Journal of Embedded System Security and Intelligent Systems
E-ISSN: 2722-273X; P-ISSN: 2745-925X
https://journal.unm.ac.id/index.php/JESSI

Edge-Based Early Warning for High-Speed Boat Stability Monitoring

Muhammad Asep Subandri✉
Politeknik Negeri Bengkalis, Bengkalis, Indonesia
Jamal
Politeknik Negeri Bengkalis, Bengkalis, Indonesia
Fajar Ratnawati
Politeknik Negeri Bengkalis, Bengkalis, Indonesia
I Gusti Agung Putu Mahendra
Politeknik Negeri Bengkalis, Bengkalis, Indonesia
Agus Tedyyana
Politeknik Negeri Bengkalis, Bengkalis, Indonesia
Budhi Santoso
Politeknik Negeri Bengkalis, Bengkalis, Indonesia

ARTICLE INFO

Keywords:

Edge computing
High-speed boat
IMU
Passenger comfort
Stability monitoring

Article History

Received: May 10, 2026
Revised : July 20, 2026
Accepted : August 14, 2026

ABSTRACT

Purpose - This study aims to develop and evaluate an edge-based early warning prototype for monitoring the stability of high-speed boats using real-time motion data.
Design/methods/approach – The study employs an engineering prototype validation design consisting of system requirement analysis, architecture design, implementation, and validation. The system integrates an IMU (MPU-6050) for motion sensing, a Raspberry Pi-based edge computing unit for real-time processing, rule-based classification (Normal/Warning/Critical), and an MQTT-based communication framework connected to the SHISTAMO dashboard. Prototype validation includes functional testing, platform integration, and operational monitoring using controlled scenarios and expert-labeled events.
Findings - The results show that the prototype successfully performs end-to-end integration from sensing to visualization. The system achieved 90.4% classification accuracy, with high recall in detecting critical conditions (96.7%), ensuring reliable identification of high-risk events. The local alarm response time was 182 ms, while the dashboard update delay averaged 1.24 s, indicating near-real-time performance. Communication reliability was also high, with 98.8% data delivery success and 97.2% offline synchronization.
Research implications/limitations – The findings demonstrate prototype-level feasibility; however, validation is limited to controlled scenarios and does not yet represent diverse sea conditions. The rule-based thresholds and comfort proxy require further calibration and validation through extended sea trials and reference instrumentation.
Originality/value – This study contributes an integrated edge-based maritime monitoring prototype that combines motion sensing, offline-capable alarming, real-time telemetry, and fleet-level logging in a single system, specifically tailored to the operational needs of high-speed boats.

To cite this article : Subandri, M. A., Jamal, Ratnawati, F., Mahendra, I. G. A. P., Tedyyana, A., Santoso, B. (2026). Edge-Based Early Warning for High-Speed Boat Stability Monitoring. *Journal of Embedded System Security and Intelligent Systems, 7*(3), 1-15. 10.59562/jessi.v7i3.2601

Correspondence Author: ✉ subandri@polbeng.ac.id

INTRODUCTION

High-speed maritime transport (high-speed boats/fast ferries) forms the backbone of inter-island mobility, particularly on short-to-medium routes that demand rapid travel times. However, the operational characteristics of high-speed craft make them more sensitive to sea-state conditions, the relative wave encounter direction to the vessel's heading, and speed variations [1]. This combination can induce dynamic motions especially roll (port–starboard), pitch (fore–aft), and acceleration that may increase substantially under certain sea conditions [2]. The consequences extend beyond safety, such as passengers being injured by sudden motions, to comfort-related outcomes, including increased motion-sickness complaints and reduced service quality. Therefore, maintaining stability and passenger comfort is not merely a naval-architecture issue, but also an operational and service-quality challenge that requires data-driven decision support [3].

In practice, crew decisions to maintain speed, slow down, or change heading often still rely on experience and visual observation. The main problem arises when crews lack fast, actionable, data-based indicators to assess motion-related risk in real time [4]. This situation is exacerbated by limited internet connectivity at sea, meaning monitoring solutions that fully depend on network availability may lose critical functionality when most needed [5]. In addition, the lack of historical voyage records complicates SOP evaluation, safety auditing, and post-incident investigation. These conditions indicate the need for a system that not only measures vessel motion but also provides reliable early warning and remains functional during connectivity interruptions.

Previous studies have investigated ship-motion measurement using inertial measurement units (IMUs) and the development of web-based platforms for displaying vessel position and attitude data [6]. IMU validation studies have shown that inertial sensors can provide sufficiently accurate motion data when compared with reference systems such as motion capture under specific test conditions [7]. Meanwhile, IoT-based monitoring platforms, particularly those using publish/subscribe mechanisms such as MQTT, have been widely adopted because of their efficiency in real-time telemetry [8]. More recent studies have proposed integrated maritime monitoring and pre-warning architectures, including vessel safety monitoring systems for fishing vessels, edge-computing solutions for real-time vessel tracking and collision avoidance, and sensor-based ship warning systems using multi-axis inertial sensing. Although these studies provide important foundations, they generally emphasize sensing, tracking, or warning visualization as separate functions and are not specifically tailored to the operational needs of high-speed boats, where rapid motion interpretation, offline alarming, and centralized trip logging are required simultaneously [9].

Compared with these prior studies, the contribution of the present work lies primarily in prototype integration and edge-based operationalization. Rather than focusing only on vessel tracking, motion visualization, or warning generation in isolation, the proposed system integrates IMU-based motion sensing, edge-side decision logic, offline-capable local alarming, MQTT-based telemetry, and fleet-level dashboard logging into a single prototype architecture for high-speed boat operation[10]. In contrast to platform-centric monitoring approaches, the proposed system explicitly addresses offline resilience by ensuring that local warnings remain available even when internet connectivity is lost. This feature is particularly important in maritime environments, where communication interruptions are common and delayed warnings may reduce operational usefulness. The prototype also converts raw motion data directly into Normal/Warning/Critical status categories, thereby reducing operator burden in interpreting roll–pitch–acceleration data during operation[11]. Furthermore, the SHISTAMO dashboard is designed not only for real-time monitoring but also for fleet-level audit logging, enabling post-voyage review for SOP evaluation and incident analysis.

