

Leveraging Cloud-Fog Automation for Autonomous Collision Detection and Classification in Intelligent Unmanned Surface Vehicles

Thien Tran^{*}, Quang Nguyen^{†‡}, Jonathan Kua^{*}, Minh Tran^{‡||}, Toan Luu^{§‡}, Thuong Hoang^{*}, and Jiong Jin[¶]

^{*}Deakin University, Australia; [†]University of Birmingham, UK; [‡]RMIT University, Vietnam;

[§]VinUniversity, Vietnam; [¶]Swinburne University of Technology, Australia; ^{||}University of Tasmania, Australia

{peter.tran, jonathan.kua, thuong.hoang}@deakin.edu.au; mxn498@student.bham.ac.uk;

minh.tranquang@rmit.edu.vn; toan.ld@vinuni.edu.vn; jiongjin@swin.edu.au

Abstract—Industrial Cyber-Physical Systems (ICPS) technologies are foundational in driving maritime autonomy, particularly for Unmanned Surface Vehicles (USVs). However, onboard computational constraints and communication latency significantly restrict real-time data processing, analysis, and predictive modeling, hence limiting the scalability and responsiveness of maritime ICPS. To overcome these challenges, we propose a distributed Cloud-Edge-IoT architecture tailored for maritime ICPS by leveraging design principles from the recently proposed Cloud-Fog Automation paradigm. Our proposed architecture comprises three hierarchical layers: a Cloud Layer for centralized and decentralized data aggregation, advanced analytics, and future model refinement; an Edge Layer that executes localized AI-driven processing and decision-making; and an IoT Layer responsible for low-latency sensor data acquisition. Our experimental results demonstrated improvements in computational efficiency, responsiveness, and scalability. When compared with our conventional approaches, we achieved a classification accuracy of 86%, with an improved latency performance. By adopting Cloud-Fog Automation, we address the low-latency processing constraints and scalability challenges in maritime ICPS applications. Our work offers a practical, modular, and scalable framework to advance robust autonomy and AI-driven decision-making and autonomy for intelligent USVs in future maritime ICPS.

Index Terms—Maritime Industrial Cyber-Physical Systems, Cloud-Fog Automation, Intelligent Unmanned Surface Vehicles

I. INTRODUCTION

The adoption of maritime Industrial Cyber-Physical Systems (ICPS) has significantly advanced Unmanned Surface Vehicles (USVs) by enabling intelligent autonomous operations in navigation, environmental monitoring, and industrial applications [1]–[3]. As ICPS become integral to maritime autonomy, their ability to detect and classify collisions in low-latency remains a critical challenge. Conventional collision detection systems are based on visual sensors (cameras, LiDAR, and radar), often susceptible to environmental disturbances such as fog, low visibility, and water splashes [4]. In addition, alternative sensing methods that rely on proprioceptive data from the Inertial Measurement Unit (IMU) have emerged, offering robustness against environmental disturbances [5]. By utilizing derived signals as input data, Artificial Intelligence (AI)-

driven mechanisms, such as Convolutional Neural Networks (CNN) classification methods, can reliably identify collisions in low-latency environments without dependence on visual sensing conditions [6]. However, onboard processing limits computational resources, resulting in high latency, scalability issues, and energy inefficiency in real-world deployment [7].

Although recent studies have explored IMU-based collision classification for USVs, existing methods are mainly based on localized/centralized processing, limiting low-latency performance and scalability. The recently Cloud-Fog Automation paradigm provides a distributed, intelligent, and scalable computing framework that enables real-time decision-making, adaptive control, and resilient automation in ICPS, particularly in complex, latency-sensitive, or resource-constrained environments [8], [9]. While widely adopted in ICPS, most USV-based ICPS architectures currently do not leverage network-centric architectures. This gap motivates us to propose and develop a distributed Cloud-Edge-IoT architecture tailored for low-latency USV collision classification and detection, while efficiently distributing computational loads, decentralization, and power efficiency [10], [11].

In this paper, we make the following key contributions:

- Proposed a hierarchical Cloud-Edge-IoT architecture tailored for maritime ICPS.
- Developed a distributed modular computing architecture for low-latency USVs' AI-driven collision classification and detection.
- Experimentally validated the classification accuracy, processing time, and latency analysis across hierarchical layers in controlled water environments.

