] Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. *An introduction to statistical learning: with applications in R*. Springer texts in statistics. Springer, New York, corrected at 8th printing 2017 edition, 2017.

[37] Felix Abramovich, Vadim Grinshtein, and Tomer Levy. Multiclass classification by sparse multinomial logistic regression. *IEEE Transactions on Information Theory*, 67(7):4637–4646, 2021.

[38] Jiaqi Liang, Guoshu Bi, and Cheng Zhan. Multinomial and ordinal logistic regression analyses with multi-categorical variables using r. *Annals of Translational Medicine*, 8(16), 2020.

[39] Nelson Lay-Raby, Hanns de la Fuente-Mella, and Omar Lameles-Corvalán. Multinomial logistic regression to estimate and predict the job opportunities for people with disabilities in chile. *Information*, 12(9), 2021.

Appendix A
SABIMS Component Description

S.NO.	COMPONENT DESCRIPTION	NAME USED IN DIAGRAMS	ABBREVIATION	ACTUAL LOCATION ON SHIP
		LEVEL SENSORS - INSIDE BILGE WELLS		
1	E/R Bilge well level sensor aft	BILGE WELL (A)	BWA	E/R Aft
2	E/R Bilge well level sensor forward port	BILGE WELL (FP)	BWFP	E/R forward port
3	E/R Bilge well level sensor forward starboard	BILGE WELL (FS)	BWFS	E/R forward starboard
4	M/E flywheel pit level sensor	M/E FLYWHEEL PIT	MEFP	E/R Aft
		LEVEL SENSORS - ON E/R TANK TOP		
5	E/R Tank top flood level sensor aft	TANK TOP LEVEL (A)	TTLA	E/R Tank top aft
6	E/R Tank top flood level sensor forward port	TANK TOP LEVEL (FP)	TTFP	E/R Tank top forward port
7	E/R Tank top flood level sensor forward starboard	TANK TOP LEVEL (FS)	TTFS	E/R Tank top forward starboard
		LEVEL SENSORS - INSIDE GIVEN TANKS		
8	Bilge Tank Level Sensor	BILGE TANK LEVEL	BTL	Bilge Tank
9	Sludge Tank Level Sensor	SLUDGE TANK LEVEL	STL	Sludge Tank
10	Aft Peak Tank Level Sensor	AFT PEAK TANK LEVEL	APTL	Aft Peak Tank
		PUMP RUNNING STATUS		
11	E/R Bilge Pump	E/R BILGE PUMP RUN STATUS	ERBP	E/R Aft Port
12	Fire & Bilge Pump	FIRE & G.S. PUMP RUN STATUS	FGSP	E/R Forward port
13	Bilge & Ballast Pump	BILGE & G.S. PUMP RUN STATUS	BGSP	E/R Forward starboard
14	Main S.W. Pump	MAIN S.W. PUMP RUN STATUS	MSWP	E/R Forward starboard
		OWS & ANCILLARY EQUIPMENT STATUS		
15	Oily Water Separator	OILY WATER SEPARATOR STATUS	OWS	E/R Aft Port
16	15 Ppm Oil Alarm Monitor	15 PPM OIL ALARM MONITOR STATUS	OAM	E/R Aft Port
17	3-Way Changeover Valve-OWS Overboard discharge	3-WAY CHANGEOVER VALVE POSITION STATUS	COV	E/R Aft Port
18	Oily Water Outflow Flow meter	OILY WATER OUTFLOW MONITOR	FM	E/R Aft Port
		PUMPS AUTO/MANUAL MONITORING		
19	E/R Bilge Pump	E/R BILGE PUMP AUTO/MANUAL STATUS	ERBPAM	@ ECR Console
20	Fire & Bilge Pump	FIRE & G.S. PUMP AUTO/MANUAL STATUS	FGSPAM	@ ECR Console
21	Bilge & Ballast Pump	BILGE & G.S. PUMP AUTO/MANUAL STATUS	BGSPAM	@ ECR Console
22	Main S.W. Pump	MAIN S.W. PUMP AUTO/MANUAL STATUS	MSWPAM	@ ECR Console
23	Oily Water Separator	OWS AUTO/MANUAL STATUS	OWSAM	@ ECR Console
		OTHER COMPLIANCE MONITORING SENSORS		
24	Programmable Geo-Fencing GPS monitor	GEO-FENCE MONITOR	GPSM	Bridge ECDIS console
25	Main Engine Sub-Telegraph for ship condition	EN-ROUTE MONITOR	ERM	Bridge and ECR console
26	List angle measurement	LIST MONITOR	LM	Navigation Bridge/ECR
27	Trim value measurement	TRIM MONITOR	TM	Navigation Bridge/ECR
		VALVES		
28	E/R Bilge pump suction valve	----	ERBPSV	@ E/R bilge pump
29	E/R Bilge pump priming valve	----	ERBPPV	@ E/R bilge pump
30	Oily Water Separator inlet valve	----	OWSV	@ OWS
31	Fire & G.S. Pump Direct Suction Valve (from BWFP)	----	FGSPDSV	@BWFP
32	Fire & G.S. Pump Direct Suction Valve	----	FGSPBSV	@FGSP
33	Fire & G.S. Pump priming Valve	----	FGSPPV	@FGSP
34	Fire & G.S. Pump overboard discharge Valve	----	FGSPODV	@ Ship's side
35	Fire & G.S. Pump ballast line discharge Valve	----	FGSPBV	@FGSP
36	Bilge & G.S. Pump Direct Suction Valve (from BWFS)	----	BGSPDSV	@BWFS
37	Bilge & G.S. Pump Direct Suction Valve	----	BGSPBSV	@BGSP
38	Bilge & G.S. Pump priming Valve	----	BGSPPV	@BGSP
39	Bilge & G.S. Pump overboard discharge Valve	----	BGSPODV	@ Ship's side
40	Bilge & G.S. Pump ballast line discharge Valve	----	BGSPBV	@BGSP
41	Main Sea Water pump suction valve	----	MSWPSV	@ MSWP
42	Emergency Bilge Suction Valve	----	EBSV	@ MSWP
43	Main Sea Water pump discharge valve	----	MSWPDV	@ MSWP
44	Bilge Well Aft suction valve	----	BWASV	@BWA
45	Main Engine Flywheel Pit suction valve	----	MEFPSV	@ M/E Aft end, below flywheel
46	Bilge Well Forward Port suction valve	----	BWFPSV	@BWFP
47	Bilge Well Forward Starboard suction valve	----	BWFSSV	@BWFS
48	Aft Peak Tank suction valve	----	APTSV	@Aft Peak Tank
49	Bilge Tank suction valve	----	BTSV	@Bilge Tank
50	Bilge Tank filling valve	----	BTFV	@Bilge Tank
51	Sludge tank filling valve from E/R bilge pump	----	STFV	@Sludge Tank
		PUMP DISCHARGE PRESSURE SENSORS		
52	E/R bilge pump discharge pressure	----	ERBPDP	@ ERBP
53	Fire & G.S. pump discharge pressure	----	FGSPDP	@ FGSP
54	Bilge & G.S. pump discharge pressure	----	BGSDP	@ BGSD
55	Main sea Water pump discharge pressure	----	MSWPDP	@ MSWP