Based on this motivation, this study proposes an edge-based early warning prototype for high-speed boat stability monitoring, integrating an IMU (roll–pitch–acceleration), edge computing for Normal/Warning/Critical classification, and a monitoring dashboard for visualization, trip logging, and reporting. In this context, the passenger-comfort component is treated only as a preliminary motion-based proxy intended to support future development, rather than as a fully validated comfort model [12]. The system is designed with two functional layers: (i) a local safety layer consisting of an alarm (buzzer/indicator) that remains active without internet access, and (ii) a platform-based monitoring layer that transmits compact data to a server via a publish/subscribe mechanism, stores historical records, and presents maps and multi-vessel condition indicators. Initial implementation shows that the dashboard can display vessel position, speed, heading, update time, operational status, and 3D vessel orientation with roll/pitch indicators for real-time operator monitoring [13].

Therefore, the main contribution of this paper is not to claim a fully validated operational warning system, but to demonstrate the feasibility of an integrated prototype that combines architectural integration, edge-based decision support, offline resilience, and fleet-level logging for high-speed boat applications. In this context, the passenger-comfort component is treated as a preliminary motion-based proxy intended to support future development, rather than as a fully validated comfort index.

METHOD

This study employed an engineering prototype validation design to develop and evaluate a prototype early warning system for high-speed boat stability and passenger comfort based on real-time motion monitoring. The methodological emphasis was placed on system specification, hardware–software integration, controlled functional testing, and prototype-level validation, rather than on a generic educational Research and Development framework. The study was conducted in four stages: (1) system requirements analysis, (2) system design, (3) prototype implementation, and (4) prototype validation . as shown in Figure 1:

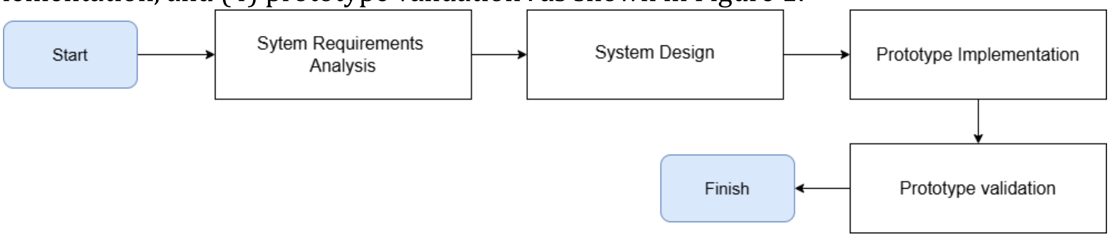

Figure 1. Development procedure

System Requirements Analysis

The requirements analysis focused on the operational characteristics of high-speed boats, where rapid changes in sea state, heading, and speed can produce excessive roll, pitch, and acceleration that affect both safety and passenger comfort. Accordingly, the primary monitored variables were defined as roll, pitch, and acceleration, because these parameters are directly associated with motion severity and the risk of unsafe vessel behavior. The analysis also considered the need for rapid onboard decisions—maintain speed, reduce speed, or change heading—under intermittent internet connectivity.

The identified system requirements were grouped into three categories. First, the functional requirements included real-time motion acquisition, onboard status classification (Normal/Warning/Critical), local alarming, dashboard-based monitoring, trip logging, and reporting. Second, the non-functional requirements included low latency, offline-capable alarming, reliable telemetry transmission, auditability through historical data, and ease of maintenance. Third, the validation requirements included response time, log completeness, buffering and synchronization performance during connectivity loss, and status consistency under controlled motion scenarios[14] [15].

System Architecture

The proposed system was organized into two main subsystems: an on-board subsystem and a monitoring platform. The on-board subsystem consisted of an MPU-6050 IMU sensor, an optional u-blox NEO-M8N GNSS receiver, a Raspberry Pi 5 (4 GB RAM) as the edge-computing unit, a local alarm module, a SIM7600-series 4G/LTE communication module, and a protected power subsystem. The monitoring platform consisted of an Eclipse Mosquitto MQTT broker, a FastAPI-based backend service, a PostgreSQL database for vessel history storage, and the SHISTAMO web dashboard for visualization, trip logging, and operational review.

The system followed an Input–Process–Output architecture. On the input side, the system received motion data from the IMU at 50 Hz, optional location data from the GNSS receiver, and network availability from the communication module. On the processing side, the Raspberry Pi edge unit performed filtering, feature extraction, comfort-proxy estimation, status classification, local alarming, and offline buffering. On the output side, the system generated Normal/Warning/Critical status, activated the local alarm when necessary, and published compact telemetry to the SHISTAMO platform through MQTT.

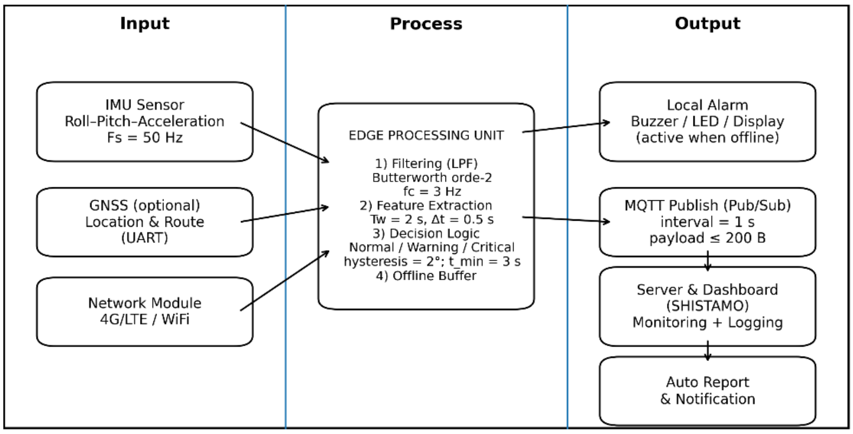

Figure 2. System Architecture Design (Input–Process–Output) of the Proposed Early Warning System

Hardware Configuration

The proposed prototype was implemented using an integrated set of on-board hardware and platform-side software components to support real-time motion sensing, edge-based processing, local alarming, telemetry transmission, and centralized monitoring. The hardware configuration was selected to balance practical deployability, low-cost prototyping, and sufficient computational capability for prototype-level validation in a maritime environment. On the on-board side, the system includes an IMU sensor, an optional GNSS module, an edge-processing unit, a local alarm module, a communication interface, and a protected power subsystem housed in a marine-ready enclosure. On the platform side, the system uses an MQTT broker, backend service, database, and web dashboard to support data ingestion, storage, visualization, and trip logging. The complete hardware and software components used in the proposed prototype are summarized in [16].