The remaining paper is structured as follows: Section II presents the proposed system architecture, while Section III details the research methodology. Section IV describes the experimental setup, followed by Section V, which presents experimental results and discussions. Finally, Section VI concludes the paper and outlines future research directions.

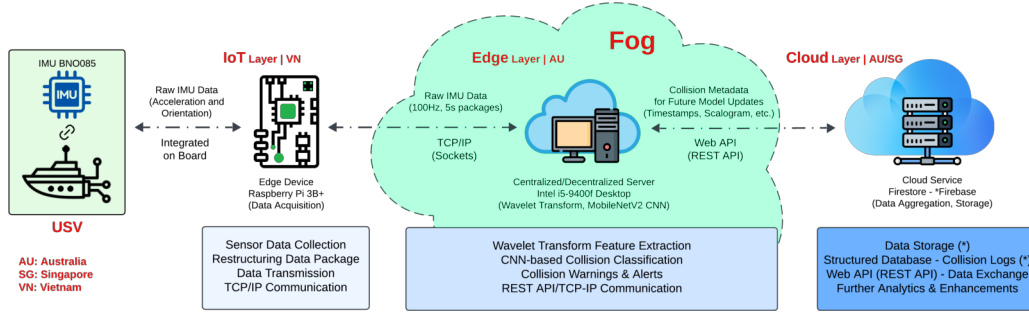


Fig. 1. Distributed Cloud-Edge-IoT architecture leveraging Cloud-Fog Automation for USVs' collision detection and classification.

II. SYSTEM ARCHITECTURE

To address the computational and system performance limitations associated with onboard processing, we propose a distributed Cloud-Edge-IoT computing architecture based on Cloud-Fog Automation to improve power efficiency, scalability, manageability, and complexity constraints. The system is structured into three hierarchical layers: the Cloud Layer (data aggregation for future enhancement), the Edge Layer (AI-driven detection and classification), and the IoT Layer (data acquisition). In this architecture, the fog computing domain encompasses all distributed processing that occurs between the Cloud and IoT layers, enabling adaptive, real-time decision-making close to the data source. This distributed architecture ensures computational load balancing, low-latency decision-making, and scalability for ICPS in maritime applications.

Figure 1 illustrates the overall system architecture, highlighting the distinct roles of each layer and the communication flow throughout the network. This hierarchical layered processing ensures that low-latency collision classifications are performed at the Edge layer, minimizing delays while offloading long-term storage functions to the Cloud layer.

The proposed framework integrates proprioceptive sensing IMU, AI-driven classification, and distributed data processing to improve low-latency collision classification for USVs. Each layer is designed to perform specialized tasks, ensuring a seamless and hierarchical computational workflow:

- **Cloud Layer:** Functions as a structured data storage node, maintaining essential records and structured datasets for future analytics, predictive maintenance, re-training of AI models, and human validation reference.
- **Edge Layer:** Conducts feature extraction using Wavelet Transform, applies a CNN model for collision classification, and then provides low-latency warnings.
- **IoT Layer:** Acquires raw proprioceptive IMU data and performs basic data restructuring before transmitting to the Edge Layer for collision classification and detection.

A. Cloud Layer: Data Aggregation for Future Enhancement

The Cloud Layer (hosted on the Cloud Server) serves as a structured data repository for collision event logs, sensor data, and related classification results. Unlike conventional Cloud-based architectures, where AI processing is conducted

remotely, this work retains AI inference at the Edge Layer [9], [12] while utilizing the Cloud Layer exclusively for:

- Long-term storage of classified collision data.
- Dataset aggregation for future AI model refinements and predictive maintenance research.
- System-wide analytics for monitoring the performance of the distributed computing architecture.

It is noted that the Cloud Layer does not yet perform computations or model retraining, but remains a scalable component for further system enhancements through predictive analytics, and federated learning-driven optimization [13], [14].

