Received XX Month, XXXX; revised XX Month, XXXX; accepted XX Month, XXXX; Date of publication XX Month, XXXX; date of current version XX Month, XXXX.

Digital Object Identifier 10.1109/OJCOMS.2022.1234567

Low-Complexity Multi-Step Signal Prediction for Underwater Acoustic Communications: A Joint Reservoir Computing and Transfer Learning Approach

Senthan Prasanth[1], ***Student Member, IEEE,*** **Adnan A. Cheema**[2], ***Senior Member, IEEE,*** **Gökhan Seçinti**[3], ***Senior Member, IEEE,*** **Dac-Binh Ha**[4], **Berk Canberk**[5], ***Senior Member, IEEE,*** **Octavia A. Dobre**[1], ***Fellow, IEEE*** **and Trung Q. Duong**[1], ***Fellow, IEEE***

[1]Memorial University, St. John's, NL A1B 3X5, Canada, (e-mail: {sprasanth, odobre, tduong}@mun.ca)
[2]Ulster University, BT15 1AP, Belfast, U.K. (e-mail: a.cheema@ulster.ac.uk)
[3]Istanbul Technical University, Turkey (e-mail: secinti@itu.edu.tr)
[4]Duy Tan University, Vietnam (e-mail: hadacbinh@duytan.edu.vn).
[5]Edinburgh Napier University, UK, (e-mail: b.canberk@napier.ac.uk)

CORRESPONDING AUTHOR: TRUNG Q. DUONG (tduong@mun.ca).

This work was supported in part by the Canada Excellence Research Chair (CERC) Program CERC-2022-00109, in part by the Natural Sciences and Engineering Research Council of Canada (NSERC) Discovery Grant Program RGPIN-2025-04941, in part by Canada First Research Excellence Fund Transforming Climate Action, Grant TCA-LRP-20241-MUN-1.3-03, and in part by Canada Research Chairs Program CRC-2022-00187. The work of D.-B. Ha is partially funded by the Vietnam National Foundation for Science and Technology Development (NAFOSTED) under grant number 102.04-2025.72. The work of B. Canberk is partially supported by The Scientific and Technological Research Council of Türkiye (TÜBİTAK) 1515 Frontier R&D Laboratories Support Program for Türk Telekom 6G R&D Lab under project number 5249902.

ABSTRACT Underwater acoustic communication faces significant challenges due to multipath propagation, signal attenuation, and environmental variability. Accurate signal prediction supports adaptive transmission by forecasting upcoming samples and enabling efficient resource allocation. This work presents a complexity-reduced echo state network (ESN) framework for underwater acoustic signal prediction using real-world shallow-water at-sea experimental data. By reducing the reservoir to 200 neurons and optimizing hyperparameters, the model achieves strong performance with low computational complexity. Single-step prediction on the first hydrophone (H1) yields a test R^2 of 0.9978. Furthermore, multi-step forecasting on H1 is performed using a recursive prediction strategy and evaluated up to 6 steps ahead, achieving $R^2 = 0.8232$ at step 6. Using transfer learning, we extend the H1-trained model across 14 additional hydrophones in the vertical array by retraining only the readout layer, which reduces retraining time and computational complexity. Cross-dataset validation between different shallow-water environments further demonstrates robustness to changes in propagation conditions. Across the vertical array, single-step transfer learning achieves $R^2 \geq 0.997$ for most hydrophones, while cross-dataset transfer yields R^2 values typically in the range 0.93-0.97.

INDEX TERMS Underwater acoustic communication, echo state networks, multi-step forecasting, reservoir computing, transfer learning, echo state network, time series.

I. Introduction

Underwater operations are growing fast across scientific, economic, and military sectors worldwide. Therefore, there is an increasing need for innovative methods for underwater communication. The three most common types of underwater communications include optical [1], [2], [3], electromagnetic [4], and acoustic [5], [6], [7] communications. Among them, optical communications, in terms of bandwidth, have the largest potential for data transfer. However, it experiences significant power loss due to light absorption and scattering in water [1]. In practical underwater environments, optical links are typically limited to a few meters to tens of meters due to absorption and scattering, with longer ranges possible only in very clear water [2]. Electromagnetic waves have inherent disadvantages for underwater communication, particularly because they undergo severe attenuation in seawater due to high conductivity, and practical long-range links often require very low frequencies [3]. Although low-frequency radio transmission may be used, the size of the antennas required for such implementations would be impractically large [8]. Moreover, optical methods are sensitive to water turbidity caused by variations in clarity and suspended particles [1], [2]. These effects make optical and electromagnetic underwater links difficult to deploy reliably over practical distances. These constraints make long-range, reliable underwater links difficult, so most systems rely on acoustic signaling despite the severe impairments of the underwater acoustic channel [3]. Consequently, underwater acoustic communication (UWAC) is widely adopted in ocean exploration, military and defense systems, marine research, offshore oil and gas operations, and environmental monitoring [4], [5], [9], [10], [11].