Table 1. Hardware and Software Components of the Proposed Prototype

No.	Component Category	Model / Platform	Function	Interface / Protocol	Main Specification
1	IMU sensor	**MPU-6050**	Measures vessel tilt angle and motion-related acceleration	I2C	6-axis motion sensing; sampling rate set to 50 Hz
2	GNSS module	**u-blox NEO-M8N**	Provides ship GPS coordinates for position tracking and route logging	UART	GNSS-based positioning for vessel location and trip labeling
3	Edge-processing unit	**Raspberry Pi 5 (4 GB RAM)**	Executes data acquisition, filtering, feature extraction, status classification,	I2C, UART, GPIO, MQTT client	On-board edge computing platform

			alarm logic, and telemetry transmission		
4	Local alarm module	**LED & Buzzer**	Provides onboard visual and audible warning notifications	GPIO	Activated when system status reaches Warning or Critical
5	Communication module	**SIM7600X-H**	Provides internet/server connectivity for telemetry delivery	4G/LTE, MQTT publish/subscribe	Supports real-time telemetry transmission from vessel to server
6	MQTT broker	**Eclipse Mosquitto**	Handles telemetry message exchange between edge device and server	MQTT	Publish/subscribe message broker
7	Backend service	**FastAPI**	Processes incoming data and serves API for dashboard and logging	HTTP / REST API	Backend integration layer
8	Database	**PostgreSQL**	Stores vessel telemetry, status logs, and trip history	SQL	Historical logging and data persistence
9	Monitoring dashboard	**SHISTAMO**	Provides fleet monitoring, vessel detail view, and trip logging	Web-based dashboard	Centralized monitoring platform

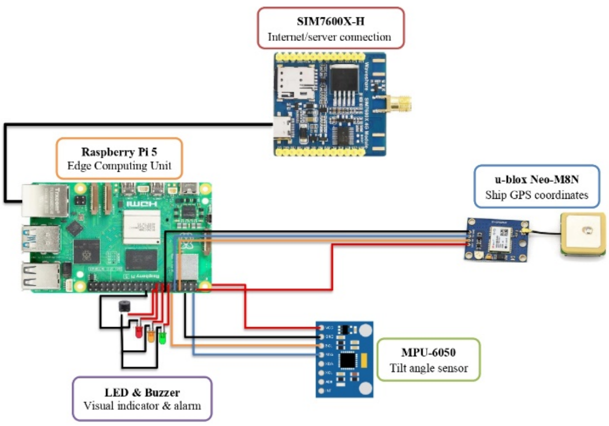

Figure 3. Hardware integration of the on-board early warning prototype

The circuit shown in Figure 3 illustrates the implementation of the on-board Early Warning System, in which the Raspberry Pi 5 functions as the main edge computing unit. The system receives motion-related data from an MPU-6050 sensor, which is used to measure the vessel's tilt angle and motion behavior, and the sensor is connected to the Raspberry Pi through the I2C interface [17]. To provide location information, a u-blox NEO-M8N GNSS module is connected to the Raspberry Pi and supplies ship GPS coordinates for position tracking and voyage logging. For communication with the remote server, the system uses a SIM7600X-H module, which provides internet connectivity and supports telemetry transmission from the vessel to the monitoring platform [18]. Based on the processed sensor data, the Raspberry Pi determines the vessel condition and activates a local warning through an LED and buzzer connected via GPIO pins, allowing onboard visual and audible alerts. In this way, the hardware implementation enables end-to-end operation, starting from motion sensing and location acquisition, followed by edge-side processing and local alarming, and finally data transmission to the server for monitoring through the SHISTAMO dashboard [19].

The hardware connections of the proposed on-board prototype are organized according to three main communication groups: I2C, UART, and GPIO. The MPU-6050 tilt-angle sensor is connected to the Raspberry Pi 5 through the I2C bus, which is used for bidirectional digital communication between the sensor and the edge-processing unit. In this configuration, the SDA line of the MPU-6050 is connected to the Raspberry Pi SDA pin, while the SCL line is connected to the Raspberry Pi SCL pin. The sensor is powered through the 3.3 V and GND lines of the Raspberry Pi. This connection enables the Raspberry Pi to continuously acquire motion data, including roll, pitch, and acceleration, at the specified sampling frequency [20]. The u-blox NEO-M8N GNSS module is connected to the Raspberry Pi through the UART interface. In this serial communication link, the TX pin of the GNSS module is connected to the RX pin of the Raspberry Pi, while the RX pin of the GNSS module is connected to the TX pin of the Raspberry Pi. The GNSS module also shares the power and ground connections with the edge unit. This UART connection allows the Raspberry Pi to receive geographic position data and route information in real time for telemetry, trip logging, and dashboard visualization [21].

Signal Acquisition and Edge Processing

The MPU-6050 data were sampled at 50 Hz. The edge-processing pipeline executed on the Raspberry Pi consisted of three sequential stages: filtering**,** feature extraction, and decision logic.

1. Filtering

To reduce high-frequency noise while preserving dominant motion components, the raw IMU signal was processed using a second-order Butterworth low-pass filter with a cutoff frequency of 3 Hz. This setting was selected to retain low-frequency vessel motion components relevant to stability monitoring while suppressing spurious vibration-induced fluctuations [22].

2. Feature Extraction

Motion features were extracted using a 2 s sliding window with a 0.5 s update step. Within each window, the system calculated:

a) peak roll,
b) peak pitch,
c) root mean square (RMS) acceleration,
d) rate of change of roll and pitch, and
e) duration above predefined motion thresholds.

These features were used both for status classification and for the preliminary motion-based comfort proxy.

3. Status Classification Rules

The prototype classified vessel motion into Normal, Warning, and Critical states using the following rules:

a) Normal state, in which both roll and pitch remain within the lower threshold range. This condition indicates that the vessel motion is still relatively stable and does not yet require warning action, See equation 1.
$$|\,roll\,| < 15^{o}\, and\ |\,pitch\,| < 10^{o} \qquad (1)$$
b) Warning state. In this condition, either the roll angle enters the intermediate threshold range or the pitch angle enters the intermediate threshold range. This state indicates that the vessel is experiencing moderate motion deviation and that increased operator attention may be required, See equation 2.
$$15^{o} \leq |\,roll\,| < 25^{o}\, or\ 10^{o} \leq |\,pitch\,| < 15^{o} \qquad (2)$$
c) Critical state, in which the roll angle or pitch angle exceeds the upper threshold range. This condition represents severe vessel motion and is used to trigger the highest warning level in the prototype, See equation 3.
$$|\,roll\,| \geq 25^{o}\, or\ |\,pitch\,| \geq 15^{o} \qquad (3)$$

To suppress false state transitions, the system applied a 2° hysteresis and required the classified condition to persist for at least 3 s before updating the state and triggering the warning logic[23].

