COCE 2025

The Second International Conference on Technologies for Marine and Coastal Ecosystems

ISBN: 978-1-68558-329-3

October 26th – 30th, 2025

Barcelona, Spain

COCE 2025 Editors

Mahdi Abdelguerfi, University of New Orleans, USA

COCE 2025

Forward

The Second International Conference on Technologies for Marine and Coastal Ecosystems (COCE 2025), held between October 26th, 2025, and October 30th, 2025, in Barcelona, Spain, constituted a leading event dedicated to exploring and promoting innovative solutions and technologies for oceanic research and conservation. This conference served as a platform for collaboration among researchers, engineers, policymakers, and industry leaders, aiming to showcase cutting-edge advancements in smart ocean monitoring, data analytics, and sustainable ocean engineering.

COCE 2025 highlighted the transformative potential of technologies such as Artificial Intelligence (AI), Internet of Things (IoT), robotics, and data analytics in addressing critical challenges facing our oceans and marine ecosystems.

We take the opportunity to warmly thank all the members of the COCE 2025 technical program committee, as well as all the reviewers. The creation of such a high-quality conference program would not have been possible without their involvement. We also kindly thank all the authors who dedicated much of their time and effort to contribute to COCE 2025. We truly believe that, thanks to all these efforts, the final conference program consisted of top-quality contributions. We also thank the members of the COCE 2025 organizing committee for their help in handling the logistics of this event.

We hope that COCE 2025 was a successful international forum for the exchange of ideas and results between academia and industry for the promotion of progress of technologies for marine and coastal ecosystems.

COCE 2025 Chairs

COCE 2025 General Chairs
Jaime Lloret Mauri, Universitat Politecnica de Valencia, Spain
Petre Dini, IARIA, USA/EU

COCE 2025 Steering Committee
Cesar Bordehore, University of Alicante, Spain
Kiseon Kim, Gwangju Institute of Science and Technology, Korea
Eduardo Jorge Belda Perez, Universitat Politecnica de Valencia, Spain
Mahdi Abdelguerfi, The University of New Orleans, USA
Brandon Bethel University of The Bahamas, The Bahamas
Julian Simeonov, Ocean Sciences Division, U.S. Naval Research Laboratory, Stennis Space Center, USA
Alex Sheremet, University of Florida, USA
Kun Yang, Zhejiang Ocean University, China

COCE 2025
Committee

COCE 2025 General Chairs

Jaime Lloret Mauri, Universitat Politecnica de Valencia, Spain
Petre Dini, IARIA, USA/EU

COCE 2025 Steering Committee

Cesar Bordehore, University of Alicante, Spain
Kiseon Kim, Gwangju Institute of Science and Technology, Korea
Eduardo Jorge Belda Perez, Universitat Politecnica de Valencia, Spain
Mahdi Abdelguerfi, The University of New Orleans, USA
Brandon Bethel University of The Bahamas, The Bahamas
Julian Simeonov, Ocean Sciences Division, U.S. Naval Research Laboratory, Stennis Space Center, USA
Alex Sheremet, University of Florida, USA
Kun Yang, Zhejiang Ocean University, China

COCE 2025 Technical Program Committee

Mahdi Abdelguerfi, The University of New Orleans, USA
Lavanya Addepalli, Universitat Politecnica de Valencia, Spain
Chiguk Ahn, (Former Director and Vice President) Korea Institute of Aids to Navigation, Korea
Christhian N. Aldana, Universidad Nacional de Frontera, Peru
Eduardo Jorge Belda Perez, Universitat Politecnica de Valencia, Spain
Brandon Bethel, University of The Bahamas, The Bahamas
Cesar Bordehore, University of Alicante, Spain
Lismer Andres Caceres Najarro, Chosun University, Korea
Rafael Caparó, Universidad Nacional de Ingeniería, Peru
Bok Kyoung Choi, Korea Institute of Ocean Science and Technology (KIOST), Korea
Edson S. Gomi, University of São Paulo, Brazil
Kiseon Kim, Gwangju Institute of Science and Technology, Korea
Jeong-A Lee, Chosun University, Korea
Alvaro Liebana Carrascosa, Universitat Politecnica de Valencia, Spain
Satish R. Jondhale, Savitribai Phule Pune University, Pune, India
Alberto Ivars Palomares, Universitat Politecnica de Valencia, Spain
Carlos Antonio Puig Mengual, Universitat Politecnica de Valencia, Spain
Amine Rghioui, Université Ibn Zohr, Morocco
Juan A. Montiel-Nelson, University of Las Palmas de Gran Canaria | Institute for Applied Microelectronics, Spain
Minh Tuan Nguyen, Posts and Telecommunications Institute of Technology (PTIT), Vietnam
V. Pandiyaraju, VIT University, Chennai, India
Pujan Pokhrel, Amazon AWS & GulfSCEI, USA
Edwin Saavedra, Universidad Nacional de Frontera, Peru
Fermin Saavedra Cano, Universidad Nacional de Frontera, Peru

Yesenia Saavedra, Universidad Nacional de Frontera, Peru
María-Teresa Sebastiá-Frasquet, Universitat Politecnica de Valencia, Spain
Alex Sheremet, University of Florida, USA
Vinie Lee Silva Alvarado, Universitat Politecnica de Valencia, Spain
Julian Simeonov, Ocean Sciences Division, U.S. Naval Research Laboratory, Stennis Space Center, USA
Norsuzila Ya'acob, Universiti Teknologi MARA, Malaysia
Chan-su Yang, Korea Institute of Ocean Science and Technology (KIOST), Korea
Vincent Xiaochuan Yu, University of New Orleans, USA

Copyright Information

Table of Contents

Evaluating Hyperparameter Selection Techniques for the Ratio-Coupled Loss Function

Austin B. Schmidt
GulfSCEI
University of New Orleans
New Orleans, United States
email: sbaustin@uno.edu

Pujan Pokhrel
GulfSCEI
University of New Orleans
New Orleans, United States
email: ppokhre1@uno.edu

Md Meftahul Ferdaus
GulfSCEI
University of New Orleans
New Orleans, United States
email: mferdaus@uno.edu

Elias Ioup
Center for Geospatial Sciences
Naval Research Laboratory
John C. Stennis Space Center, United States
email: elias.ioup@nrlssc.navy.mil

Mahdi Abdelguerfi
GulfSCEI
University of New Orleans
New Orleans, United States
email: gulfsceidirector@uno.edu

Julian Simeonov
Marine Geosciences Division
Naval Research Laboratory
John C. Stennis Space Center, United States
email: julian.a.simeonov.civ@us.navy.mil

***Abstract*—When forecasting fixed-location observation nodes with statistical surrogate models, combining datasets during training loss calculation has been shown to improve model accuracy. However, traditional methods for tuning the data ratio, such as grid search or random search, are computationally prohibitive. An alternative online methodology for optimizing the data during training has been previously investigated. While both approaches have been independently validated, they have never been directly compared to each other or to other search techniques. This paper presents a direct comparison to evaluate whether the online approach can serve as a viable replacement for conventional search methods. The Cahn-Hilliard physical equation provides a controlled testing environment for this analysis. The results show that the optimization algorithm may require additional improvements before an out-of-the-box approach is appropriate. However, using the derived optimal hyperparameter in an offline setup provides an improvement in accuracy, which implies the methodology is worthwhile when under time constraints.**

Keywords-Ratio-Coupled Loss; Surrogate Model; Hyperparameter Tuning; Cahn-Hilliard.

I. Introduction

Fixed-location forecasting helps fill data gaps in ocean buoys, improve weather station predictions, and reduce uncertainty in tsunami detection [1][2][3]. Expanding the capability of machine learning models to forecast fixed-location time series is therefore an interesting and challenging problem. In the case of Partial Differential Equations (PDEs) and numerical models, machine learning surrogates provide an efficient alternative to direct numerical simulations, particularly for complex oceanographic tasks, such as fluid flow modeling [4]. Surrogate models serve as computationally efficient approximators, allowing for rapid inference without the high cost of solving PDEs from first principles. This is especially valuable in scenarios where real-time forecasting is required or when computational resources are constrained. The Cahn-Hilliard equation is investigated in this study as a controlled test case for evaluating surrogate modeling approaches. This equation describes phase separation processes and serves as a benchmark for studying non-linear PDE behavior, making it a suitable candidate for testing the efficacy of the proposed methodologies. Improvements to surrogate modeling of the Cahn-Hilliard system have potential implications for broader applications in oceanographic and geophysical flow models [5].

Combining data sources in machine learning often improves model performance across various contexts. From a data perspective, representations of physical phenomena are inherently flawed as they are only approximations of underlying behaviors. Sensors are known to be noisy and have measurable errors [6]. Similarly, numerically modeled data can include discretization errors or miscalculations from nonlinear interactions [7]. Therefore, finding ways to combine multiple sources of training data improves model stability and robustness to outlier data. Combining data within the loss function of a model through a ratio of error is shown to be particularly effective [8]. An importance-weighting hyperparameter controls error flow, allowing models to adapt to either data source. Selecting the best hyperparameter has been repeatedly shown to be a principal challenge with this methodology [8][9][10]. Therefore, the main novelty in this work comes from the comparison of multiple hyperparameter techniques to the convex ratio-loss function identified in [11]. In that work, an online algorithm to select the best value of a ratio-inducing hyperparameter was proposed. The paper focused on optimization mathematics and the impact of noise levels on model convergence. To establish its usefulness in modeling PDEs and real-world data, a benchmarking study must be conducted to compare against similar ratio-coupled loss. Accordingly, this work makes the following contributions.

- Hyperparameter search techniques are evaluated to identify the most effective method for error reduction and computational efficiency.
- The convex ratio-coupled loss function is compared to its non-convex counterpart to assess its impact on model accuracy.

Copyright (c) IARIA, 2025. ISBN: 978-1-68558-329-3

- A surrogate model for the Cahn-Hilliard equation is developed to demonstrate the feasibility of surrogate modeling for nonlinear PDEs.

The paper is organized as follows: Section II reviews related work, highlighting comparable research and contrasting it with this study's objectives. Section III details the methodology. Section IV presents experimental results and their implications. Finally, Section V summarizes key contributions and outlines future directions.

II. Related Work

The Cahn-Hilliard equation originally modeled the phase separation process of binary alloys [12]. In modern research, Cahn-Hilliard equations have uses spanning from problems in material sciences to fluid dynamics [13]. In the context of oceanographic modeling, Cahn-Hilliard formulations have been coupled with the Navier-Stokes equations [5]. In those cases, Navier-Stokes governs fluid velocities while Cahn-Hilliard handles the relative density of fluid atoms. The work conducted in this paper proposes a methodology for forecasting fixed-location Cahn-Hilliard observations points. Future extensions of the work can eventually lead to a coupled environment to better model fluid mixtures.

The combination of varying data sources when modeling is seen in various contexts. Data assimilation improves analysis of physical systems and is recently combined with deep learning models [14][15][16]. Physics-Informed Neural Networks (PINNs) integrate training data with governing equations to enhance convergence and model robustness [17]. They have also been applied to solving Cahn-Hilliard equations with backward-compatible PINNs and adaptive-sampling PINNs [18][19]. Data can also be directly combined as input data from multiple observed and numerical sources [20][21]. The ratio-coupled loss function in this work combines data in the loss function during the training phase, like PINN models. However, the data is collected ahead of time and a ratio of loss from each source is used to regularize the training process [8]. This method is simpler than implementing PINN models as it does not require direct physical knowledge. It is also more flexible than using multiple input variables since additional data is only needed at training time.

Statistical models are frequently used as surrogate models for numerically derived and observed data. For example, Transformers are used to model significant wave heights observed by free floating buoys [22]. Recurrent neural networks are also a valid choice as they can manage long-term dependencies. The Long-Short Term Memory (LSTM) unit is the recurrent unit type featured in this work. LSTM units are used when forecasting many observed ocean parameters, such as sea surface temperature, salinity, significant wave heights, and others [8][10][23]. LSTMs are used in numerical surrogate contexts for modeling epidemic spread, turbulent flows derived from Navier-Stokes, and fluid-particle systems [24][25][26]. The LSTM unit is also used to model PDEs directly [11]. For example, mixed LSTM and convolution layers were used for pattern discovery to model the Cahn-Hilliard equation, achieving good agreement [27].

The investigation of hyperparameter search methodologies is inspired by common problems identified in other ratio-coupled loss research. Similar research uses a single λ parameter with grid search to find the most performant hyperparameter [8][10]. However, both works mention that long training times make the λ selection process difficult. This problem is exacerbated when using the multiple-λ ratio-coupled loss function [9]. As the number of hyperparameters combinations intractably increased, a bounded random search was used to explore the search space. A method for selecting the optimal hyperparameter with an online algorithm was finally proposed in a setting like the one investigated in this work [11]. However, this method was never validated against other search methodologies, focusing instead on the optimization problem itself. This paper extends the ideas in those works by comparing the optimized method with other search techniques.

III. Methodology

The following section describes all major techniques used to support the major claims. Within the section, the Cahn-Hilliard dataset details are outlined, the ratio-coupled loss function is detailed, the hyperparameter search techniques are compared, and the deep learning architecture used is described. The validation parameters and all experimental details are provided for reproduction.

A. Cahn-Hilliard Equation

In this work, the Cahn-Hilliard equation is used to model the concentration of binary fluids as they separate over time. The spontaneous phase separation of each fluid is demonstrated over 100 evolution steps and used to train surrogate models. The mathematical definition of the simple Cahn-Hilliard equation used to generate the training and testing data follows as,

$$\partial_t c = \nabla^2 \left(c^3 - c - \gamma \nabla^2 c \right), \tag{1}$$

where c is a scalar field taking values on the interval $[-1, 1]$ and γ sets the squared interfacial width. In the simulations used for this work, $\gamma = 1.0$. The implementation of this equation is given by the Python package py-pde [28]. The equation is evolved on a 20×20 sized grid with a random initialization of values on the interval $[-1, 1]$ for 100 time steps. The separation of fluid concentrations over the entire evolution period is seen in Figure 1. The figure displays how the random initialization dynamically splits into pure domains over the 100 epochs.

Along the evolution of the grid, Figure 1 also displays the target testing nodes. These nodes indicate the 80 fixed-location observation nodes reserved for final testing. The entire 20 × 20 grid yields 400 available observation points. Then, 350 of these points are randomly reserved yielding an 87.5% data coverage. Subsequently, 100 of those points are reserved as observation nodes for testing and validation data, at an 80/20 split. When modeling with machine learning algorithms, the amount of data is responsible for how well the surrogate model

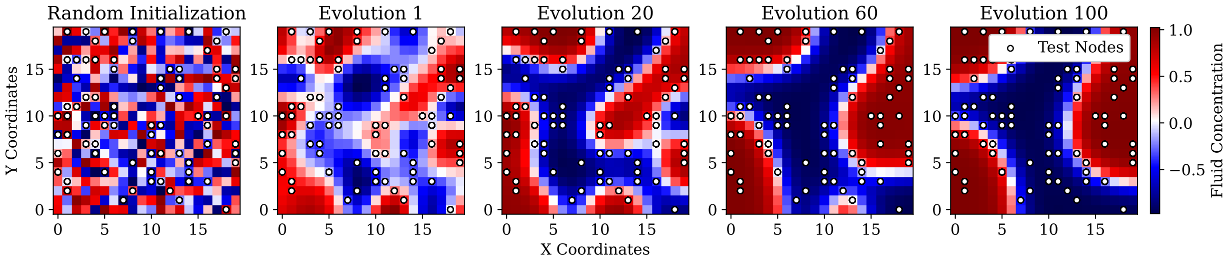


Figure 1. Evolution of the Cahn-Hilliard equation over 100 evolutions. Displays the locations of the 80 reserved observation testing nodes.

can generalize learned behaviors. A lower percentage of data coverage would negatively affect overall model accuracy.

To represent multiple sources of imperfect data, one final processing step is applied to the time series. First, the data is split into three separate instances. One instance represents noisy observations of local conditions and Gaussian noise is applied to the data at a fixed standard deviation rate of 0.01. This subset of data is used as the input to the model and when calculating error in the training loop. A second instance of the data represents larger inaccuracies from an additional, secondary source of data and has an applied Gaussian standard deviation of 0.1. This second source of data is used as a source of regularization when using the ratio-coupled loss function described in the upcoming section. The final instance of the data has no Gaussian noise applied to it and is used to calculate the final metrics of each model. This methodology follows the experimental setup in [11], which first introduced the optimized hyperparameter search explored in this work.

B. Ratio-Coupled Loss Function

A ratio-coupled loss function allows for two sources of data to be used for each feature when training a model. The hyperparameter λ controls the ratio of error generated when compared to either data source. The set of coupled features is defined as $S \subseteq \{1, \ldots, d\}$ where $d \in \mathbb{N}$. The cost function is formally introduced in (2)-(4) as,

$$\Delta_1 = g(\hat{y}, y_o), \tag{2}$$

$$\Delta_2 = g(\hat{y}, y_m), \tag{3}$$

$$\Omega_{\text{ratio-coupled loss}} = \lambda * \Delta_1 + (1 - \lambda) * \Delta_2. \tag{4}$$

In (2), the predicted value and the reserved values y_o are used to generate the first error term. Similarly, the same score is calculated for (3) by comparing the prediction and the secondary data source y_m. In the context of this work, y_o represents the reserved data with Gaussian noise of 0.01, y_m represents the reserved data with Gaussian noise of 0.1, and $\hat{y}$ represents the predicted output from the surrogate model. Therefore, the two Δ terms defined in (2) and (3) represent the error between the prediction and each of the dual sources as calculated with g. In this case, the error formulation g is Huber loss,

$$g_\delta(a) = \begin{cases} \frac{1}{2}a^2 & \text{if } |a| \leq \delta, \\ \delta(|a| - \frac{1}{2}\delta) & \text{if } |a| > \delta, \end{cases},$$

where a is the residual between predicted values and $\delta = 1.35$. Huber loss is used in this context to prevent model underfitting by penalizing inferences close to zero, which is the mean value across the Cahn-Hilliard formulation. The error terms are finally weighted by the hyperparameter λ as outlined in (4). The coupled feature loss is characterized by the ratio of the two Δ values that measure predicted error across multiple sources of truth. So, the selected λ value represents a ratio to determine the importance provided by either source. The hyperparameter is constrained to a ratio such that $\lambda \in [0.0, 1.0]$. The ratio of data that produces the most performant model is unknown and must be tuned by finding an optimal value of λ.

C. Hyperparameter Search Techniques

Four hyperparameter search techniques are compared in this work to validate whether the optimized λ technique is viable for improving the modeling of nonlinear fluid dynamic equations. To this end, a simple grid search, a random search, a Bayesian search, and the highlighted online optimized search are implemented for validation.

1) Grid Search: The grid search technique uses a range of λ hyperparameter values at fixed intervals. Each of these values is statically used to train a model. After each selection of λ is used, the results are compared and the best λ value is determined. When using a ratio-coupled loss function, $\lambda \in [0, 1]$, so the grid search is constrained to this interval. Given a fixed step size of 0.1, the grid search evaluates the hyperparameter values $\lambda \in \{0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0\}$. A smaller step size can be selected, although this may be computationally expensive in cases where the amount of data or model size is large.

2) Random Search: A random search explores the hyperparameter space by randomly selecting hyperparameters at a desired resolution for a fixed number of values. This technique is not guaranteed to select an optimal hyperparameter but can give good coverage on a variety of λ values, given enough iterations. In the case of this experiment, 20 random λ values are selected from a uniform distribution with a precision of 0.01.

3) Bayesian Search: Bayesian optimization is a popular hyperparameter search technique in deep learning. The optimization is used to select the minimizing hyperparameter based on a specified criterion [29]. Therefore, the formulation is set up as,

$$\lambda = \arg \min_{\lambda \in [0,1]} f(\lambda). \tag{5}$$

The search space is defined such that $\lambda \in [0.0, 1.0]$. The criterion function $f(\lambda)$ is defined as the evaluation of a partially trained surrogate when making predictions on the validation dataset. That is, the error value on the validation data should be minimized by the selected λ. Continuing, Bayesian optimization is derived from Bayes' theorem. Given evidence data E, the posterior probability $P(M|E)$ of a model M is proportional to the likelihood $P(E|M)$ of observing E given model M, multiplied by the prior probability $P(M)$. This is expressed as,

$$P(M|E) \propto P(E|M)P(M). \tag{6}$$

The evidence and models terms expressed in (6) are the loss calculation and surrogate trained using a specific λ, respectively. Using this formulation, Bayes optimization iteratively searches the λ space for an optimal value. The specific implementation of the Bayesian optimization used in this work comes from the Python library GPyOpt [30]. The optimization algorithm selects the most performant λ over multiple trial iterations. Each candidate model is trained for 50 epochs, to save computational resources. There is an initial selection of eight random λ values and subsequent values are selected based on the expected improvement that a new λ value will provide. The optimization algorithm is run for a total of 30 iterations before selecting the λ that minimizes the objective function, $f(\lambda)$. Upon the selection of a minimizing λ value, a final model is trained using the full number of epochs for final evaluation.

4) Optimized Search: The final technique compared in this case study is the online optimized λ search that was first explored in [11]. The essential idea is that while the model weights are being trained, the λ hyperparameter is slowly optimized to the value that minimizes the loss function. To allow for direct optimization of the λ value, a modification must be made to the loss formulation described in (4). To make the function differentiable with respect to λ, square terms are added around each ratio term. Therefore, the convex ratio-coupled loss is defined as,

$$\Omega_{\text{convex ratio-coupled loss}} = (\lambda * \Delta_1)^2 + ((1-\lambda) * \Delta_2)^2. \tag{7}$$

The optimized value of the convex loss function is constrained such that the minimal value is guaranteed to exist when $\lambda \in [0.0, 1.0]$, which is one benefit of this method. Also, optimizing the λ simultaneously with model weights means that lengthy tuning times are reduced to the training of a single model. Any optimization scheme may be used, but the Adam optimizer and TensorFlow's gradient tape implementation allows λ to be easily optimized as the model is trained. However, it should be noted that by changing the loss function formulation, the model weights may not converge as they would with the basic coupled loss function. This crucial point is the motivation to compare the optimized search technique with other methodologies.

D. Machine Learning Model Architecture

Machine learning architecture is static among all experimental hyperparameter search types. The full architecture is displayed in Table I and is made up of five main layers and an input layer. The input to the model is a vector of four features, which include the current fluid concentration, the X and Y coordinates, and the timestep being modeled. Following, the LSTM unit is used in the next four layers. LSTM units are recurrent layers that contain input, forget, and update gates, which aid in learning time series dependencies [23]. Although the time horizon for a single forecast step will be one, the LSTM layers use additional parameters, making them a viable choice for surrogate modeling. Each of the LSTM layers uses the hyperbolic tangent activation function, allowing layer outputs in the range $[-1, 1]$. The final layer is a simple densely connected layer with a linear activation, to output the predicted fluid concentration in the next timestep. Dropout layers are placed between each LSTM layer with a dropout value of 0.2. These layers randomly drop weights during the training phase to help prevent overfitting of the model. Finally, the Adam optimization function is used to optimize the model weights.

TABLE I. LSTM MODEL ARCHITECTURE BY LAYER. THE TOTAL NUMBER OF TRAINABLE PARAMETERS IS 526,241. N REPRESENTS THE BATCH SIZE.

Layer Type	Shape	Parameters	Activation
Input Layer	(N, 4, 1)	0	None
Reshape	(N, 1, 4)	0	None
LSTM	(N, 1, 256)	267,264	Tanh
Dropout	(N, 1, 256)	0	None
LSTM	(N, 1, 128)	197,120	Tanh
Dropout	(N, 1, 128)	0	None
LSTM	(N, 1, 64)	49,408	Tanh
Dropout	(N, 1, 64)	0	None
LSTM	(N, 1, 32)	12,416	Tanh
Dense	(N, 1)	33	Linear

Before training, all input data is normalized with respect to the training data, including the testing data. To analyze the results, all data is transformed back to the original scale. When using the model to make predictions over multiple time horizons, a rolling forecast methodology is used. That is, only the first forecast uses fresh sensor values. The consecutive forecasts use the predicted fluid concentration as an input to the next prediction. This continues until the entire horizon has been predicted. Only then are fresh observation node values provided again. To improve model stability over multiple horizons, the same methodology is used when training the model. The loss function is summed over multiple predictions in a rolling-style forecast and the accumulated error is used to back propagate the model weights. To improve the model stability for rolling forecasts, the model accumulates training loss in the same way. That is, given a horizon size of eight, the training loop accumulates loss over eight steps while using the model's prediction as input values. This method of training supports long term predictions in the model by improving forecast stability.

E. Validation Metrics

There are two main metrics considered in this work. The first is the average Root Mean Square Error (RMSE) across all observational nodes reserved for testing. RMSE is preferred over other error scores because it highly penalizes large deviations. The RMSE formulation is defined as $\sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}$. The parameter N is the number of test samples, y is the ground truth, and $\hat{y}$ is the prediction vector. Another method for analyzing the results is the time taken in seconds for each hyperparameter search to produce its most optimal result. Therefore, the best method should balance between minimizing the RMSE as well as requiring the least amount of time to produce those results.

IV. Results

After running each of the hyperparameter searches, 34 test cases were considered. By training each model with the same initial seed, only differences caused by varying λ values affect model convergence. Therefore, the effectiveness of each search methodology is considered by ranking the RMSE values over the full 12-step prediction horizon. The top ten results discovered are compared in Table II. In the table, it is noted that the top result uses the optimized λ search, with one major caveat. The online algorithm did not produce an optimal model by itself. Using the λ value derived from the online algorithm as a static λ value worked very well. In that case, the top result highlights an optimized λ in a static training environment. This is significant as it shows the optimized algorithm may not be suitable for directly training a model. However, this suggests it may be suitable for estimating the most performant λ overall. The grid and random searches both found values of λ that would be considered suitable. That is, they reduced the error beyond that found when no ratio-coupled regularization is used ($\lambda = 1.0$). Finally, the Bayesian search yielded interesting results by estimating a λ value that produced the lowest single horizon step RMSE overall. It seems reasonable to suggest that a longer training time per iteration of the Bayes search would result in an optimal λ value, like those found in the top four results.

TABLE II. Top 10 results sorted by the calculated RMSE over the full forecast horizon.

Rank	λ Search	λ Value	Full Horizon RMSE	Single Step RMSE
1	Optimized*	0.95961833	0.045182	0.013879
2	Grid	0.9	0.045754	0.014290
3	Random	0.89	0.045981	0.014298
4	Random	0.95	0.046696	0.014519
5	Grid	1.0	0.047465	0.014367
6	Bayes	0.7319939	0.047566	0.013782
7	Random	0.79	0.048094	0.014380
8	Grid	0.7	0.048827	0.014427
9	Random	0.68	0.048832	0.014167
10	Grid	0.8	0.048988	0.014345

Following, consider the top performing result for each of the λ search techniques in Table III. In this table, the total time taken for each search algorithm to provide the most performant result is given. It is notable that grid search and random search both linearly increase with each test case considered. Given a high number of tested λ values, increasingly better results can be found. However, this becomes prohibitively expensive as the resolution of the hyperparameter increases. Comparatively, the Bayesian search takes little time to explore the hyperparameter space. This is because only 50 epochs are used when searching for candidate values, which would require approximately 5% of the training time. Since the results are worse, it is likely that a higher number of epochs could balance time taken and the RMSE score. In the case of the optimized search, it takes exactly one training cycle of about 1,500 seconds to train the model and λ value. Given the inferior performance, the trade-off of time taken to RMSE is not very impressive. Encouragingly, using the optimized λ with the traditional ratio-coupled loss and a static training setup gives the best overall results, and it only requires enough time to train two models.

TABLE III. Comparison of the best results for each search technique.

Rank	λ Search	λ Value	Full Horizon RMSE	Single Step RMSE	Total Time Taken ($\approx$ s)
1	Optimized*	0.95961833	0.045182	0.013879	3,109
2	Grid	0.9	0.045754	0.014290	18,701
3	Random	0.89	0.045981	0.014298	30,644
7	Bayes	0.7319939	0.047566	0.013782	5,344
34	Optimized	0.95961833	0.055039	0.014998	1,582

An example observational node forecast is given in Figure 2. This example displays the inferences generated by the best performing models outlined in Table III. Every 12 forecast steps, the model is provided with new observation values, as denoted by the refresh points in the figure. The model does well at matching the fluid concentration over time, especially after the first 20 evolutions. This is in part because of the high coverage of training data as well as the rolling forecast implemented in the training algorithm. The best performing models mostly agree on forecasts. Observed differences are mainly in how close they fit to the actual curve. It is notable that there are 80 different observation nodes and each of them behaves differently, depending on how the Cahn-Hilliard equation evolves over time, as seen in Figure 1. Observation nodes that monitor unstable regions are notoriously more difficult to model in the long term.

Lastly, consider how the average error changes over the Cahn-Hilliard evolution in Figure 3. Comparing the single step error (left) and the 12-step error (right) shows two main behaviors. First, the instability early has the highest rate of error. It is difficult for the models to generalize how the evolution of a chaotic and randomly initialized system will begin to separate. Following, Cahn-Hilliard has two main phases, which are simpler to model. Either the fluid concentration has already separated into a stable group or is transitioning into a stable group. Both behaviors are more easily modeled. Finally, error seemingly rises by the end of the evolution period. This is attributed to the fact that some locations complete the phase separation process, resulting in some binary regions disappearing. Consider evolutions 60 through 100 in Figure

Copyright (c) IARIA, 2025. ISBN: 978-1-68558-329-3

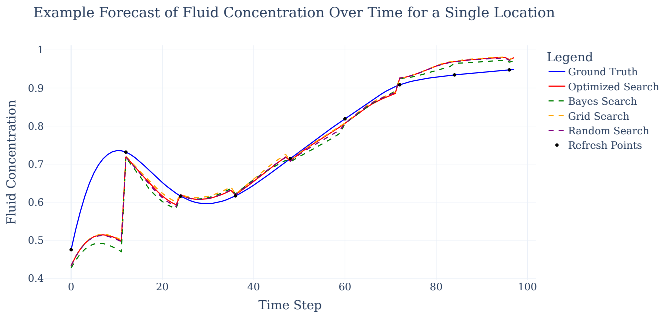


Figure 2. Example plot of a single observation node with a 12-step forecast horizon. Fresh initial values are seen every twelfth step.

1 as an example. The most performant λ values consistently seem to come from the Bayesian search λ and the optimized static λ. Surprisingly, Bayesian search performed well in general but suffered the most when initial conditions were chaotic.

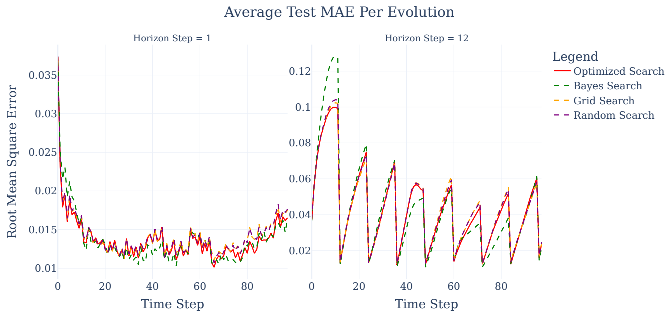


Figure 3. Evolution of average error over the evolution period. The single horizon error (left) and the full horizon error (right) highlight when the forecast problem is most difficult for each selected model.

Overall, each search methodology was compared based on the RMSE and time taken to produce its most optimal result. In the case of the optimized search, the resulting model performed poorly. However, the λ value produced from the algorithm yielded best results overall. This shows potential for estimating data-specific hyperparameters in a two-stage process. Although, if this methodology is to be useful in a wide range of tasks, formal investigations into methods for improving the online training approach are necessary. Through the investigation of a dataset distorted with Gaussian data, an obvious question arises. It is currently unknown whether other distributions of noise are optimizable in this way. Further investigations on real-world data and data distorted with other noise distributions should be validated against.