B. Edge Layer: AI-driven Classification and Decision-Making

The Edge Layer (implemented on a centralized/decentralized computing unit) is responsible for data feature extraction and AI-driven collision classification. This layer processes incoming IMU data and classifies collision types using a CNN trained on labeled impact datasets. The Edge Layer facilitates low-latency AI processing by leveraging a distributed and powerful computational infrastructure. Although processing tasks are not localized directly onboard, this approach effectively reduces reliance on cloud connectivity, thus significantly enhancing operational responsiveness and reliability. Key processing steps include:

- Wavelet Transform to extract time-frequency domain features from IMU signals.
- CNN-based classification model optimized for low-latency inference, distinguishing between different stages of impacts and collision position.
- A Decision-making framework that generates collision alerts and action recommendations based on the detection classification results.

C. IoT Layer: Data Acquisition and Preprocessing

The IoT Layer consists of a single-board computer connected to an IMU. The IMU continuously captures linear acceleration and angular velocity data, providing proprioceptive sensing for collision event detection. By offloading complex processing to the Edge Layer, the IoT Layer remains computationally lightweight and energy-efficient, crucial for overall system performance enhancement. To ensure data integrity and low-latency processing, this layer only performs basic data restructuring and data transmission via Wi-Fi.

III. METHODOLOGY

A. Data Acquisition and Preprocessing

The IoT Layer (implemented on a Raspberry Pi 3B+) is responsible for IMU data acquisition and preliminary preprocessing. The system utilizes an Adafruit IMU BNO085, which provides 9-degree-of-freedom (9-DOF) orientation data by capturing 1) Linear acceleration (m/s^2), 2) Angular velocity (rad/s), 3) Orientation (Euler angles), etc. To ensure high temporal resolution, the IMU samples data at 100Hz, transmitting the combined data package of a 5-second window with a window offset of 3 seconds to the Edge Layer. The raw data is restructured, packaged, and transmitted from the IoT to the Edge using the TCP/IP protocol. The raw dataset contains noise and interference, mitigated at the IoT layer by the BNO085 IMU's integrated MCU, which performs preprocessing and sensor fusion before data use.

B. Feature Extraction and AI-driven Classification

The Edge Layer (Intel i5-9400f Desktop) receives proprioceptive IMU data, performs advanced feature extraction using Morlet Wavelet Transform techniques [15], and classifies collisions utilizing a MobileNetV2-based CNN model.

1) *Wavelet Transform-Based Feature Extraction:* To effectively capture both time-domain and frequency-domain patterns, this process employs the Wavelet Transform for feature extraction. This method offers distinct advantages over Fourier-based analysis by providing localized, time-sensitive frequency decomposition. While Fourier analysis provides high-frequency resolution, wavelets balance resolution with precise time localization, making them particularly suitable for transient event detection and making them well-suited for impact identification in USVs [16]. The extracted wavelet coefficients are converted into scalograms, 2D visual representations of energy distribution, and input to the CNN classifier. Figure 2 illustrates the application of the continuous Wavelet Transform to ship collision data. The selection of the wavelet function plays a crucial role in determining the resolution of the continuous Wavelet Transform across the time and frequency domains, an aspect that, together with identifying appropriate wavelets for specific signal characteristics.

2) *Convolutional Neural Network (CNN) Model:* A custom CNN architecture [17] is trained on labeled wavelet scalogram, distinguishing between three types of collision and default stage: 1) Bow (Frontal); 2) Port (Left-Side); 3) Starboard (Right-Side); 4) None (Default) as shown in Figure 3. The classification task utilizes a lightweight CNN architecture (MobileNetV2), a convolutional neural network optimized for embedded systems and resource-constrained hardware [18]. The MobileNetV2 CNN model employs inverted residuals with linear bottlenecks, beginning with a standard 3x3 convolutional layer followed by bottleneck blocks composed of a 1x1 expansion convolution, depthwise convolution, and a 1x1 projection convolution to maintain efficient, low-dimensional representations. It processes six input channels (3 linear acceleration, 3 angular acceleration), receives

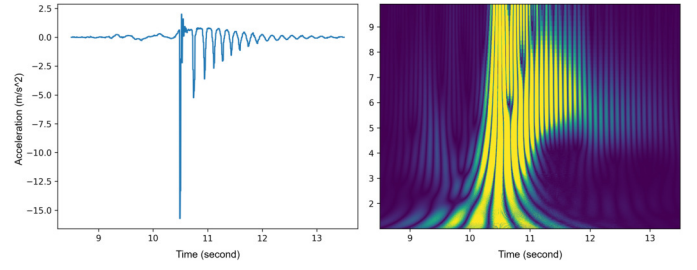


Fig. 2. X-axis raw acceleration data (left) and continuous Wavelet Transform scalogram (right) for a ship collision event.