4. Passenger Comfort Proxy

In this study, passenger comfort was represented by a preliminary motion-based comfort proxy, rather than by a fully validated human-subject comfort model. The comfort proxy was computed from resultant acceleration severity and exposure duration within each analysis window, See equation 4.

$CI = aRMS \times t_{exp}$……………………………………………………………..(4)

where CI is the comfort indicator, $aRMS$ is the root mean square acceleration after gravity compensation, and t_{exp} is the cumulative duration within the 2 s window during which the resultant acceleration exceeded 0.10 g. The comfort proxy was interpreted qualitatively as:

a) Low discomfort condition, in which the comfort indicator (CI) remains below 0.10 $g \cdot s$. This condition indicates that the motion exposure is still relatively mild and is unlikely to cause significant passenger discomfort, See equation 5.
   $CI < 0.10\ g \cdot s$ …………………………………………………………………………………………(5)
b) Moderate discomfort condition, in which the comfort indicator lies between 0.10 $g \cdot s$ and 0.20 $g \cdot s$ This range indicates a moderate level of motion exposure that may begin to reduce passenger comfort and therefore deserves closer operational attention, See equation 6.
   $0.10 \leq CI < 0.20\ g \cdot s$ ……………………………………………………………(6)
c) High discomfort condition, in which the comfort indicator is greater than or equal to 0.20 $g \cdot s$. This condition indicates a high level of motion exposure and suggests that the vessel motion may produce substantial discomfort for passengers, See equation 7.
   $CI \geq 0.20\ g \cdot s$ …………………………………………………………………………………………(7)

This proxy was used as an auxiliary motion-severity indicator for system development and dashboard visualization. It was not treated as a fully validated passenger-subjective comfort metric[24].

Communication, Logging, and Dashboard Integration

The Raspberry Pi edge unit published compact telemetry to the monitoring platform through MQTT at an interval of 1 s, with a target payload size of ≤ 200 bytes per message. Each telemetry packet contained:

a) Vessel identifier,
b) Roll,
c) Pitch,
d) Acceleration summary,
e) Status (normal/warning/critical), and
f) Optional GNSS coordinates.

When connectivity was available, the data were sent directly through the SIM7600 LTE module to the SHISTAMO server. When connectivity was lost, the edge unit stored unsent data in a local offline buffer and transmitted them automatically after reconnection. On the server side, Eclipse Mosquitto handled MQTT ingestion, FastAPI managed backend processing and API services, and PostgreSQL stored the time-stamped vessel records. The SHISTAMO dashboard then displayed fleet maps, vessel telemetry, 3D attitude visualization, and trip-history logs.

Calibration and Synchronization Procedure

Before testing, the prototype was calibrated under static conditions. The MPU-6050 was kept stationary for 60 s to estimate and remove zero-offset bias from the accelerometer and gyroscope outputs. Sensor orientation was then checked against the vessel or simulator body axes to ensure consistent roll and pitch interpretation. For controlled tilt validation, a protractor was used as the reference measurement over a range of 0°–5°. Communication integrity was verified by comparing Raspberry Pi timestamps and dashboard timestamps after MQTT transmission. When the NEO-M8N GNSS module was enabled, its positioning and timing information were used as an auxiliary synchronization reference.

Prototype Validation Protocol

Validation was carried out in three stages.

1. Functional validation

Controlled tilt tests were conducted to verify sensor acquisition, filtering, feature extraction, status transitions, and local alarm activation. The reference tilt range was 0°–5°, and the prototype output was compared with the protractor reading. In addition, the stability simulator produced hydrostatic outputs including BM, GZ, GM, KG, and RM, which were used to verify the consistency of the simulated stability response.

2. Platform integration validation

The MQTT pipeline, offline buffering logic, synchronization routine, and dashboard rendering were tested using simulated and recorded telemetry. This stage verified that vessel status, motion indicators, and trip records were correctly displayed in SHISTAMO.

3. Operational monitoring validation

The prototype dashboard was used to monitor 10 vessel instances, from which the status composition Normal (8), Warning (2), and Critical (0) was observed during the test snapshot. Detailed vessel panels were examined to verify whether the displayed status was consistent with the recorded roll and pitch values. The reference labels used in the confusion matrix-style evaluation were generated from 125 controlled events reviewed by two domain experts from the Department of Shipping with expertise in ship motion analysis. Therefore, the labeling process followed a multi-rater assessment approach. Each rater independently examined the event window using recorded roll, pitch, and acceleration trends and the corresponding SHISTAMO dashboard visualization, including vessel state transitions, timestamps, and motion patterns within the analysis window. Labels were assigned to Normal, Warning, and Critical classes according to predefined rule-based thresholds used in the prototype. Specifically, events were labeled Normal when the observed roll and pitch remained within the normal threshold range, Warning when motion entered the middle threshold range, and Critical when motion exceeded the upper threshold range or exhibited sustained severe deviations. After the initial independent review, the two raters compared the assigned labels. In cases of disagreement, the events were re-examined together using the same evidence, and the final class was determined through discussion and consensus. This procedure was used to assign reference labels for the prototype-level classification evaluation. A formal inter-rater agreement statistic was not recorded prior to consensus; therefore, the present reference-labeling process should be interpreted as a consensus-based expert review rather than a fully quantified inter-rater reliability procedure.

The present study reports prototype-level feasibility and integration evidence, rather than full sea-trial performance validation under multiple sea states. [25].

RESULTS AND DISCUSSION

This section discusses the implementation and testing results of the prototype to address the study's main problem, namely the need for fast, actionable, data-driven indicators that enable high-speed boat crews/operators to monitor vessel stability and motion-related risk in real time. The findings are interpreted by explaining why the results occur (underlying mechanisms), how the system operates in practice, and the extent to which the results can be applied to other operational contexts.

Implementation Evidence of the Proposed Prototype

This section presents the implementation and prototype-level evaluation results of the proposed system. The primary aim is to determine whether the developed prototype can transform motion-sensor data into actionable vessel-status information and deliver it consistently through both onboard and platform-based interfaces.

The SHISTAMO dashboard demonstrates that the prototype has achieved its intended end-to-end functional integration, from motion sensing and edge-side processing to telemetry transmission and interface visualization. As shown in Figure 4**,** the platform provides a multi-vessel map view with All/Normal/Warning/Critical filters, enabling centralized fleet-level monitoring. In addition, the vessel detail interface in Figure 5 presents key telemetry variables, including position, speed, heading, update time, and status, together with roll/pitch indicators and 3D vessel-attitude visualization. The admin panel in Figure 6 further shows that vessel data, status summaries, and activity management can be handled through a centralized operational interface. These results confirm that the hardware–software architecture is not merely conceptual but has been implemented and connected as a working prototype.