When considering the other search methodologies, Bayesian search also showed promise, given enough training epochs in the investigative phase. Grid search and random search are both valid methods for λ discovery, given enough time to explore enough model iterations. Overall, the best models performed very similarly, with minor variations in performance. This is to be expected when using the exact same model architecture. However, the main differences are seen when comparing the computational and time resources required to find the best result.

The Cahn-Hilliard equation is a good test bed for understanding how non-linear PDEs can be solved using surrogate modeling methods. Given the interesting initial results, a more generalized surrogate that can model Cahn-Hilliard under varying initial and forcing conditions is an interesting challenge. Future work that focuses on the coupling of these forecasts with Navier-Stokes to improve wave modeling, is a natural goal.

V. Conclusion and Future Work

In this work, surrogate models were investigated to forecast observation nodes of a Cahn-Hilliard equation with a random initialization. The models were trained using the ratio-coupled loss function and a rolling forecast-style training procedure. A selection of common hyperparameter search methodologies were compared to an online hyperparameter tuning algorithm. This algorithm adds quadratic terms to the ratio-coupled loss, to make the hyperparameter optimizable. Given varying amounts of noise added to two datasets, the search algorithms identified values for λ that were most performant. In all variations of the experiments, the average RMSE of all test cases and time taken to find performant hyperparameters were used to determine the best results. The comparison of search methodologies revealed that while the optimized search performed poorly overall, the λ value it produced led to the best results, highlighting the potential for a two-stage hyperparameter estimation process. Bayesian search also showed promise with sufficient training epochs, while grid and random searches remained valid given enough computational resources.

Future work will focus on three main identified weak points. Most importantly, investigations into improving the convex ratio-coupled loss function should be explored. Early stopping mechanisms to provide a stable training environment can be considered. Also, using the ratio-coupled loss function for model weights while the convex function for online tuning of the λ values might work in some situations. Next, the optimization algorithm should be validated on other datasets. In this case the models were not trained on different initializations of Cahn-Hilliard, which leads to poor generalization. Noise across varying data sources is not necessarily Gaussian, so future work would benefit from implementing different distributions of noise. Similarly, the methodology should be validating using real world data. Lastly, more advanced model architecture should be considered to understand how well the ratio-coupled loss function reacts when using more sophisticated models. Overall, there is a wide range of research directions to consider in the future.

References

[1] S. S. Kolukula and P. Murty, “Enhancing observations data: A machine-learning approach to fill gaps in the moored buoy data,” *Results in Engineering*, p. 104 708, 2025.

[2] D. S. Roy, “Forecasting the air temperature at a weather station using deep neural networks,” *Procedia computer science*, vol. 178, pp. 38–46, 2020.

[3] M. Angove *et al.*, “Ocean observations required to minimize uncertainty in global tsunami forecasts, warnings, and emergency response,” *Frontiers in Marine Science*, vol. 6, p. 350, 2019.

[4] L. Sun, H. Gao, S. Pan, and J.-X. Wang, “Surrogate modeling for fluid flows based on physics-constrained deep learning without simulation data,” *Computer Methods in Applied Mechanics and Engineering*, vol. 361, p. 112 732, 2020.

[5] F. Boyer, C. Lapuerta, S. Minjeaud, B. Piar, and M. Quintard, “Cahn–hilliard/navier–stokes model for the simulation of three-phase flows,” *Transport in Porous Media*, vol. 82, pp. 463–483, 2010.

[6] W. L. Oberkampf, S. M. DeLand, B. M. Rutherford, K. V. Diegert, and K. F. Alvin, “Error and uncertainty in modeling and simulation,” *Reliability Engineering & System Safety*, vol. 75, no. 3, pp. 333–357, 2002.

[7] H. Y. Teh, A. W. Kempa-Liehr, and K. I.-K. Wang, “Sensor data quality: A systematic review,” *Journal of Big Data*, vol. 7, no. 1, p. 11, 2020.

[8] A. B. Schmidt, P. Pokhrel, M. Abdelguerfi, E. Ioup, and D. Dobson, “Forecasting buoy observations using physics-informed neural networks,” *IEEE Journal of Oceanic Engineering*, pp. 1–20, 2024. DOI: 10.1109/JOE.2024.3378408.

[9] A. B. Schmidt *et al.*, “Physics-regularized buoy forecasts: A multi-hyperparameter approach using bounded random search,” in *COCE 2024: The First International Conference on Technologies for Marine and Coastal Ecosystems*, ThinkMind Digital Library, 2024.

[10] E. Sandner *et al.*, “A multiple-location modeling scheme for physics-regularized networks: Recurrent forecasting of fixed-location buoy observations,” *COCE 2024: The First International Conference on Technologies for Marine and Coastal Ecosystems*, 2024.

[11] A. B. Schmidt, P. Pokhrel, M. Abdelguerfi, E. Ioup, and D. Dobson, “An algorithm for modelling differential processes utilising a ratio-coupled loss,” *TechRxiv*, 2024.

[12] J. Kim, S. Lee, Y. Choi, S.-M. Lee, and D. Jeong, “Basic principles and practical applications of the cahn–hilliard equation,” *Mathematical Problems in Engineering*, vol. 2016, no. 1, p. 9 532 608, 2016.

[13] J. Shen and X. Yang, “Numerical approximations of allen-cahn and cahn-hilliard equations,” *Discrete Contin. Dyn. Syst*, vol. 28, no. 4, pp. 1669–1691, 2010.

[14] P. Pokhrel, M. Abdelguerfi, and E. Ioup, “A machine-learning and data assimilation forecasting framework for surface waves,” *Quarterly Journal of the Royal Meteorological Society*, vol. 150, no. 759, pp. 958–975, 2024.

[15] J. Brajard, A. Carrassi, M. Bocquet, and L. Bertino, “Combining data assimilation and machine learning to emulate a dynamical model from sparse and noisy observations: A case study with the lorenz 96 model,” *Journal of computational science*, vol. 44, p. 101 171, 2020.

[16] M. Adrian, D. Sanz-Alonso, and R. Willett, “Data assimilation with machine learning surrogate models: A case study with fourcastnet,” *arXiv preprint arXiv:2405.13180*, 2024.

[17] M. Raissi, P. Perdikaris, and G. E. Karniadakis, “Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations,” *Journal of Computational physics*, vol. 378, pp. 686–707, 2019.

[18] C. L. Wight and J. Zhao, “Solving allen-cahn and cahn-hilliard equations using the adaptive physics informed neural networks,” *arXiv preprint arXiv:2007.04542*, 2020.

[19] R. Mattey and S. Ghosh, “A novel sequential method to train physics informed neural networks for allen cahn and cahn hilliard equations,” *Computer Methods in Applied Mechanics and Engineering*, vol. 390, p. 114 474, 2022.

[20] E. Jang, Y. J. Kim, J. Im, and Y.-G. Park, “Improvement of smap sea surface salinity in river-dominated oceans using machine learning approaches,” *GIScience & Remote Sensing*, vol. 58, no. 1, pp. 138–160, 2021.

[21] G. Ibarra-Berastegi *et al.*, “Wave energy forecasting at three coastal buoys in the bay of biscay,” *IEEE Journal of Oceanic Engineering*, vol. 41, no. 4, pp. 923–929, 2016.

[22] P. Pokhrel, E. Ioup, J. Simeonov, M. T. Hoque, and M. Abdelguerfi, “A transformer-based regression scheme for forecasting significant wave heights in oceans,” *IEEE Journal of Oceanic Engineering*, vol. 47, no. 4, pp. 1010–1023, 2022. DOI: 10.1109/JOE.2022.3173454.

[23] Q. Zhang, H. Wang, J. Dong, G. Zhong, and X. Sun, “Prediction of sea surface temperature using long short-term memory,” *IEEE geoscience and remote sensing letters*, vol. 14, no. 10, pp. 1745–1749, 2017.

[24] D. A. David, L. Street, S. Ramakrishnan, and M. Kumar, “Lstm-based estimation of time-varying parameters in a spatiotemporal pde model for prediction of epidemic spread,” *IFAC-PapersOnLine*, vol. 58, no. 28, pp. 468–473, 2024.

[25] A. T. Mohan and D. V. Gaitonde, “A deep learning based approach to reduced order modeling for turbulent flow control using lstm neural networks,” *arXiv preprint arXiv:1804.09269*, 2018.

[26] A. Hajisharifi *et al.*, “An lstm-enhanced surrogate model to simulate the dynamics of particle-laden fluid systems,” *Computers & Fluids*, vol. 280, p. 106 361, 2024.

[27] E. Kiyani, S. Silber, M. Kooshkbaghi, and M. Karttunen, “Machine-learning-based data-driven discovery of nonlinear phase-field dynamics,” *Physical Review E*, vol. 106, no. 6, p. 065 303, 2022.

[28] D. Zwicker, “Py-pde: A python package for solving partial differential equations,” *Journal of Open Source Software*, vol. 5, no. 48, p. 2158, 2020. DOI: 10.21105/joss.02158.

[29] J. Wu *et al.*, “Hyperparameter optimization for machine learning models based on bayesian optimization,” *Journal of Electronic Science and Technology*, vol. 17, no. 1, pp. 26–40, 2019.

[30] GPyOpt Project, *Gpyopt: A bayesian optimization framework in python*, http://github.com/SheffieldML/GPyOpt, [retrieved: September, 2025], 2016.

Multi-Source Constrained Machine Learning for Oceanic Parameters Forecasting

Pujan Pokhrel
GulfSCEI,
University of New Orleans
New Orleans, United States
email: ppokhre1@uno.edu

Austin B. Schmidt
GulfSCEI,
University of New Orleans
New Orleans, United States
email: sbaustin@uno.edu

Md Meftahul Ferdaus
GulfSCEI,
University of New Orleans
New Orleans, United States
email: mferdaus@uno.edu

Elias Ioup
Center for Geospatial Sciences
Naval Research Laboratory
Mississippi, United States
email: elias.z.ioup.civ@us.navy.mil

Julian Simeonov
Sediment Dynamics Section
Naval Research Laboratory
Mississippi, United States
email: julian.a.simeonov.civ@us.navy.mil

Mahdi Abdelguerfi
GulfSCEI,
University of New Orleans
New Orleans, United States
email: gulfsceidirector@uno.edu

***Abstract*—This study evaluates multi-source fusion techniques for environmental forecasting, focusing on their effectiveness in predicting oceanic and atmospheric variables. Three neural network architectures are examined: a baseline Long Short Term Memory (LSTM) model, a Softmax Fusion model, and a Lagrangian Fusion model. A central component of the approach is the incorporation of physics-based constraints during training to ensure physically consistent predictions. Results based on Root Mean Squared Error (RMSE) indicate that fusion-based models consistently outperform the baseline for wave-related and thermodynamic variables such as air and water temperature. RMSE reductions for these variables range from approximately 5% to over 40%, driven by the models' ability to enforce spatiotemporal smoothness and reduce spatial variability. In contrast, wind components show higher RMSE in fusion models, highlighting a trade-off between global physical consistency and the accurate representation of localized, high-variance wind phenomena. These findings demonstrate the advantages of fusion architectures for improving buoy-based wave and thermodynamic forecasts, while suggesting that future work on wind predictions may benefit from adaptive regularization or hybrid loss functions to capture both global coherence and local detail better.**

Keywords-multi-source fusion, physics-informed neural networks, constrained machine learning.

I. INTRODUCTION

Accurate short-term forecasting of oceanic and atmospheric variables is important for a wide range of applications, including maritime navigation, coastal hazard mitigation, and climate monitoring [1][2]. Traditional Numerical Weather Prediction (NWP) uses partial differential equations based on physics [3], which provide physically consistent forecasts. However, these models require substantial computation and exhibit sensitivity to initial conditions, restricting their real-time utility. In contrast, data-driven methods based on Long Short Term Memory Networks (LSTMs) have demonstrated strong performance in forecasting buoy-observed quantities such as gush speed, wave height, pressure, and sea surface temperature [4]–[8]. These methods offer faster inference and ease of deployment, but frequently rely on single-source input, not exploiting auxiliary information from satellite retrievals or reanalysis datasets that can offer broader spatial and temporal perspectives.

In practice, buoy measurements are often noisy and sparse, and simulations may not capture the full dynamics at relevant scales. These limitations motivate the study of fusion architectures, which aim to produce well-calibrated initial states that physical laws can later constrain during forecasting. Multisource fusion combines diverse observations, such as buoy recordings, ECMWF Reanalysis v5 (ERA5) reanalysis, and NOAA Global Forecast System (GFS) analysis data to generate more accurate estimates of surface wind components (u_{10}, v_{10}). Adaptive fusion models, especially those that use softmax-based weight mechanisms, dynamically allocate trust among sources in response to measurement discrepancies. However, the model outputs are not always physically plausible. The weights can be negative or do not sum up to one, and the forecasts may violate the principles of energy or mass conservation [9]. Physics-Informed Neural Networks (PINNs) address these issues by embedding PDEs into loss functions. Still, they often require explicit physical equations and incur heavy training costs, which limit their applicability in noisy or incomplete data regimes [9].

Schmidt *et al.* [6][7] and Pokhrel *et al.* [10][11] studied fusion models in the context of physics-based forecasting. In these models, numerical model predictions are integrated with observations in a weighting scheme to generate fused predictions. Specifically, Pokhrel *et al.* extended this to multiple data sources in [12]. In this study, we extend the previous studies by using differentiable constraints to obtain physically consistent predictions. This study focuses on architectures that both fuse multisource data effectively and enforce physical constraints on learned outputs. This study frames fusion as a method for generating physically meaningful initial estimates that remain valid when subjected to downstream constraints. Our primary contribution is the Lagrangian Fusion Model, which employs an augmented Lagrangian formulation to constrain fusion weights (ensuring non-negativity and unit sum) to enforce consistency between predicted wind vectors and observed Gust

Copyright (c) IARIA, 2025. ISBN: 978-1-68558-329-3

Speed (GST $\approx \sqrt{u_{10}^2 + v_{10}^2}$), without requiring explicit PDE formulas.

To validate our approach, three end-to-end architectures are compared:

1) **Baseline Model**: a sequence-to-sequence LSTM that forecasts Gust Speed (GST), Wave Heights (WVHT), Pressure (PRES), Water Temperature (WTMP), and wind components (u_{10} and v_{10}) using buoy data alone. This model benchmarks single-source performance under noisy conditions.
2) **Softmax Fusion Model**: adds softmax-weighted fusion of multisource inputs for u_{10} and v_{10}, while maintaining direct forecasting for other variables. It demonstrates the benefit of fusing information, along with enforcing the physical constraints. The weights of different sources are generated implicitly using the softmax layer in this approach.
3) **Lagrangian Fusion Model**: enhances the softmax approach by incorporating augmented Lagrangian penalties to enforce physically meaningful fusion weights and outputs, ensuring meteorological consistency and robustness in the presence of noisy measurements.

All fusion models are trained using buoy-observed targets and multi-source wind input. Physics-inspired loss terms, including constraint penalties and temporal smoothing, are integrated to improve the forecast fidelity. Evaluated on three years of data (2021–2023) with an 80/20 train/test split and rolling-horizon validation, the Lagrangian Fusion Model achieves substantial improvements while maintaining physically consistent behaviors. Our study advances the state-of-the-art by presenting a robust and interpretable fusion strategy that combines data-driven flexibility with physical realism, generating reliable initial states for downstream constrained forecasting or data assimilation frameworks.

This paper is organized as follows. Section 2 reviews related work. Section 3 details the dataset, model architectures, and training procedures. Section 4 presents the experimental results and their analysis. Section 5 discusses the limitations of the study and outlines directions for future research. Finally, Section 6 concludes the study.

II. Related Work

A. Neural Forecasting for Marine Variables

LSTMs have become important in marine environmental forecasting because of their ability to model temporal dependencies. For example, Bonino *et al.* combined buoy and reanalysis inputs in an LSTM framework and achieved a significant error reduction in sea surface temperature prediction compared to purely physical models [4]. Park *et al.* and Kim *et al.* showed comparable improvements in wave height and wind speed forecasting using buoy-only LSTM [13]. These models demonstrate the benefits of data-driven modeling but remain constrained by single-source inputs and lack of explicit physical constraints, which our work overcomes through multi-source fusion and constrained optimization.

B. Multisource Fusion Techniques

Optimal interpolation and classical data assimilation methods combine multiple sources of observational data by modeling error covariances [14]. Recent deep learning-based fusion approaches learn weighting schemes directly from data. Scher and Messori fused satellite and reanalysis data using convolutional architectures for global weather forecasts, reporting improved accuracy in spatial detail [15]. Shaw *et al.* used Spatiotemporal Dynamic Graph (SDG) networks to fuse multiple features to improve prediction accuracy for significant wave heights prediction [16]. However, neither approach enforces explicit physical constraints on fusion outputs. Our Softmax Fusion Model extends these works by enabling component-specific weighting for wind vectors, while our Lagrangian Fusion Model further ensures weight validity and meteorological consistency.

C. Physics-Informed Learning and Constrained Optimization

PINNs integrate physical laws into deep networks via loss-term penalties, enabling PDE-informed prediction capabilities [9]. While effective, they require explicit PDE formulations and tend to be computationally expensive, particularly in noisy or incomplete-data regimes [9]. Constrained optimization frameworks, such as those reviewed by Kotari *et al.*, embed structural constraints into network parameters using Lagrange multipliers, without relying on full physical equations [17]. Gramacy *et al.* demonstrated this approach in fluid dynamics by enforcing mass conservation using augmented Lagrangian methods [18]. Our Lagrangian Fusion Model adopts this strategy, applying constraints to both fusion weights and output predictions, ensuring that noisy, fused initial estimates remain physically consistent and usable for downstream data assimilation or forecasting frameworks.

Furthermore, we frame fusion as a preconditioning step that enhances subsequent constraint enforcement and data-assimilation efforts. This is supported by assimilation literature, which highlights the importance of good-quality initial states for robust real-time forecasts and correction of model deficiencies [19]–[21].

III. Methodology

A. Dataset

Table 1 presents the essential summary statistics (count, mean, standard deviation, minimum, and maximum) for each variable. Note that the fusion models are weighted using ERA-5 and NOAA wind fields in the cost function and not used as a part of input to the models.

Table 1 displays the count, mean, standard deviation, minimum, and maximum for each feature. In particular, the buoy-measured wind direction and wind speed, each with 29,778 samples, have means of 223.27° (std 96.07°) and 6.06 m/s (std 3.01 m/s), respectively. The zonal and meridional wind components at 10 meters ($u10\_buoy$ and $v10\_buoy$) were derived from the gust speed and wind direction. These wind-related variables are fused in the loss function using either fixed Lagrangian weights or learned weights via a Softmax

TABLE 1. BASIC SUMMARY STATISTICS FOR ALL VARIABLES

Variable	Meaning (Units)	Mean	Std	Min	Max
WDIR	Wind Direction (degrees)	223.27	96.07	2.44	360.00
WSPD	Wind Speed (m/s)	6.48	2.94	0.00	21.00
GST	Gust Speed (m/s)	8.38	3.61	0.21	26.61
WVHT	Wave Height (m)	2.12	1.03	0.35	10.08
DPD	Dominant Wave Period (s)	10.42	2.83	3.67	22.29
APD	Average Wave Period (s)	6.66	1.27	3.29	14.11
MWD	Mean Wave Direction (radians)	3.45	1.61	0.02	6.28
WTMP	Water Temperature (°C)	18.82	7.29	2.90	30.39
DEWP	Dew Point Temperature (°C)	15.08	6.74	-9.40	28.52
ATMP	Air Temperature (°C)	17.99	7.11	-3.26	30.84

layer. The data set comprises a total of 94,442 observations that span the period from January 1, 2021, to December 31, 2023 and the points are sampled every 3 hours. We utilize global predictions for ERA-5 and NOAA datasets.

B. Data Preprocessing

The raw time-series data is preprocessed through several key steps to ensure temporal continuity and improve model performance. First, the features that are redundant, irrelevant, or represent metadata not useful for forecasting are removed. To maintain the integrity of temporal sequences, rows with missing values are discarded instead of applying imputation, which can introduce artifacts. All continuous variables are standardized to have zero mean and unit variance, which facilitates faster convergence and stabilizes training in recurrent architectures. We then segment the data using 80/20 time-based split for training and testing. This setup supports multi-step forecasting aligned with short-term marine prediction needs.

Since the buoy observations lack corresponding numerical simulations, their values are predicted directly. The fusion targets in this study are the 10-meter wind components, u_{10} and v_{10}, estimated using multi-source integration of buoy measurements, ERA5 reanalysis, and NOAA data products. Although this work focuses on a single variable pair, the approach is extendable to other features, provided suitable numerical models are available for integration.

C. Model Architectures

This study evaluates three neural network architectures built on a sequence-to-sequence (Seq2Seq) LSTM backbone for multi-step environmental forecasting. All models share a common structure and incorporate physics-based constraints during training.

The architecture consists of an LSTM encoder and decoder, with an optional attention layer to improve temporal representation. The overall design is:

$$\begin{aligned} \text{Encoder:} \quad & \mathbf{h}_t, (\mathbf{h}_T, \mathbf{s}_T) = \text{LSTM}(\mathbf{x}_t, \mathbf{h}_{t-1}) \\ \text{Attention:} \quad & \mathbf{z}_t = \text{Attention}(\{\mathbf{h}_t\}) \\ \text{Decoder:} \quad & \mathbf{d}_t, (\mathbf{d}_*, \mathbf{s}_*) = \text{LSTM}(\mathbf{z}_t, \mathbf{d}_{t-1}) \end{aligned} \tag{1}$$

Here, $\mathbf{x}_t \in \mathbb{R}^{D_{in}}$ is the input at time t, and the encoder summarizes the input sequence into hidden states, and s represents the state at the last layer. The attention layer (when used) reweights the encoder output to form a context vector $\mathbf{z}_t$, which helps the decoder focus on relevant time steps during prediction. The decoder then produces latent outputs for the forecast window.

To encourage physically meaningful predictions, a physics-based loss term is added in the fusion models:

$$\mathcal{L}_{\text{physics}} = \sum_{c \in \mathcal{C}} w_c \cdot \phi_c(\hat{\mathbf{Y}}) \tag{2}$$

Each constraint $c \in \mathcal{C}$ has a weight w_c and penalty function ϕ_c that measures violations based on model outputs $\hat{\mathbf{Y}}$. Details are provided in Section III-D.

1) Baseline Model: The Baseline Model employs a straightforward prediction head applied to the decoder's latent representation:

$$\hat{\mathbf{Y}} = \text{Linear}(\text{ReLU}(\text{Linear}(\mathbf{d}))) \tag{3}$$

In this model, the constraints of physics are strictly enforced through the $\mathcal{L}_{\text{physics}}$ term, acting as a soft regularization.

2) Softmax Fusion Model: The Softmax Fusion Model extends the Baseline by incorporating a dynamic, attention-based fusion mechanism for wind component prediction. This model learns softmax-normalized weights to combine wind estimates from multiple input data sources (e.g., buoy, ERA5and NOAA inputs). The architecture integrates source-aware fusion with physics-constrained training:

$$\begin{aligned} \mathbf{\Lambda}_u &= \text{Softmax}(\text{MLP}_u(\mathbf{d})) \\ \mathbf{\Lambda}_v &= \text{Softmax}(\text{MLP}_v(\mathbf{d})) \\ \hat{\mathbf{Y}}_{\text{direct}} &= g_d(\mathbf{d}) \\ \hat{\mathbf{Y}}_{\text{wind}} &= g_w(\mathbf{d}) \\ \hat{u}_{\text{fused}} &= \sum_s \lambda_u^{(s)} u_s \\ \hat{v}_{\text{fused}} &= \sum_s \lambda_v^{(s)} v_s \\ \mathcal{L}_{\text{total}} &= \mathcal{L}_{\text{MSE}} + \alpha \mathcal{L}_{\text{physics}} \end{aligned} \tag{4}$$

where $\mathbf{\Lambda}_u$ and $\mathbf{\Lambda}_v$ are the learned softmax weights for the wind components u and v of the sources s, respectively. u_s and v_s represent the wind components from source s. g_d and g_w are multi-layer perceptrons (MLPs) that predict direct variables and wind residuals, respectively. The softmax mechanism facilitates the differentiable and interpretable weighting of each modality in each forecast timestep, thereby improving the robustness to noise and variations in data quality between different sources. Figure 1 displays the architecture of the fusion models. Note that the buoy data is used to make predictions and the other datasets are only used during training.

3) Lagrangian Fusion Model: While the Softmax Fusion approach implicitly ensures non-negativity and sum-to-one constraints for fusion weights, it may limit the model's expressiveness by tightly coupling all weights through a single normalization. To address this, we adopt an augmented Lagrangian formulation that allows for explicit control over constraints, enabling the model to learn fusion weights more flexibly.

The Lagrangian Fusion Model learns raw (unnormalized) weights $\mathbf{\Lambda}_u$ and $\mathbf{\Lambda}_v$ for wind component fusion. The fusion loss is then defined as:

$$\begin{aligned}
\mathbf{\Lambda}_u &= \mathrm{Linear}_u(\mathbf{d}) \\
\mathbf{\Lambda}_v &= \mathrm{Linear}_v(\mathbf{d}) \\
\mathcal{L}_{\text{constraints}} =& \mu_u \left(\sum_s \lambda_u^{(s)} - 1\right) + \frac{\rho}{2}\left(\sum_s \lambda_u^{(s)} - 1\right)^2 \\
&+ \sum_s \nu_u^{(s)}(-\lambda_u^{(s)}) + \frac{\rho}{2}\sum_s(-\lambda_u^{(s)})^2 \\
&+ \mu_v \left(\sum_s \lambda_v^{(s)} - 1\right) + \frac{\rho}{2}\left(\sum_s \lambda_v^{(s)} - 1\right)^2 \\
&+ \sum_s \nu_v^{(s)}(-\lambda_v^{(s)}) + \frac{\rho}{2}\sum_s(-\lambda_v^{(s)})^2 \\
\mathcal{L}_{\text{total}} &= \mathcal{L}_{\text{MSE}} + \alpha\mathcal{L}_{\text{physics}} + \beta\mathcal{L}_{\text{constraints}}
\end{aligned} \tag{5}$$

Here, μ_u, μ_v are Lagrange multipliers for the sum-to-one equality constraint, and $\nu_u^{(s)}, \nu_v^{(s)}$ are Lagrange multipliers for the non-negativity inequality constraints. ρ is the penalty parameter of the augmented Lagrangian. This formulation allows for decoupled optimization of weights while explicitly enforcing physical constraints.

D. Physics-Based Constraints

To promote physically consistent forecasts, the fusion models incorporate a set of differentiable physics-based constraints within the loss term $\mathcal{L}_{\text{physics}}$. These constraints are grounded in well-established relationships from wave physics and are summarized in Table 2.

E. Constraint Integration

Each constraint is implemented as a differentiable component of the total loss, allowing for gradient-based optimization. Denote each constraint as $\phi_c(\hat{\mathbf{Y}})$, then the total physics-informed penalty is:

$$\mathcal{L}_{\text{physics}} = \sum_c w_c \cdot \phi_c(\hat{\mathbf{Y}}) \tag{6}$$

where w_c is a fixed weight determined through empirical tuning. The wave development constraint and wave steepness limit are assigned lower weights (e.g., $w_c = 0.1$), reflecting their supportive role in shaping output realism without overwhelming the primary forecasting loss. The gravitational constant $g = 9.8\,\mathrm{m/s^2}$ is used in the wave steepness constraint, derived from classical wave theory.

This constraint formulation ensures that model predictions remain physically plausible, especially for wave-related parameters where empirical structure is well understood.

F. Training Framework

All models were trained using a consistent framework to ensure a fair comparison of their architectural merits. The key configuration parameters are as follows:

- **Optimizer:** Adam optimizer with an initial learning rate (η) of 5×10^{-4}. For the Lagrangian model, separate Adam optimizers were used for main model parameters and Lagrangian multipliers, both with an initial learning rate of 5×10^{-4}.
- **Learning Rate Schedule:** A step-down learning rate scheduler was applied, reducing the learning rate by a factor of 0.5 (for Baseline and Softmax) or 0.05 (for Lagrangian) every 25 epochs.
- **Regularization:** Gradient clipping with a maximum L2 norm of 1.0 was applied to prevent exploding gradients during training.
- **Batch Size:** Training was performed using a batch size of 64 sequences.
- **Physics Weight:** The weighting factor α for the $\mathcal{L}_{\text{physics}}$ term was set to 10^{-4} for all models. For the Lagrangian model, an additional weight $\beta = 0.1$ was applied to the $\mathcal{L}_{\text{constraints}}$ term.
- **Epochs:** Each model was trained for 50 epochs.

IV. Results and Discussion

This section presents a comprehensive analysis of the proposed models' performance, focusing on quantitative evaluation of first-step prediction (3 hours ahead) RMSE and a discussion of the observed trends and their implications.

Table 3 shows the first-step Root Mean Squared Error (RMSE) for the Baseline, Softmax, and Lagrangian models across a suite of oceanic and atmospheric variables. A clear trend of improvement is observed for the fusion-based models compared to the Baseline, especially for wave-related and temperature variables.

The most substantial relative improvements occurred in the prediction of air temperature (ATMP), sea surface temperature (WTMP), and average wave period (APD). For example, the RMSE for WTMP dropped from 0.2286 to 0.1793 with the Softmax model. These gains can be attributed to:

1) **Spatiotemporal Smoothness:** Variables like temperature evolve more smoothly in space and time, aligning well with the inductive bias of fusion models that leverage weighted averaging and regularization.

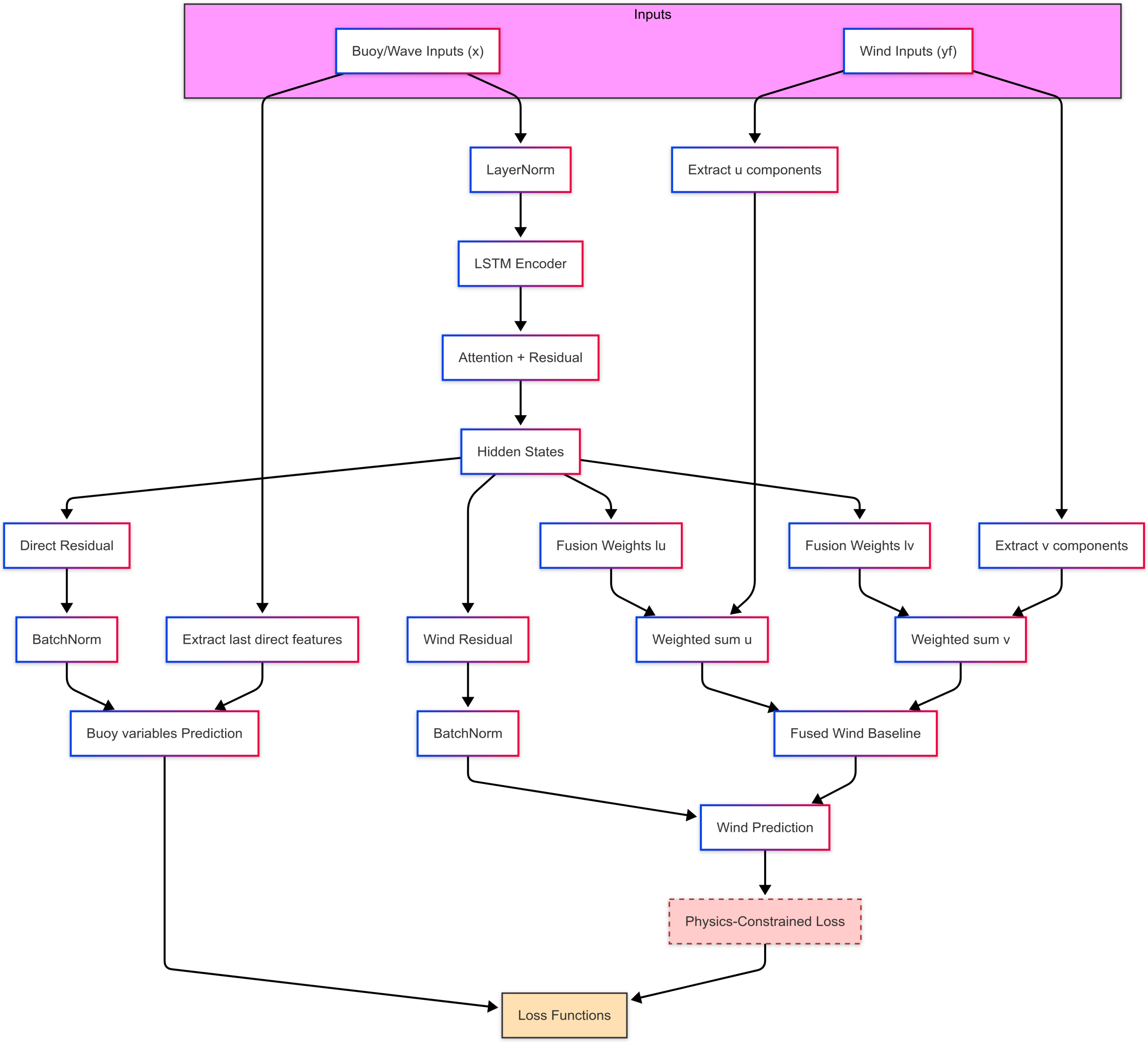


Figure 1. Model Architecture. Note that the wind inputs from multiple sources and fusion weights training is used only during training in the loss function calculation. The testing loop proceeds as normal.