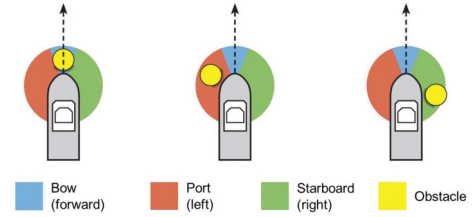


Fig. 3. Three main impact types of USV collision classification [6].

192x150 RGB scalograms in a 5-second duration (X-axis) and under 10Hz (Y-axis), with normalized continuous Wavelet Transform coefficient range. These details ensure clarity regarding the temporal length and input channels, enhancing the reproducibility of our research methodology.

C. Low-Latency Communication and System Integration

The proposed architecture employs TCP/IP and REST API communication protocols over the Internet to facilitate seamless integration [19], as illustrated in Figure 1. Specifically, the IoT Layer transmits a restructured IMU data package to the Edge Layer for immediate processing and collision classification. Subsequently, the Edge Layer forwards classified collision metadata and relevant analytical data to the Cloud Layer (Firestore), serving primarily as a repository for data aggregation, long-term analytics, and future scalability.

The data transmission includes timestamped collision events, IMU data, classified collision types, Wavelet Transform outputs, and human-validated reference. This structured communication pipeline ensures robust and efficient integration across all layers for the distributed computing processes, optimizing low-latency responsiveness and analytical capabilities.

IV. EXPERIMENTAL SETUP

A. Hardware and Software Configurations

The system is implemented using three hierarchical layers, each equipped with distinct hardware and software components. Table I below summarizes the system configurations. The IoT Layer collected 100Hz IMU data and transmits processed data to the Edge Layer. The Edge Layer performed Wavelet Transform-based feature extraction and CNN-based collision classification, forwarding classified results to the Cloud Layer for data aggregation and future analytics.

B. Testing Environment and Collision Emulation

To ensure realistic and controlled testing conditions, the experimental setup utilized a model-scale USV deployed in a calm water environment. The testbed is designed to replicate real-world maritime conditions while maintaining repeatability and controlled impact scenarios. Figure 4 presents that the USV is fitted with an IMU BNO085 sensor and a Raspberry Pi 3B+ at the end of the shipdeck.

We use a model-scale vessel to represent the USV in our work. The IMU sensor is securely mounted to the hull, ensuring minimal external disturbances affect the measurements. The USV is manually controlled to impact the predefined collision points for data testing collections. To minimize external disturbances, our experiments are conducted in an 8m×4m×2m enclosed water basin to provide a controlled water surface with predictable impact dynamics across multiple trials, and consistency in environmental conditions for repeatability.

The testing framework consisted of three predefined impact zones, replicating real-world collision scenarios. Each scenario is repeated at least 10 times per impact type, ensuring statistical reliability in classification accuracy. The IMU sensor recorded acceleration and angular velocity at the moment of impact, providing raw motion data for analysis.

C. Evaluation Metrics and Data Collection

The system performance is evaluated using four primary metrics, as summarized in Table II. These metrics are selected to assess the accuracy, responsiveness, and communication efficiency of the proposed system. To ensure reliable evaluation, data is systematically collected during multiple experimental trials conducted under controlled conditions. Each trial involved recording IMU data, classification outputs, and precisely measuring processing time and latency across layers. Metrics are quantified through confusion matrix analysis, system timestamps, and Round-Trip Time (RTT) analysis to provide an overall performance of the proposed system. In this study, the RTT analysis is used to evaluate the full processing time of Edge-to-Cloud data transmission.

TABLE I
EXPERIMENTAL SETUP AND CONFIGURATION DETAILS OF THE DISTRIBUTED CLOUD-EDGE-IOT COMPUTING ARCHITECTURE.