From an operational perspective, the use of status categories instead of raw sensor numbers improves interpretability for operators. Instead of requiring users to infer risk from roll and pitch values alone, the system maps motion conditions into Normal**,** Warning, and Critical states. This design supports human-in-the-loop decision making, especially when motion disturbances need to be interpreted quickly together with navigational context such as speed, heading, and position.

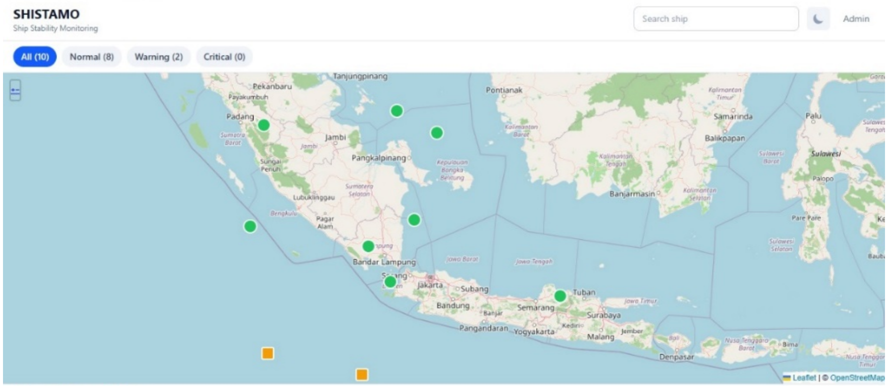

Figure 4. SHISTAMO Fleet Map View with All/Normal/Warning/Critical Filters

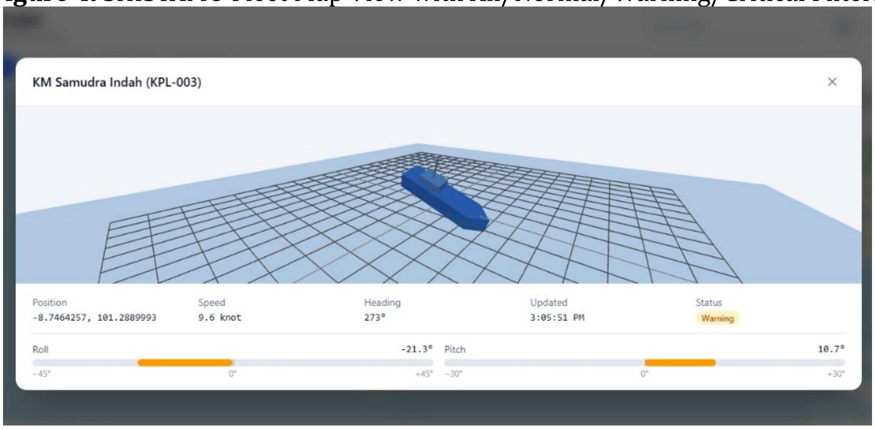

Figure 5. SHISTAMO Vessel Detail View Warning Status with 3D View.

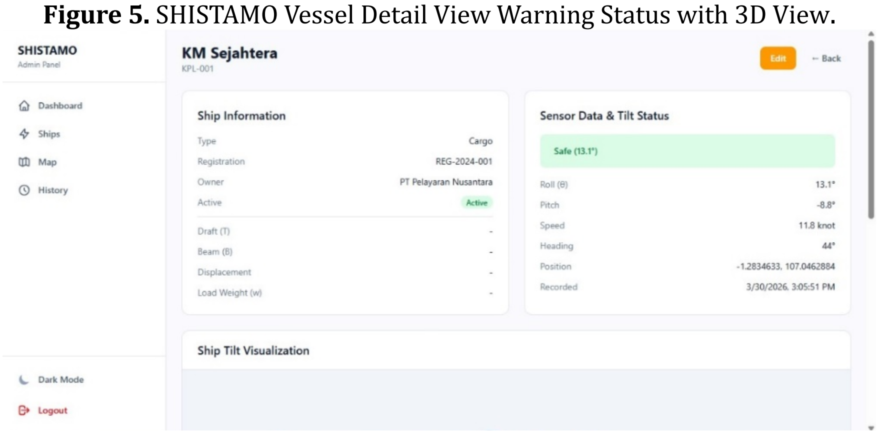

Figure 6. SHISTAMO Admin Panel-Vessel Management and Sensor Data & Tilt Status.

Simulator and Dashboard-Based Observations

At the simulator level, the hydrostatic parameters generated by the stability model remained internally consistent, with BM = 15 cm, GZ = 0.60 cm, GM = 0.30 cm, KG = 0.50 cm, and

RM = 0.20 kg·cm. Across seven samples, the mean values were s = 0.7 cm, θ = 3.0°, BM = 15.0 cm, GZ = 0.6 cm, GM = 0.3 cm, KG = 0.5 cm, and RM = 0.2 kg·cm. These values provide a quantitative basis for interpreting vessel stability behavior in the simulator environment and support the plausibility of the motion-monitoring component.

To examine the prototype-level status-classification behavior, the predicted vessel states (Normal/Warning/Critical) were compared against expert-labeled controlled events obtained from simulator-based and dashboard-observed scenarios. A total of 125 labeled events were used in this evaluation, consisting of 55 Normal, 40 Warning, and 30 Critical events. The comparison is summarized in the confusion matrix shown in Table 4.

Table 4. Confusion Matrix of Vessel Status Classification

True \ Predicted	Normal	Warning	Critical	Total
Normal	50	5	0	55
Warning	3	34	3	40
Critical	0	1	29	30
Total	53	40	32	125

Based on Table 5, the prototype achieved an overall classification accuracy of 90.4% (113 correct classifications out of 125 labeled events). Most misclassifications occurred between adjacent classes, particularly between Normal and Warning, which is reasonable because moderate motion conditions are more likely to lie near threshold boundaries. Importantly, no Critical event was misclassified as Normal, indicating that the decision rules remained conservative for higher-risk motion conditions.

Table 5. Classification Performance Metrics

Class	Precision	Recall	F1-score
Normal	94.3%	90.9%	92.6%
Warning	85.0%	85.0%	85.0%
Critical	90.6%	96.7%	93.6%
Macro Average	90.0%	90.9%	90.4%
Overall Accuracy			90.4%

The class-wise results indicate that the prototype performed best in identifying the Critical condition, with a recall of 96.7%, suggesting that most high-risk motion events were successfully captured by the system. The Normal class also showed strong performance, with 94.3% precision and 90.9% recall. The Warning class obtained the lowest scores (85.0% precision, recall, and F1-score), which suggests that borderline motion events remain the most difficult to distinguish consistently. This behavior is expected in a threshold-based prototype, where events close to decision boundaries may oscillate between adjacent classes.