Two primary challenges to communication systems within an underwater acoustic environment are the limited bandwidth and the high signal propagation delay associated with transmission in water [4], [5], [6], [7]. The speed at which sound travels is significantly less than that of electromagnetic waves, resulting in longer propagation times compared with electromagnetic wave propagation through the air [1], [2]. Furthermore, the transmission of sound through aquatic environments is subject to changes in frequency, multipath propagation, high attenuation, and fading phenomena [12], [13]. Moreover, signal degradation could happen from external interference and inter-symbol interference [12]. Underwater signal characteristics are additionally influenced by environmental variables, including salinity and temperature [14], [15], while noise from humans and sounds generated by marine organisms further impair signals [16]. All of these factors combine to create a very challenging and harsh environment for communicating via acoustic signals.

To address these channel characteristics, underwater communication requires specialized signal processing methods such as adaptive equalization, channel coding, and related algorithms [4], [7]. For UWAC systems, accurate channel prediction is very important because sound travels slowly in water and channels change quickly [4], [5]. Prediction facilitates proactive resource allocation and adaptive transmission methods while minimizing errors caused by channel variability and propagation delays [6], [7]. However, creating precise predictive models under these challenging conditions remains a significant research challenge and requires further investigation [4], [5].

Physics-based models of the underwater acoustic channel use complex mathematics and need a lot of computing power to run [17], [18]. Machine learning (ML) and deep learning (DL) based models that are used for predicting the behavior of an underwater acoustic channel can be very accurate [6], [19], [20], [21], [22]. But they typically require a lot of training data and computing power [6], [19]. Reservoir computing (RC) architectures, especially the echo state networks (ESNs), offer computationally efficient output through fixed reservoir architectures and training only the readout layer [5], [7]. However, critical challenges remain for practical UWAC deployment. Currently, ESNs are implemented with large reservoir sizes, which makes them computationally intensive [7]. Additionally, when utilizing multiple hydrophones, separate models must be created for each hydrophone location, and this creates a demand for additional computing power and training data. The use of transfer learning between hydrophones at different locations may reduce this computational complexity, but it has not been widely explored. Moreover, extended multi-step forecasting with long temporal horizons has not drawn much attention in underwater acoustic applications. Further, cross-dataset validation to assess model generalization across different underwater conditions also remains unaddressed.

colour blue Therefore, to address the above issues, we propose a complexity-reduced ESN framework for predicting underwater acoustic received signals using real shallow-water at-sea experimental data. In practical UWAC systems, a prediction model should not only be accurate but also lightweight for real-time use and scalable across multiple hydrophones and different environments. Unlike previous UWAC studies that mainly focused on transmitter-to-receiver mapping or single-hydrophone forecasting, the proposed framework follows a receiver-only approach and supports recursive multi-step prediction, readout-only transfer across a vertical hydrophone array, and cross-dataset validation. Such received-signal prediction can also support the estimation of underwater acoustic channel characteristics. The main contributions of this paper are as follows:

- We propose a low-complexity single-step prediction model based on a single hydrophone using an ESN, achieving a very high R^2 score with a compact reservoir.
- We extend and comprehensively evaluate the proposed model for recursive multi-step forecasting, up to 6 steps ahead.
- We apply spatial transfer learning across the remaining 14 hydrophones by keeping the reservoir fixed and retraining only the linear readout, enabling efficient deployment across depth.
- We validate cross-dataset generalization by training the model on one shallow-water dataset and evaluating it on a second shallow-water dataset collected under different propagation conditions.

The remainder of this paper is organized as follows. Section II reviews related work on underwater acoustic channel modeling and prediction. Section III describes the experimental setup and provides details of the dataset and preprocessing steps. Section IV presents the proposed ESN-based prediction methodology. Section V reports and discusses the experimental results under the considered configurations. Finally, Section VI concludes the paper and outlines directions for future work.

II. Related Works

Various approaches have been developed for underwater acoustic channel prediction, ranging from physics-based models to data-driven ML techniques. This section reviews the existing research in underwater acoustic signal modeling using conventional methods, ML and DL approaches, along with their strengths and limitations. However, practical deployment still raises open questions on model complexity, scalability across hydrophones, and generalization across environments, which motivates us to develop the proposed models.