Layer	Device	Software	Function
IoT Layer	Raspberry Pi 3B+, IMU BNO085	Raspberry Pi OS, Sockets	IMU data acquisition (100Hz) and dataset restructuring
Edge Layer	Intel i5-9400f Desktop	TensorFlow, PyWavelets, Python, MobileNetV2	Feature extraction (Wavelet Transform), CNN classification, low-latency collision decision-making
Cloud Layer	Firestore (Firebase)	Web API, Firebase Database	Structured data storage, aggregation for future analytics and model retraining

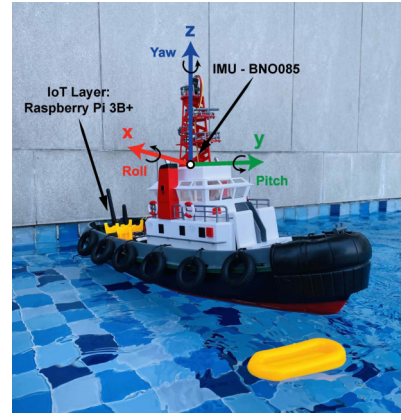


Fig. 4. Model-scale USV with integrated onboard system, featuring a BNO085 IMU and Raspberry Pi 3B+.

V. RESULTS AND DISCUSSIONS

A. Classification Performance

The classification accuracy of the CNN model, based on the MobileNetV2 framework, is evaluated using a confusion matrix derived from 50 trials, ensuring a balanced dataset with at least 10 trials conducted for each impact type to minimize bias and ensure fairness in the classification results. The results indicated that the proposed system successfully distinguishes between all collision types, as shown in Figure 5.

Table III summarizes the classification performance across four distinct impact cases. Overall, the system achieved 86% aggregate accuracy (43/50). The unified precision, recall, and F1-score reflected balanced predictive capabilities across all classes.

Individually, the 'None' and 'Port' impacts showed the highest precision (0.909), indicating strong accuracy in correctly classifying these specific scenarios. The 'Bow' impact achieved the highest recall (0.917), demonstrating reliable detection of actual collision events. However, "Starboard"

TABLE II
PERFORMANCE EVALUATION METRICS

Metric	Definition	Measurement Method
Classification Accuracy	Correctly classified collision event rate	Confusion matrix analysis
IoT Processing Time	Processing time of IMU data collection and restructuring	Timestamped processing before transmission to Edge
IoT-to-Edge Latency	Transmission time IMU data package from IoT to Edge	Timestamped transmission events log
Edge Layer Processing Time	Processing time of feature extraction and CNN classification	Time stamps captured at data arrival and CNN output
Edge-to-Cloud Latency	Transmission time from Edge to Cloud for storage	Measure RTT of structured data transmission

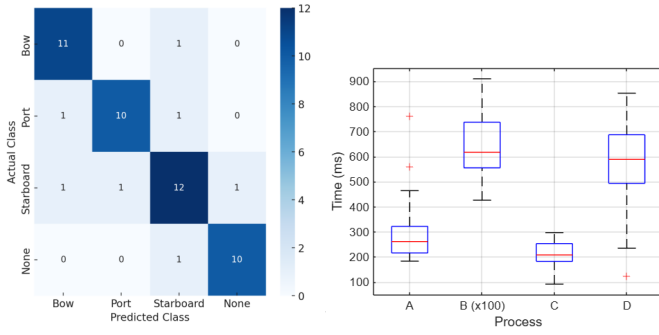


Fig. 5. Confusion matrix for USV collision classification (left) and Processing time/latency analysis across layers (right).

TABLE III
CLASSIFICATION PERFORMANCE SUMMARY

Impact Case	Total Cases	Correct	Precision	Recall	F1-Score
Bow	12	11	0.846	0.917	0.880
Port	12	10	0.909	0.833	0.870
Starboard	15	12	0.800	0.800	0.800
None	11	10	0.909	0.909	0.909
Overall	50	43	0.861	0.860	0.859

impacts showed lower performance, with an F1-score of 0.800, suggesting that classification accuracy for this class could benefit from additional training or refined feature extraction.