2) **Low Spatial Variability:** ATMP and WTMP are governed by large-scale synoptic systems, making them more amenable to fusion from multiple sources. Fusion reduces noise and integrates consistent large-scale patterns effectively.

These variables also tend to have higher signal-to-noise ratios, enhancing the ability of models to generalize from reanalysis and auxiliary data sources.

Moderate improvements are also seen in wave-related variables (WVHT, DPD, APD) and wind speed (WSPD), suggesting that fusion helps smooth out some of the high-frequency noise or discrepancies among sources.

Interestingly, wind vector components (u_{10} and v_{10}) show degraded performance under fusion. These variables are subject to rapid and localized changes influenced by turbulence, topography, and mesoscale systems. Pointwise buoy observations may not align with gridded inputs (like ERA5), and fusion could oversmooth sharp transitions or localized bursts, reducing apparent accuracy.

Fusion models, especially the Lagrangian variant, demonstrate strong performance for coherent variables like temperature and wave metrics. Their inductive bias toward smoothness and large-scale consistency aligns with the physics and statistical characteristics of these variables. However, for high-variance, small-scale phenomena like wind vectors, fusion may inadvertently degrade performance. This does not imply

Copyright (c) IARIA, 2025. ISBN: 978-1-68558-329-3

TABLE 2. PHYSICS CONSTRAINTS APPLIED TO FUSION MODELS

Constraint	Formula	Basis	Weight
Wave Development Limit	$\mathrm{ReLU}(\mathrm{WVHT} - 1.3 \times 0.02 \times \mathrm{GST}^2)$	Pierson–Moskowitz spectrum	0.3
Wave Period Alignment	$\mathrm{ReLU}(\lvert\mathrm{DPD} - 0.55 \times \mathrm{GST}\rvert - 1.5)$	JONSWAP spectrum relation	0.4
Wave Steepness Limit	$\mathrm{ReLU}(\mathrm{WVHT} - 0.1 \times \frac{g \cdot \mathrm{DPD}^2}{2\pi})$	Non-breaking wave criterion	0.3

TABLE 3. FIRST-STEP PREDICTION (3 HOURS AHEAD) ROOT MEAN SQUARED ERROR (RMSE) COMPARISON (BEST VALUES IN **BOLD**)

Variable	Baseline	Softmax	Lagrangian
GST	**1.1134**	1.1217	1.1196
WVHT	0.1988	0.1961	**0.1945**
DPD	0.9134	0.9038	**0.9032**
APD	0.3305	0.2996	**0.2964**
MWD	0.4713	0.4362	**0.4353**
ATMP	0.3783	0.3767	**0.3691**
WTMP	0.2286	**0.1793**	0.1799
WSPD	0.9521	**0.9477**	0.9510
DEWP	0.5302	**0.5268**	0.5304
u_{10}	**6.5590**	9.9120	9.8987
v_{10}	**5.8451**	14.8446	14.8560

model failure but highlights a resolution mismatch and the challenges of noisy labels in wind observations. To address this, future work should explore hybrid or multi-scale loss structures that balance smoothness with local adaptability. Adaptive regularization or uncertainty-aware weighting may help retain fidelity in turbulent regimes while maintaining the benefits of fusion elsewhere.

A. Additional Physics Constraints

Beyond wind speed consistency and temporal smoothness, the following physics-based penalties, computed from the GFS or ERA5 fields, can further improve forecasts. Each constraint is tagged with the approach it benefits most and a brief rationale:

- **Boundary-Layer Momentum Balance (Turbulence-driven).** In the lowest ∼100 m the Coriolis force is negligible, so we can enforce a non-rotating, turbulent momentum balance forced by the wind at the layer top. Let τ be the turbulent stress tensor and $\mathbf{F}_{\mathrm{turb}}$ the wind-driven forcing. We then penalize deviations from
$$\rho \frac{\partial \mathbf{u}}{\partial t} + \rho\,(\mathbf{u}\cdot\nabla)\mathbf{u} = -\nabla p + \nabla\cdot\tau + \rho\,\mathbf{F}_{\mathrm{turb}},$$
via
$$\mathcal{L}_{\mathrm{BL}} = \mathbb{E}\Big[\big\|\rho\,\partial_t\hat{\mathbf{u}} + \rho\,(\hat{\mathbf{u}}\cdot\nabla)\hat{\mathbf{u}} + \nabla\hat{p} - \nabla\cdot\hat{\tau} - \rho\,\hat{\mathbf{F}}_{\mathrm{turb}}\big\|^2\Big].$$
(Best addressed via high-frequency observations + turbulence parameterizations.) This leverages detailed measurements to capture rapid, subgrid-scale fluctuations that a coarse grid would miss.
- **Hydrostatic Balance (Grid-resolved).** Using pressures at multiple levels (e.g. 925 hPa, 850 hPa) we can approximate
$$\frac{\Delta\hat{p}}{\Delta z} = -\,\hat{\rho}\,g, \quad \hat{\rho} = \frac{\hat{p}}{R_d\,\hat{T}}.$$
Anchoring $\hat{p}_{\mathrm{surf}}$ with buoy measurements, we can penalize any violation via
$$\mathcal{L}_{\mathrm{hydro}} = \mathbb{E}\Big[\big(\tfrac{\Delta\hat{p}}{\Delta z} + \hat{\rho}\,g\big)^2\Big].$$
(Resolved accurately on a typical numerical grid.) Large-scale pressure gradients vary smoothly over kilometers, so a grid model naturally enforces hydrostatic balance without needing high-frequency corrections.
- **Moisture Consistency (Hybrid).** We can enforce Clausius–Clapeyron and relative-humidity relations from ERA5 dewpoint and air temperature:
$$e_s(T) = 6.112\exp\!\Big(\tfrac{17.67\,T}{T+243.5}\Big), \quad \mathrm{RH} \approx 100\,\frac{e(\mathrm{DEWP})}{e_s(\mathrm{ATMP})}.$$
The loss penalizes squared errors between $(\hat{T}, \hat{T}_d, \hat{\mathrm{RH}})$ and these thermodynamic constraints:
$$\mathcal{L}_{\mathrm{moist}} = \mathbb{E}\Big[\big\|\big(\hat{T}, \hat{T}_d, \hat{\mathrm{RH}}\big) - \mathrm{thermo}\big\|^2\Big].$$
(Combines observations for humidity with grid-model thermodynamics.) Observations provide accurate moisture content but grid models capture large-scale transport, so a hybrid enforces both.
- **Surface Energy Budget (Hybrid).** Conservation of heat at the surface implies
$$\rho\,c_p\,\frac{\partial T}{\partial t} \approx H_s + H_l + R_n - Q_h,$$
where R_n is net radiation and H_s, H_l, Q_h are turbulent fluxes. We penalize
$$\mathcal{L}_{\mathrm{energy}} = \mathbb{E}\Big[\big(\rho\,c_p\,\partial_t\hat{T} - (H_s + H_l + R_n - Q_h)\big)^2\Big].$$
(Requires both grid-scale radiative fields and observations of fluxes.) Radiative inputs are well resolved on

Copyright (c) IARIA, 2025. ISBN: 978-1-68558-329-3

the grid, but surface fluxes fluctuate rapidly and need observational constraints.

- **Mass Continuity (Grid-resolved).** Assuming incompressible horizontal flow, any nonzero divergence is penalized:

  $$\mathcal{L}_{\text{mass}} = \mathbb{E}\big[(\nabla\cdot\, \hat{\mathbf{u}})^2\big].$$

  Horizontal winds can be taken directly from ERA5 or buoy observations, but this balance is naturally enforced on a resolved grid. *(Grid models handle continuity inherently over large scales.)*

B. Limitations and Future Work

The current model enforces only wind speed and smoothness physics, leaving out important couplings such as moisture, energy, and large-scale balance. It is a single-location model without spatial gradients, which limits its ability to conserve mass or energy across a region. Moreover, the coarser resolution of ERA5 and GFS fields compared to buoy data can introduce alignment errors. In such cases, it is better to handle numerical data and observations separately, as is common in data assimilation. Moreover, since buoy data represent turbulence scale phenomena which is not incorporated in global/regional models, physics loss can be independently applied to the fusion of numerical data, apart from observations.

From the experimental setup described in the paper, the method succeeds when the true physical parameter lies within the span of the available data sources. A natural concern is what happens if the optimal value falls outside this range, i.e., if all data sources are biased. In such scenarios, incorporating the governing PDEs as an additional "data source" within the framework ensures that predictions remain physically consistent.

Future work includes incorporating spatial embeddings using convolutional or graph modules to enforce conservation of mass, momentum, and energy over regions. We also plan to apply multiscale constraints that address synoptic-scale pressure and hydrostatic balance as well as subdaily energy and moisture budgets. Adaptive scheduling of Lagrangian multipliers will help gradually increase physics penalties during training to avoid conflicts with data-driven loss. Additional extensions include modeling ocean-atmosphere coupling with a mixed-layer sea surface temperature model and improving uncertainty quantification through ensembles or Bayesian networks calibrated with ERA5 error statistics.

V. Conclusion

This study developed and evaluated multi-source fusion architectures for accurate short-term forecasting of oceanic and atmospheric variables, addressing key limitations of existing data-driven and traditional numerical weather prediction models. Integrating heterogeneous data sources (buoy, ERA5, NOAA) through fusion models consistently improved predictive performance compared to a Baseline LSTM model relying on single-source observations. Multi-source fusion architectures offer a compelling improvement over single-source models for short-term forecasting of oceanic and atmospheric variables. Across a diverse set of targets, fusion consistently improves precision for parameters characterized by smooth, large-scale dynamics, most notably wave metrics and temperature fields, by leveraging complementary information from reanalysis, and observational products (buoys). This trend underscores the ability of models to take advantage of spatio-temporal coherence and reduce noise when integrating heterogeneous inputs.

Although fusion models showed a trade-off in performance for wind components (u_{10}, v_{10}), highlighting the challenge of balancing global physical consistency with capturing localized, high-variance wind phenomena, the overall benefits for buoy forecasting systems are clear. These findings emphasize the value of physics-informed constrained machine learning in enhancing the reliability and interpretability of environmental forecasts. Future work will extend the constraint set to include large-scale balances (boundary layer momentum, hydrostatic) and energy budgets, and will explore dynamic fusion weights and spatial embeddings to better reconcile global coherence with local fidelity. This research lays the groundwork for robust, physically consistent data-driven forecasting systems vital to maritime and climate-related applications.

Acknowledgment

This work was partly supported by the U.S. Department of the Navy, Naval Research Laboratory under contract N00173-20-2-C007. The work of Austin Schmidt was funded by a SMART (Science, Mathematics and Research for Transformation) Department of Defense (DoD) scholarship for service.

References

[1] M. J. Kaiser and A. G. Pulsipher, "The impact of weather and ocean forecasting on hydrocarbon production and pollution management in the gulf of mexico," *Energy policy*, vol. 35, no. 2, pp. 966–983, 2007.

[2] A. J. Hobday, C. M. Spillman, J. Paige Eveson, and J. R. Hartog, "Seasonal forecasting for decision support in marine fisheries and aquaculture," *Fisheries Oceanography*, vol. 25, pp. 45–56, 2016.

[3] P. Bauer, "What if? numerical weather prediction at the crossroads," *Journal of the European Meteorological Society*, vol. 1, p. 100 002, 2024, ISSN: 2950-6301. DOI: https://doi.org/10.1016/j.jemets.2024.100002. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2950630124000024.

[4] G. Bonino, G. Galimberti, S. Masina, R. McAdam, and E. Clementi, "Machine learning methods to predict sea surface temperature and marine heatwave occurrence: A case study of the mediterranean sea," 2, vol. 20, 2024, pp. 417–432. DOI: 10.5194/os-20-417-2024. [Online]. Available: https://os.copernicus.org/articles/20/417/2024/.

[5] S. Scher and G. Messori, "Weather and climate forecasting with neural networks: Using general circulation models (gcms) as a study ground," *Geoscientific Model Development*, vol. 12, pp. 2797–2809, 2019. DOI: 10.5194/gmd-12-2797-2019.

[6] A. B. Schmidt, P. Pokhrel, E. Ioup, D. Dobson, and M. Abdelguerfi, "An algorithm for modelling differential processes utilising a ratio-coupled loss," *Authorea Preprints*, 2024.

[7] A. B. Schmidt, P. Pokhrel, M. Abdelguerfi, E. Ioup, and D. Dobson, “Forecasting buoy observations using physics-informed neural networks,” *IEEE Journal of Oceanic Engineering*, 2024.

[8] A. B. Schmidt *et al.*, “Evaluating hyperparameter selection techniques for the ratio-coupled loss function,” *Authorea Preprints*, 2025.

[9] M. Raissi, P. Perdikaris, and G. E. Karniadakis, “Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations,” *Journal of Computational Physics*, vol. 378, pp. 686–707, 2019.

[10] P. Pokhrel, M. Abdelguerfi, and E. Ioup, “A machine-learning and data assimilation forecasting framework for surface waves,” *Quarterly Journal of the Royal Meteorological Society*, vol. 150, no. 759, pp. 958–975, 2024.

[11] P. Pokhrel, E. Ioup, M. T. Hoque, M. Abdelguerfi, and J. Simeonov, “Forecasting rogue waves in oceanic waters,” in *2020 19th IEEE International Conference on Machine Learning and Applications (ICMLA)*, IEEE, 2020, pp. 634–640.

[12] P. Pokhrel, A. Schmidt, M. Abdelguerfi, and E. Ioup, “Generalized multi-source assimilation: A framework for cross-modal integration and source optimization,” *Authorea Preprints*, 2025.

[13] F. C. Minuzzi and L. Farina, “A deep learning approach to predict significant wave height using long short-term memory,” *Ocean Modelling*, vol. 181, p. 102 151, 2023, ISSN: 1463-5003. DOI: https://doi.org/10.1016/j.ocemod.2022.102151. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1463500322001652.

[14] T. Archambault, A. Filoche, A. Charantonis, D. Bereziat, and S. Thiria, “Learning of sea surface height interpolation from multi-variate simulated satellite observations,” *arXiv*, 2024. arXiv: 2310.07626 [cs.LG]. [Online]. Available: https://arxiv.org/abs/2310.07626.

[15] X. Sun *et al.*, “Fuxi weather: A data-to-forecast machine learning system for global weather,” Aug. 2024. DOI: 10.48550/arXiv.2408.05472.

[16] S. Yin, T. Wu, J. Yu, X. He, and L. Xu, “A significant wave height prediction method with ocean characteristics fusion and spatiotemporal dynamic graph modeling,” *Acta Oceanologica Sinica*, vol. 43, pp. 13–33, Mar. 2025. DOI: 10.1007/s13131-024-2450-4.

[17] J. Kotary, F. Fioretto, P. Van Hentenryck, and B. Wilder, “End-to-end constrained optimization learning: A survey,” *arXiv preprint arXiv:2103.16378*, 2021.

[18] R. B. Gramacy *et al.*, “Modeling an augmented lagrangian for blackbox constrained optimization,” *Technometrics*, vol. 58, no. 1, pp. 1–11, 2016.

[19] F. Zhang, S. Zhang, and S. Yang, “Data assimilation in numerical weather and climate models,” *Advances in Meteorology*, vol. 2015, p. 626 893, 2015. DOI: 10.1155/2015/626893.

[20] E. Kalnay, “Atmospheric modeling, data assimilation and predictability,” in *Cambridge University Press*, Cambridge, UK, 2003.

[21] M. A. Balmaseda and D. L. T. Anderson, “Impact of initialization strategies and observations on seasonal forecast skill,” *Geophysical Research Letters*, vol. 36, no. 1, p. L01702, 2009. DOI: 10.1029/2008GL035561.

ISBN: 978-1-68558-329-3

Simulation-Based Study of Propulsion and Control of an AUV in Current-Affected Environments

Aymen Mefti, Mahfoudh Cerdoun
Ecole Militaire Polytechniques
Alger, Algeria
email: aymen21mefti@gmail.com
email: cerdoun.mahfoudh@gmail.com

Djahida Boucetta
Royal Military Academy RMA
Brussel, Belgium
email: djahida.boucetta@mil.be

Abstract— **This study presents a numerical investigation into the hydrodynamic behavior and control of an Autonomous Underwater Vehicle (AUV) operating in current-affected marine environments. Initially, a self-propulsion test has been conducted to determine the optimal propeller rotational speed required to overcome the AUV's hydrodynamic resistance during a steady, straight-line motion. Subsequently, the effect of the lateral marine current has been examined, introducing additional transverse resistance that require dynamic adjustments in both rudder deflection and propeller rotational speed to keep a fixed forward speed. Therefore, a parametric analysis of the AUV's response to varying control configurations is investigated, focusing on the effects of rudder deflection angles and propeller rotational speeds on the surge, sway, and yaw motion. The overall numerical approach is validated using the propeller open water experimental data. The results highlight the effectiveness of coupled hydrodynamic simulation and control input strategies in predicting and managing AUV behavior in complex and dynamic marine environments.**

Keywords-Autonomous Underwater Vehicle (AUV); Self propulsion; Marine Currents; Hydrodynamic Simulation; Propeller Performance.

I. INTRODUCTION

Autonomous Underwater Vehicles (AUVs) have become indispensable tools across a broad spectrum of scientific and industrial domains. Their deployment enables operations that are hazardous, impractical, or impossible for human divers, particularly in deep-sea exploration where AUVs facilitate the discovery and detailed mapping of previously inaccessible marine environments [1]. Beyond exploration, AUVs play a crucial role in conducting oceanographic experiments, performing seabed surveys for geological and geographical research, and acquiring high-resolution data in real time [2]. In response to these growing demands, considerable research efforts have been directed toward enhancing the hydrodynamic efficiency and autonomous capabilities of AUVs. A key challenge lies in comprehensively understanding the complex fluid dynamics around AUVs, which is complicated by the presence of multiple interacting components such as propellers, rudders, and control fins [3].

Moreover, the remote and often prolonged nature of AUV missions imposes stringent requirements on autonomy and control systems. AUVs must operate independently in dynamic and often unpredictable underwater environments, necessitating sophisticated control algorithms, robust navigation systems, and efficient energy management strategies to maximize operational range and mission duration [4] [5].

The AUVs have attracted considerable research interest. Honaryar and Ghiasi [6] introduced a bio-inspired hull design modeled after the catfish *Hypostomus*. Based on numerical simulations and experimental studies, they demonstrated an approximately 99% improvement in the hydrodynamic stability. Yu et al. [7] investigated the Underwater Radiated Noise (URN) through a coupling of Computational Fluid Dynamics (CFD) with Lighthill's acoustic analogy, and these investigations allow identification of cavitation and propeller vibrations as principal noise sources. Similarly, Wu et al. [8] performed physics-based simulations of a free-running propeller-driven AUV. The results revealed transient thrust fluctuations and wake structures that are critical for understanding propulsion efficiency. Environmental interactions, such as wave-current coupling, further complicate AUV performance [9]-[14]. Ding et al. [9] numerically analyzed the DARPA Suboff submarine with a pump-jet propulsor operating near the free surface. Min et al. [10] introduced a hybrid CFD and system identification approach to model multi-propeller AUV maneuvering, enabling accurate extraction of hydrodynamic parameters for control system design. Raja et al. [11] explored unmanned amphibious systems integrating aerial and marine domains via multi-domain simulations, highlighting innovative propulsion and control strategies for enhanced marine surveillance capabilities. The effects of control surface deflections and hydrodynamic interactions among multiple AUVs have also been extensively studied. Dantas and de Barros [12] employed Reynolds-Averaged Navier-Stokes (RANS) CFD to capture nonlinear hydrodynamic forces on hulls and control surfaces, whereas Rattanasiri et al. [13] and Hong et al. [14] analyzed hydrodynamic interactions within tandem AUV formations.

Further studies have examined propeller-induced interactions and wave effects [15]-[21]. Liu et al. [15] reported thrust enhancements of up to 7% for two propeller-driven Unmanned Underwater Vehicles (UUVs) in formation. Tian et al. [16] employed two-phase CFD to analyze wave impacts on an axisymmetric AUV near the surface, revealing lift and drag sensitivity to wave parameters. Liu et al. [17] numerically evaluated the UUV's hydrodynamics and self-propulsion near the seabed.

Despite the extensive literature concerning the hydrodynamics of AUVs, a significant research gap persists in thoroughly examining the coupled effects of propulsion and stability under time-varying marine currents. Previous studies have primarily addressed individual aspects such as hull design, propeller performance, or specific environmental interactions. However, the combined influence of lateral current forces on AUV propulsion and trajectory stability has not been systematically investigated using high-fidelity numerical simulations. This omission is critical, as lateral currents can substantially modify hydrodynamic loads, induce yaw or drift, and ultimately degrade navigational accuracy.

The present study attempts to fill this gap by conducting a simulation-based investigation of AUV propulsion and stability in current-affected environments. Employing RANS modeling, the work quantifies hydrodynamic responses and assesses how lateral disturbances impact straight-line motion. By providing detailed insights into the coupled behavior of the AUV and its operational environment, this research contributes to an integrated understanding of AUV performance under realistic marine conditions.

The remainder of this paper is organized as follows. Section II provides a detailed description of the numerical modeling framework, including the AUV geometry, computational domain and boundary conditions, mesh configuration, and the numerical procedure. Section III presents the validation of the simulation methodology against experimental propeller data. Section IV discusses the results, while Section V concludes the study and outlines directions for future work.

II. Numerical Modelling

The numerical simulations conducted in this study—both for the self-propulsion analysis and for the AUV operating under current-affected conditions—were performed using the commercial CFD software ANSYS CFX [18]. Coupling with MATLAB was employed to support parameter control and data processing. This section provides a detailed description of the AUV's geometric components, including the hull, rudders, and propeller, as well as the numerical setup and modeling approach

A. AUV Geometry

The full scale self-propelled AUV utilized in this study consists of three main components: the hull, the finned rudder plates, and the propeller, as illustrated in Figure 1. The shapes of the nose and tail sections are determined from [19], L_h, L_c and L_c are the lengths of nose, body and tail sections, respectively. D_A is the maximum diameter, n is an exponential parameter which can be varied to give different body shapes ($n = 1.8$ in the current study), and θ is the included angle at the tip of the tail. The total length of the AUV is 2 meters. Detailed dimensional parameters are provided in Table 1.

The rudder blades (Figure 2) employ a flat-plate airfoil profile, characterized by a circular leading edge that has a radius of 4 mm and a trailing edge with a radius of 1.6 mm. The trailing edge of every rudder blade is positioned 1925 mm from the nose of the AUV. The chord length of the blade varies linearly across the span, starting at 82 mm at the hub and tapering down to 7 mm at the tip. The DTMB 4119 model propeller (Figure 3) has been selected to propel the vehicle. Its specific geometric details are presented in Table 2.

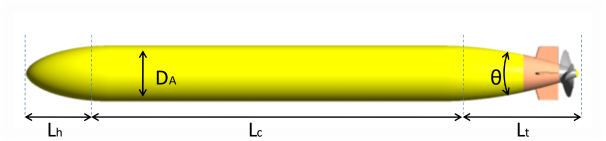


Figure 1. The AUV Geometry.

TABLE 1. MAIN PARAMETERS OF AUV HULL.

Parameters	Value
D_A	0.2 m
L_A	2m
L_h	0.3m
L_c	1.2m
L_t	0.5m
θ	20°

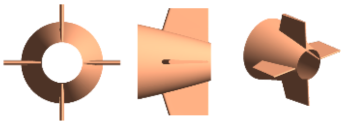


Figure 2. The Rudder Geometry.

TABLE 2. DETAILED PARAMETERS OF THE DTMB 4119 PROPELLER.

D(m)	0.1829
Z	3
Skew(°)	0
Rake(°)	0
Blade section	NACA 66 a=0.8
Rotation direction	Right

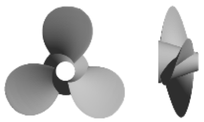


Figure 3. The DTMB 4119 Marine Propeller.

B. Computational domain and Boundary conditions

The computational domain (Figure 4) consists of a cylindrical volume that surrounds the AUV. Its dimensions are selected to reduce the influence of boundary effects on the flow field solution: the domain extends a distance of L_A upstream from the inlet, $3L_A$ downstream from the outlet,

Copyright (c) IARIA, 2025. ISBN: 978-1-68558-329-3

and *10*$\mathrm{D_A}$ in the radial direction outwards the far field, where $\mathrm{L_A}$ is the length of the AUV is and $\mathrm{D_A}$ is its diameter. To accurately capture the complex flow patterns near the propeller, the computational domain is segmented into two subdomains: One that rotates around the propeller and another that remains stationary, representing the outer flow domain. The flow around the propeller is modeled using a Moving Reference Frame (MRF) approach within the rotating region. The interface between the rotating and stationary subdomains is handled using a General Grid Interface (GGI) with a "frozen rotor" condition, which facilitates the transfer of flow variables and guarantees precise coupling between the two regions while maintaining stable and realistic flow dynamics.

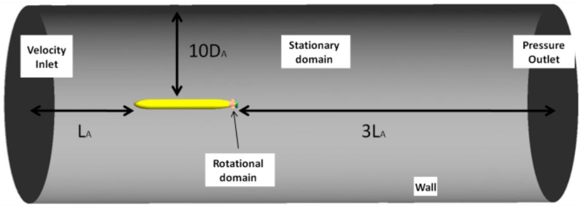


Figure 4. Computational domain with boundary conditions.

The boundary conditions were defined as presented in Figure 4. A no-slip wall condition was applied on the hull, rudder, and propeller surfaces. The propeller was modeled as a rotating wall. The presence of a lateral marine current introduces an additional velocity component, $V_{current}$, (Figure 5) which introduces additional hydrodynamic loads on the AUV. To maintain the desired forward speed and trajectory, it becomes necessary to adjust the rudder blade orientation angles and the propeller rotation speed accordingly. This control strategy forms the focus of the second part of our study.

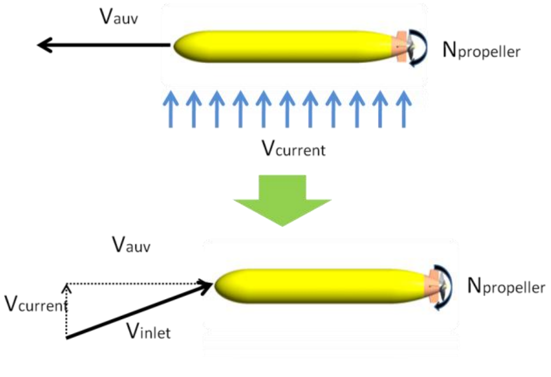


Figure 5. Modeling of the Effect of Lateral Marine Currents.

To account for the effect of the lateral marine current, the computational domain was modified by rotating the AUV about the *z*-axis by an angle $\alpha_{\mathrm{current}}$, defined as $\alpha_{\mathrm{current}} = \tan^{-1}(\mathrm{V_{current}}/\mathrm{V_{auv}})$. Additionally, the inlet velocity was updated to incorporate both the AUV's advance speed and the current velocity, as illustrated in Figure 5. Figure 6 presents the modified computational domain used in the second part of this study.

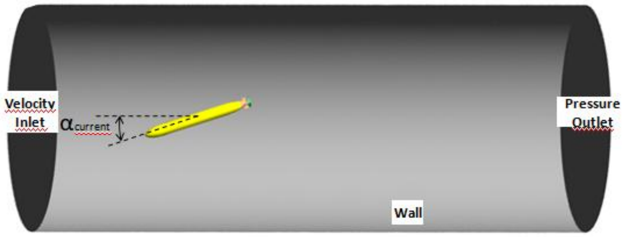


Figure 6. Computational domain considering the effect of lateral marine current.

C. Mesh Setup

A hybrid mesh was employed for the computational domain, consisting of structured elements around the propeller and rudder, and unstructured elements for the hull. Structured elements were also applied within the boundary layer regions surrounding the hull, propeller, and rudder to accurately capture near-wall effects. Additionally, local mesh refinement was implemented in regions with abrupt geometric changes, Such as the leading and trailing edges, as shown in figure 7. The computational domain consists of approximately 9 million elements, of which about 3 million belong to the rotary domain and 6 million to the stationary domain.

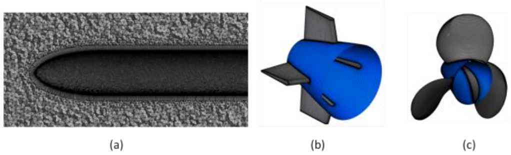


Figure 7. Surface Mesh: - (a) Hull - (b) Rudder - (c) Propeller.

D. Numerical Procedure

The self-propulsion point of the AUV, in straight-line mode, is defined as a combination of AUV constant speed and propeller Revolutions Per Minute (RPM) rate in which the propeller thrust force matches AUV hydrodynamic resistance. This resistance arises from various components, including the hull, control surfaces, such as rudders, and external appendages like sensors. From a mathematical standpoint, this condition corresponds to the propeller rotational rate at which the net force on the vehicle vanishes, consistent with Newton's second law (1).

$$T + R_{hull} + R_{ruddder} = 0 \qquad (1)$$

Here, *T* denotes the thrust generated by the propeller, while R_{hull} and R_{rudder} represent the hydrodynamic resistance of the hull and rudder, respectively. The procedure to determine the self-propulsion condition is based on an iterative process in which a numerical simulation is performed at each iteration. Once the simulation converges, the relevant hydrodynamic forces are extracted through post-processing. These values are then used to evaluate (1), which serves as the convergence criterion. Based on the obtained results, the propeller’s rotational velocity is updated using the secant method (2). This process is repeated iteratively until the self-propulsion condition is satisfied.

$$\begin{cases} N_{i+1} = N_i - F_i \times \frac{N_i - N_{i-1}}{F_i - F_{i-1}} \\ F_i = T_i + R_{hull_i} + R_{rudder_i} \end{cases} \tag{2}$$

When an AUV traveling along a straight path at steady speed encounters a current-affected environment with lateral flow (Figure 8), it experiences additional hydrodynamic forces. These forces induce sway and yaw motions, causing the vehicle to deviate from the intended trajectory and potentially compromising its directional stability. In the absence of corrective control actions, the AUV will drift and may gradually veer off from its original heading. Consequently, active control systems are essential for preserving trajectory and heading stability under such flow-disturbed conditions.