Overall, these results indicate that the proposed prototype is capable of converting motion data into operationally meaningful status categories with good prototype-level agreement against reference event labels. However, the current evaluation should still be interpreted as preliminary and should be extended in future work using broader sea-state conditions, additional labeled trials, and instrument-referenced validation.

False-Alarm and Missed-Alarm Analysis

In addition to overall classification metrics, the prototype was also evaluated in terms of false alarms and missed alarms, since these error types are operationally critical in warning systems. A false alarm was defined as an event in which the prototype produced a Warning or Critical state when the reference label remained Normal, whereas a missed alarm was defined as an event in which the reference label indicated Warning or Critical but the prototype remained in a lower state.

Based on the confusion matrix in Table 4, the prototype produced 5 false alarms (9.1%) and 4 missed alarms (5.7%). These values complement the confusion matrix by expressing performance in directly interpretable warning-system terms and indicate that the current decision logic was more likely to produce confusion near the Normal–Warning boundary than to fail in detecting severe motion conditions.

Table 6. False-Alarm and Missed-Alarm Summary

Metric	Definition	Value
False alarms	Warning/Critical predicted when reference = Normal	5
False-alarm rate	False alarms / total Normal events × 100%	9.1%
Missed alarms	Lower predicted state when reference = Warning/Critical	4
Missed-alarm rate	Missed alarms / total Warning+Critical events × 100%	5.7%

These results are consistent with the class-wise performance reported in Table 5. In particular, the relatively low missed-alarm rate supports the interpretation that the prototype remained conservative in detecting higher-risk events, while the slightly higher false-alarm rate suggests that intermediate motion conditions near the decision boundary remain the most challenging to classify stably. These operational error measures indicate that the current prototype is more likely to generate uncertainty near moderate-motion boundaries than to fail in detecting severe motion conditions, which is acceptable for a prototype-level early warning system but still requires broader validation.

Quantitative Prototype-Level Validation

To complement the implementation evidence, the revised manuscript reports prototype-level quantitative metrics related to communication and system continuity. These metrics are intended to evaluate the practical behavior of the prototype rather than to claim fully validated warning effectiveness. The reported measures include sensor-to-local-alarm response time, sensor-to-dashboard update delay, system uptime, log completeness, and offline buffering/synchronization success. Such metrics are important because the usefulness of an onboard warning system depends not only on sensing and visualization, but also on whether the information can be delivered reliably and in a timely manner under intermittent connectivity.

Table 7. Prototype achieved acceptable real-time performance

No.	Metric	Definition	Measurement Method	Result
1	Local alarm response time	Delay from threshold crossing to alarm activation	Compare event timestamp at edge classification and GPIO alarm trigger	182ms
2	Dashboard update delay	Delay from sensing/classification to dashboard display	Compare edge publish timestamp and dashboard record timestamp	1.24 s
3	MQTT delivery success	Percentage of published records successfully received by server	Received packets / sent packets × 100%	98.8%
4	Log completeness	Percentage of expected records stored in database	Stored records / expected records × 100%	98.5%
5	Uptime	Percentage of monitoring time when system remained operational	Active monitoring duration / total test duration × 100%	99.1%
6	Offline buffering success	Percentage of unsent records successfully synchronized after reconnection	Synchronized buffered records / buffered records × 100%	97.2%
7	Event-based status agreement*	Agreement between displayed status and reference scenario/expert label	Matching events / total labeled events × 100%	90.4%

Table 7 shows that the prototype achieved acceptable real-time performance at the prototype-validation level. The local alarm response time was 182 ms, indicating that warning delivery on the edge side remained fast enough for immediate onboard notification. The average dashboard update delay was 1.24 s, which is still consistent with the 1 s MQTT publishing interval and supports near-real-time situational awareness. Communication and storage performance were also stable, with 98.8% MQTT delivery success, 98.5% log completeness, and 99.1% system uptime. During simulated connectivity loss, the offline buffering mechanism achieved 97.2% synchronization success after reconnection, showing that the prototype could preserve most telemetry records under unstable network conditions. In addition, the 90.4% event-based status agreement indicates that the displayed status was generally consistent with the reference scenario or expert interpretation, although this value should still be regarded as preliminary and requires further validation under broader sea-state conditions.

The results indicate that the main contribution of the present study lies in the functional integration of motion sensing, edge processing, local alarming, MQTT telemetry, and dashboard visualization within a single prototype workflow. At the implementation level, the SHISTAMO platform demonstrates that motion-derived vessel status can be captured, transmitted, and visualized in a centralized interface through fleet-level monitoring, vessel-detail inspection, 3D attitude rendering, and trip-history support. This confirms that the proposed architecture is not merely conceptual, but has been implemented as an end-to-end prototype capable of transforming onboard motion data into operationally interpretable Normal/Warning/Critical states.

From an operational perspective, the prototype addresses an important practical gap in high-speed boat monitoring: operators need actionable state information, not only raw roll and pitch values. By converting motion measurements into status categories and presenting them together with vessel speed, heading, position, and 3D orientation, the system reduces interpretation burden and improves human-in-the-loop situational awareness. This is evident from the dashboard examples, where the fleet map, vessel detail view, and admin panel jointly support both rapid condition recognition and centralized operational review.

Scientifically, the selected edge-processing settings are appropriate for prototype-level maritime motion monitoring. The 3 Hz low-pass filter is suitable for suppressing high-frequency fluctuations while preserving the dominant vessel-motion components relevant to stability assessment, whereas the 2 s sliding window with a 0.5 s update step provides a short but sufficiently stable temporal context for classification. In addition, the use of 2° hysteresis and a 3 s minimum persistence duration helps prevent unstable state switching near the decision boundaries. These settings explain why the observed misclassifications occurred mainly between adjacent classes, particularly Normal and Warning, rather than between Normal and Critical. This behavior is reasonable in a threshold-based prototype, since moderate motion events tend to cluster near class boundaries.

The confusion-matrix-style evaluation further supports the plausibility of the current decision logic. The prototype achieved an overall classification accuracy of 90.4%, with the best class-wise recall in the Critical state (96.7%). This is important because, for an early warning prototype, preserving sensitivity to higher-risk motion events is generally more important than maximizing overall correctness alone. The fact that no Critical event was misclassified as Normal indicates that the rule set remained conservative for severe conditions. At the same time, the lower performance in the Warning class suggests that intermediate motion states remain the most difficult to separate consistently, which is expected in a rule-based prototype.