Technically, slight variations in collision force and angle occasionally cause detection and classification errors. To address this issue, a continuous CNN model retraining facilitated by the Cloud Layer is recommended for enhancing system performance. Moreover, integrating attention mechanisms into the MobileNetV2 architecture could enhance the model's focus on critical scalogram features. For instance, H. Cheng et al. [20] demonstrate that a fused spatial-channel attention mechanism in MobileNet improves classification accuracy by prioritizing relevant image regions. Moreover, sensor placement influences detection and classification outcomes, as a centrally positioned IMU ensures stable and reliable data capture, minimizing external effects. In summary, the results underscore the effectiveness and reliability of the proposed architecture while identifying areas such as the "Starboard" class that require further optimization to enhance overall system performance.

B. Processing Time and System Latency Analysis

System latency and processing times are measured across the three computational layers: Cloud (data aggregation), Edge (AI-driven classification), and IoT (data acquisition), as illustrated in Table IV and Figure 5. The results analysis across layers underscored the effectiveness of adopting Cloud-Fog Automation within the proposed architecture. The IoT Layer exhibited negligible processing latency, whereas IoT-to-Edge transmission introduced moderate latency (< 300 ms), reflecting the typical delays encountered in cross-regional network communication. Edge-to-Cloud latency varies due to geographical proximity and server location differences.

Notably, the Edge Layer achieved minimal processing latency (6.44 ms), ensuring reliable low-latency AI-driven collision detection and classification. Edge-to-Cloud communication latency varied substantially by cloud server location, with Australia-based cloud services exhibiting lower RTT values (210.15 ms) compared to Singapore-based services (583.66 ms) due to geographical distance and network constraints.

RTT measurements are utilized rather than one-way latency measurements due to their practicality and reliability in accurately assessing communication delays. One-way latency measurement often requires precise clock synchronization between geographically distributed nodes, introducing complexity and potential measurement inaccuracies [21]. RTT inherently accounts for both transmission and response delays, providing a more straightforward and representative measure of responsiveness and network communication performance [22].

TABLE IV
PROCESSING TIME AND SYSTEM LATENCY SUMMARY

Process	Mean (ms)	Std Dev. (ms)
IoT Layer (VN)	1.00	± 1.00
A – IoT-to-Edge (VN $\rightarrow$ AU)	282.96	± 100.79
B – Edge Layer (AU)	6.44	± 1.08
C – Edge-to-Cloud (AU $\leftrightarrow$ AU*)	210.15	± 47.79
D – Edge-to-Cloud (AU $\leftrightarrow$ SG*)	583.66	± 152.11

VN: Vietnam; AU: Australia; SG: Singapore – Locations | * Cloud Service

These findings emphasize the importance of geographically optimized cloud service selection to maintain low-latency communication within USV collision classification systems. Nevertheless, the latency observed between the Edge Layer and Cloud Layer does not negatively impact low-latency collision detection and classification capabilities, as critical AI-driven processing tasks are executed efficiently at the Edge Layer. Collectively, the results validated the robustness and effectiveness of the proposed distributed Cloud-Edge-IoT computing architecture for maritime autonomous operations.

VI. CONCLUSIONS AND FUTURE WORK

In this paper, we proposed and developed a distributed Cloud-Edge-IoT computing architecture tailored for low-latency AI-driven collision detection and classification in USVs within the context of maritime ICPS. By leveraging the Cloud-Fog Automation paradigm, we proposed a computational framework that is distributed across decentralized hierarchical layers to effectively mitigate the limitations of the conventional onboard processing approach.

We integrated Wavelet-based feature extraction with a MobileNetV2-based CNN model, achieving 86% accuracy. Our experimental validation demonstrated effective computational performance, reduced onboard processing loads, and improved low-latency collision classification performance. Specifically, the IoT Layer reliably captured IMU data packages, the Edge Layer executed low-latency classification, and

the Cloud Layer efficiently aggregated data for future enhancements. Experimental results analysis further confirmed the efficiency of the proposed distributed computing architecture, achieving low processing and transmission times at observed IoT and Edge layers (*mean*: 282.96 + 6.44 ms, *SD*: 100.79 + 1.08 ms). Network latency assessments revealed variations depending on cloud server location, emphasizing the importance of geographic proximity. Specifically, RTT latency for local (AU-based) communication (*mean*: 210.15 ms) is significantly lower than international cloud locations, such as Singapore (AU-SG, *mean*: 583.66 ms).