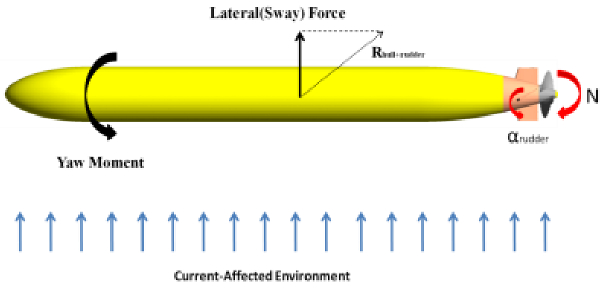


Figure 8. Effect of Cross-Current on AUV Stability and Trajectory.

A series of numerical simulations was conducted to assess the AUV's response to lateral current disturbances while implementing different corrective control measures. In particular, two control parameters were investigated: the deflection angle of the upper and lower rudder blades, referred to as α_{rudder} throughout this study, and the rotational propeller rate, denoted as *N(rpm)*.

III. Validation

To validate the numerical methodology, the hydrodynamic performance of the E779A marine propeller is evaluated through numerical simulations. This specific propeller has been extensively investigated both numerically and experimentally in the literature [20] [21]. Simulations were performed to compute the open water characteristics, which are then compared with the experimental data available in [20]. The open water characteristics illustrate the propeller's performance in a uniform flow field, and are expressed using several key dimensionless parameters: the advance coefficient *J*, the thrust coefficient K_T, the torque coefficient K_Q, and the open water efficiency η. During the simulations, the rotational speed is maintained at a constant value of 11.7881 rps, while the inflow velocity is varied to alter the advance coefficient *J* and three operating conditions were tested. The comparison of the numerical results to the experimental data for thrust and torque coefficients is summarized in Table 3.

Table 3. Comparison of Thrust and Torque Coefficients under Non-Cavitating Open-Water Conditions.

J	$K_{T_{NUM}}$	$10K_{Q_{NUM}}$	$er_{K_T}(\%)$	$er_{K_Q}(\%)$
0.348	0.409	0.685	0.362	1.587
0.747	0.217	0.413	-2.118	1.99
0.946	0.119	0.267	-5.17	4.871

The deviations in thrust and torque coefficients range from 0.36% to 5.17%, indicating a significant level of concordance with experimental data and affirming the model's validity under open-water conditions.

IV. Results and Discussion

A. *Self-Propelled AUV*

The initial study aims to determine the self-propulsion condition of the Autonomous Underwater Vehicle (AUV) described above. This is achieved using a Computational Fluid Dynamics (CFD) approach based on RANS equations, in conjunction with a MATLAB script, following the computational procedure outlined in Figure 9. The AUV is assumed to travel in a straight trajectory at a steady speed of 5 m/s. The self-propulsion condition is obtained using the secant method, which requires two initial estimates for the propeller rotational speed, denoted as N0 and N1. These initial speeds are established at 1400 rpm and 1500 rpm, respectively. The convergence criterion is defined as $e = \frac{|F|}{T} < 0.001$, where F represents the residual force and T is the thrust, ensuring that the solution reaches an acceptable level of accuracy.

Figure 9 illustrates the behavior of the Secant method during the numerical assessment of the self-propulsion condition. In Figure 9(a), the normalized residual force, is plotted against the iteration number to assess convergence. The method required a total of six iterations, including the two initial guesses, to reach convergence. Figure 9(b) depicts the residual force F(N) as a function of the propeller rotational speed N. The red markers trace the sequence of estimates generated by the Secant method, clearly indicating convergence toward the zero-residual point. The dashed horizontal line at F(N)=0 represents the condition of exact thrust-resistance equilibrium. The intersection confirms that the propeller generates

Copyright (c) IARIA, 2025. ISBN: 978-1-68558-329-3

exactly the thrust required to overcome the vehicle's hydrodynamic resistance.

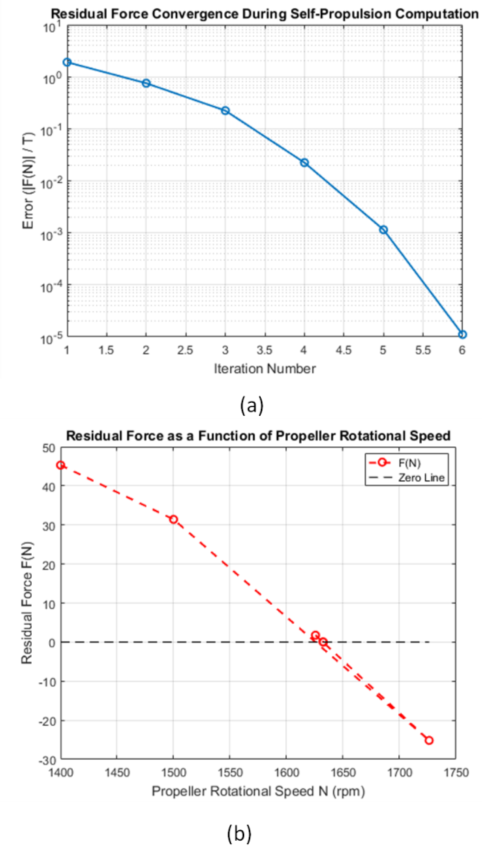


Figure 9. Convergence and residual force behavior during the computation of the self-propulsion condition using the Secant method: (a) Residual force convergence over iterations; (b) Residual force F(N) as a function of propeller rotational speed.

Table 6 displays the propeller thrust alongside the corresponding hydrodynamic resistance forces generated by the hull and rudder at an AUV velocity of 5 m/s. As expected, the hull resistance constitutes the majority of the total resistance, primarily due to its extensive wetted surface area. Approximately 67.5% of the hull resistance is attributed to viscous forces, as demonstrated by the wall shear stress distribution illustrated in Figure 10. Elevated shear stress values are particularly prominent near the bow and along the midsection of the hull. In comparison, the rudder contributes around 21.1% to the total resistance, with pressure forces being the predominant component. This is supported by Figure 11, which shows the static pressure distribution along the AUV's mid-plane and highlights a pronounced pressure gradient between the rudder's leading and trailing edges. This behavior is largely influenced by the geometry of the rudder, which is characterized by a thin profile featuring rounded leading and trailing edges. Figure 12 further illustrates this effect, depicting a relatively smooth flow along the hull surface, in contrast to the significantly disturbed flow observed in the wake region downstream of the rudder.

TABLE 6. PROPELLER THRUST AND THE CORRESPONDING HULL AND RUDDER RESISTANCE FORCES AT AN AUV SPEED OF 5 M/S.

	T	R_hull	R_rudder
Value(N)	81.28	-60.18	-21.10
Pressure Force (%)	104.6%	32.5%	84%
Viscous Force (%)	-4.6%	67.5%	16%

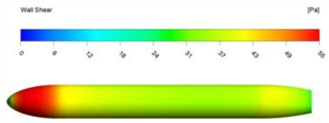


Figure 10. Wall shear stress distribution over the hull surface.

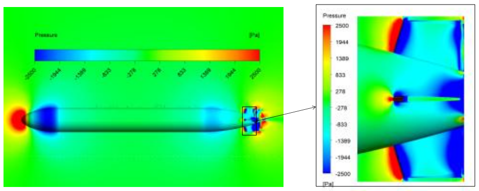


Figure 11. Pressure field around the AUV.

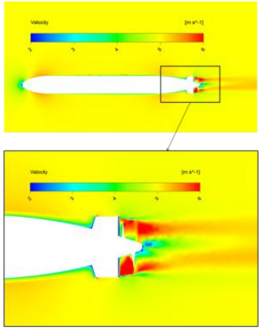


Figure 12. Velocity field around the AUV.

B. *AUV in Lateral Current Affected Environment*

The second part of this study investigates the behavior of the AUV in an environment influenced by a lateral current, with particular focus on the effects of varying rudder deflection angles and propeller rotational speeds. To achieve this, a series of simulations were conducted using different control configurations, combining various rudder deflection angles and propeller speeds. The objective was to identify the optimal control settings that enable the AUV to sustain a straight and steady trajectory, in accordance with Newton's second law of motion.

Figure 13 presents the net forces acting on the AUV in the x-direction (surge) and y-direction (sway), as well as the net moment around the z-axis (yaw), for various control configurations.

1) *Effect of Propeller Rotational Speed on Surge Force*

As the propeller's rotational speed increases, it produces increased thrust. This thrust counteracts the hydrodynamic resistance imposed by the hull and rudder, leading to a reduction in the net surge force as illustrated in Figure 13(a).

Thus, an inverse correlation is observed between propeller speed and the net force in the x-direction.

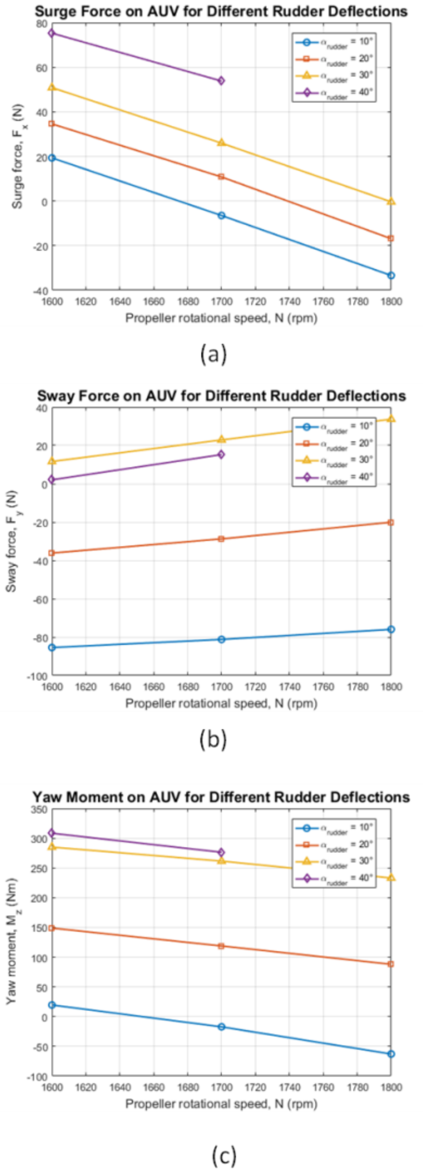


Figure 13. AUV Hydrodynamic Response in (a) Surge, (b) Sway, and (c) Yaw under Varying Control Inputs.

2) *Effect of Rudder Deflection on Surge and Sway Forces*

Increasing the rudder deflection angle amplifies the pressure gradient acting on the rudder blades. This effect is visualized in Figure 14, which shows the pressure field around the rudder at N=1800 rpm for two distinct rudder deflection angles α_{rudder}=20°and α_{rudder} =30°. A clear increase in pressure gradient across the rudder is apparent with increased deflection, leading to enhanced pressure forces induced by the rudder. As a result, both surge and sway forces increase with rudder deflection at a given propeller speed as illustrated in Figure 13(a) and 13(b).

3) *Rudder's Role in Compensating the Yaw Moment*

To counteract the negative yaw moment generated by the hull, the rudder produces a moment in the positive z-direction. This is achieved by deflecting the rudder to generate a pressure-induced moment that opposes the hull's contribution. For the configuration with a rudder deflection angle of α_{rudder} =20° and rotational speed of N=1800 rpm, the rudder contributes up to 99% of the total yaw moment. This dominance of the rudder's pressure force explains the observed increase in net yaw moment with increasing rudder deflection, as illustrated in Figure 13(c).

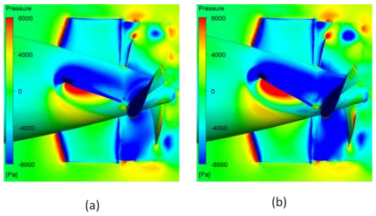


Figure 14. Pressure Field around AUV for N=1800 rpm in Current-Affected Environment: a) α_{rudder}=20°and b) α_{rudder} =30°.

4) *Interaction between Propeller Speed and Rudder-Induced Forces*

For a fixed rudder deflection angle, increasing the propeller rotational speed accelerates the flow around the rudder, consequently amplifying the pressure differential across its surface.

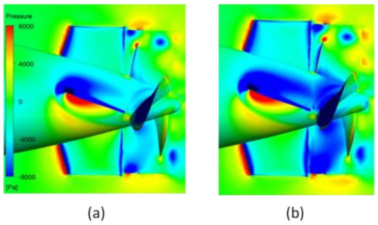


Figure 15. Pressure Field around AUV with α_{rudder}=20°. in Current-Affected Environment: -(a):N=1600 rpm and b) N=1800 rpm

This phenomenon is also depicted in Figure 15, which contrasts the pressure field around the rudder at α_{rudder}=20° for N=1600 rpm and N=1800 rpm. The increased flow velocity at higher propeller speeds results in a more pronounced pressure gradient across the rudder, thereby increasing the sway force.

Copyright (c) IARIA, 2025. ISBN: 978-1-68558-329-3

V. CONCLUSION AND FUTURE WORK

This study presents a simulation-based analysis of the propulsion and stability characteristics of an Autonomous Underwater Vehicle (AUV) during both straight, constant-speed motion and in current-affected environments. The AUV model, consisting of the hull, rudder, and propeller, was simulated under steady-state conditions using the RANS. The CFD methodology was validated against experimental data from open-water propeller tests, showing good agreement with experimental results, thereby confirming the reliability of the adopted numerical approach. A self-propulsion condition was established by coupling CFD with the secant root-finding method to determine the required propeller rotational speed and corresponding thrust to balance the hydrodynamic resistance forces. This method demonstrated high accuracy and robust convergence. Additionally, the behavior of the AUV in a lateral current environment was explored through a series of simulations that varied control inputs—particularly focusing on rudder deflection angles and propeller rotational speeds. The outcomes were analyzed with particular attention to the resulting surge, sway, and yaw responses. Findings reveal a significant interdependence between the control parameters and the hydrodynamic forces and moments, highlighting the importance of precise actuation for maintaining trajectory stability in flow-disturbed environments. As a future direction, this research will be extended through the integration of optimization tools and Artificial Intelligence (AI) algorithms. The goal is to identify optimal control configurations that enable the AUV to effectively navigate complex and dynamic oceanic environments, which include strong marine currents and surface wave disturbances, thus enhancing its operational robustness and mission success.

REFERENCES

[1] R. B. Wynn et al., "Autonomous underwater vehicles (AUVs): their past, present and future contributions to the advancement of marine geoscience," *Marine Geology*, vol. 352, pp. 451–468, Jun. 2014.

[2] K. J. Campbell, S. Kinnear, and A. Thame, "AUV technology for seabed characterization and geohazards assessment," *The Leading Edge*, vol. 34, no. 2, pp. 170–178, 2015.

[3] F.-d. Gao, C.-y. Pan, and Y.-y. Han, "Numerical computation and analysis of unsteady viscous flow around autonomous underwater vehicle with propellers based on sliding mesh," *Journal of Central South University*, vol. 19, pp. 944–952, 2012.

[4] G. Antonelli and E. Cataldi, "Recursive adaptive control for an underwater vehicle carrying a manipulator," in *Proceedings of the 22nd Mediterranean Conference on Control and Automation*, 2014.

[5] T. I. Fossen, *Handbook of Marine Craft Hydrodynamics and Motion Control*. Hoboken, NJ, USA: Wiley, 2011.

[6] A. Honaryar and M. Ghiasi, "Design of a bio-inspired hull shape for an AUV from hydrodynamic stability point of view through experiment and numerical analysis," *Journal of Bionic Engineering*, vol. 15, no. 6, pp. 950–959, 2018.

[7] C. Yu, R. Wang, X. Zhang, and Y. Li, "Experimental and numerical study on underwater radiated noise of AUV," *Ocean Engineering*, vol. 201, p. 107111, 2020.

[8] L. Wu et al., "Hydrodynamic performance of AUV free running pushed by a rotating propeller with physics-based simulations," *Ocean Engineering*, vol. 202, p. 107205, 2020..

[9] Y. Ding, X.-r. Li, Y. Yan, and W.-Q. Wang, "Investigation of the self-propulsion performance of a submarine with a pump-jet propulsor in a surface wave–current coupling environment," *Ocean Engineering*, vol. 298, p. 119306, 2024.

[10] F. Min, G. Pan, and X. Xu, "Modeling of autonomous underwater vehicles with multi-propellers based on maximum likelihood method," *Journal of Marine Science and Engineering*, vol. 8, no. 6, p. 429, 2020.

[11] V. Raja et al., "Multi-domain based computational investigations on advanced unmanned amphibious system for surveillances in international marine borders," *Aerospace*, vol. 9, no. 11, p. 652, 2022.

[12] J. L. D. Dantas and E. A. de Barros, "Numerical analysis of control surface effects on AUV manoeuvrability," *Applied Ocean Research*, vol. 40, pp. 118–126, 2013.

[13] P. Rattanasiri, P. A. Wilson, and A. B. Phillips, "Numerical investigation of a pair of self-propelled AUVs operating in tandem," *Ocean Engineering*, vol. 100, pp. 126–137, 2015.

[14] L. Hong, R. Fang, X. Cai, and X. Wang, "Numerical investigation on hydrodynamic performance of a portable AUV," *Journal of Marine Science and Engineering*, vol. 9, no. 8, p. 812, 2021.

[15] X. Liu, Y. Hu, Z. Mao, W. Ding, and S. Han, "Numerical investigation on interactive hydrodynamic performance of two adjacent unmanned underwater vehicles (UUVs)," *Journal of Marine Science and Engineering*, vol. 11, no. 11, p. 2088, 2023.

[16] W. Tian, B. Song, and H. Ding, "Numerical research on the influence of surface waves on the hydrodynamic performance of an AUV," *Ocean Engineering*, vol. 183, pp. 40–56, 2019.

[17] X. Liu, Y. Hu, Z. Mao, and W. Tian, "Numerical simulation of the hydrodynamic performance and self-propulsion of an UUV near the seabed," *Applied Sciences*, vol. 12, no. 14, p. 6975, 2022.

[18] ANSYS Inc., *ANSYS CFX User's Guide*, Release 18.2, Canonsburg, PA, USA.

[19] T.-H. Joung, K. Sammut, F. He, and S.-K. Lee, "Shape optimization of an autonomous underwater vehicle with a ducted propeller using computational fluid dynamics analysis," *International Journal of Naval Architecture and Ocean Engineering*, vol. 4, pp. 44–56, 2012.

[20] F. Salvatore, F. Pereira, M. Felli, D. Calcagni, and F. Di Felice, *Description of the INSEAN E779A Propeller Experimental Dataset (INSEAN Technical Report No. 2006-085)*. Rome, Italy: INSEAN – Italian Ship Model Basin, 2006.

[21] D. Boucetta and O. Imine, "Numerical simulation of the cavitating flow around marine co-rotating tandem propellers," *Brodogradnja: Shipbuilding*, vol. 70, no. 1, pp. 47–58, 2019.

Operationalizing and Testing Machine Learning Models for Acoustic Target Classification

Nils Olav Handegard, Silje Smith-Johnsen, Arne Johannes Holmin, Cristian Muños Mas
Institute of Marine Research
Bergen, Norway
email: nilsolav@hi.no; silje.smith-johnsen@hi.no; arnejh@hi.no; cristian.munoz.mas@hi.no

Ingrid Utseth
Norwegian Computing Centre
Oslo, Norway
email: utseth@nr.no

Daniel Dondorp
Sopra Steria
Bergen, Norway
email: Daniel.Dondorp@hi.no

***Abstract*—Acoustic trawl surveys use echosounders to collect acoustic backscatter, which is categorized and combined with trawl samples to generate abundance indices for fisheries assessment models. Machine learning models are being developed to automate the acoustic target classification step, and it is necessary to evaluate their performance in comparison to manual processes and earlier model versions. The data processing pipeline consists of several stages, utilizing various software, versions, and libraries. Docker containers provide flexibility, especially for advanced pipeline steps. Some steps use Python virtual environments. Clearly defining data models between processing steps is necessary and adopting community standards where applicable is recommended. We have set up a system to combine and run the pipeline steps, and we have used it to compare different ML models. We are currently working to further streamline the process.**

Keywords- acoustic trawl surveys; machine learning; micro services; containerisation; data linage; MLOps; CI/CD.

I. INTRODUCTION

Data from acoustic trawl surveys are an important source of information for fisheries assessments [1]. Data are collected with echosounders mounted on research vessels and individual fish are sampled, usually from trawls. The echosounder data are calibrated and used to enumerate the abundance of fish, whereas the physical samples are used for ground truthing the acoustic registrations and to obtain age and other biological parameters used by the assessment models. More recently, autonomous platforms are being used to augment the acoustic data collection [2].

The process of allocating the acoustic backscatter to species, the Acoustic Target Classification (ATC) step, is in most cases a manual process. There are methods to automate ATC [3], and both the shape of the marks on the echogram [4], as well as the response between different echosounder frequencies [5][6] are being used.

Machine learning methods are well suited for this process, and convolutional neural networks have been developed for ATC [7][8]. Training supervised machine learning models requires training data, and the historical records of manually annotated acoustic data is a rich source of information but requires careful preparations before use. The data sets are highly imbalanced since most of the echograms contain no fish, and different methods for training these methods have been developed [9]. Methods that require less labels during training are also being developed, and both semi supervised and fully supervised models [10][11] are available [12].

To estimate the effect of the different models on the survey results, the different models are run on the same data set and are further processed using the standard data processing pipeline for the survey. This way we can evaluate the effect of the different models and compare them to the official estimates [13].

To efficiently run the data processing pipeline with various configurations and a range of different software and software environments, a digital infrastructure is needed. The objective is to have a pipeline that is modular, has sufficient data provenance, i.e., track the dataset and model used in the different runs, and has the flexibility to be run with different settings and input data to evaluate different methods. The objective of this paper is to describe the current approach, and the plans for extending and operationalizing the pipeline. We start by describing the different steps in the data processing pipeline, followed by how we operationalize it. Finally, we describe the plans and future developments.

II. THE DATA PROCESSING PIPELINE

The different steps in the pipeline (Figure 1) have been described earlier, c.f. the supplementary material of [13], and are briefly summarized here. The different steps have their own git repositories and are versioned and maintained separately.

ISBN: 978-1-68558-329-3

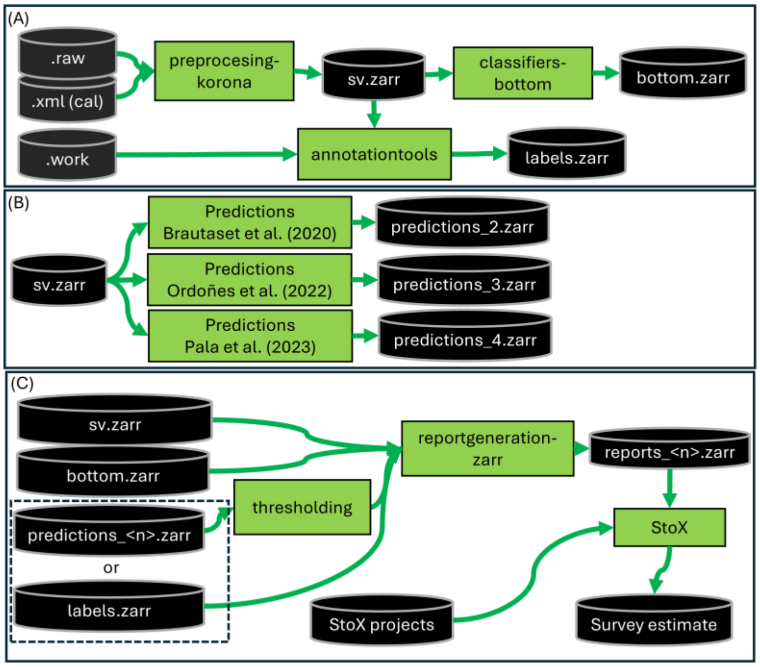


Figure 1. The data procesing pipeline for automated acoustic target classificaiton. Figure reprinted from [13] under CC-BY 4.0. (A) The preprosessing step converting the the raw data and annotations and creating a mask that removes data below the sea bed. (B) The Machine learning models (C) The survey estimation step.

A. Data preprocessing

The data are preprocessed (Figure 1A) from the native data format from the echosounder manufacturer (Kongsberg Discovery, Horten, Norway) to a self-documented gridded format (Figure 2). The data is gridded to the same grid as the primary 38 kHz frequency. We have used the Zarr store [14], which is a chunked cloud friendly format similar to NetCDF. The Python Xarray package [15] can work seamlessly between NetCDF and Zarr, and offer a convenient method for working with the data. The data processing is done using Korona (Marec, Bergen Norway) combined with the python libraries Zarr and Xarray, and results in a three-dimensional gridded data structure with dimensions “range”, “time” and “acoustic frequency”.

The labels are read from the Large Scale Survey System (LSSS) [16] “work” files (Figure 3). LSSS is the standard tool at the Institute of Marine Research (IMR) for annotating the acoustic data. The annotations are converted to the same grid as the echosounder files resulting in a gridded data structure with dimensions “range”, “time” and “acoustic category”, where the “range” and “time” dimensions are identical to the preprocessed echosounder data. A bottom detection algorithm is used to mask the data below the seabed.

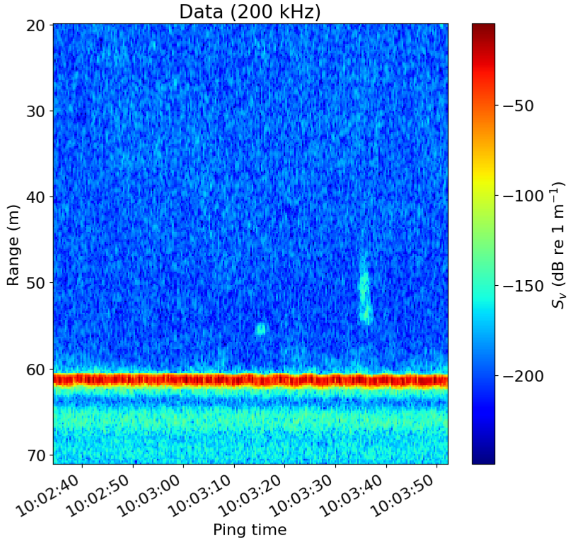


Figure 2. The 200 kHz channel slice in range and time from the regridded data structure.

ISBN: 978-1-68558-329-3

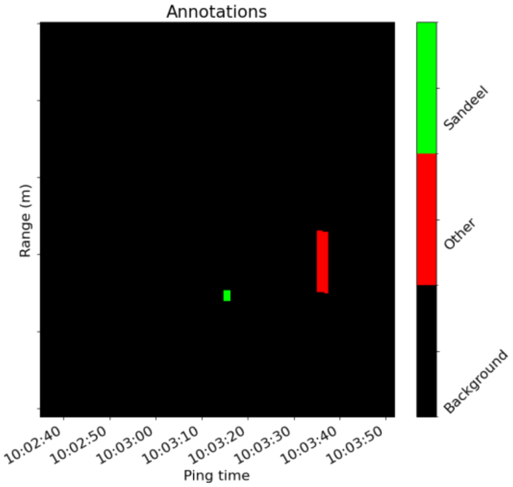


Figure 3. The annotations converted from the LSSS system.

B. Running ML model predictions

Several machine learning models are developed, mostly using PyTorch [17], and trained on segments of processed acoustic data to detect and classify acoustic categories and background. To prevent empty water segments from dominating the training dataset, an effective sampling strategy must be used during training [7][9].

The ML models were trained on a NVIDIA GTX 1080 Ti GPU using the PyTorch framework. The inference is run on a Dell PowerEdge R730 server with 80x Intel(R) Xeon(R) CPU E5-2698, 768GB memory, two Tesla P100-PCIE-12GB graphical processing units running CUDA version 13.0 and rootless docker.

The models return predictions of size 256 x 256 samples, and the data segments are extracted using a sliding window with 40 samples overlap to mitigate edge effects. SoftMax is applied to the predictions to obtain normalized class scores for each sample, and the predictions are subsequently stitched together [7] to a data structure that is the same as the annotations, except that the values are SoftMax predictions instead of binary annotation data.

C. Model thresholding and abundance estimates

To convert the SoftMax predictions to binary masks identical to the annotations, a thresholding approach is applied, c.f. [13] for details. The predictions can then be used interchangeably with the annotations to calculate the integrated backscatter (nautical area scattering coefficient, NASC) in a 0.1 nautical mile by 10 m depth channels.

The survey estimates are calculated using the StoX software [18], where the NASC values from the manual annotation process are replaced with the ML generated reports to evaluate the effect on the survey estimate. This has allowed us to compare the performance of different ML models in relation to the survey estimate. The general impression is that which training years are used are more important than the various model and training approaches, c.f. [13] for more details.

III. OPERATIONALIZING THE PIPELINE

Each step in the pipeline is an individual step connected by data structures shared by the connected components. The individual steps use various software and tools with different programming languages and versions of libraries. Maintaining computational environments for all the individual steps can be challenging to set up in a conventional compute environment, especially when the pipelines are composed of components across different disciplines or organizational units.

In our case, each individual step in the pipeline is currently handled somewhat different depending on its maturity. The preprocessing steps (Figure 1A) are embedded in individual Docker containers, and the Docker image is defined in a versioned Dockerfile that describes the environment. The machine learning models (Figure 1B) and the thresholding and report generation steps (Figure 1C) are run in Python using virtual environments (venv), and the environments are specified in a requirements file in individual git repositories. A working python environment is needed to run these. The survey estimate is packaged in a Docker container running R and StoX [18] and are specified in a Dockerfile.

In cases where a step is fully coded in python and uses a set of python libraries, the use of Python virtual environments works well. However, different environments are used for different steps, and the approach is restricted to python only. Container technologies, e.g., Docker, to package the individual steps are a powerful technology that allows us to set up the compute environment with a range of different software and programming languages. It also allows the different steps in the pipeline to be upgraded and rolled out independently of each other. This is highly flexible and allows a pipeline to be coded up using any programming language or software as long as it can be run in a container, i.e., headless and scripted.

After the individual steps are coded and set up in separate containers or environments, they need to be orchestrated. This must also include documentation of the data provenance, i.e., which versions of the various data and algorithms were used to generate the result. Except for the preprocessing step, we are currently manually running the steps using shell scripts and ad hoc provenance.

We have deployed an automated continuous integration and continuous delivery (CI/CD) pipeline using a GitLab instance running on IMRs servers. This approach improves efficiency and reproducibility and keep track of version control throughout the preprocessing stages. When a new version of the preprocessor is made, the data set is updated. We do not keep earlier versions of the data sets, but we keep track of which version of the preprocessor that was used. The pipeline facilitates the orchestration of the three preprocessing tasks (Figure 1A), where each step is maintained in separate Git repositories and connected to a continuous integration and continuous delivery (CI/CD) pipeline. Each time modifications are made to the repositories, it triggers a job within the pipeline, and this is manually initiated to execute the preprocessing workflow.

The current CI/CD pipeline is effective in converting raw data to preprocessed data. However, it remains static in configuring experiments and managing different combinations of data and processing steps. It lacks flexibility for efficiently modifying or replacing components within the processing chain and for handling data provenance.

IV. Conclusion and Future work

Setting up the pipelines is a work in progress, and the maturity of the different steps varies. The preprocessing pipeline is at a more advanced stage, and we use that to test various approaches for orchestration.

Regardless of technology, the data model that sits between the steps must be agreed upon and specified. This can be a challenging task unless the pipeline is well established. Community data conventions and open data formats should be used where possible. Since the pipeline is scripted, reprocessing the data is straight forward, but downstream code must be adjusted to accommodate changes in data formats.

The containerization of the processing steps is particularly useful since it allows separation of concerns. Different groups and have full flexibility to use and maintain their part of the pipeline as they prefer, but it is important to define the steps to reflect the organizational units.

The CI/CD is still in the works. Currently we have a system for the preprocessing step where the data is updated when a new version of the preprocessor is available. We are currently working on methods to more flexibly set up data processing pipelines where we can use different models or preprocessed data to generate the results.

In conclusion, we have a clearly defined data lineage for the problem. The containerization approach is particularly useful and helps separation of concerns. The data models need to be upgraded to community standards where applicable, and we need a better orchestration of the pipeline to set up and track experiments and test the effects on new algorithms for the data processing.