1) Physics-Based Underwater Acoustic Channel Modeling

Early underwater acoustic channel modeling relied primarily on physics-based approaches [17]. Bellhop and similar ray-tracing models simulate acoustic wave propagation through detailed environmental parameters. These models require extensive information about water properties, bathymetry, and sound speed profiles. The Bellhop ray model provides an intuitive means for modeling sound propagation in the ocean among various existing mathematical approaches based on ray, normal-mode, and parabolic curve methods. Furthermore, physics-based models often demand substantial computational resources to achieve accurate predictions [15], [17], [18]. In practice, the need for detailed environmental inputs and the associated computational cost make real-time adaptation difficult, which has driven interest toward data-driven predictors that learn channel behavior directly from measured signals.

Mathematical approximations are often unable to reflect the complexity of real underwater environments, especially at sea [6], [18], [12]. The majority of the models above are based on mathematical assumptions and approximations rather than being formulated using real underwater communication data. As a result, it may deviate in practice due to parameter mismatch and nonstationarity. Moreover, these models encounter difficulties with dynamic environmental fluctuations and multipath effects [12], [15], [18]. The computational burden and modeling constraints have motivated research scholars to explore data-driven alternatives capable of learning channel behavior directly from measurements [6]. Furthermore, the effective implementation of UWAC requires predictors that maintain efficiency with constrained computational resources and can scale beyond a single hydrophone or a specific depth. These needs motivate lightweight architectures and transfer learning strategies that reduce retraining effort.

2) Machine Learning and Deep Learning for UWAC

Data-driven approaches using ML have gained considerable attention for underwater channel modeling and broadly for key use cases in 6G [4], [6], [23], [24], [25], [26]. These methods learn complex channel patterns directly from measurements without requiring detailed environmental parameters [7], [22]. It is difficult to deploy such ML-based models when there is limited processing capability, limited training data, or limited time for retraining, especially in realistic environments. Deep neural networks have demonstrated strong performance in underwater channel estimation tasks [4], [6], [19]. The work in [19] employed deep neural networks for channel estimation in underwater acoustic orthogonal frequency-division multiplexing (OFDM) systems. Results showed improved R^2 scores over traditional least squares and minimum mean square error methods. The deep neural network outperformed the least squares method and achieved performance comparable to the minimum mean square error method in terms of bit error rate and normalized mean square error. Similarly, DL-based single-carrier communications were investigated for time-varying underwater channels. The DL approach enabled transmitted symbol recovery without requiring explicit channel estimation and equalization [20], [21]. Moreover, signal detection techniques using model-driven DL were proposed for full-duplex cognitive underwater systems [27]. This approach addressed self-interference cancellation challenges in full-duplex underwater acoustic communications.

Recurrent neural networks (RNNs) and their variants provide promising solutions for temporal sequence prediction in underwater acoustic channels [4], [6]. RNNs can capture temporal dependencies in underwater acoustic signals through their recurrent connections. The recurrent architecture maintains memory of previous inputs to inform current predictions. Long short-term memory (LSTM) networks address vanishing gradient problems that occur in standard RNNs during extended sequence processing. LSTM architectures incorporate gating mechanisms to selectively retain or forget information over long sequences. However, these architectures require extensive training data for optimal performance [4]. Training times are considerably longer than conventional methods because RNNs typically train their recurrent weights using backpropagation through time, which unrolls the network across multiple time steps and propagates gradients through the sequence. This computational cost limits real-time deployment on resource-constrained underwater platforms [6], [7], [19]. These limitations motivate the investigation of more efficient architectures that reduce training complexity.

Among data-driven approaches, ESNs are computationally efficient approaches for predicting underwater acoustic signals [7]. ESNs belong to the RC paradigm with fixed reservoir architectures. The reservoir layer processes temporal dynamics through random initialization with fixed weights throughout training and testing, while only the readout layer is trained using simple linear regression or ridge regression [28], [29]. Compared to traditional RNNs that utilize backpropagation through time to strengthen recurrent connections, ESNs offer several advantages, such as improving the ease of learning and reducing the time required for the learning process [6]. Additionally, ESNs achieve competitive R^2 scores compared to traditional methods [5], [7]. These computational advantages make ESNs well-suited to underwater scenarios with limited resources, where training time and processing capability are constrained.

In [7], an experiment with ESN was applied to real experimental data from controlled tank and lake environments for modeling the underwater acoustic channel. From the observation, it was noticed that the ESN architecture is faster in training compared to LSTM network while maintaining competitive prediction performance. This is mainly due to the fixed reservoir, which captures complex temporal patterns through a high-dimensional state space. Furthermore, the model effectively captured the time-varying multipath characteristics inherent in underwater acoustic channels. In addition, the study further demonstrated that RC provides a viable alternative to DL approaches for underwater acoustic applications. However, the research focused primarily on a single-hydrophone sequence-to-sequence prediction scenario. While the study tested different reservoir sizes, the best results came from relatively large state dimensions, which increased computational complexity and highlighted the need for further optimization [7]. Moreover, the researchers did not systematically investigate how to reduce the complexity through hyperparameter tuning. Moreover, accommodating multi-hydrophone configurations, transfer learning across spatial positions, and extended multi-step forecasting remained unexplored in this work. These limitations motivate the design of complexity-reduced ESNs with proper hyperparameter tuning, together with efficient adaptation across hydrophones in a vertical array at different depths. In addition, evaluating recursive multi-step prediction over increasing horizons and testing whether learning models can generalize across different shallow-water conditions are critical.