Our proposed architecture not only demonstrates the potential for maritime autonomy but also provides a generalized architecture applicable to broader ICPS applications. By decentralizing computational processes and reducing latency, our framework can support diverse ICPS applications, including industrial networked robotics, precision manufacturing, industrial automation, resource management, and smart factory. For instance, industries utilizing networked robotics and autonomous systems can benefit from enhanced low-latency decision-making, advanced analytics, predictive maintenance, and dynamic system adaptation, leveraging distributed computing enabled by the proposed architecture. Moreover, critical sectors such as healthcare, infrastructure monitoring, and disaster response can utilize Cloud-Fog Automation's scalability and low-latency advantages to achieve robust teleoperation and AI-empowered autonomous capabilities [23]–[25].

Our work contributes to the future development of flexible, scalable, and resilient intelligent systems. The proposed architecture offers a foundational framework that can be adapted to a wide range of industrial sectors, thereby advancing intelligent scalability and operational efficiency in next-generation ICPS. Future work will extend our architecture's capabilities beyond data aggregation, hence incorporating advanced predictive analytics, federated learning, and continuous AI model fulfillment to enhance low-latency decision-making, adaptive autonomous task offloading, and proactive autonomy [26].

REFERENCES

- [1] N. Tabish and T. Chaur-Luh, "Maritime Autonomous Surface Ships: A Review of Cybersecurity Challenges, Countermeasures, and Future Perspectives," *IEEE Access*, vol. 12, p. 17114–17136, 2024.
- [2] O. K. Kinaci, "Ship Digital Twin Architecture for Optimizing Sailing Automation," *Ocean Engineering*, vol. 275, p. 114128, 2023.
- [3] Y. Feng, B. Hu, H. Hao, Y. Gao, Z. Li, and J. Tan, "Design of Distributed Cyber-Physical Systems for Connected and Automated Vehicles with Implementing Methodologies," *IEEE Transactions on Industrial Informatics*, vol. 14, p. 4200–4211, 2018.
- [4] S. Thombre, Z. Zhao, H. Ramm-Schmidt, J. M. V. García, T. Malkamäki, S. Nikolskiy, T. Hammarberg, H. Nuortie, M. Z. H. Bhuiyan, S. Särkkä, and V. V. Lehtola, "Sensors and AI Techniques for Situational Awareness in Autonomous Ships: A Review," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, p. 64–83, 2022.
- [5] P. Nath and A. Malepati, "IMU-Based Accident Detection and Intimation System," in *2018 2nd International Conference on Electronics, Materials Engineering Nano-Technology (IEMENTech)*, p. 1–4, 2018.
- [6] Q. Nguyen, T. Luu, T. Tran, C. Nguyen, M. Tran, and H. Nguyen, "Collision Classification for Unmanned Surface Vehicle Using Inertial Measurement Unit Data," in *2024 2nd International Conference on Mechatronics, Control and Robotics (ICMCR)*, p. 11–15, 2024.
- [7] C. Ieracitano, N. Mammone, F. Spagnolo, F. Frustaci, S. Perri, P. Corsonello, and F. C. Morabito, "An Explainable Embedded Neural System for On-Board Ship Detection from Optical Satellite Imagery," *Engineering Applications of Artificial Intelligence*, vol. 133, p. 108517, 2024.
- [8] J. Jin, Z. Pang, J. Kua, Q. Zhu, K. H. Johansson, N. Marchenko, and D. Cavalcanti, "Cloud-Fog Automation: The New Paradigm towards Autonomous Industrial Cyber-Physical Systems," *IEEE Journal on Selected Areas in Communications*, p. 1–1, 2025.
- [9] H. Lyu, J. Yan, J. Zhang, Z. Pang, G. Yang, and A. J. Isaksson, "Cloud-Fog Automation: Heterogenous Applications Over New-Generation Infrastructure of Virtualized Computing and Converged Networks," *IEEE Industrial Electronics Magazine*, vol. 18, p. 30–42, 2024.