Acknowledgment

This project is supported by the Research Council of Norway (CRIMAC grant no. 309512).

References

[1] D. MacLennan and E. J. Simmonds, Fisheries Acoustics, 2nd ed. in Fish and aquatic resources series 10. London: Chapman & Hall, 2005.

[2] N. O. Handegard et al., “Uncrewed surface vehicles (USVs) as platforms for fisheries and plankton acoustics,” ICES J. Mar. Sci., pp. 1712–1723,, 2024, doi: 10.1093/icesjms/fsae130.

[3] R. Korneliussen, Ed., “Acoustic target classification,” ICES Coop. Res. Rep., vol. 344, pp. 104, 2018, doi: 10.17895/ices.pub.4567.

[4] D. G. Reid, “Report on Echo Trace Classification,” International Council for the Exploration of the Sea, Copenhagen, Denmark, ICES Cooperative Research Report No. 238, pp. 107, 2000. doi: 10.17895/ices.pub.5371

[5] R. J. Korneliussen and E. Ona, “Synthetic echograms generated from the relative frequency response,” ICES J. Mar. Sci. J. Cons., vol. 60, no. 3, pp. 636–640, 2003, doi: 10.1016/S1054-3139(03)00035-3.

[6] R. J. Kloser, T. Ryan, P. Sakov, A. Williams, and J. A. Koslow, “Species identification in deep water using multiple acoustic frequencies,” Can. J. Fish. Aquat. Sci., vol. 59, pp. 1065–1077, 2002, doi: 10.1139/f02-076.

[7] O. Brautaset et al., “Acoustic classification in multifrequency echosounder data using deep convolutional neural networks,” ICES J. Mar. Sci., vol. 77, no. 4, pp. 1391–1400, 2020, doi: 10.1093/icesjms/fsz235.

[8] A. Ordoñez, I. Utseth, O. Brautaset, R. Korneliussen, and N. O. Handegard, “Evaluation of echosounder data preparation strategies for modern machine learning models,” Fish. Res., vol. 254, pp. 106411, 2022, doi: 10.1016/j.fishres.2022.106411.

[9] A. Pala, A. Oleynik, I. Utseth, and N. O. Handegard, “Addressing class imbalance in deep learning for acoustic target classification,” ICES J. Mar. Sci., vol. 80, pp. 2530–2544, 2023, doi: 10.1093/icesjms/fsad165.

[10] C. Choi et al., “Semi-supervised target classification in multi-frequency echosounder data,” ICES J. Mar. Sci., vol. 78, no. 7, pp. 2615–2627, 2021, doi: 10.1093/icesjms/fsab140.

[11] C. Choi, M. Kampffmeyer, N. O. Handegard, A.-B. Salberg, and R. Jenssen, “Deep Semisupervised Semantic Segmentation in Multifrequency Echosounder Data,” IEEE J. Ocean. Eng., pp. 1–17, 2023, doi: 10.1109/JOE.2022.3226214.

[12] A. Pala, A. Oleynik, K. Malde, and N. O. Handegard, “Self-supervised feature learning for acoustic data analysis,” Ecol. Inform., vol. 84, pp. 102878, 2024, doi: 10.1016/j.ecoinf.2024.102878.

[13] N. O. Handegard, A. J. Holmin, A. Pala, I. Utseth, and E. Johnsen, “Integrating and assessing machine learning acoustic target classification models for fish survey estimations,” ICES J. Mar. Sci., vol. 82, no. 5, pp. 1712–1723,, 2025, doi: 10.1093/icesjms/fsaf069.

[14] Zarr.dev: Zarr core specification. Accessed: Sep. 19, 2025. [Online]. Available from: https://zarr-specs.readthedocs.io/en/latest/v3/core/index.html

[15] S. Hoyer and J. Hamman, “xarray: N-D labeled Arrays and Datasets in Python,” J. Open Res. Softw., vol. 5, no. 1, 2017, doi: 10.5334/jors.148.

[16] R. J. Korneliussen, Y. Heggelund, G. J. Macaulay, D. Patel, E. Johnsen, and I. K. Eliassen, “Acoustic identification of marine species using a feature library,” Methods Oceanogr., vol. 17, pp. 187–205,2016, doi: 10.1016/j.mio.2016.09.002.

[17] A. Paszke et al.: Automatic differentiation in PyTorch, 2017, [Online]. Available from: https://openreview.net/forum?id=BJJsrmfCZ

[18] E. Johnsen et al., “StoX: An open source software for marine survey analyses,” Methods Ecol. Evol., vol. 10, no. 9, pp. 1523–1528, 2019, doi: 10.1111/2041-210X.13250.

ISBN: 978-1-68558-329-3

Semi-Supervised Object Detection for Marine Monitoring using Temporal Information

Viljar Holm Elvevoll[1], Kim Tallaksen Halvorsen[2], and Ketil Malde[1,2]
[1] Department of Informatics, University of Bergen, Norway
[2] Institute of Marine Research, Bergen, Norway
e-mail: viljarhe00@hotmail.no, {kim.halvorsen | ketil.malde}@hi.no

***Abstract*—Accurate and sustainable monitoring of marine biodiversity is crucial for effective fisheries management and conservation. Traditional fish population assessments, relying on manual annotation and invasive techniques, are labor-intensive and potentially harmful to marine ecosystems. This work presents a Semi-Supervised Learning (SSL) approach that leverages extensive unlabeled underwater video data to significantly enhance object detection performance for fish species. By integrating the YOLOv8 object detector with Multi-Object Tracking (MOT) algorithms, specifically ByteTrack, a novel methodology is proposed to generate high-quality pseudolabels from temporal sequences. Iterative training incorporating these pseudolabels consistently improved model precision and recall, with the best-performing approach (ByteTrack with an extrapolated heuristic) demonstrating average precision of 90%, recall of 70%, mAP50 of 74%, and mAP50-95 of 59%. Notably, scores improved substantially over the baseline supervised model on all metrics. These results underscore the potential of temporally informed pseudolabeling in enhancing fish detection accuracy and robustness, reducing reliance on manual annotations and supporting sustainable marine monitoring practices.**

Keywords-Image classifcation; machine learning; species recognition; semi-supervised learning

I. Introduction

Traditionally, methods for monitoring marine ecosystems include trawling, netting, and manual visual surveys by divers, which are labor intensive, costly, and often disruptive to habitats or producing bycatch. Less invasive methods using underwater cameras, such as Baited Remote Underwater Video (BRUV) and Remote Underwater Video (RUV), are often an attractive alternative [1][2]. To process the large volumes of collected video data, it is necessary to use automated analysis tools, typically object detection models [3][4]. However, training such models requires large amounts of high-quality, labeled data, which is costly to produce.

To address this limitation, we here investigate semi-supervised learning [5], an automated method to iteratively generate training data sets using predictions from preliminary models (pseudolabels) that are considered sufficiently reliable. In contrast to earlier work, we selected the pseudolabeled data to use based on temporal information (*i.e.*, tracking) rather than more commonly used confidence scores. This is particularly advantageous in this setting, since an abundance of temporally contiguous video or image data can be produced, but expert annotation is time consuming and requires skilled curators.

The rest of the paper is structured as follows. In Section II, we describe the data set and the method for generating pseudolabels, as well as the training regime. In Section III, we present the results, and select the best performing method to investigate further. In Section IV, we discuss the results and their implications, and propose an explanation for the observations, before we conclude in Section V.

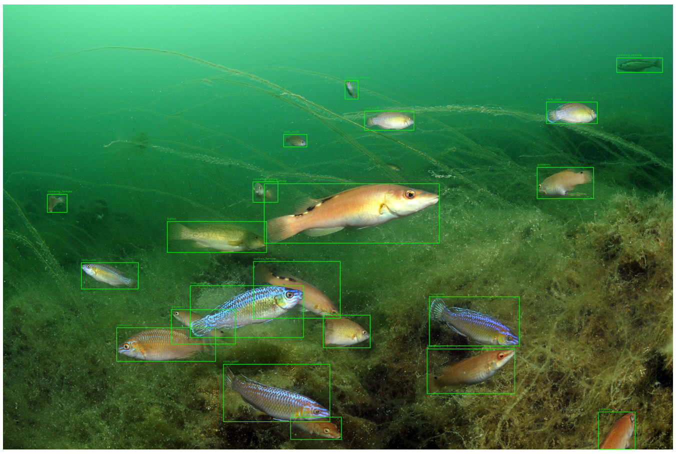


Figure 1. An example from the data set showing several annotated fish of various species (Photo by Erling Svendsen, used with permission).

II. Methods

For this study, we used a data set consisting of 1248 images from a combination of sources (RUVs, photos by divers) under different conditions and with variable resolutions (see Figure 1 for an example). The annotation by experts from the Institute of Marine Research include 10 categories (Figure 2): corkwing wrasse (*Symphodus melops*; male and female), two-spotted goby (*Pomatoschistus flavescens*), goldsinny (*Ctenolabrus rupestris*, rock cook (*Centrolabrus exoletus*, cuckoo wrasse (*Labrus mixtus* male and female), pollack (*Pollachius pollachius*), ballan wrasse (*Labrus bergylta*), and *unknown* fish that could not be labeled to the species level due to low visibility or by being too distant from the camera. In addition, six unannotated videos from similar habitats were used as sources of pseudolabeled frames for the semi-supervised training. As some species were not present or very scarce in the unlabeled data, the semi-supervised method is only trained on five of the classes: male and female corkwing, two-spotted goby, goldsinny, and ballan.

The object detection model used was YOLOv8 (Ultralytics, 2023), a state-of-the-art single pass object detector [6], combined with two advanced tracking algorithms ByteTrack [7] and DeepSort [8] to construct an iterative pipeline:

1. Base Model Training: An initial YOLOv8 model (YOLOv8x, the largest variant) was trained using a smaller,

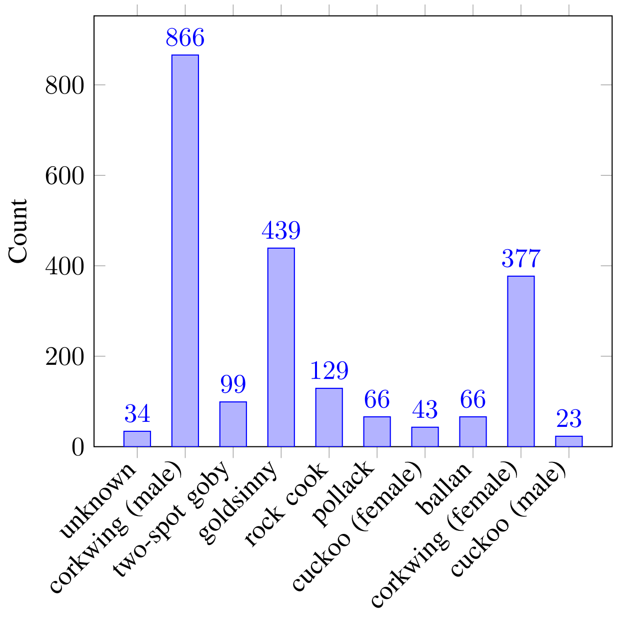


Figure 2. The distribution of the classes of annotated objects in the data set.

manually annotated dataset. This provides the baseline for performance evaluation.

2. Pseudolabel Generation using Temporal Information: The trained model is then used to process unlabeled underwater video recordings and extract pseudolabeled data as illustrated in Figure 3. By integrating Multi-Object Tracking (MOT) algorithms (ByteTrack and DeepSORT), frames with objects that are missed or given low score by the detector can still be identified with high confidence. Two heuristics were investigated for selecting frames to generate pseudolabels:

A. Interpolated Intermediate Labels: This heuristic infers an object's presence in intermediate frames if it is detected by the model in preceding and subsequent frames, and the MOT algorithm assigns the same track ID. This method yields fewer but potentially highly accurate pseudolabels.

B. Extrapolated Labels: This more inclusive approach requires an object to be natively detected at least three times consecutively. All subsequent detections of that object by the MOT algorithm are included, forming an unbroken chain, even if the native model fails to detect it in every frame. This significantly increases the volume of pseudolabeled data.

3. Iterative Retraining: The pseudolabels generated are combined with the original labeled dataset, and the model is retrained. It is crucial to retain the original data to prevent catastrophic forgetting of classes not present in the video recordings. The learning rate of the AdamW optimizer is reset at the start of each new training phase to facilitate rapid adjustment and prevent trapping in suboptimal local minima. This iterative process (pseudolabel generation followed by retraining) is repeated for multiple cycles to progressively improve the model.

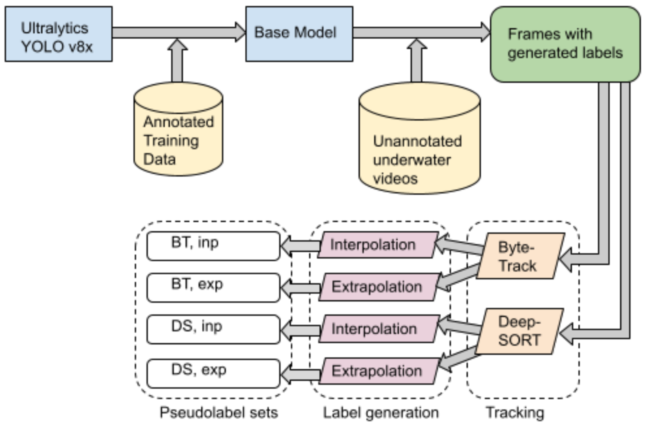


Figure 3. The process for generating the base model and the four pseudolabeled data sets (step 1 and step 2), using automated labeling of video data, tracking, and label interpolation and extrapolation.

TABLE I. PERFORMANCE METRICS AFTER 100 EPOCHS OF TRAINING.

Model	Prec	Recall	mAP50	mAP50-95
DS, inp	0.74988	0.56890	0.61023	0.47017
DS, exp	0.66251	0.56209	0.57969	0.45028
BT, inp	0.70776	0.53875	0.58142	0.44878
BT, exp	0.88792	0.62665	0.69120	0.52972

III. RESULTS

The study rigorously evaluated four configurations: ByteTrack with interpolated intermediate labels, DeepSORT with interpolated intermediate labels, ByteTrack with extrapolated labels, and DeepSORT with extrapolated labels. Each configuration was run for 100 epochs (50 supervised, followed by 50 semi-supervised with pseudolabels) which yielded the results in Table I. We see that all models perform adequately, but ByteTrack with extrapolated labels consistently outperformed the other models.

In order to explore the limits of semi-supervised training, the baseline and ByteTrack with extrapolation models were trained for 250 epochs. In Figure 4, we can see how the different components of the loss rapidly decrease both for training (top row) and validation (bottom row) data, while the four different performance measures increase correspondingly. We also observe five distinct jumps in the graphs, these are caused by introduction of new data and resetting of the learning rate for each iteration, which cause an initial worsening of scores before the model gradually converges again.

The performance statistics on the test set after 250 epochs is shown in Table II. We see that using semi-supervised training with ByteTrack and the extrapolated pseudolabeling scheme results in substantial improvements for all metrics.

Per class improvements are shown in Figure 5. As expected, classes present in the semi-supervised training data (shown in solid colors) see substantial improvements on all metrics. Classes not present (shown with faded colors) see slight degradation in precision, and mAP, but surprisingly recall improves also for these classes.

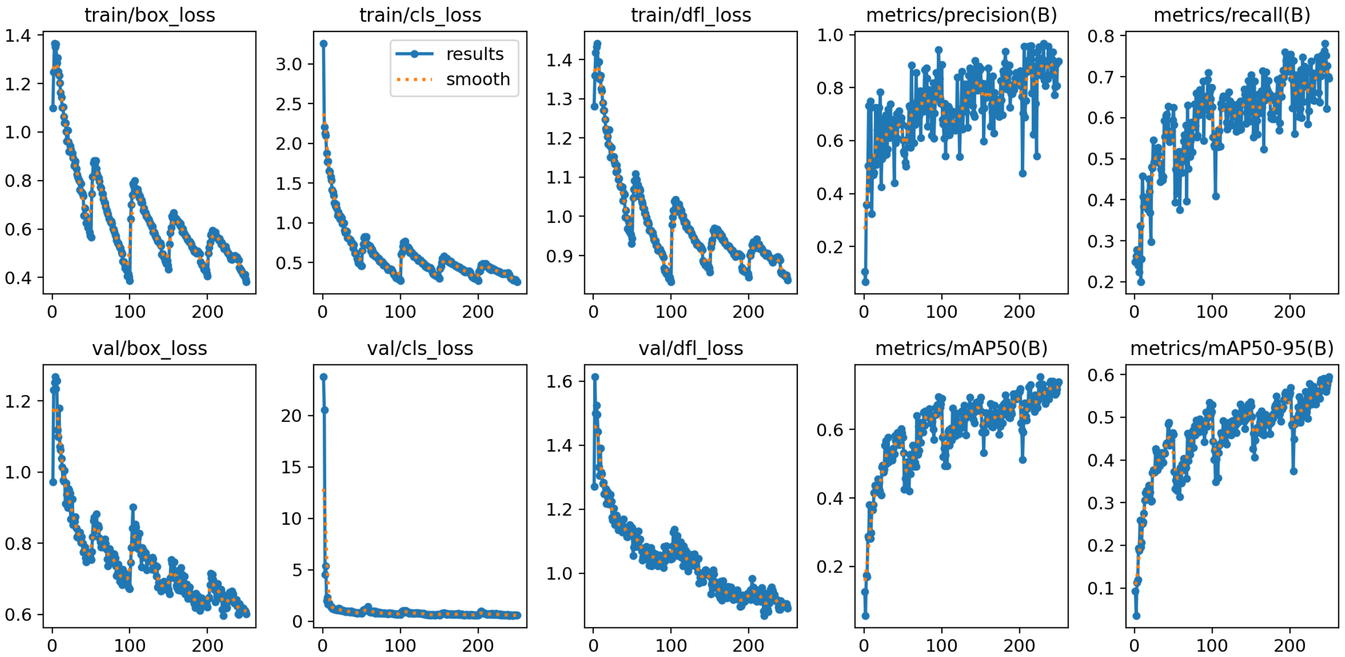


Figure 4. Training from ByteTrack tracks with extrapolated labels for 250 epochs (one iterations of supervised followed by four iterations of semi-supervised training).

TABLE II. Performance of baseline and ByteTrack with extrapolation models after extensive (250 epochs) training.

Model	Prec	Recall	mAP50	mAP50-95
Base	0.75652	0.52468	0.57954	0.48811
BT, exp	0.90011	0.69644	0.73863	0.59468

IV. Discussion

The most effective approach was ByteTrack combined with the extrapolated heuristic. This configuration consistently outperformed all other tested methods, as well as the baseline supervised model. The results demonstrate that leveraging temporal information through pseudolabeling significantly enhances fish detection accuracy and consistency. The substantial improvements in precision, recall, and mAP for the pseudolabeled classes, coupled with minimal negative impact on other classes, validate the effectiveness of this approach in mitigating annotation scarcity.

A crucial insight from this study is the progressive mitigation of initial model biases through iterative pseudolabeling. For instance, a systematic error where parts of the monitoring equipment were misclassified as "corkwing male" in early iterations (Figure 6) was effectively corrected and eliminated in later iterations using ByteTrack with extrapolated labels. This highlights the self-correcting nature of the temporal semi-supervised framework, guiding the model towards more accurate predictions over time.

The choice of MOT algorithm also proved critical. ByteTrack consistently outperformed DeepSORT in this semi-supervised setup. We suspect the discrepancy is caused by the use of Kalman filters in DeepSORT, which can interpolate predictions even when the object is lost by the detection model. While beneficial in predictable scenarios, this can lead to inaccurate pseudolabels for fish due to their often erratic movements, possibly creating an "off-policy" learning situation akin to the "Deadly Triad" in Reinforcement Learning, which can impede stable convergence. ByteTrack, by contrast, relies solely on the detector's predictions, ensuring a stronger alignment between pseudolabels and the model's current capabilities, thus avoiding such instability.

Semi-supervised learning has been used effectively in many different settings, but selecting pseudolabeled data to train on can be difficult. Using classifier confidence is an option [9], but tends in our experience to improve the classifier where it is already strong. Using augmentation [10] or taking class balance into account [11] may help to mitigate this, but by extracting presudolabels from temporal information removes (or at least reduces) the dependence on the classifier itself from the selection process. Although temporal pseudolabeling methods have been attempted before (*e.g.*, [12]), our approach distinguishes itself by targeting the model's weaknesses rather than reinforcing its strengths. By relying on MOT algorithms to generate labels specifically where the base model fails to detect objects, it directly addresses gaps in detection capability.

V. Conclusion and future work

We have successfully demonstrated the significant potential of semi-supervised learning leveraging temporal information for enhancing object detection in marine life monitoring. By integrating YOLOv8 with ByteTrack, a robust methodology was developed to generate high-quality pseudolabels from unlabeled video data, substantially improving model performance

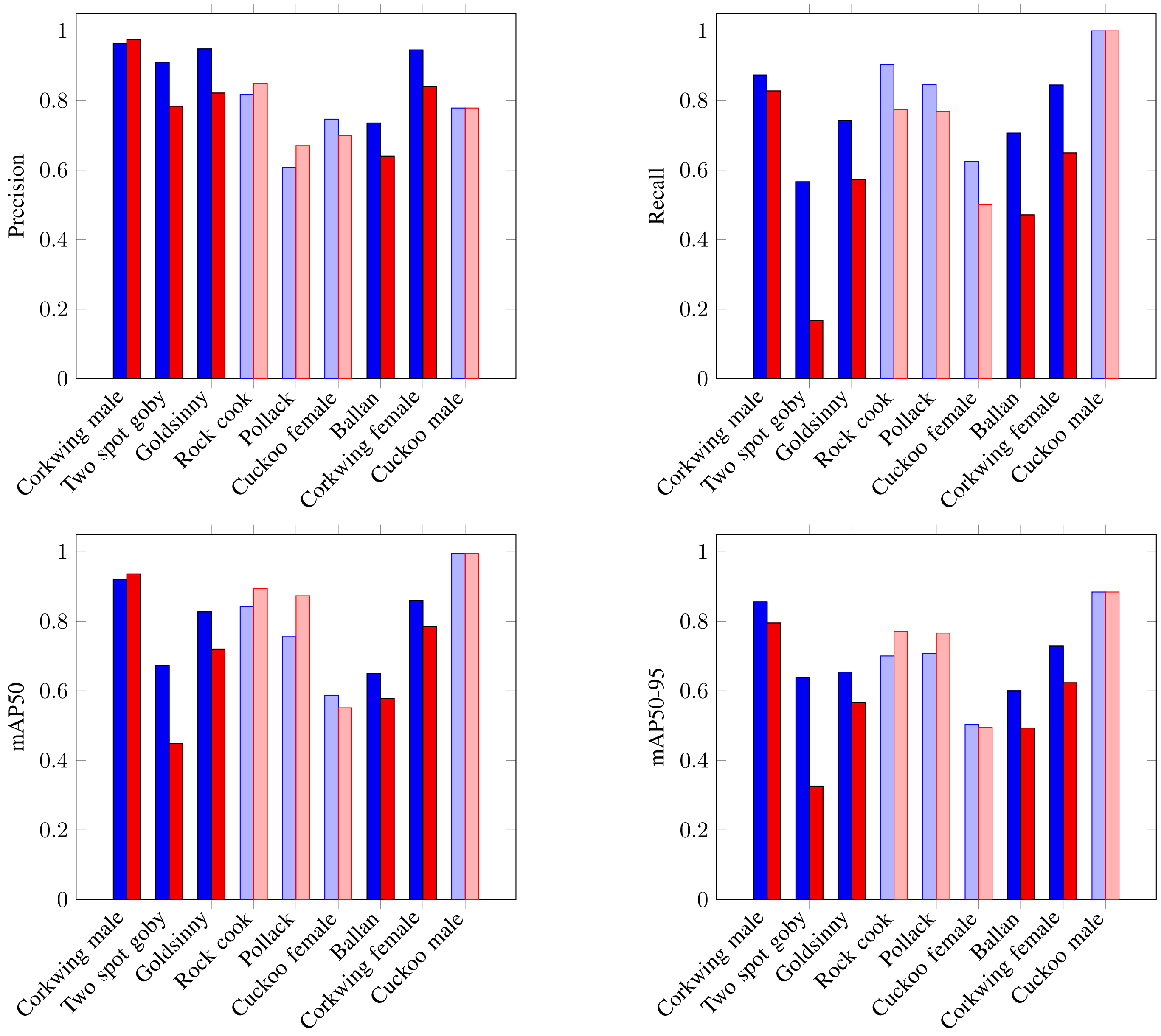


Figure 5. Precision (top left), recall (top right), mAP50 (bottom left) and mAP50-95 (bottom right) for baseline (red) and semi-supervised (blue) models. Classes not present in the pseudolabeled data shown with faded color.

for fish species. This approach reduces the reliance on costly and labor-intensive manual annotations, paving the way for more sustainable and scalable marine life assessment practices. The insights gained regarding iterative bias mitigation and the critical role of MOT algorithm selection provide valuable directions for future research and practical deployment in real-world marine conservation efforts.

Acknowledgement

This work is based on results from the Master's degree thesis of VHE [13], which includes a more detailed exploration of the methods and data. The image data was collected and annotated as part of the CoastVision project, RCN grant number 325862.

References

[1] A. W. Bicknell, B. J. Godley, E. V. Sheehan, S. C. Votier, and M. J. Witt, "Camera technology for monitoring marine biodiversity and human impact," *Frontiers in Ecology and the Environment*, vol. 14, no. 8, pp. 424–432, 2016. DOI: https://doi.org/10.1002/fee.1322. eprint: https://esajournals.onlinelibrary.wiley.com/doi/pdf/10.1002/fee.1322. [Online]. Available: https://esajournals.onlinelibrary.wiley.com/doi/abs/10.1002/fee.1322.

Copyright (c) IARIA, 2025. ISBN: 978-1-68558-329-3

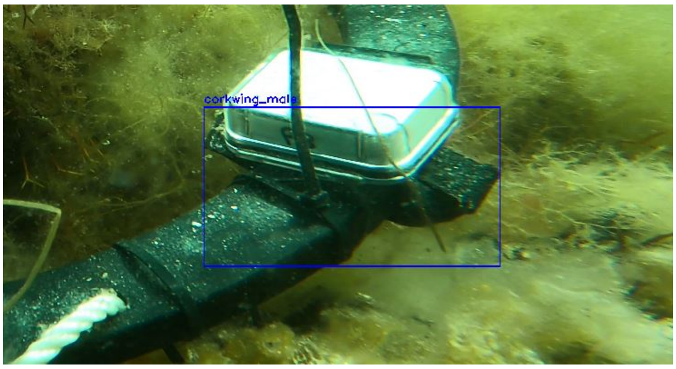


Figure 6. Equipment incorrectly predicted as “corkwing male” by the initial classifier.

[2] S. K. Whitmarsh, P. G. Fairweather, and C. Huveneers, “What is Big BRUVver up to? methods and uses of baited underwater video,” *Reviews in Fish Biology and Fisheries*, vol. 27, no. 1, pp. 53–73, 2017, ISSN: 1573-5184. DOI: 10.1007/s11160-016-9450-1. [Online]. Available: https://doi.org/10.1007/s11160-016-9450-1.

[3] H. Liu, X. Ma, Y. Yu, L. Wang, and L. Hao, “Application of deep learning-based object detection techniques in fish aquaculture: A review,” *Journal of Marine Science and Engineering*, vol. 11, no. 4, p. 867, 2023.

[4] P. Rubbens et al., “Machine learning in marine ecology: An overview of techniques and applications,” *ICES Journal of Marine Science*, vol. 80, no. 7, pp. 1829–1853, 2023.

[5] J. E. Van Engelen and H. H. Hoos, “A survey on semi-supervised learning,” *Machine learning*, vol. 109, no. 2, pp. 373–440, 2020.

[6] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, *You only look once: Unified, real-time object detection*, 2016. arXiv: 1506.02640 [cs.CV]. [Online]. Available: https://arxiv.org/abs/1506.02640.

[7] Y. Zhang et al., *Bytetrack: Multi-object tracking by associating every detection box*, 2022. arXiv: 2110.06864 [cs.CV]. [Online]. Available: https://arxiv.org/abs/2110.06864.

[8] N. Wojke, A. Bewley, and D. Paulus, *Simple online and realtime tracking with a deep association metric*, 2017. arXiv: 1703.07402 [cs.CV]. [Online]. Available: https://arxiv.org/abs/1703.07402.

[9] K. Sohn et al., “A simple semi-supervised learning framework for object detection,” *arXiv preprint arXiv:2005.04757*, 2020.

[10] J. Jeong, S. Lee, J. Kim, and N. Kwak, “Consistency-based semi-supervised learning for object detection,” *Advances in neural information processing systems*, vol. 32, 2019.

[11] M. Xu et al., “End-to-end semi-supervised object detection with soft teacher,” in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 3060–3069.

[12] R. J. Veiga et al., “Autonomous temporal pseudo-labeling for fish detection,” *Applied Sciences*, vol. 12, no. 12, p. 5910, 2022.

[13] V. H. Elvevoll, “Semi-supervised object detection using temporal information,” M.S. thesis, Department of Informatics, The University of Bergen, 2025.

Machine Learning Using Physics-Based Backscattering Modeling for Acoustic Identification of Mesopelagic Organisms

Sander Andre Berg Marx[1,*] , Babak Khodabandeloo[2,†], Ketil Malde[1,2,‡]
[1]Department of Informatics, University of Bergen, Bergen, Norway
[2]Acoustics and Observation Methodologies, Institute of Marine Research (IMR), Bergen, Norway
[*] email: sanderamarx@hotmail.com
[†] email: babak.khodabandeloo@hi.no
[‡] email: ketil.malde@hi.no

Abstract— **Target Strength (TS) is the logarithmic measure of the backscattered acoustic energy reflected toward a sound source when an acoustic wave encounters an organism. It depends on the organism's size, shape, material properties, orientation, and the frequency of the wave, and thus carries information useful for identifying and characterizing marine organisms. Advanced broadband echosounders now allow detailed TS measurements of marine organisms over nearly continuous frequency ranges, which provide valuable information for biomass estimation and ecosystem monitoring. However, interpreting these TS measurements has traditionally relied on manual classification, which makes it difficult to extract biological characteristics for target classification or ecological analysis, especially given the complexity of broadband data. Physics-based backscattering models are versatile tools for modeling the TS frequency response given the shape and material properties of the scatterers. In this study, we employ an exact prolate spheroid model, representative of many marine organisms, to simulate broadband TS spectra for training machine learning models. These models aim to classify and characterize targets based on their TS frequency signatures. A hybrid one-Dimensional Convolutional Neural Network (1D-CNN) is proposed for the simultaneous classification (gas- vs. liquid-filled) and regression of geometric properties and compared against K-Nearest Neighbors (KNN), Support Vector Machine (SVM), and Random Forrest (RF). Results show that while all models achieved perfect classification accuracy, the hybrid 1D-CNN clearly outperformed the others in parameter estimation. This demonstrates that simulation-driven machine learning can help overcome data scarcity and enable automated acoustic identification of mesopelagic organisms.**

Keywords-Acoustic target classification; machine learning; Convolutional neural network; prolate spheroid backscattering modeling.

I. INTRODUCTION

Active acoustics is a versatile tool for monitoring marine life, offering unrivaled spatial and temporal resolution compared to other methods, such as net-based biological sampling, optical systems, or video recording. However, interpreting the collected echoes to classify organism size and species remains challenging. Converting acoustic data to biomass is especially difficult when the insonified volume contains mixed species and/or a diverse size distribution within the same species [1][2][3]. Broadband echosounders provide detailed spectra, but these are difficult to interpret manually, making large-scale analysis time-consuming and thereby motivating automated approaches.