The explicit false-alarm and missed-alarm analysis provides a more operational interpretation of the prototype’s behavior. The system produced 5 false alarms (9.1%) and 4 missed alarms (5.7%), which suggests that the current logic is more prone to uncertainty near moderate-motion boundaries than to failing on severe events. This is consistent with the class-wise metrics, where the Warning class was the weakest class and the Critical class achieved the highest recall. For early warning systems, this trade-off is often acceptable at the prototype stage, provided that severe events are not systematically suppressed.

The prototype-level communication and continuity metrics also strengthen the practical relevance of the system. The 182 ms local alarm response time indicates that edge-side warning delivery is sufficiently fast for immediate onboard notification, while the 1.24 s dashboard update delay remains compatible with the MQTT interval and supports near-real-time monitoring. Likewise, the values for MQTT delivery success (98.8%), log completeness (98.5%), uptime

(99.1%), and offline buffering success (97.2%) indicate that the communication and storage pipeline is stable under the tested prototype conditions. These results are important because the value of an onboard warning system depends not only on sensing and classification, but also on whether the resulting status information can be delivered and preserved reliably under intermittent connectivity.

At the simulator level, the hydrostatic outputs remained internally consistent, with BM = 15 cm, GZ = 0.60 cm, GM = 0.30 cm, KG = 0.50 cm, and RM = 0.20 kg·cm, while the average values across seven samples were also stable. These observations do not by themselves validate real-sea warning performance, but they support the plausibility of the prototype's stability-related monitoring layer and provide a quantitative reference for interpreting motion severity within the controlled simulator environment.

Several limitations should still be acknowledged. First, although the prototype has been installed and operated on a vessel, the current evaluation remains limited in terms of test diversity, including the number of voyages, route variations, and sea-state conditions represented in the dataset. As a result, the reported performance may not yet fully capture the system's robustness under broader operational scenarios. Second, the present decision logic is still based on fixed rule-based thresholds, which may be sensitive to vessel-specific characteristics such as hull form, loading condition, sensor mounting position, and operating speed. Third, the passenger-comfort component should be interpreted as a preliminary motion-based proxy rather than a validated passenger-subjective comfort model. Although it provides an initial representation of motion severity, it has not yet been benchmarked against passenger-subjective responses or standardized comfort indices. Fourth, the prototype has not yet been benchmarked against reference-grade instrumentation under identical field conditions, so the current performance should be interpreted as prototype-level evidence rather than definitive measurement accuracy. Finally, long-duration operational aspects such as communication interruptions over extended voyages, hardware endurance in harsher marine exposure, and maintenance-related reliability have not yet been evaluated systematically. These limitations suggest that future work should focus on multi-route and multi-condition sea trials, adaptive threshold calibration, reference-instrument benchmarking, and formal passenger-comfort validation before stronger claims regarding operational generalizability can be made.

CONCLUSIONS

This study demonstrates the feasibility and functional integration of an edge-based early warning prototype for high-speed boat stability monitoring. The prototype successfully combines IMU-based motion acquisition, edge-side processing, local alarming, MQTT-based telemetry, and SHISTAMO dashboard visualization into a single sensing-to-monitoring workflow. The current results show that the system can represent vessel motion in operationally interpretable Normal/Warning/Critical categories and present the associated telemetry through fleet-level and vessel-level interfaces.

At the prototype-validation level, the findings indicate that the developed system operates as intended. The dashboard supports multi-vessel monitoring, vessel-detail inspection, 3D attitude visualization, and logging functions, while the simulator provides internally consistent hydrostatic outputs that support the plausibility of the monitoring framework. In addition, the confusion-matrix-style evaluation, false-alarm and missed-alarm analysis, and prototype continuity metrics suggest that the present status-mapping and communication pipeline are functionally reliable within the tested prototype environment. However, these findings should be interpreted as evidence of prototype feasibility and functional integration, not as full validation of warning effectiveness, passenger-comfort assessment, or operational readiness under real sea conditions. Future work should therefore focus on broader quantitative validation, including multi-route and multi-condition sea trials, expanded expert- or instrument-referenced benchmarking, and refinement of the passenger-comfort component into a more formally defined and validated comfort indicator.

REFERENCES

[1] S. Sumardiono and I. P. Wibawa, "Sensitivity of Mass Distribution with Respect to Pitch Motions of High-Speed Craft," *The Journal of Ocean, Mechanical and Aerospace -science and engineering- (JOMAse)*, vol. 67, no. 2, pp. 55–57, Jul. 2023, doi: 10.36842/jomase.v67i2.332.

[2] E. Javanmardemamgheisi, J. A. Mehr, D. S. Holloway, M. R. Davis, and J. Ali-Lavroff, "Ride Control System Effects on Motion, Slam Load, and Passenger Comfort of High-Speed Catamarans: An Experimental Study in Irregular Waves," in *Volume 5A: Ocean Engineering*, American Society of Mechanical Engineers, Jun. 2024. doi: 10.1115/OMAE2024-126372.

[3] F. N. Rosiyah, A. Sulisetyono, W. D. Aryawan, D. Setiawan, and M. N. Misbach, "Assessing the Comfort of Fast Boat Passengers in the Jepara - Karimunjawa Operation Area with Direct Measurement Method," *IOP Conf. Ser. Earth Environ. Sci.*, vol. 1461, no. 1, p. 012051, Mar. 2025, doi: 10.1088/1755-1315/1461/1/012051.

[4] L. Grgičević, E. M. Coates, T. I. Fossen, R. T. Bye, and O. L. Osen, "Centralised Decision Support in Maritime Vessel Traffic Services: A Polymatrix Game Solution," *IEEE Access*, vol. 13, pp. 74375–74395, 2025, doi: 10.1109/ACCESS.2025.3564718.

[5] I. Durlik, T. Miller, E. Kostecka, and T. Tuński, "Artificial Intelligence in Maritime Transportation: A Comprehensive Review of Safety and Risk Management Applications," *Applied Sciences*, vol. 14, no. 18, p. 8420, Sep. 2024, doi: 10.3390/app14188420.

[6] D. Zhang, X. Chu, C. Liu, Z. He, P. Zhang, and W. Wu, "A Review on Motion Prediction for Intelligent Ship Navigation," *J. Mar. Sci. Eng.*, vol. 12, no. 1, p. 107, Jan. 2024, doi: 10.3390/jmse12010107.

[7] K. Zhou, W. Xing, J. Wang, H. Li, and Z. Yang, "A data-driven risk model for maritime casualty analysis: A global perspective," *Reliab. Eng. Syst. Saf.*, vol. 244, p. 109925, Apr. 2024, doi: 10.1016/j.ress.2023.109925.