- [10] R. Mahmud, J. Jin, J. Kua, M. Afrin, S. Mistry, and A. Krishna, "Trusted Microservice Orchestration for Secure Edge Computing in Industrial Cyber-Physical Systems," *IEEE Network*, p. 1–1, 2025.
- [11] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge Computing: Vision and Challenges," *IEEE Internet of Things Journal*, vol. 3, p. 637–646, 2016.
- [12] J. Jin, K. Yu, J. Kua, N. Zhang, Z. Pang, and Q. an, "Cloud-Fog Automation: Vision, Enabling Technologies, and Future Research Directions," *IEEE Transactions on Industrial Informatics*, vol. 20, p. 1039–1054, 2024.
- [13] S. Liu, D. Lan, J. Wang, D. Hu, Z. Pang, and H. Lyu, "How Pretrained Foundation Models and Cloud-Fog Automation Empower the Recycling of Electrical Vehicles," in *2024 IEEE 22nd International Conference on Industrial Informatics (INDIN)*, p. 1–6, 2024.
- [14] J. Woo and N. Kim, "Collision Avoidance for an Unmanned Surface Vehicle Using Deep Reinforcement Learning," *Ocean Engineering*, vol. 199, p. 107001, 2020.
- [15] J. Liu, F. Zuo, Y. Guo, T. Li, and J. Chen, "Research on Improved Wavelet Convolutional Wavelet Neural Networks," *Applied Intelligence*, vol. 51, p. 4106–4126, 2021.
- [16] R. W. Liu, W. Yuan, X. Chen, and Y. Lu, "An Enhanced CNN-Enabled Learning Method for Promoting Ship Detection in Maritime Surveillance System," *Ocean Engineering*, vol. 235, p. 109435, 2021.
- [17] R. Skulstad, G. Li, T. I. Fossen, B. Vik, and H. Zhang, "A Hybrid Approach to Motion Prediction for Ship Docking-Integration of a Neural Network Model into the Ship Dynamic Model," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, p. 1–11, 2021.
- [18] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. Hen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, p. 4510–4520, 2018.
- [19] M. Ladegourdie and J. Kua, "Performance Analysis of OPC UA for Industrial Interoperability Towards Industry 4.0," *IoT*, vol. 3, no. 4, p. 507–525, 2022.
- [20] H. Cheng, J. Lian, and W. Jiao, "Enhanced MobileNet for Skin Cancer Image Classification with Fused Spatial Channel Attention Mechanism," *Scientific Reports*, vol. 14, no. 1, p. 28850, 2024.
- [21] M. Shin, M. Park, D. Oh, B. Kim, and J. Lee, "Clock Synchronization for One-Way Delay Measurement: A Survey," in *Advanced Communication and Networking*, p. 1–10, Springer Berlin Heidelberg, 2011.
- [22] H. Lv, H. Zhou, R. Wang, H. Wu, Z. Pang, and G. Yang, "Towards Intercontinental Teleoperation: A Cloud-Based Framework for Ultra-Remote Human-Robot Dual-Arm Motion Mapping," p. 132–144, Springer Nature Singapore, 2023.
- [23] T. Tran, Q. Nguyen, T. Luu, M. Tran, J. Kua, T. Hoang, and M. Dien, "Empowering Robotic Training with Kinesthetic Learning and Digital Twins in Human-Centric Industrial Systems," *Journal of Industrial Information Integration*, vol. 43, p. 100743, 2025.
- [24] Q. Nguyen, T. Tran, T. Luu, M. Tran, G. R. Emad, and T. Hoang, "Human-Centred Interfaces on Engineering Teleoperation: A Case Study of Utilising Virtual Reality (VR) with Robotic Systems," in *2024 7th International Conference on Information and Communications Technology (ICOIAC)*, p. 142–147, 2024.
- [25] T. Luu, Q. Nguyen, T. Tran, M. Q. Tran, S. Ding, J. Kua, and T. Hoang, "Effectiveness of IoT and VR for Real-Time Teleoperation of Industrial Robots," *Research Square*, 2024.
- [26] Z. Zhang, L. Wu, J. Jin, E. Wang, B. Liu, and Q. an, "Secure Federated Learning for Cloud-Fog Automation: Vulnerabilities, Challenges, Solutions, and Future Directions," *IEEE Transactions on Industrial Informatics*, p. 1–13, 2025.