When an acoustic wave encounters organisms along its propagation path, the acoustic energy scatters in different patterns depending on the organism's size, shape, material properties, orientation, and the frequency of the wave [4][5]. A portion of this energy is reflected back toward the source, referred to as backscattering. Therefore, the backscattered signal—or its logarithmic measure, known as Target Strength (TS)—carries information that can be used to identify the object from which it originated. Since the TS of an organism varies with frequency, measuring TS over a broad, continuous frequency band provides more detailed information about the scatterers, thereby enhancing the ability to characterize organisms [6]. Broadband fisheries echosounders transmit frequency-modulated pulses, offering two main advantages [7][8]: (1) measurement across a nearly continuous frequency range, and (2) enhanced range resolution through pulse compression, which improves single-target detection. Therefore, the use of broadband echosounders improves the ability to identify organisms and to resolve their spatial distribution within aggregations, both of which are crucial for interpreting and converting acoustic echoes into meaningful biological information.

Backscattering models are an essential tool for interpreting measured backscattered data from marine organisms [9]. Techniques to model acoustic backscattering range from analytical to numerical models. Numerical models can accommodate arbitrary geometries and inhomogeneous material properties, allowing detailed and realistic simulations [10]. However, their high computational cost can limit their practical use. Although most marine organisms have complex geometries and consist of inhomogeneous materials, backscattering models from canonical geometries with homogeneous material properties are often sufficient to acoustically represent key characteristics such as size, elongation, and orientation. For example, spherical models have long been used in fisheries acoustics due to their simplicity [11][12]. However, spherical models cannot capture elongation, which is a critical parameter for accurately representing many marine organisms or their main acoustic reflecting organ, e.g. gas-bladder.

Machine Learning (ML) techniques can be used to classify targets based on trained models that learn to find patterns within the broadband TS spectra. However, large, accurately labeled datasets are needed to develop robust machine learning models [13]. Acquiring such datasets is challenging and expensive in real-world marine environments [14].

While real acoustic datasets exist, they rarely provide reliable ground truth. To overcome this limitation, we use a physics-based simulation model to generate large synthetic datasets for training. On this basis, a novel hybrid one-Dimensional Convolutional Neural Network (1D-CNN) is proposed for simultaneous classification (gas-filled vs liquid) and regression of geometric properties such as size, elongation and orientation. The performance of the 1D-CNN is then compared against traditional ML models.

The remainder of the paper is organized as follows. Section II describes the methods for data generation and machine learning model training. Section III reports the results, and Section IV provides a discussion of the findings. Section V concludes the paper and outlines directions for future work.

II. METHOD

A. Physics-based acoustic backscattering modeling for synthetic data generation

To generate synthetic broadband backscattering data from objects representing mesopelagic marine organisms, with and without gas bladders, we employed an optimized version of the fluid-filled prolate spheroid backscattering model [15]. This physics-based model provides accurate backscattering for liquid—and gas—filled targets over a wide frequency range and for all incident angles, i.e. relative orientation of the prolate spheroid axis to the direction of wave propagation.

The prolate spheroids are represented by their volume (quantified by the equivalent spherical radius R_{eq}) and elongation (or aspect ratio α) instead of directly using semi major and minor axes, *a* and *b*, respectively. They are related by the following equations:

$$b = R_{eq}/(\alpha^{1/3}), \qquad (1)$$

$$a = \alpha \times b\ . \qquad (2)$$

This parameterization facilitates direct control over the size and shape of the modeled organisms, making it easier to explore a wide range of biologically relevant geometries. The range of parameters used to model different organisms are given in Table I.

TABLE I. PARAMETER RANGES USED FOR GENERATING SYNTHETIC ACOUSTIC DATA

Parameter	Targets	
	Gas-filled	**Liquid-filled**
Equivalent radius (*Req*)	0.1–5 mm	5–20 mm
Aspect ratio (α)	1.05–8.0	1.05–8.0
Spheroid Density (ρ_s)	2–80 kg/m^3	1.01–1.07 × (ρ_w) kg/m^3
Water Density (ρ_w)	1027 kg/m^3	1027 kg/m^3
Spheroid Sound Speed (c_s)	343 m/s	1.01-1.07 × (c_w) m/s
Water Sound Speed (c_w)	1500 m/s	1500 m/s
Incident angle (θ)	0.01–90°	0.01–90°
Frequency range	10–260 kHz (0.5 kHz steps)	

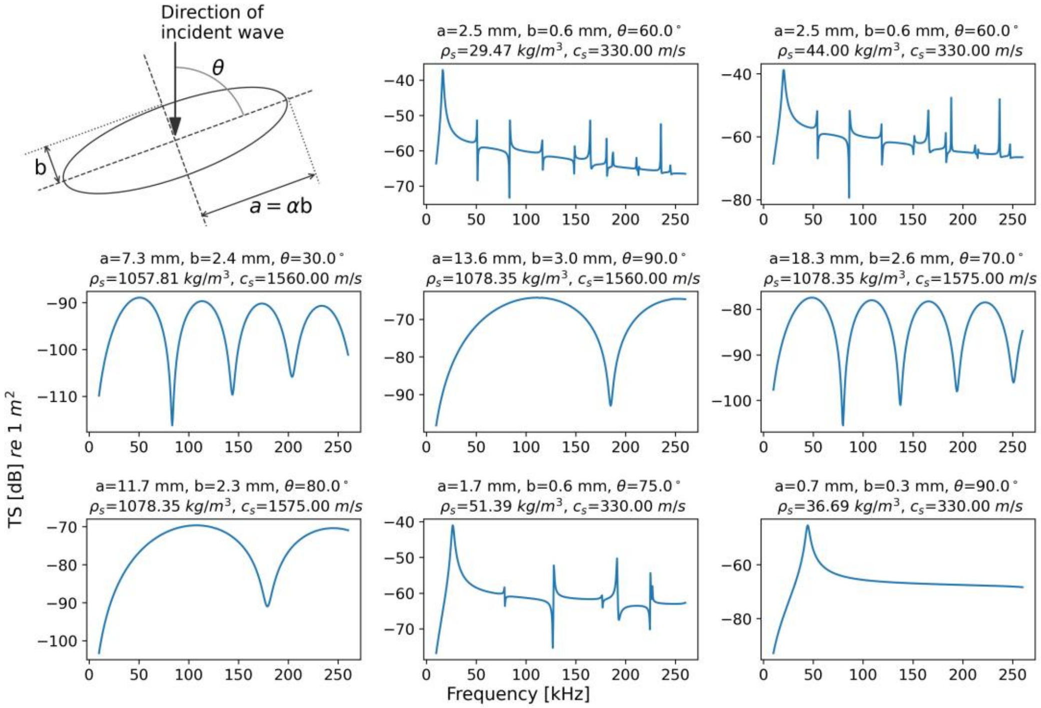


Figure 1. Examples of TS(f) for prolate spheroids with parameters within the ranges in Table I. An illustration of a prolate spheroid with semi-major and semi-minor axes, *a* and *b* respectively, along with the incident angle θ, is shown in the upper left corner.

Copyright (c) IARIA, 2025. ISBN: 978-1-68558-329-3

B. Data Preprocessing and Machine Learning Setup

The synthetic acoustic data generated by the prolate spheroid backscattering model underwent several preprocessing steps to prepare it for machine learning. The dataset consists of 30,000 simulated TS spectra, 15,000 liquid-filled and 15,000 gas-filled targets.

Each target is represented by a feature vector with corresponding labels. The input features consist of the broadband TS spectra. Each spectrum is represented as a vector of TS values, sampled at 0.5 kHz steps across the frequency range of 10 to 260 kHz. The models were trained to predict a classification label (liquid-filled or gas-filled) and three regression labels corresponding to the geometric properties of the prolate spheroid: the incident angle (θ), the semi-major axis (a), and the semi-minor axis (b).

Normalization procedures are applied separately to the feature and label columns to ensure consistent scaling and improve model stability [16]. Each frequency TS value is standardized independently using training set statistics, transforming the data to have zero mean and unit variance. The continuous regression labels (θ, a and b) are also normalized to ensure that each parameter contributes equally to model training regardless of original scale.

The complete dataset of 30,000 target instances is then partitioned into training, validation, and test subsets using an 80/10/10 split. The training set (80%) is used for training the ML models and deriving normalization statistics. The validation set (10%) is used for hyperparameter tuning and model selection during training, as well as for monitoring overfitting. Finally, the test set (10%) is reserved for the final, unbiased evaluation of the trained models' performance.

C. Machine learning models

To classify the type and estimate geometric parameters from the TS spectra, a hybrid 1D-CNN was developed. The architecture was chosen for its effectiveness in automatically extracting hierarchical features from sequential data, like acoustic frequency spectra [17].

The network architecture, illustrated in Figure 2, was designed to process sequential TS data. The architecture incorporates advanced deep learning components, specifically residual blocks [18] and Squeeze-and-Excitation (SE) attention mechanisms [20], to improve feature extraction and training stability. The TS spectra is first processed through an initial 1D convolution layer, followed by a series of residual blocks. The residual blocks contain two 1D convolutional layers, batch normalization [19] and ReLU activations. Also integrated in these residual blocks are Squeeze and Excitation (SE) attention mechanisms that perform channel-wise feature weighting by compressing information through global average pooling and then learning channel relationships via two fully connected layers. The network ends in fully connected layers that lead to a multi-head output structure with dedicated heads for classification and regression. One head for classification of target type, one for regression of the incident angle (θ) and one for regression of the size parameters (a and b).

For comparative analysis of the performance of the 1D-CNN, three additional traditional machine learning models previously used in acoustic target classification research were tested.

- K-Nearest Neighbors (KNN) [21]: An instance-based learner classifying targets based on the majority class of their k nearest neighbors in the feature space, as applied by Cotter et al. [22].
- Support Vector Machine (SVM) [23]: Seeking an optimal hyperplane to separate classes or predict continuous values, utilized by Yang et al. [24] for underwater target recognition.
- Random Forest (RF) [25]: An ensemble method constructing multiple decision trees on random subsets of data and features, aggregating their predictions, as employed by Gugele et al. [26].

These traditional models were not hyperparameter tuned and served as a baseline to evaluate the 1D-CNN.

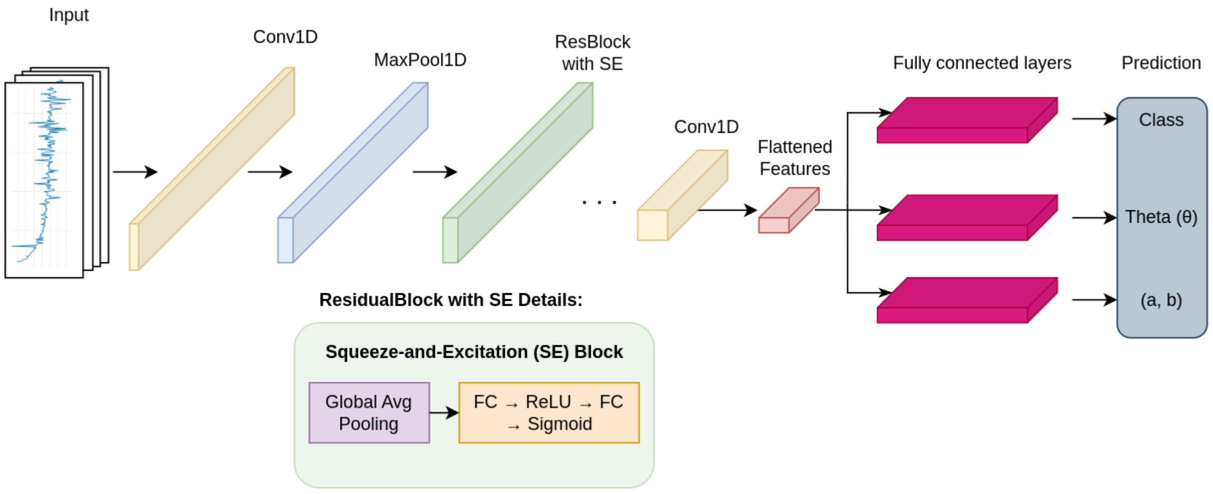


Figure 2. Hybrid one-dimensional convolutional neural network architecture for acoustic target classification.

Copyright (c) IARIA, 2025. ISBN: 978-1-68558-329-3

D. *Training and evaluation*

The 1D-CNN was trained using a custom multi-component loss function designed to address both the classification and regression tasks. The total loss (Λ) is a weighted sum of four components as shown in (3):

$$\Lambda = \omega_c \cdot \Lambda_c + \omega_r \cdot (\Lambda_\theta + \Lambda_{ab}) + \omega_{cons} \cdot \Lambda_{cons}. \qquad (3)$$

where Λ_c is cross-entropy loss for the binary classification task. For regression, Huber loss [27] was used for the angle (Λ_θ) and Mean Squared Error (MSE) for the size parameters (Λ_{ab}). To enforce the geometric relationship between the semi-major and minor axes, a consistency loss ($\Lambda_{cons.}$), also based on Huber loss, was applied by comparing the predicted aspect ratio ($\alpha_{pred} = a_{pred} / b_{pred}$) to its ground truth value. The weights were determined through hyperparameter tuning and set to $\omega_c = 0.1$, $\omega_r = 2.0$, and $\omega_{cons} = 0.5$ to prioritize the more challenging regression task.

Model performance was evaluated on the unseen test set. Classification performance was measured by accuracy, while regression performance was assessed using the coefficient of determination (R^2) and Root Mean Squared Error (RMSE) for θ, a, b, and the derived α. A composite score, averaging the classification accuracy and the R^2 scores of the three primary regression targets, was used for overall model comparison.

III. RESULTS

A. *Performance Comparison Across Models*

Using the simulated backscattering frequency responses, we first classified the targets (i.e., liquid- or gas-filled), then estimated the geometrical properties (semi-major and semi-minor axes) and the incident angle of the prolate spheroids based on their wideband target strength frequency responses, using different ML models. Since the data are simulated, the true model parameters are known, allowing us to quantitatively evaluate the estimates produced by the various ML models. The performance of the models in both classification and morphological parameter estimation is summarized in Table II.

TABLE II. COMPARATIVE REGRESSION PERFORMANCE (R^2 SCORES), CLASSIFICATION ACCURACY AND COMPOSITE SCORE ON SYNTHETIC TEST DATA

Parameter	SVM	KNN	RF	1D-CNN
R^2 for θ	0.742	0.751	0.853	**0.974**
R^2 for a	0.795	0.898	0.902	**0.997**
R^2 for b	0.911	0.931	0.942	**0.996**
Clas. Acc	1.000	1.000	1.000	**1.000**
Composite score	0.719	0.804	0.867	**0.992**

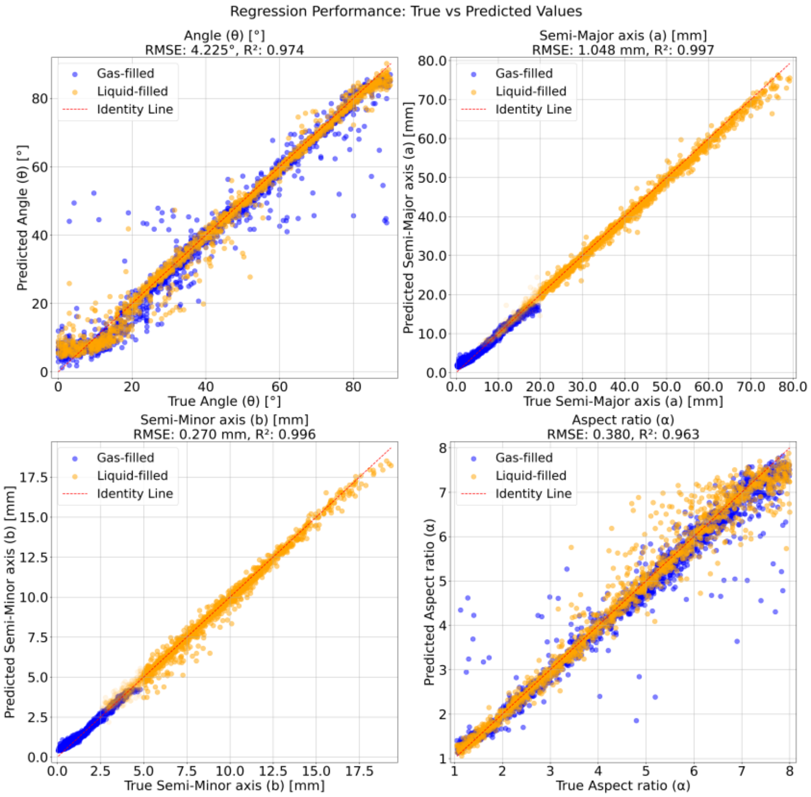


Figure 3. Regression performance of the hybrid 1D-CNN model on test data showing true values against predicted values for (top-left) angle (θ), (top-right) semi-major axis (a), (bottom-left) semi-minor axis (b), and (bottom-right) dervided aspect ratio (α = a/b). Blue points represent gas-filled targets, and orange points represent liquid targets.

B. *Detailed performance of Hybrid 1d-CNN*

All tested models achieved a perfect classification accuracy of 1.00, correctly identifying every target as either gas-filled or liquid-filled. The primary differences between the models emerged in the regression tasks. Given its superior regression performance, the results of the hybrid 1D-CNN were analyzed in further detail. Figure 3 presents scatter plots of the model's predicted values against the true values for each regression target.

The model demonstrated high precision in predicting the semi-major axis a and semi-minor axis b; data points were tightly clustered along the identity line for both gas-filled and liquid-filled targets, indicating the model's robustness across target types and size ranges. Predictions for the incident angle θ were also strong, though with more visible scatter compared to the size parameters a and b, with notable gas-filled outliers. The derived aspect ratio ($\alpha=a/b$) also showed good performance, but with some increased scatter for targets with higher aspect ratios ($\alpha > 4$).

C. *Analysis of prediction outliers*

Outlier analysis revealed consistent patterns for both incident angle (θ) and aspect ratio (α).

For incident angle (θ) estimation, the largest errors occurred with small, gas-filled targets with simple TS spectra containing few distinct resonance peaks. The model therefore defaulted toward predicting the training mean (45°), creating the horizontal band of outliers seen in Figure 3. In contrast, liquid-filled outliers showed smaller angle errors and were typically targets with low aspect ratio ($\alpha \approx 1.05$), where the orientation is less defined physically as the target approaches a perfect sphere.

A similar pattern emerged for aspect ratio (α) prediction, where most outliers were again small, gas-filled targets with spectrally simple signatures. This result aligns with the physical principle that accurate estimation of elongation requires at least two distinct resonance peaks [28], a feature these outlier spectra lacked.

IV. DISCUSSION

The tested machine learning models, including shallow learners (KNN, SVM, and RF) and a deep learner (1D-CNN), successfully classified gas- versus liquid-filled targets. This is primarily due to the distinct differences in the TS(f) responses between gas- and liquid-filled targets, as observed in Figure 1. In contrast, the models showed varying levels of performance in the regression task. Among them, the 1D-CNN demonstrated the highest accuracy in estimating the model parameters, evidenced by high R^2 scores for a, b, and θ (see Table II). However, some outliers were observed in the predictions, where the parameters were not correctly estimated. This was especially the case for gas-filled targets in the prediction of angle θ and the computed aspect ratio α. Further investigation of the outliers revealed that they mostly corresponded to small targets (i.e. small R_{eq}) (see [29] for more details). The TS frequency response of small gas-filled targets is known to be insensitive to shape and incident angle near the resonance frequency [30], explaining the model's difficulty in the 10–260 kHz frequency band. An example TS(*f*) response of such small targets is shown in the lower right panel of Figure 1. For deep learning approaches such as neural networks, large amounts of data are typically required; therefore, the use of optimized code for computing TS(*f*) was critical. Although the shallow learners achieved lower R^2 scores, it is important to note that less effort was devoted to tuning these models compared to the 1D-CNN. It is possible that more extensive hyperparameter optimization and validation could lead to improved performance for the shallow models.

V. CONCLUSION AND FUTURE WORK

This study demonstrates that using machine learning, more particularly a 1D-CNN trained exclusively on physics-based simulated data, can accurately classify and estimate geometric properties of acoustic targets. We demonstrated that the hybrid 1D-CNN outperforms traditional machine learning methods such as KNN, SVM, and RF. While all models achieved perfect classification of gas- and liquid-filled targets, the 1D-CNN was notably more accurate in the regression task of estimating the targets' geometric properties, including semi-major/minor axes (a, b) and incident angle (θ).

Outlier analysis revealed specific challenges, particularly in estimating geometric parameters of small, gas-filled targets, with simpler spectral features lacking distinct resonance peaks. Addressing these limitations by expanding frequency ranges or incorporating additional acoustic parameters in future research could enhance performance further. Although this study focused on simulated data, preliminary tests on a small real dataset [29] showed encouraging results, indicating potential for application to in situ measurements.

We conclude that this simulation-driven approach is a powerful and viable strategy for overcoming data-scarcity in marine acoustics. It represents a promising step toward the automated, non-invasive classification of marine organisms, with potential application in real-time classification of mesopelagic organisms during acoustic surveys, supporting biomass estimation and reducing the need for manual analysis.

ACKNOWLEDGMENT

S. M. acknowledges the use of the "Birget" high-performance computing cluster, provided by the Department of Informatics, University of Bergen, which was essential for this research. B.K. and K.M. gratefully acknowledge funding from the Center for Research-based Innovation in Marine Acoustic Abundance Estimation and Backscatter Classification (CRIMAC, Project No. 309512; www.crimac.no). This work is based on the master’s thesis of S.M. conducted at the University of Bergen.

REFERENCES

[1] C. F. Greenlaw, "Acoustical estimation of zoo-plankton populations", Limnol. Oceanogr., vol. 24, pp. 226–242, 1979.

[2] A. C. Lavery et al., "Determining dominant scatterers of sound in mixed zooplankton populations", J. Acoust. Soc. Am. vol. 122, pp. 3304–3326, 2007.

[3] R. J. Kloser, T. E. Ryan, J. W. Young, and M. E. Lewis, "Acoustic observations of micronekton fish on the scale of an ocean basin: potential and challenges", ICES J. Mar. Sci., vol. 66, pp. 998–1006, 2009, https://doi.org/10.1093/icesjms/fsp077.

[4] R. Hickling, "Analysis of echoes from a solid elastic sphere in water", J. Acoust. Soc. Am., vol. 34, pp. 1582–1592, 1962, https://doi.org/10.1121/1.1909055.

[5] J.J. Faran, "Sound scattering by solid cylinders and spheres", J. Acoust. Soc. Am., vol. 23, pp. 405–417, 1951, https://doi.org/10.1121/1.1906780.

[6] D. Chu and T.K. Stanton, "Application of pulse compression techniques to broadband acoustic scattering by live individual zooplankton", J. Acoust. Soc. Am., vol. 104, pp. 39–55, 1998, https://doi.org/10.1121/1.424056.

[7] M. E. Zakharia, "Wide band fisheries sounder; From individual echoes analysis to classification of schools at sea", J. Acoust. Soc. Am., vol. 103, pp. 3068–3068, 1998, https://doi.org/10.1121/1.422849.

[8] J. E. Ehrenberg, and T. C. Torkelson, "FM slide (chirp) signals: a technique for significantly improving the signal-to-noise performance in hydroacoustics assessment systems", Fisheries Research vol. 47, pp. 193–199, 2000.

[9] A. C. Lavery, T. K. Stanton, D. E. McGehee, and D. Chu, "Three-dimensional modeling of acoustic backscattering from fluid-like zooplankton", J. Acoust. Soc. Am., vol. 111, pp. 1197–1210, 2002 https://doi.org/10.1121/1.1433813.

[10] J. M. Jech et al., "Comparisons among ten models of acoustic backscattering used in aquatic ecosystem research", J. Acoust. Soc. Am., vol. 138, pp. 3742–3764, 2015, https://doi.org/10.1121/1.4937607.

[11] H. Medwin, "Sounds in the sea: From ocean acoustics to acoustical oceanography", Cambridge University Press, 2005.

[12] R. H. Love, "Resonant acoustic scattering by swimbladder-bearing fish", J. Acoust. Soc. Am., vol. 64, pp. 571–580, 1978.

[13] L. Alzubaidi et al., "A survey on deep learning tools dealing with data scarcity: definitions, challenges, solutions, tips, and applications", Journal of Big Data, vol. 10, art. 46, 2023, https://doi.org/10.1186/s40537-023-00727-2.

[14] R. Kubilius, G. J. Macaulay, and E. Ona, "Remote sizing of fish-like targets using broadband acoustics", Fisheries Research vol. 228, art. 105568, 2020, https://doi.org/10.1016/j.fishres.2020.105568.

[15] B. Khodabandeloo, Y. Heggelund, B. Ystad, S. A. B. Marx, and G. Pedersen, "High-precision model and open-source software for acoustic backscattering by liquid- and gas-filled prolate spheroids across a wide frequency range and incident angles: Implications for fisheries acoustics", Journal of Sound and Vibration vol. 616, art. 119227, 2025, https://doi.org/10.1016/j.jsv.2025.119227.

[16] D. Singh and B. Singh, "Investigating the impact of data normalization on classification performance", Applied Soft Computing vol. 97, art. 105524, https://doi.org/10.1016/j.asoc.2019.105524.

[17] S. Kiranyaz et al., "1D convolutional neural networks and applications: A survey", Mechanical Systems and Signal Processing, vol. 151, art. 107398, 2021, https://doi.org/10.1016/j.ymssp.2020.107398.

[18] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition", IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770–778, 2016, https://doi.org/10.1109/CVPR.2016.90.

[19] S. Ioffe, and C. Szegedy, "Batch normalization: accelerating deep network training by reducing internal covariate shift", Proceedings of the 32nd International Conference on International Conference on Machine Learning, vol. 37, pp. 448–456, 2015.

[20] J. Hu, L. Shen, and G. Sun, "Squeeze-and-Excitation Networks", IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 7132–7141, 2018, https://doi.org/10.1109/CVPR.2018.00745.

[21] E. Fix, and J. L. Hodges, "Discriminatory Analysis. Nonparametric Discrimination: Consistency Properties", International Statistical Review / Revue Internationale de Statistique, vol. 57, pp. 238–247, 1989 https://doi.org/10.2307/1403797.

[22] E. Cotter, C. Bassett, and A. Lavery, "Classification of broadband target spectra in the mesopelagic using physics-informed machine learning", The Journal of the Acoustical Society of America, vol. 149, pp. 3889–3901, 2021, https://doi.org/10.1121/10.0005114.

[23] M. A. Hearst, S. T. Dumais, E. Osuna, J. Platt, and B. Scholkopf, "Support vector machines", IEEE Intelligent Systems and Their Applications, vol. 13, pp. 18–28, 1998, https://doi.org/10.1109/5254.708428.

[24] H. Yang et al., "Underwater acoustic target recognition using SVM ensemble via weighted sample and feature selection", 13th International Bhurban Conference on Applied Sciences and Technology (IBCAST), pp. 522–527, 2016, https://doi.org/10.1109/IBCAST.2016.7429928.

[25] L. Breiman, "Random Forests", Machine Learning, vol. 45, pp. 5–32, 2001, https://doi.org/10.1023/A:1010933404324.

[26] S. M. Gugele et al., "Differentiation of two swim bladdered fish species using next generation wideband hydroacoustics", Sci Rep, vol. 11, art. 10520, 2021, https://doi.org/10.1038/s41598-021-89941-7.

[27] P. J. Huber, "Robust Estimation of a Location Parameter," Ann. Math. Statist. vol. 35, pp. 73 - 101, 1964, https://doi.org/10.1214/aoms/1177703732

[28] B. Khodabandeloo et al., "Mesopelagic fish gas bladder elongation, as estimated from wideband acoustic backscattering measurements", J. Acoust. Soc. Am., vol. 151, pp. 4073–4085, 2022, https://doi.org/10.1121/10.0011742.

[29] S. A. B. Marx, "Learning acoustic target classification from simulation", Master's Thesis, University of Bergen, 2025.

[30] B. Khodabandeloo, M. D. Agersted, T. A. Klevjer, G. Pedersen, and W. Melle, "Mesopelagic flesh shear viscosity estimation from in situ broadband backscattering measurements by a viscous–elastic model inversion", ICES Journal of Marine Science, vol. 78, no. 9, pp. 3147-3161, 2021, https://doi.org/10.1093/icesjms/fsab183.

Accelerating HEC-RAS: A Recurrent Neural Operator for Rapid River Forecasting

Edward Holmberg
GS Center for Env. Informatics
University of New Orleans,
Louisiana, USA
eholmber@uno.edu

Pujan Pokhrel
GS Center for Env. Informatics
University of New Orleans,
Louisiana, USA
ppokhre1@uno.edu

Elias Ioup
Center for Geospatial Sciences
Naval Research Laboratory,
Mississippi, USA
elias.z.ioup.civ@us.navy.mil

Ken Pathak
US Army Corps of Engineers
Vicksburg District,
Mississippi, USA
ken.pathak@usace.army.mil

Steven Sloan
US Army Corps of Engineers
Vicksburg District,
Mississippi, USA
steven.sloan@usace.army.mil

Kendall Niles
US Army Corps of Engineers
Vicksburg District,
Mississippi, USA
kendall.niles@usace.army.mil

Jay Ratcliff
GS Center for Env. Informatics
University of New Orleans,
Louisiana, USA
jratclif@uno.edu

Maik Flanagin
US Army Corps of Engineers
New Orleans District,
Louisiana, USA
maik.c.flanagin@usace.army.mil

Christian Guetl
CoDiS-Lab ISDS
Graz University of Technology,
Graz, Austria
c.guetl@tugraz.at

Julian Simeonov
Ocean Sciences Division
Naval Research Laboratory,
Mississippi, USA
julian.a.simeonov.civ@us.navy.mil

Mahdi Abdelguerfi
GS Center for Env. Informatics
University of New Orleans,
Louisiana, USA
gulfsceidirector@uno.edu

***Abstract*—Physics-based solvers such as the Hydrologic Engineering Center's River Analysis System (HEC-RAS) provide high-fidelity river forecasts but are too slow for on-the-fly decision-making during floods. We present a machine learning (ML) surrogate that treats HEC-RAS as a data generator and couples a Gated Recurrent Unit (GRU) for short-term memory with a geometry-aware Fourier Neural Operator (Geo-FNO) for long-range spatial coupling. Trained on 71 reaches of the Mississippi River Basin and evaluated on a year-long hold-out, the surrogate achieves a median absolute stage error of 0.28 ft. For a full 71-reach ensemble forecast, it reduces wall-clock time from 139 to 40 minutes (3.5×). By reading native HEC-RAS files and operating on a compact eight-channel feature interface, the model delivers operational speed while preserving fidelity, enabling rapid "what-if" ensemble guidance.**

***Index Terms*—Fourier Neural Operator; Surrogate Modeling; HEC-RAS; Gated Recurrent Units.**

I. Introduction

During a flood, the U.S. Army Corps of Engineers (USACE) must make critical, time-sensitive decisions—from issuing evacuation orders to scheduling gate operations—within minutes. This operational tempo is fundamentally at odds with the hours-long wall-clock times required by physics-based solvers such as the Hydrologic Engineering Center's River Analysis System (HEC-RAS) to simulate unsteady flow [7], [8]. While pre-computed scenario libraries or reduced-order models offer one workaround, they are often too coarse to capture the specific hydrograph that unfolds in real time [9], [11]. The central challenge, therefore, is to deliver the fidelity of an HEC-RAS simulation at a speed that enables rapid, on-the-fly ensemble forecasting.

We address this challenge by reframing the HEC-RAS workflow itself. Instead of relying on its iterative solver, we treat its native project files as a direct source of training data for a deep-learning surrogate. We propose an autoregressive model that couples a Gated Recurrent Unit (GRU) for short-term temporal memory with a geometry-aware Fourier Neural Operator (Geo-FNO) for long-range spatial dependencies. This hybrid architecture learns the coupled spatio-temporal dynamics of river flow by ingesting a minimal eight-channel vector representing dynamic state, static geometry, and boundary forcings, and then advances the system state hour by hour.