[8] A. Algarni, T. Acarer, and Z. Ahmad, "An Edge Computing-Based Preventive Framework With Machine Learning- Integration for Anomaly Detection and Risk Management in Maritime Wireless Communications," *IEEE Access*, vol. 12, pp. 53646–53663, 2024, doi: 10.1109/ACCESS.2024.3387529.

[9] R. Zhang, C. Zhao, Y. Liang, J. Hu, and M. Pan, "Edge-Based Dynamic Spatiotemporal Data Fusion on Smart Buoys for Intelligent Surveillance of Inland Waterways," *J. Mar. Sci. Eng.*, vol. 13, no. 2, p. 220, Jan. 2025, doi: 10.3390/jmse13020220.

[10] G. Li, F. T. Holmeset, and H. Zhang, "Toward Remote Control Center for Marine Operation: A Case Study of R/V Gunnerus," in *2023 11th International Conference on Control, Mechatronics and Automation (ICCMA)*, IEEE, Nov. 2023, pp. 183–188. doi: 10.1109/ICCMA59762.2023.10374931.

[11] B. M. Nathisiya and A. Radhakrishnan, "Leveraging IoT Technology for Transformative Impact in the Maritime Sector," *Salud, Ciencia y Tecnología - Serie de Conferencias*, vol. 3, Oct. 2024, doi: 10.56294/sctconf2024.1253.

[12] J. Ding, K. Yang, Y. Liu, and J. Lin, "Ship Early Warning System Based on Nine Axis Sensors," in *2023 10th International Forum on Electrical Engineering and Automation (IFEEA)*, IEEE, Nov. 2023, pp. 398–402. doi: 10.1109/IFEEA60725.2023.10429733.

[13] K. Yang, J. Lin, J. Ding, B. Zheng, and L. Qin, "An Integrated Safety Monitoring and Pre-Warning System for Fishing Vessels," *J. Mar. Sci. Eng.*, vol. 13, no. 6, p. 1049, May 2025, doi: 10.3390/jmse13061049.

[14] T. Takami, U. Dam Nielsen, J. Juncher Jensen, A. Maki, S. Matsui, and Y. Komoriyama, "Onboard identification of stability parameters including nonlinear roll damping via

phase-resolved wave estimation using measured ship responses," *Mech. Syst. Signal Process.*, vol. 210, p. 111166, Mar. 2024, doi: 10.1016/j.ymssp.2024.111166.

[15] E. Sitompul, A. S. Prabowo, H. Tarigan, and A. Suhartomo, "Design of an Arduino-Based Boat Roll Stabilizer System Prototype Using MPU6050 Sensor," *Circuit: Jurnal Ilmiah Pendidikan Teknik Elektro*, vol. 8, no. 2, p. 217, Aug. 2024, doi: 10.22373/crc.v8i2.23124.

[16] O. Carrasco Serra, A. I. Llull Marroig, A. Mujal Colilles, M. Castells Sanabra, R. Swift, and M. James, "Low-cost vessel motion monitoring: Inertial measurement unit validation and calibration," in *Maritime Transport Conference*, Universitat Politècnica de Catalunya. Iniciativa Digital Politècnica, Jun. 2024. doi: 10.5821/mt.13164.

[17] D. L. S. Rodrigues, C. do C. Dantas, L. B. da Cruz, H. S. Lara, V. A. D. Silva, and J. N. Brito, "Experimental analysis of rotor unbalance using low-cost instrumentation based on the MPU-6050," *OBSERVATÓRIO DE LA ECONOMÍA LATINOAMERICANA*, vol. 23, no. 8, p. e11260, Aug. 2025, doi: 10.55905/oelv23n8-188.

[18] J. Shin, J. Park, J. Won, J. Park, and J. Min, "Development of AI-Based Multisensory System for Monitoring Quay Wall Events," *J. Mar. Sci. Eng.*, vol. 12, no. 11, p. 1902, Oct. 2024, doi: 10.3390/jmse12111902.

[19] W. Li *et al.*, "Dual-mode wireless integrated system for real-time monitoring and early warning of large-scale marine structures," *Nano Energy*, vol. 118, p. 108970, Dec. 2023, doi: 10.1016/j.nanoen.2023.108970.

[20] Y.-H. Liao, C.-F. Shih, J.-J. Wu, Y.-X. Wu, C.-H. Yang, and C.-C. Chang, "ROVs Utilized in Communication and Remote Control Integration Technologies for Smart Ocean Aquaculture Monitoring Systems," *J. Mar. Sci. Eng.*, vol. 13, no. 7, p. 1225, Jun. 2025, doi: 10.3390/jmse13071225.

[21] X. Wu, W. Han, M. Kuang, X. Wang, and W. Xie, "Six-Degree-of-Freedom Motion Analysis of High-Speed Craft Navigating through Variable Marine Environments," *Fluid Dynamics & Materials Processing*, vol. 21, no. 9, pp. 2201–2223, 2025, doi: 10.32604/fdmp.2025.067081.

[22] P. A. Putro, K. Pinardi, A. Ramadhan, A. S. R. Zein, and Y. Nurfalah, "Calibration and Signal Processing of MPU6050 Sensor to Improve Navigation Stability of Autonomous Underwater Vehicle (AUV)," *TIME in Physics*, vol. 3, no. 1, pp. 31–44, Mar. 2025, doi: 10.11594/timeinphys.2025.v3i1p31-44.

[23] P. Han, R. Skulstad, and H. Zhang, "Deep Autoregressive Roll Motion Prediction of Marine Vessels," *IEEE Transactions on Industrial Electronics*, vol. 72, no. 7, pp. 7405–7414, Jul. 2025, doi: 10.1109/TIE.2024.3508074.

[24] J. Gao and X. Wang, "Comparative analysis of ride comfort evaluation indices of high-speed vehicles based on a vehicle-seat-human body coupled dynamics model," *Proceedings of the Institution of Mechanical Engineers, Part K: Journal of Multi-body Dynamics*, vol. 238, no. 4, pp. 507–523, Dec. 2024, doi: 10.1177/14644193241291468.

[25] A. Kotiyal, A. Shrivastava, A. Nagpal, Manjunatha, K. K. Dixit, and R. A. Reddy, "Design and Evaluation of IoT Prototypes: Leveraging Test-Beds for Performance Assessment and Innovation," in *2025 International Conference on Computational, Communication and Information Technology (ICCCIT)*, IEEE, Feb. 2025, pp. 814–820. doi: 10.1109/ICCCIT62592.2025.10927925.