Our primary contribution is a true plug-in surrogate that requires no re-meshing or data conversion, reading native HEC-RAS files directly. This is enabled by a minimalist, reusable interface—a compact feature set sufficient for stable, multi-day forecasts. Evaluated across 71 reaches of the Mississippi River Basin, the model achieves a **3.45**× end-to-end speedup while maintaining a median absolute stage error of **0.28 ft** on a year-long, unseen hold-out, elevating autoregressive neural operators from academic prototypes to operationally promising engines for rapid ensemble flood guidance.

The rest of the paper is structured as follows. Section II details the HEC-RAS data pipeline. Section III reviews related work. Section IV presents the model architecture. Section V outlines the experimental setup, followed by results in Section VI. Section VII discusses findings and limitations, and Section VIII concludes.

II. Background: HEC-RAS as a Data Preprocessor

A. HEC-RAS: The Industry-Standard Solver

HEC-RAS, the U.S. Army Corps of Engineers' River Analysis System, is widely regarded as the industry-standard platform for river hydraulics [8]. Under the hood, it solves the one-dimensional Saint-Venant equations[7] using an implicit Newton–Raphson finite-difference scheme, with several inner iterations per global time step to balance continuity and momentum [8]. This strategy delivers high numerical accuracy but at a steep computational cost: full-reach unsteady-flow simulations typically require hours to days of wall-clock time [8], [11].

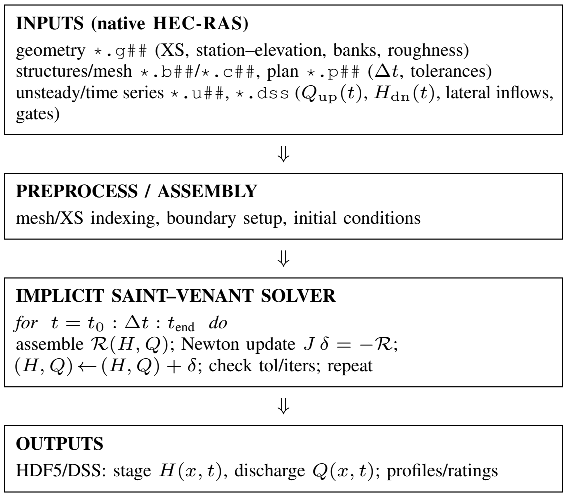

INPUTS (native HEC-RAS)
geometry $\star$.g## (XS, station–elevation, banks, roughness)
structures/mesh $\star$.b##/$\star$.c##, plan $\star$.p## (Δt, tolerances)
unsteady/time series $\star$.u##, $\star$.dss ($Q_{\mathrm{up}}(t)$, $H_{\mathrm{dn}}(t)$, lateral inflows, gates)
$\Downarrow$
PREPROCESS / ASSEMBLY
mesh/XS indexing, boundary setup, initial conditions
$\Downarrow$
IMPLICIT SAINT–VENANT SOLVER
for $t = t_0 : \Delta t : t_{\text{end}}$ *do*
assemble $\mathcal{R}(H, Q)$; Newton update $J\,\delta = -\mathcal{R}$;
$(H, Q) \leftarrow (H, Q) + \delta$; check tol/iters; repeat
$\Downarrow$
OUTPUTS
HDF5/DSS: stage $H(x, t)$, discharge $Q(x, t)$; profiles/ratings

Figure 1. HEC–RAS numerical pipeline (robust non-TikZ rendering).

B. Novel Use Case: From Solver to Pre-Processor

This work treats HEC-RAS not as an end-to-end simulation tool, but as a powerful data-generation engine. By leveraging its mature GIS and project-management capabilities [8], we can assemble consistent geometries, meshes, and boundary hydrographs directly from the native project bundle [8], [12], [25], [26]. We then export this curated file set into a machine-learning pipeline. The surrogate ingests these inputs, learns the hydraulic relationships and returns reach-scale forecasts in seconds rather than hours [11]. In this workflow HEC-RAS becomes a build tool for high-quality training data, while the surrogate supplies the speed needed for rapid what-if analyses.

C. The HEC-RAS File Ecosystem

The key to this approach is the structured, information-rich file bundle that constitutes a standard HEC-RAS project. These files contain all the static, quasi-static, and dynamic information required to train a robust surrogate model, as summarized in Table I.

TABLE I. HEC-RAS FILE BUNDLE ORGANISED BY INFORMATION TYPE. '##' DENOTES VERSION INDICES.

Files	Class	Key contents
Static geometry		
$\star$.g##	XS/1-D	Station–elevation pairs, banks, centre-line
$\star$.c##	2-D mesh	Cell polygons, bed elevation, roughness zones
$\star$.b##	1-D structs	Bridge and culvert shapes, pier spacing
Quasi-static metadata		
$\star$.p##	Plan	Geometry/flow linkage, solver tolerances
Dynamic time-series		
$\star$.u##	Unsteady flow	Hydrograph pointers, gate schedules, run window
$\star$.dss	DSS	Upstream $Q(t)$, downstream $H(t)$, lateral inflows

D. Key Hydraulic Terminology

To interpret the model inputs and outputs, we define the following core terms:

Reach
: A contiguous channel segment between two network break-points (e.g. a confluence or control structure)[8]. Our model operates on a single reach at a time, advancing from an *upstream node* (boundary inflow Q_{up}) to a *downstream node* (boundary stage H_{dn}).

Stage (H)
: The water-surface elevation at a cross-section, referenced to a project datum such as NAVD 88[7]. Units: metres.

Discharge (Q)
: The volumetric flow rate through a cross-section, defined as positive in the downstream direction[7]. Units: $\mathrm{m}^3\,\mathrm{s}^{-1}$.

III. Related Work

Our work builds on advances in three key areas: data-driven hydraulic modeling, autoregressive sequence prediction, and neural operators for scientific computing.

A. Data-Driven Surrogates for River Hydraulics

Early data-driven surrogates for river hydraulics often relied on feed-forward neural networks or polynomial meta-models to emulate one or two cross-sections at a time [11]. More recent studies have scaled to full reaches by coupling convolutional encoders with graph neural networks [10], and physics-informed neural networks have now been demonstrated for single-reach stage prediction [24]; yet many approaches remain restricted to steady-flow conditions or simplified rectangular channels [9], [11]. In contrast, our study targets the entire unsteady-flow

Copyright (c) IARIA, 2025. ISBN: 978-1-68558-329-3

regime of the Mississippi River model, encompassing 71 distinct reaches and thousands of irregularly spaced, natural-geometry cross-sections.

B. Autoregressive Models for Temporal Dynamics

Autoregressive (AR) models, which forecast the next state by feeding back their own previous outputs, form the backbone of classical time-series analysis [20]. The closed-loop structure is computationally efficient for long-horizon roll-outs, but a known weakness is *error accumulation*: small mistakes are recycled and amplified, ultimately drifting the forecast away from reality [21].

To mitigate this, modern hydrology has shifted from classical ARMA models to Recurrent Neural Networks (RNNs) such as Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRUs) [17], [22]. GRUs use update and reset gates to regulate information flow, capturing temporal dependencies while remaining parameter-efficient. When applied to river networks these RNNs typically predict each gage independently, failing to capture the spatial physics that connect them [22]. Our work addresses this by embedding a GRU within a spatial operator, allowing the recurrence to span both time and space. Furthermore, we anchor the AR loop at every step with the true boundary hydrographs (Q_{up}, H_{dn}), providing a strong physical constraint that drastically reduces long-term drift.

C. Neural Operators for Spatial Dependencies

To model the spatial physics, we turn to the Fourier Neural Operator (FNO), which learns mappings between function spaces via global convolutions in the spectral domain [2]. By modulating Fourier modes directly, FNOs capture long-range spatial dependencies with high efficiency and are essentially discretisation invariant [2], [4]. The Geo-FNO variant extends this concept to irregular meshes by injecting coordinate information into the spectral block, making it well suited to the non-uniform cross-section spacing found in river models [3]. Previous studies have already employed two-dimensional FNOs for rapid flood-inundation mapping [23]; here we adopt a one-dimensional Geo-FNO specifically tailored to the chain-like topology of a river reach.

D. Positioning This Work

Combining recurrent networks with neural operators is an emerging and powerful tool for modeling complex spatio-temporal systems [18].

A key aspect of our work is its training methodology. We show that the network learns the underlying hydraulic behavior implicitly from the data itself. This is achieved through a carefully engineered eight-channel feature vector that encodes the system's essential physical drivers: the channel geometry ($z_{\text{bed}}, z_{\text{bank}}$), frictional properties (n_{man}), and the mass and energy constraints imposed by boundary hydrographs ($Q_{\text{up}}, H_{\text{dn}}$).

The success of this approach, using a standard mean-squared error objective with a smoothness regularizer, demonstrates that meticulous feature engineering is a powerful and efficient tool for instilling physical consistency in a data-driven surrogate.

IV. Methodology

We build a one–hour-ahead, autoregressive surrogate that advances using the last $L{=}12$ hours of state. At each step, the network consumes this history and outputs the next-hour stage and discharge $(\hat{H}_{t+1}, \hat{Q}_{t+1})$; the prediction is appended to the history and the loop repeats over the forecast horizon (Figure 2).

A. Input Feature Vector

At hour t and cross-section i, we form a per-section feature vector $x_t(i)$ composed of three groups:

$$\begin{aligned} x_t(i) = [\,&\underbrace{H_t(i),\, Q_t(i)}_{\text{dynamic (2)}} \\ &\mid\ \underbrace{z_{\text{bed}}(i),\, z_{\text{bank}}(i),\, n_{\text{man}}(i),\, x_{\text{coord}}(i)}_{\text{static (4)}} \\ &\mid\ \underbrace{Q_{\text{up}}(t),\, H_{\text{dn}}(t)}_{\text{boundary (2, broadcast over } N)}\ ]. \end{aligned} \tag{1}$$

This *base* interface has $C_{\text{in}}{=}8$ channels. In our implementation, we also include two lightweight auxiliaries: depth $D_t(i) = \max\{H_t(i) - z_{\text{bed}}(i), 0\}$ and a seasonal phase, yielding $C_{\text{in}}{=}10$ total channels. We assemble training tensors of shape $[B, L, N, C_{\text{in}}]$ with $L{=}12$ hours. The 1-D coordinate is also passed as a positional input to the encoder, so its first layer receives $(C_{\text{in}}{+}1)$ inputs.

B. Network Architecture: A Recurrent Neural Operator

The surrogate employs a hybrid architecture that couples a GRU for temporal feature extraction with a Geometry-Aware 1-D Fourier Neural Operator (FNO) for spatial dependencies.

1) **Encoder:** A linear layer lifts the 10-channel input vector and its spatial coordinate x_{coord} (11 total inputs) to a 96-dimensional latent space.
2) **Temporal Block:** A single-layer GRU (hidden size 96) processes the 12-hour encoded sequence at each cross-section, capturing temporal dynamics and outputting its final hidden state.
3) **Spatial Block:** The resulting tensor of final hidden states (shape $[B, N, 96]$) is processed by a 1-D FNO. The FNO applies a global convolution in the frequency domain across the spatial dimension (N), efficiently modeling long-range dependencies. We use up to 48 Fourier modes, proportional to N.
4) **Decoder:** A final linear layer maps the 96-dimensional FNO output to the two target variables: the predicted stage ($\hat{H}_{t+1}$) and discharge ($\hat{Q}_{t+1}$) for the next hour.

C. Rationale for the GRU–Geo-FNO Architecture

Figure 3 highlights that river hydraulics can be viewed as two coupled 1-D signals: (i) a *spatial* profile along chainage at a fixed time (bed and stage across all cross-sections), and (ii) a *temporal* trace at a fixed cross-section (stage/flow through time). We therefore split modeling duties accordingly.

Spatial coupling (Geo-FNO). At each step, we form an ordered vector over the N cross-sections and apply a 1-D

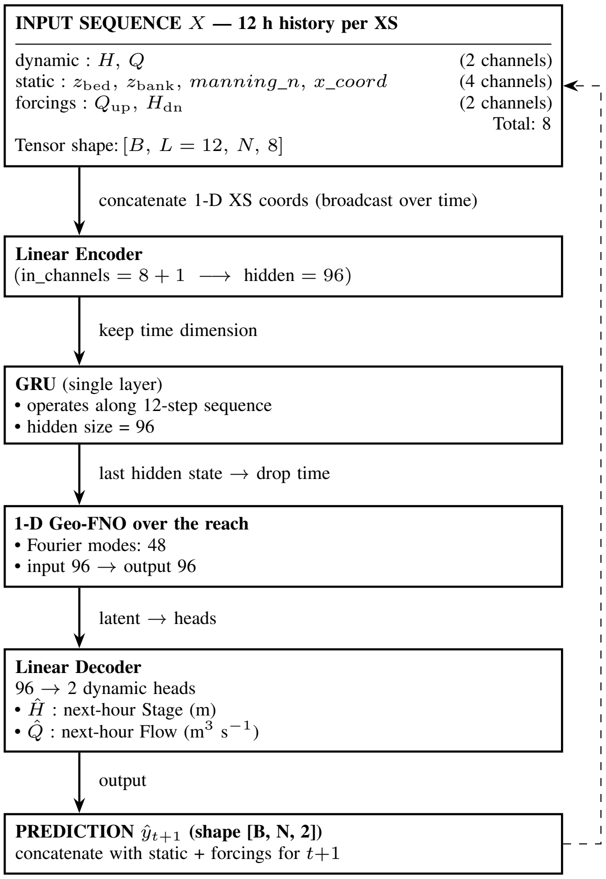

INPUT SEQUENCE X — 12 h history per XS
dynamic : H, Q (2 channels)
static : z_{bed}, z_{bank}, $manning\_n$, x_coord (4 channels)
forcings : Q_{up}, H_{dn} (2 channels)
Total: 8
Tensor shape: $[B,\ L=12,\ N,\ 8]$
concatenate 1-D XS coords (broadcast over time)
Linear Encoder
(in_channels $= 8+1 \longrightarrow$ hidden $= 96$)
keep time dimension
GRU (single layer)
• operates along 12-step sequence
• hidden size = 96
last hidden state $\rightarrow$ drop time
1-D Geo-FNO over the reach
• Fourier modes: 48
• input 96 $\rightarrow$ output 96
latent $\rightarrow$ heads
Linear Decoder
96 $\rightarrow$ 2 dynamic heads
• $\hat{H}$: next-hour Stage (m)
• $\hat{Q}$: next-hour Flow (m^3 s^{-1})
output
PREDICTION $\hat{y}_{t+1}$ (shape [B, N, 2])
concatenate with static + forcings for $t+1$

Figure 2. Autoregressive GRU-GeoFNO surrogate architecture. Vertical arrows share a common inset; the dashed loop feeds predictions back as inputs for the next step.

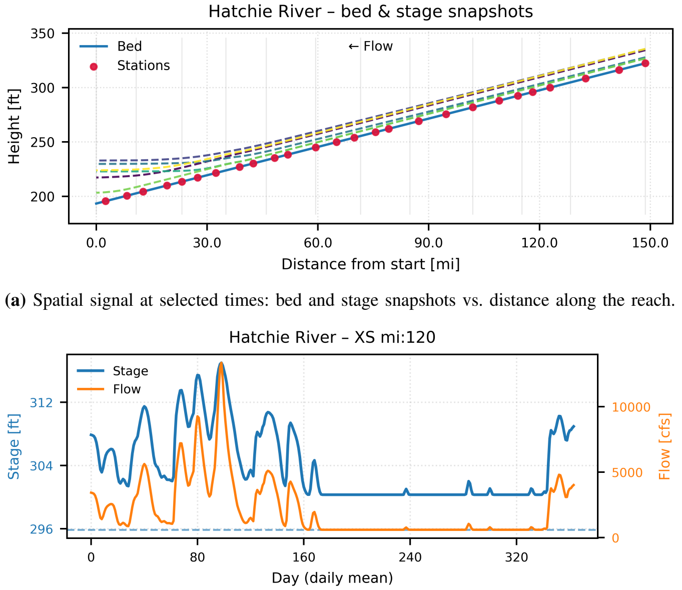
Fourier neural operator across this spatial axis. Because the FNO performs a global spectral convolution, every output depends on all cross-sections simultaneously, enabling upstream–downstream interactions (e.g., backwater/attenuation) to be learned in a single pass. Injecting the 1-D coordinate (x_{coord}) makes the operator geometry-aware, accommodating irregular cross-section spacing without re-meshing.

Temporal memory (GRU). For each cross-section, short-term dynamics are encoded by a single-layer GRU that processes a 12-hour window and returns a compact latent state. This summarizes the recent hydrograph (rising/falling limbs, lags) and supplies the per-section features that the FNO then exchanges across space.

Result. The GRU provides local temporal context; the Geo-FNO propagates that context globally along the reach. This division cleanly matches the physics of 1-D hydraulics while remaining efficient and discretization-robust.

D. Training Objective and Inference

Training Objective. The network minimizes a composite loss that combines a data-fidelity term with physics-informed regularizers to promote stable, realistic predictions. The total



(a) Spatial signal at selected times: bed and stage snapshots vs. distance along the reach.

(b) Temporal signal at one cross-section: daily stage (fill shows depth) and discharge.

Figure 3. River hydraulics as 1-D signals in space and time: space is modeled by a Geo-FNO, time by a GRU.

loss $\mathcal{L}$ is

$$\begin{aligned}\mathcal{L} = {} & W_H\, \mathcal{L}_{\text{Huber}}(\hat{H}_n, H_n) + W_Q\, \text{MSE}(\hat{Q}_n, Q_n) \\ & + \lambda_n\, \|\Delta^2 \hat{H}_n\|_2^2 + \lambda_t\, \|\Delta_t^2 \hat{H}\|_2^2 \\ & + \lambda_\Delta\, \|\Delta_t \hat{H} - \Delta_t H\|_2^2 + \lambda_{\text{bias}}\, \|\overline{\hat{H}} - \overline{H}\|_2^2. \qquad (2)\end{aligned}$$

Here, $(\cdot)_n$ denotes normalization by training-set statistics. The stage (H) term uses a Huber loss (with β=0.02); discharge (Q) uses the *mean squared error (MSE)*. The regularizers act on stage in physical units: spatial smoothness (Δ^2), temporal smoothness (Δ_t^2), change-in-stage matching (Δ_t), and mean-bias control ($\bar{\cdot}$).

Here, $(\cdot)_n$ denotes variables normalized by training-set statistics. The primary data-fidelity loss is computed in this normalized space, using a Huber loss (β=0.02) for stage (H) and MSE for discharge (Q). The four regularizers, which act only on stage, are computed in real physical units to enforce consistent behavior:

- **Spatial Smoothness:** Penalizes the second spatial difference (Δ^2) of the normalized stage prediction to discourage noisy outputs.
- **Temporal Smoothness:** Penalizes the second temporal difference ($\Delta_t^2 \hat{H} = \hat{H}_t - 2H_{t-1} + H_{t-2}$) to reduce high-frequency oscillations over time.
- **Change in Stage:** Encourages the predicted one-hour change in stage ($\Delta_t \hat{H} = \hat{H}_t - H_{t-1}$) to match the true change.
- **Mean Bias:** Penalizes deviations in the spatial mean of the predicted stage ($\overline{\hat{H}}$) from the true mean ($\overline{H}$).

We use weights W_H=1, W_Q=0.3, and regularization coefficients λ_n=$3e-5$, λ_t=$3e-5$, λ_Δ=0.35, and λ_{bias}=$8e-4$. An Exponential Moving Average (EMA) of the model weights (decay 0.995) is maintained and used for all evaluations.

Inference. For multi-step forecasting, we use a 12-hour-seeded autoregressive loop. The model's prediction $(\hat{H}_{t+1}, \hat{Q}_{t+1})$ is used to construct the input for the next step by combining it with the derived depth $\hat{D}_{t+1} = \max(\hat{H}_{t+1} - z_{\text{bed}}, 0)$, the static geometric features, the seasonal phase, and the *true* boundary forcings for hour $t+1$. This complete feature vector is then re-normalized before being passed to the model.

V. Experimental Setup

A. Study Area and Data Sources

We use the U.S. Army Corps of Engineers (USACE) HEC-RAS model of the Mississippi River Basin, which contains 71 distinct one-dimensional river reaches. Data is extracted from the project's HDF5 files, which provide static geometry (cross-section shape, roughness) and hourly simulation results (Stage H, Flow Q) for three major flood years. All data were converted from imperial to SI units (metres, m³/s).

Year	Primary Flood Event	# Hourly Snapshots
2002	June–Sept. Moderate Flood	8,737
2008	May 50-Year Flood	8,783
2011	April Historic Flood	8,737

B. Training and Evaluation Protocol

A separate surrogate model is trained for each of the 71 river reaches using a strict temporal data split.

a) Data Splits: The 2002 and 2008 simulations (17.5k hours) form the **training set**. The first quarter of 2011 (2.2k hours) serves as the **validation set** for early stopping. The entire **2011 year** is the final **test set**.

b) Implementation Details: Each model is trained for up to 120 epochs using the AdamW optimizer ($lr = 2 \times 10^{-4}$) and a batch size of 64.

c) Evaluation via Autoregressive Rollout.: Final performance is measured via an autoregressive rollout on the year-long 2011 test set, mimicking a real-world forecast. After seeding the model with an initial 12-hour history of true data, it iteratively predicts the next 8,725 hourly steps. At each step, the model's prediction is combined with the true boundary forcings $(Q_{\text{up}}, H_{\text{dn}})$ to form the input for the subsequent step.

d) Evaluation Protocol.: The primary evaluation metric is the **Mean Absolute Error (MAE)** in predicted stage, as it provides a direct, interpretable measure of the average error in feet, which is most relevant for operational flood guidance.

$$\text{MAE} = \frac{1}{T}\sum_{t=1}^{T}|y_t^{\text{pred}} - y_t^{\text{true}}| \qquad (3)$$

Reporting convention. Unless noted otherwise, we report *per-reach* MAE in feet (Table III).

VI. Results

We evaluate each of the 71 per-reach surrogates via a full-length autoregressive rollout on the unseen 2011 hold-out year. Results are reported in wall-clock time for the full ensemble and in Mean Absolute Error (MAE, feet) for stage.

A. Computational Speedup

A primary goal of this work is to accelerate forecasting. Table II summarizes the end-to-end wall-clock time required for a 1-year (8,737-hour) ensemble forecast across all reaches. The surrogate completes this task in **40 minutes** compared to **139 minutes** for the HEC-RAS solver, a **3.45**× speedup. This acceleration is operationally significant, enabling minute-scale “what-if" analysis.

It is important to note that this benchmark represents a conservative estimate of potential gains. While the neural network itself is highly parallelizable, our current rollout implementation is a single-threaded Python loop running on a CPU. **Porting this autoregressive data-handling pipeline to a GPU would unlock substantial further acceleration**, representing a key avenue for future optimization.

TABLE II. INFERENCE TIME FOR 1-YEAR, 71-REACH FORECAST.

Model	Wall-Clock Time
HEC-RAS 5.0.1	139 minutes
Recurrent FNO Surrogate	**40 minutes**
Speedup Factor	**3.45**×

B. Predictive Accuracy

We report accuracy using Mean Absolute Error (MAE) in feet, as this metric directly answers the operational question: “Is the predicted stage within a tolerable deviation of the HEC-RAS result?”

a) Per-Reach Performance: Table III details the MAE for each of the 71 reaches. The performance is strong across the majority of the basin, with a median MAE of **0.28 ft** and an InterQuartile Range (IQR) of **0.06–1.04 ft**. Overall, **91.5%** of reaches (65 of 71) achieve a MAE of 2.0 ft or less, meeting a key fidelity target for stage guidance.

b) Error Analysis: While most reaches perform well, a smaller subset of primarily smaller tributaries drives a long tail in the error distribution. Only 6 of 71 reaches (8.4%) exceed a MAE of 2.0 ft. These challenging cases, such as the ‘Hatchie River‘ and ‘St. Francis‘, are known to exhibit more complex hydraulics or have sparser data representation, and they are the focus of ongoing improvements.

TABLE III. PER-REACH MEAN ABSOLUTE STAGE ERROR (MAE) FOR THE 2011 HOLD-OUT YEAR, WITH THE NUMBER OF CROSS-SECTIONS (N) AND REACH LENGTH (MILES). SORTED ASCENDING BY MAE

Reach ID (River—Reach)	XS	Len [mi]	MAE [ft]
CouleeDesGrues—1	56	5.47	0.0066
Black River—R1	3	0.24	0.0131
OldRiver6—1	2	0.12	0.0131
BayouBourdeaux—1	71	6.23	0.0164
OldRiver2—1	14	6.95	0.0164
Ouachita River—R2	79	24.16	0.0164
OldRiver4—1	78	6.06	0.0230
Ouachita River—R1	237	50.89	0.0295
BayouJeansonne—1	42	3.27	0.0328

Continues on next column/page

Table III continued from previous column/page

Reach ID (River—Reach)	XS	Len [mi]	MAE [ft]
BigCanal—1	40	3.07	0.0361
OldRiverOutflow—SidneyMurray	8	19.39	0.0427
Mississippi—Below Loosahatch	2	0.60	0.0459
OldRiverOutflow—OverBank-Aux	6	1.51	0.0459
OldRiverOutflow—RedRiv-SidMur	10	3.91	0.0492
YazooRiver—Reach1	202	59.60	0.0525
OldRiver3—1	43	3.28	0.0558
DUMMY—1	3	0.04	0.0558
outlet—1	2	100.00	0.0591
OldRiver1—1	24	3.22	0.0623
Mississippi—Below Wolf	13	18.90	0.0722
Mississippi—Below Hatchie	14	7.67	0.0755
RedRiver—BelowBlack	9	8.24	0.0755
Mississippi—Below Nonconnah	11	6.88	0.0820
OldRiverOutflow—LowSill	4	0.45	0.0984
Boeuf River—R1	78	24.17	0.0984
Mississippi—Upper Miss	63	29.63	0.0984
OldRiverOutflow—Auxiliary	5	10.10	0.1017
Ohio River—Lower SOHS	2	0.20	0.1083
Ohio River—LD 52-53	10	10.23	0.1115
BayouJoson—1	70	25.79	0.1411
Mississippi—Below Obion	30	24.48	0.1509
Mississippi—Below St. Fran	31	39.52	0.1509
Tensas River—R1	16	7.89	0.2395
Cumberland River—Cumberland River	56	10.17	0.2461
Forked Deer—Forked Deer	42	4.62	0.2493
Tennessee River—Tennessee River	138	28.06	0.3117
Forked Deer—North Fork	58	6.10	0.3150
RedRiver—AboveBlackRiver	7	7.19	0.3314
Mississippi—Below Big Muddy	51	76.37	0.4197
RedRiver—BelowNatch	9	24.66	0.4364
Straight Slough—Straight Slough	5	1.22	0.4495
Mississippi—Below Cairo	72	112.65	0.4495
Obion River—Below Forked	83	12.34	0.4724
White River—White River	90	24.36	0.5229
Mississippi—Below Arkansas	81	100.58	0.5528
Atchafalaya—BelowOldRiver	172	30.99	0.6201
St. Francis—Above SS	16	3.54	0.6412
OldRiver5—1	7	0.20	0.6537
Little River—R1	11	9.86	0.7874
Big Muddy—Reach-1	29	25.65	0.8333
Black River—R3	35	23.70	0.8825
Nonconnah Cr—Nonconnah Cr	57	14.62	1.0335
Cat Oua R—R1	41	8.21	1.0542
YazooRiver—Reach2	53	12.65	1.2172
Mississippi—Below Vicksburg	42	137.20	1.2412
Arkansas River—Arkansas River	31	28.06	1.2967
YazooRiver—Reach1.5	43	10.63	1.3419
Wolf River—Wolf River	140	15.40	1.3484
BayNatch—1	160	12.32	1.4633
White River—Below Cache	77	15.34	1.4699
W-Wit—1	7	0.07	1.8209
Morganza Outlet—To Atchafalaya	26	25.08	1.9127
Forked Deer—South Fork	56	5.18	2.0046
Black River—R2	70	26.39	2.0965
Cache River—Cache River	72	49.58	2.1428
Loosahatchie—Loosahatchie	146	35.33	2.1555
Ohio River—OHS	3	0.13	2.2605
Obion River—Reach_1	198	33.77	3.6059
St. Francis—Below SS	21	5.61	3.9633
Hatchie River—Hatchie River	161	41.16	5.7316

C. Ablation Studies: Validating Architectural Choices

To validate our design, we conducted ablation studies on the ‘Cache River—Cache River‘ reach, a case spanning **49.6 miles** and discretized into **71 cross-sections**. We systematically removed key architectural blocks and feature groups from our full proposed model to quantify their contribution. The results, summarized in Table IV, demonstrate that each component is critical for achieving high fidelity.

TABLE IV. ABLATION RESULTS ON THE CACHE RIVER.

Model Configuration	Component Removed	Stage MAE (ft)
Full Proposed Model	**(All components included)**	**2.14**
Architecture Ablations		
No FNO Block	(Replaced with pointwise MLP)	9.12
No GRU Block	(Replaced with last-frame encoding)	11.49
Feature Ablations		
No Boundary Forcings	Q_{up}, H_{dn} channels	10.45
No Static Geometry	$z_{bed}, z_{bank}, n_{man}, x_{coord}$	12.62
No Derived Depth	$D = \max(H - z_{bed}, 0)$	**17.82**
No Seasonal Phase	s_t feature	16.04

a) Impact of Architecture: Both the temporal and spatial blocks of the network are essential. Removing the FNO and using a simple multilayer perceptron (MLP) head (‘noFNO‘) degrades performance by over 4x (MAE 9.12 ft vs. 2.14 ft), confirming that a global spatial operator is necessary to capture long-range hydraulic dependencies. Similarly, removing the GRU's temporal memory (‘noGRU‘) and encoding only the last known time step increases error by over 5x (MAE 11.49 ft), validating the need to process a sequence history.

b) Impact of Features: The ablation results confirm that physically-informed feature engineering is vital. Removing the derived auxiliary channels had the most severe impact: omitting the derived depth channel increased MAE by over 8x to 17.82 ft, while removing the seasonal phase increased it by over 7x to 16.04 ft. This highlights that providing the model with features that encapsulate non-obvious physical context is critical. Removing the core boundary conditions or static geometry also caused a catastrophic drop in performance, confirming that every channel in our proposed feature vector contributes meaningfully to the final accuracy.

D. Qualitative Case Study: Error Propagation

To understand the nature of the errors in challenging cases, we examine the full-year autoregressive rollout for the ‘Hatchie River—Hatchie River‘ reach, which had the highest MAE. Figure 4 shows the forecast hydrographs at twelve evenly spaced cross-sections (XS) along the reach.

A clear spatial pattern emerges. At the upstream end (e.g., XS 0, XS 7), the surrogate tracks the HEC-RAS ground truth with high fidelity, capturing the primary flood waves accurately. However, performance degrades progressively downstream. At the midpoint (e.g., XS 42), minor deviations appear. By the downstream end (e.g., XS 63, XS 70, XS 77), the model becomes unstable, introducing large, high-frequency oscillations and diverging significantly from the ground truth. This suggests a pattern of spatial error propagation, where small inaccuracies from upstream are amplified as they are passed downstream by the model.

VII. Discussion

Our results show that a recurrent neural operator can emulate year-long HEC-RAS runs with operational fidelity while accelerating an ensemble forecast by $3.45\times$. The remaining failure modes concentrate in a small set of hydraulically

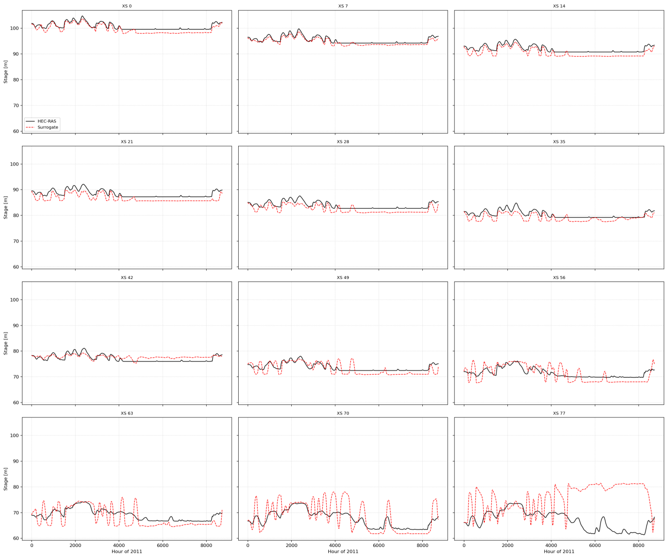


Figure 4. Year-long autoregressive rollout on the ‘Hatchie River‘ reach at 12 evenly spaced cross-sections. The surrogate (red, dashed) tracks the HEC-RAS ground truth (black, solid) well at the upstream end (top-left panels) but accumulates error and develops instabilities downstream (bottom-right panels).

complex reaches, yielding a heavy-tailed error distribution (Table III). We summarize the key lessons and scope.

a) Exposure bias despite boundary anchoring.: The Hatchie River case (Fig. 4) illustrates classic autoregressive drift: small upstream errors accumulate and are propagated downstream by the FNO’s global spatial coupling. Clamping the true boundary hydrographs $(Q_{\text{up}}, H_{\text{dn}})$ at every step acts as a strong physical prior mirroring HEC-RAS inputs and prevents catastrophic divergence in most reaches. The residual instabilities on Hatchie indicate that internal dynamics can overwhelm this anchoring. This suggests two complementary remedies: (i) training-time strategies that reduce exposure bias (e.g., scheduled sampling, noise injection on inputs) and (ii) richer boundary/forcing information or data assimilation for reaches with complex internal hydraulics.

b) Hydraulic regime, not size, drives difficulty.: Geometric scale (reach length, cross-section count) shows no reliable relationship with MAE in Table III. Long, well-gauged main-stem reaches (e.g., *Mississippi—Below Vicksburg*) are modeled accurately, whereas shorter tributaries with backwater effects or prolonged low-flow spells (e.g., *Hatchie River*, *St. Francis—Below SS*) dominate the tail. In practice, the governing factor is the frequency and persistence of regimes under-represented in training, not the number of cross-sections.

c) Operational scope and limits.: Today, the surrogate is best used as a *scenario-analysis accelerator* for known hydrologic regimes: (i) models are trained and evaluated per reach, so network-scale feedbacks across confluences are not yet represented; (ii) performance depends on the hydrologic diversity seen in training generalization to far out-of-distribution events is fragile; and (iii) evaluation assumes true boundary forcings; operational deployment will inherit uncertainty from boundary forecasts. Addressing these gaps will likely require graph neural operators for topology-aware coupling, targeted data augmentation to balance low-flow/backwater regimes, and experiments with perturbed or forecast boundary conditions to quantify skill degradation.

d) Takeaway.: Accuracy at minute-scale cost came from three ingredients: (1) an architecture that separates temporal memory (GRU) from global spatial coupling (Geo-FNO), (2) physics-aware features including derived depth and seasonal phase and (3) a stabilizing loss. The ablations substantiate each ingredient’s contribution and explain where the current model fails, thus charting a concrete path to basin-scale, autonomous forecasting.

VIII. Conclusion

We presented a recurrent neural–operator surrogate for 1-D HEC-RAS that delivers year-long, reach-wide forecasts at operational fidelity. On the unseen 2011 hold-out across 71 reaches, the model achieves a **3.45**× end-to-end speedup (40 vs. 139 minutes) while maintaining a **median stage MAE of 0.28 ft**, with **91.5% of reaches** $\leq$ **2 ft**.

Accuracy at minute-scale cost follows from three ingredients validated by ablations: (i) a GRU for short-term memory coupled to a Geo-FNO for global spatial coupling; (ii) physics-aware features, especially derived depth and seasonal phase; and (iii) a stabilizing loss together reducing Cache River MAE from 12.9 ft (plain MSE) to 2.14 ft and preventing drift.

Current scope is intentionally conservative: models are trained per reach (no network-scale feedbacks yet), skill degrades for out-of-distribution regimes, and rollout stability assumes true boundary forcings. These constraints point to clear upgrades: topology-aware operators for basin coupling, data/augmentation to balance challenging hydraulic regimes, assimilation or forecasted boundaries to quantify resilience, and a GPU rollout pipeline to unlock further wall-clock gains.

The surrogate is a practical *scenario-analysis accelerator* today and a potentially viable path toward basin-scale, near-real-time flood guidance with some further enhancements.

References

[1] M. Raissi, P. Perdikaris, and G. E. Karniadakis, “Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations,” *Journal of Computational Physics*, vol. 378, pp. 686–707, 2019.

[2] Z. Li, *et al.*, “Fourier neural operator for parametric partial differential equations,” in *Proc. ICLR*, 2021.

[3] Z. Li, D. Z. Huang, B. Liu, and A. Anandkumar, “Fourier neural operator with learned deformations for PDEs on general geometries,” *arXiv preprint arXiv:2207.05209*, 2022.

[4] N. Rahaman, *et al.*, “On the spectral bias of neural networks,” in *Proc. 36th Int. Conf. Machine Learning (ICML)*, vol. 97, pp. 5301–5310, 2019.

[5] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 770–778.

[6] M. Zoch, *et al.*, “Physics-informed neural network surrogate models for river stage prediction,” *arXiv:2503.16850*, 2025.

[7] H. Chanson, *Hydraulics of Open Channel Flow*. Butterworth-Heinemann, 2004.

[8] G. W. Brunner, “HEC-RAS river analysis system, hydraulic reference manual,” U.S. Army Corps of Engineers, Hydrologic Engineering Center, Davis, CA, Tech. Rep. CPD-69, 2021.

[9] P. Benner, S. Gugercin, and K. Willcox, “A survey of projection-based model reduction methods for parametric dynamical systems,” *SIAM Review*, vol. 57, no. 4, pp. 483–531, 2015.

[10] H. Bi, J. Lin, and X. Zhao, “Data-driven surrogate modeling in computational fluid dynamics: A review,” *Computers and Fluids*, vol. 258, p. 104764, 2023.

[11] Z. Takbiri-Borujeni, J. T. Smith, and M. D. White, “Challenges of using surrogate models for environmental simulations: A case study in hydrology,” *Environmental Modelling and Software*, vol. 134, p. 104882, 2020.

[12] M. Flanagin, *et al.*, “Hydraulic splines: A hybrid approach to modeling river channel geometries,” *Computing in Science and Engineering*, vol. 9, no. 5, pp. 4–15, 2007.

[13] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, “PointNet: Deep learning on point sets for 3D classification and segmentation,” in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 652–660.

[14] M. Abdelguerfi, *3D Synthetic Environment Reconstruction*. Springer, 2012.

[15] M. Chung, *et al.*, “The geospatial information distribution system (GIDS),” in *Succeeding with Object Databases*. John Wiley & Sons, 2001, pp. 357–378.

[16] Z. Li, *et al.*, “Graph neural operator for PDEs,” *arXiv:2010.08895*, 2020.

[17] K. Cho, *et al.*, “Learning phrase representations using RNN Encoder–Decoder for statistical machine translation,” in *Proc. EMNLP*, Doha, Qatar, 2014, pp. 1724–1734.

[18] J. Brandstetter, D. E. Worrall, and M. Welling, “Message passing neural PDE solvers,” in *Proc. Int. Conf. on Learning Representations (ICLR)*, 2022. [Online]. Available: https://arxiv.org/abs/2202.03376

[19] J. Pathak, *et al.*, “FourCastNet: A global data-driven high-resolution weather model using adaptive Fourier neural operators,” in *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. [Online]. Available: https://arxiv.org/abs/2202.11214

[20] G. E. P. Box and G. M. Jenkins, *Time Series Analysis: Forecasting and Control*. San Francisco, CA: Holden–Day, 1970.

[21] S. Bengio, O. Vinyals, N. Jaitly, and N. Shazeer, “Scheduled sampling for sequence prediction with recurrent neural networks,” in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 28, Montréal, Canada, 2015, pp. 1171–1179.

[22] F. Kratzert, D. Klotz, J. Herrnegger, G. Hochreiter, and S. Nearing, “Toward improved predictions in ungauged basins: Exploiting the power of machine learning,” *Water Resources Research*, vol. 55, no. 12, pp. 11 344–11 364, 2019.

[23] A. Stenta, J. Smith, and L. Hennig, “FloodFNO: Fourier neural operators for real-time inundation mapping,” in *Proc. IAHR World Congress*, 2023.

[24] M. Zoch, *et al.*, “Physics-informed neural network surrogate models for river stage prediction,” in *Proc. ALLDATA 2025 – 11th Int. Conf. on Big Data, Small Data, Linked Data and Open Data*, 2025.

[25] R. Wilson, *et al.*, “Geographical data interchange using XML-enabled technology within the GIDB system,” in *XML Data Management: Native XML and XML-Enabled Database Systems*. Morgan Kaufmann, 2003, pp. 235–261.

[26] M. Abdelguerfi (ed.), *3D Synthetic Environment Reconstruction*. Springer, 2001, vol. 611 of the NATO Science Series.

Integrated Design and Field Validation of MoBI: An Intelligent Buoy for Environmental Data Acquisition

Alessio Chirigu[1], Alberto Mancosu[1], Sara Pinna[2], Mariella Sole[2], Matteo Anedda[2], and Daniele D Giusto[2]

[1]Hedya srl - Cagliari, Italy
[2]University of Cagliari, Department of Electrical and Electronic Engineering - UdR CNIT Cagliari, Cagliari, Italy
[1]Emails: alessio.chirigu@hedya.it, alberto.mancosu@hedya.it
[2]Emails: s.pinna62@studenti.unica.it, mariella.sole@unica.it, matteo.anedda@unica.it, ddgiusto@gmail.com

***Abstract*—This study details an advanced marine monitoring system based on smart buoys, designed to detect environmental conditions in coastal and open-sea areas. To meet the critical need for secure and accurate on-site data collection, the solution utilizes embedded hardware, multi-parameter sensors, and redundant data transmission technologies. This work is perfectly aligned with the conference topics focusing on intelligent sensing and autonomous systems for environmental protection. The main contribution lies in introducing a novel system architecture and presenting real-world results obtained during field tests. The conclusions demonstrate that the platform is robust, easily expandable, and represents an effective approach for optimizing the quality of information and the reliability of its communicationin maritime surveillance scenarios.**

Keywords-smart buoy; environmental monitoring; LoRaWAN; LTE; marine data.

I. Introduction

Marine environmental monitoring represents one of the major challenges for the sustainable management of natural resources and for mitigating the risks associated with climate change and human activities. A significant contribution in this field is represented by the coastal monitoring framework proposed in [1], which introduced an IoT-based architecture integrating smart buoys and onshore stations for real-time environmental and crowd data collection along the Sardinian coast. Building upon these concepts, the EcoMonitoring project extends this approach by combining smart buoys and surface drones, enabling efficient, scalable, and low-impact data acquisition in coastal and open-sea areas [2].

A key component of the system is the Monitoring Buoy (MoBI). Built from High-Density Polyethylene (HDPE) using 3D printing technology, MoBI is designed to operate in a semi-autonomous mode: it can be towed by an Unmanned Surface Vehicle (USV) to predefined locations, where it performs targeted environmental measurements. The buoy is equipped with sensors for both wave dynamics and water quality parameters, and it can acquire, process, and transmit data even under limited connectivity conditions. This approach is consistent with other recently developed intelligent systems for marine monitoring [3].

This paper provides a detailed description of the MoBI system, including its architecture, operational logic, communication strategies, and the results of field testing. The remainder of the document is structured as follows: Section II, hardware and software architecture, communication strategies, and control logic; Section III, acquisition of environmental data through the multiparametric probe; Section IV, wave monitoring subsystem and signal processing approach; Section V, results of field campaigns and performance analysis; Section VI, conclusions and future development directions.

II. System Concept and Architecture

The MoBI system has been designed to provide a robust, adaptable, and cost-effective platform for maritime and coastal environmental monitoring. Its architecture integrates low-power processing (on internal components), continuous interfacing with multiple sensors, and a dual transmission mode to ensure data consistency even in cases of weak or absent connectivity. At the technological core of the system are two processing units: a Raspberry Pi 3B+ micro-computer and an Arduino Mega 2560 R3 microcontroller. These units are housed inside the buoy's sealed hull and powered by a stabilized 5V supply. A voltage booster converts the supply to 12V, required for the multiparametric probe. The Raspberry Pi handles higher-level tasks, including probe management, data standardization, and log creation. The Arduino, on the other hand, manages the acquisition of fundamental data from orientation and positioning sensors: GPS, accelerometer, gyroscope, and magnetometer. It also controls the LoRa module for emergency communications. Similar modular IoT-based buoy architectures combining embedded controllers and multiple communication interfaces have been described in [4]. Communication and synchronization between the Raspberry Pi and the Arduino Mega are achieved through a USB serial connection, enabling bidirectional data and command exchange. This setup also allows the Raspberry Pi to directly power the Arduino Mega through the same USB cable used for data transmission, thereby reducing wiring complexity. The protocol used for communication is Universal Asynchronous Receiver-Transmitter (UART), a full-duplex asynchronous system that enables simultaneous transmission and reception of data on separate channels. In a classic UART configuration, the transmit (TX) pin is connected to the receive (RX) pin of the counterpart device. With USB, this bridging is indirectly managed by a USB-to-Serial converter chip, which translates UART signals

ISBN: 978-1-68558-329-3

for USB compatibility. Communication is handled by dedicated software libraries. Since UART is asynchronous, it does not require a shared clock. To ensure reliable data exchange, both devices must be configured with the same baud rate (bits per second), set during software initialization. On the Arduino Mega, the Serial library manages communication, leveraging its four available hardware serial ports. The USB port is mapped to Serial0 (pins 0 and 1), leaving the remaining ports available for external modules (e.g., one is reserved for GPS). On the Raspberry Pi, the Python pySerial library handles access to the USB serial port and manages asynchronous command exchange in ASCII format. This lightweight method ensures effective coordination between the two devices, avoiding overload and ensuring smooth acquisition, storage, and transmission of data. To guarantee uninterrupted data flow, MoBI employs a dual transmission strategy. When 4G/LTE connectivity is available, the Raspberry Pi transmits environmental data to a remote server. In cases of weak or absent coverage, data is sent via LoRa [5] [6]. This strategy leverages the complementary features of 4G/LTE and LoRa in terms of speed, coverage, and energy efficiency. 4G/LTE serves as the primary channel, offering high data transfer rates, low latency, and wide coverage across coastal and inland areas, thus ensuring reliable and immediate communication. LoRa acts as a fallback option, enabling data transmission in areas with limited or no cellular coverage. Its low power consumption and long range make it ideal for remote or hard-to-access environments. This dual system enhances the buoy's resilience and versatility, ensuring continuous operation and efficient data delivery under diverse conditions. In all cases, data are also logged locally on an SD card, allowing deferred transmission in case both primary and backup channels fail. At the end of each mission, the system checks whether the remote platform has received the data and, if necessary, initiates retransmission via 4G. The electrical system includes a charge controller and voltage converters (12V, 5V, 3.3V) that supply all electronic modules. Comparable surface gateway platforms designed for shallow-water underwater networks, integrating solar power management and embedded processing, have been presented in [7]. From a structural perspective, the MoBI buoy (shown in Figure 1) is built from HDPE using 3D printing. Its modular design facilitates maintenance and component replacement/upgrades. The internal arrangement protects the electronics from water while ensuring optimal exposure of antennas (LTE, GPS, and LoRa) and correct placement of the probe in the water. Similar smart buoy solutions have been reported in [8].

The control software (firmware) is written in Python (for Raspberry Pi) and C++ (for Arduino), following a finite state machine (FSM) approach to manage the operational protocol. Each mission is structured into the following phases: sensor and communication preparation, navigation to target points, data collection, and transmission. Figure 2 illustrates the complete operational software flow of MoBI. Upon startup, serial communication between the Arduino Mega and Raspberry Pi is established, enabling constant exchange of information and commands.

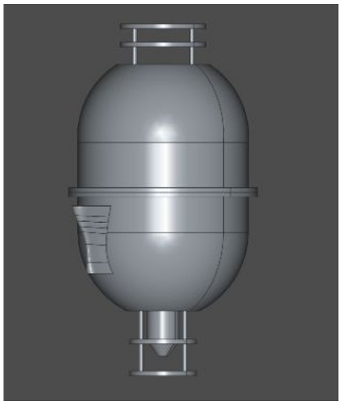


Figure 1. MoBI buoy.

The mission begins with a request to the remote platform to obtain the GPS coordinates of the measurement sites. Upon receiving the start command (START), the Raspberry Pi forwards the coordinates to the Arduino. After acknowledgment, the Arduino initializes the inertial sensors, GPS module, and LoRa communication. Once a stable GPS signal (fix) is acquired, the Arduino sends a confirmation (GPSFIX), indicating operational readiness.

The displacement phase then begins: the buoy, either manually or USV-towed, moves toward the first measurement point. During transit, the Arduino Mega continuously updates the GPS position and computes the distance to the destination. When this distance falls below the predefined threshold, a STOP command is issued to signal target arrival and transition to the next phase. In the measurement phase, the Raspberry Pi sends a start acquisition command (STARTDATA) to the Arduino, triggering the wave motion analysis algorithm. During this interval, inertial parameters such as wave period, direction, and height are recorded.

Simultaneously, the Raspberry Pi collects data from the environmental probe, which measures parameters such as temperature, pH, conductivity, dissolved oxygen, and turbidity. At the end of acquisition, the Arduino sends the wave data to the Raspberry Pi, completing the collection process.

The data gathered by both units are then managed in the transmission phase, which depends on connectivity: if 4G is available, the data are sent immediately (SEND4G); otherwise, they are transmitted via LoRa to the receiving node (SENDLORA).

After transmission, the system checks whether additional points need to be visited. If so, the next destination is set, and the cycle repeats. If all sites have been covered, the mission is concluded. The transmission frequency is variable and directly related to the number of configured measurement points, since data are sent after each acquisition cycle.

III. Integrated Environmental Data Acquisition

The MoBI system is engineered to collect both physico-chemical properties of water and wave dynamics data. Its acquisition workflow is structured into automated cycles,

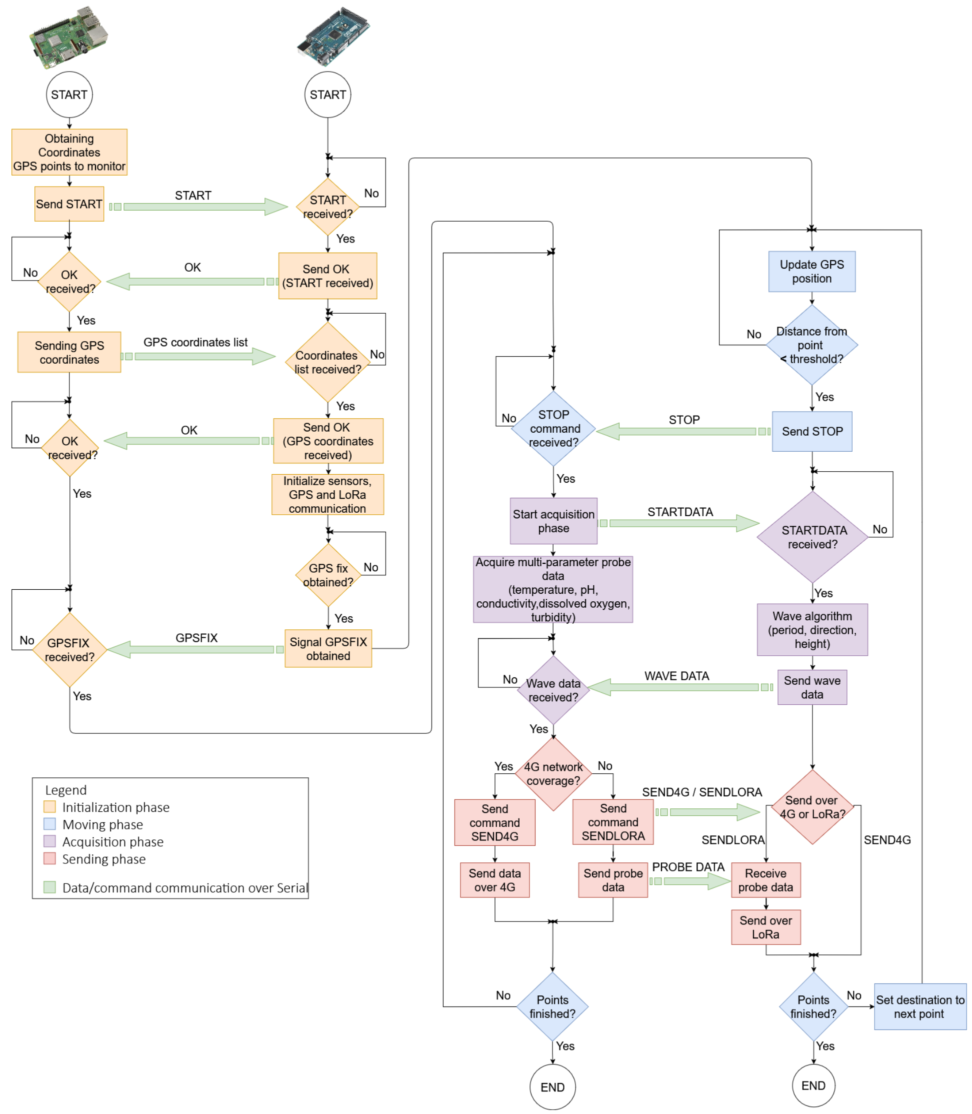


Figure 2. Software design.

guaranteeing measurement accuracy and seamless integration with data transmission and storage systems.

For monitoring environmental conditions, the buoy is equipped with a multiparametric probe WMP6, connected to a Raspberry Pi via a USB serial interface RS485. This sensor can record in real time several key indicators of water quality, including temperature, pH, electrical conductivity, dissolved oxygen, and turbidity. The probe remains submerged throughout the mission, ensuring continuous exposure to the measured environmental conditions.

During each cycle, the Raspberry Pi sends interrogation commands to the probe. The acquired values are processed by a software module that uses regular expressions to extract and validate each measurement. For every parameter, the system performs three consecutive readings a few seconds apart; the results are then averaged and stored in a structured JSON file, ready for transmission or local storage. Similar solutions, based on modular and open-source architectures, have already been implemented in long-term water monitoring projects.

Special attention has been given to turbidity, a parameter that often proves critical in saltwater testing. In several cases, the probe produced irregular or negative values during the initial immersion phase. To prevent distortions, a preliminary stabilization period was introduced before the actual sampling phase, in order to ensure representative measurements.

The collected data are saved on an SD card and simultaneously queued for transmission to the remote server via 4G/LTE network. If the connection is unavailable, the system switches to the LoRa channel as an alternative. In all cases, local SD storage guarantees data availability for retransmission in case of transfer failure. At the end of each measurement cycle, the system checks that all data packets have been successfully received by the platform and, if necessary, selectively resends the missing ones. Each measurement is timestamped and associated with

GPS coordinates and a unique mission identifier.

The acquisition cycle can be executed for preconfigured measurement points defined on the web platform or manually activated in the field. This flexibility allows the system to adapt to a variety of operational scenarios, ranging from short missions with a few points to extended campaigns involving dozens of successive readings.

IV. Monitoring of Wave Dynamics

Beyond monitoring the physico-chemical properties of water, the MoBI system also records wave motion dynamics, a crucial dataset for assessing coastal stability and ensuring maritime safety. This functionality relies on a suite of inertial sensors (accelerometer and gyroscope) mounted on an X-NUCLEO-IKS01A3 board, complemented by a magnetometer that provides directional orientation.

The wave analysis employs a four-phase algorithm designed to identify complete wave cycles through the examination of vertical acceleration. Specifically, fluctuations along the Z-axis are tracked to discern the acceleration and deceleration phases during both the cresting and trough movements. Similar low-cost smart buoy systems for real-time wave height measurement have been proposed in [9].

When a full cycle is detected, the system determines the wave period by measuring the interval between two successive peaks, while the wave height is estimated using double numerical integration of the accelerometer data. Although this technique is sensitive to sensor drift and noise, the level of precision achieved is sufficient for practical field applications. Comparable strategies have been implemented in professional-grade coastal wave monitoring systems [2].

Wave measurements are collected over a configurable time window, defined at the start of each deployment. The acquired data are then averaged to produce a representative statistical profile while minimizing transmission overhead. As with other environmental parameters, wave-related data are georeferenced and stored in JSON format, enabling subsequent processing.

Field evaluations confirmed the system's capability to reliably detect waves taller than 5–10 cm, even under minor buoy oscillations or environmental noise. These findings are consistent with results obtained from other low-cost prototypes tested in lagoon conditions [3].

Nevertheless, during periods of very calm water, small inaccuracies in wave period estimation were observed, primarily due to difficulties in distinguishing buoy micro-movements from actual wave patterns. Future improvements to the algorithm are planned to address this limitation.

V. Test Campaign and Experimental Results

To assess the performance of the MoBI system in realistic operational contexts, a series of field trials was conducted between January and March 2025 in the coastal area of Cagliari, with particular reference to the Sant'Elmo pier and the Palma channel. The campaign was aimed at verifying system stability, evaluating the accuracy and consistency of the acquired data, and testing the robustness of the communication subsystems under variable environmental conditions. The following section outlines the testing objectives and adopted methodology, and subsequently provides a synthesis of the key experimental outcomes.

A. Objectives and Methodology

The testing campaign aimed to evaluate several critical aspects of the system:

- Accuracy of GPS positioning throughout missions;
- Proper acquisition, validation, and structuring of environmental measurements;
- Reliability of 4G/LTE communication and the effectiveness of LoRa as a backup channel;
- Performance of the multiparameter probe in saline conditions;
- Seamless integration of the platform with backend systems, including the dashboard and data storage.

During the field operations, the buoy was transported to designated locations and maintained in a stationary position to facilitate data collection. Each measurement cycle involved logging coordinates, stabilizing the probe, and performing repeated sampling. Data were transmitted in real time whenever a 4G connection was available, with LoRa serving as an alternative communication channel when the primary network was inaccessible.

B. Main Results

Positioning. The GPS module exhibited an average positional deviation of around 10 meters with respect to the predefined target points. While this discrepancy falls within the operational tolerance of the platform, subsequent tests using a Global Navigation Satellite System (GNSS) module demonstrated improvements, reducing both the time required to obtain a satellite fix and the mean positioning error.

Environmental Data Quality. The multiparametric probe provided stable and reliable measurements for all parameters, with the exception of turbidity, which intermittently yielded zero or negative readings. To ensure the integrity of the dataset, such invalid measurements were systematically excluded from the analysis.

Communication Reliability. The system effectively transmitted data over LTE in regions with sufficient network coverage. In areas lacking connectivity, the fallback and retransmission mechanisms functioned as designed: data were temporarily stored locally and subsequently transmitted via LoRa. Comparable LoRa-based communication systems for marine buoy monitoring achieved stable GPS data transmission with low packet loss, confirming the suitability of this protocol for long-range maritime links [10]. Upon mission completion, the system verified the presence of any missing records by querying the platform. Any absent data were retrieved from the SD card and retransmitted over 4G until successful reception was confirmed.

Integration and Visualization. The acquired data were subsequently uploaded to a cloud-based platform for visualization and aggregation, where they were transformed into interactive charts and geospatial maps. The dashboards enabled real-time monitoring of parameters such as temperature, pH, and dissolved oxygen, providing operators with immediate insight into water quality. The integration of LoRa communication with cloud-based dashboards has also been successfully implemented in professional systems, such as the CB-150 buoy deployed in Green Bay [5].

Operational Observations. From an operational perspective, the field deployment of the buoy demonstrated the effectiveness of its modular design, which simplified post-mission inspections and battery maintenance. No instances of hardware malfunction or water penetration were detected, confirming the reliability and watertight performance of the 3D-printed HDPE hull.

VI. Conclusion and Next Step

The MoBI system has demonstrated itself as a dependable, versatile, and effective platform for environmental monitoring in marine and coastal environments. Its dual-processor configuration, comprehensive sensor suite, and redundant communication architecture allowed the system to successfully address the operational challenges encountered during field deployments.

Field testing validated the platform's robustness in terms of data acquisition, transmission, and integration with backend systems. Environmental measurements were generally consistent and reliable, while the wave monitoring module performed adequately for waves exceeding 5 cm in height. The software infrastructure supported continuous data logging and ensured recovery even during temporary network outages.

Insights from the field campaigns also identified several areas for enhancement, which will inform future developments:

- optimization of power management to reduce energy consumption [6];
- additional testing under more demanding wave conditions;
- expanded compatibility with diverse sensor protocols and automated long-term mission management;
- native integration with cloud platforms to enable historical data analysis and real-time alerts.

In summary, MoBI constitutes a significant advancement toward the automation of environmental monitoring, merging technological reliability with user-friendly operation in a compact system ready for deployment in real-world coastal scenarios.

Acknowledgment

This research activity was partially funded by the European Union - NextGenerationEU and by the Ministry of University and Research (MUR), National Recovery and Resilience Plan (NRRP), Mission 4, Component 2, Investment 1.5, project "RAISE - Robotics and AI for Socio-economic Empowerment" (ECS00000035).

References

[1] R. Girau et al., "Coastal monitoring system based on social internet of things platform", *IEEE Internet of Things Journal*, vol. 7, no. 2, pp. 1260–1272, 2020. DOI: 10.1109/JIOT.2019.2954202.

[2] J. F. Martínez-Osuna et al., "Coastal buoy data acquisition and telemetry system for monitoring oceanographic and meteorological variables in the gulf of mexico", *Measurement*, vol. 183, p. 109 841, 2021. DOI: 10.1016/j.measurement.2021.109841.

[3] F. Campagnaro et al., "Monitoring the venice lagoon: An iot cloud-based sensor network approach", *IEEE Journal of Oceanic Engineering*, vol. 50, pp. 570–582, 2024. DOI: 10.1109/JOE.2024.3459483.

[4] A. Babic, M. Orec, and N. Miskovic, "Developing the concept of multifunctional smart buoys", in *Proceedings of the MTS/IEEE OCEANS Conference*, IEEE, 2021, pp. 1–8. DOI: 10.1109/OCEANS44145.2021.9705916.

[5] NexSens Technology, *Using data buoys to monitor southern green bay*, [retrieved: June, 2025], 2025.

[6] C. Delgado, J. M. Sanz, C. Blondia, and J. Famaey, "Batteryless LoRaWAN communications using energy harvesting: Modeling and characterization", *IEEE Internet of Things Journal*, vol. 8, no. 4, pp. 2694–2711, 2021. DOI: 10.1109/JIOT.2020.3019140.

[7] S. Falleni et al., "Design, development, and testing of a smart buoy for underwater testbeds in shallow waters", in *Proceedings of the IEEE Global Oceans 2020: Singapore–U.S. Gulf Coast*, IEEE, 2020, pp. 1–9. DOI: 10.1109/IEEECONF38699.2020.9389065.

[8] J. Likozar and A. Jaklic, "Smart anchor buoy: Design and implementation", in *Proceedings of the 2023 IEEE Third International Conference on Signal, Control and Communication (SCC)*, IEEE, 2023, pp. 1–5. DOI: 10.1109/SCC59637.2023.10527710.

[9] M. Wyawahare, A. Patil, M. S. Balan, U. Anuse, and A. Aadarsh, "Jal-tarang: Smart buoy for wave height measurement", in *Proceedings of the 2023 IEEE International Carnahan Conference on Security Technology (ICCST)*, IEEE, 2023, pp. 1–6. DOI: 10.1109/ICCST59048.2023.10474257.

[10] N. Sa'adah, A. D. Ramadhani, and D. P. Riananda, "Tracking position on gps smart buoy system using lora communication", in *Proceedings of the 6th International Conference on Vocational Education and Electrical Engineering (ICVEE)*, IEEE, 2023, pp. 304–309. DOI: 10.1109/ICVEE59738.2023.10348269.