Research in [5] investigates one of the RC variants, known as liquid state machines (LSMs), which represents an approach for modeling UWAC. In this work, LSMs employ spiking neural networks with fixed recurrent connections to process temporal information. Unlike traditional ESNs that use continuous-valued activations, LSMs utilize spiking neurons that communicate through discrete events. The reservoir transforms input signals into high-dimensional spatiotemporal state patterns that capture complex channel dynamics. The fixed reservoir configuration of an LSM architecture makes it well-suited for handling changes over time in underwater acoustic channels, and the study validated LSM effectiveness for underwater acoustic channel modeling tasks [5]. However, the work does not emphasize reduced-parameter deployment or transfer across multi-hydrophone array, which remains an important challenge for scalable real-world use cases.

Beyond UWAC-specific applications, various advanced ESN architectures have been developed for common time series prediction tasks [28], [30]. An example of this is shown in [28], where long-short-term ESN architectures are used to predict multiple time scales by utilizing multiple reservoirs with different time constants. Also, an architecture that couples a long-term reservoir with a short-term reservoir enhances time series prediction performance. Moreover, this research study addresses the limitations of standard ESNs that use a single time scale for prediction, resulting in low performance. At the same time, multi-reservoir ESN systems combine multiple reservoirs with sequence resampling techniques used to enhance R^2 scores [30]. Multi-reservoir ESN systems partition the input sequence across multiple reservoirs and aggregate their outputs. Such advanced architectures could potentially benefit underwater acoustic prediction tasks by capturing multiple temporal scales present in channel variations [5], [7]. However, their application to underwater acoustic systems remains largely unexplored, particularly under complexity-reduced constraints and in settings that require scalable deployment across multi-hydrophone array. Table 1 gives a compact comparison between different UWAC studies and proposed work using the same set of criteria.

In summary, prior RC-based UWAC studies have shown that reservoir-based models can effectively capture underwater acoustic signal dynamics [5], [7]. However, practical gaps remain in terms of model complexity, scalable adaptation across multi-hydrophone array, and unified evaluation of recursive multi-step forecasting and cross-dataset generalization. Motivated by these gaps, the proposed work adopts a complexity-reduced ESN framework with an Rx-only sequence-to-one formulation, recursive rollout, and readout-only transfer learning across hydrophones and datasets.

III. Experimental Setup, Dataset and Preprocessing

In this section, firstly, we explain the real acoustic data from a UWAC that we employ for our experiments. The dataset description includes the details on deployment configuration, signal characteristics, and hydrophone array geometry. Then, we detail how we have formulated the signals from the watermark benchmark framework. Finally, we outline the preprocessing pipeline that converts the raw hydrophone recordings into normalized training sequences for our proposed model.

TABLE 1: Comparison of representative UWAC studies and this work across practical modeling choices and the main gaps targeted in this paper

Study	Real Data	Reservoir Computing	Prediction Type	Seq-to-Seq Mapping	Multi-Step Recursive	Complexity Reduction	Spatial Transfer	Cross-Dataset Validation
[5]	✓	✓	Rx-only	✓	✗	✗	✗	✗
[6]	✓	✗	Tx→Rx	✓	✗	✗	✗	✗
[7]	✓	✓	Tx→Rx	✓	✗	✗	✗	✗
This work	✓	✓	Rx-only	✗	✓	✓	✓	✓

Real Data: Based on measured underwater recordings rather than purely synthetic data. **Reservoir Computing**: Uses a reservoir-based model such as an ESN or an LSM. **Prediction Type**: Tx→Rx denotes transmitter-to-receiver mapping; Rx-only denotes forecasting using only the received signal. **Seq-to-Seq Mapping**: Fixed-window input sequence mapped to an output sequence in a single forward pass (e.g., encoder–decoder or direct multi-horizon outputs). This work uses a sequence-to-one formulation (window → single next sample), as described in Section III-C and Equation (2). **Multi-Step Recursive**: Multi-horizon forecasting via recursive rollout, where each prediction is fed back as input for the next step (e.g., horizons up to 6 as detailed in Section IV-D Phase 4). **Complexity Reduction**: Explicit reduction of model size via hyperparameter tuning. **Spatial Transfer**: Transfer across hydrophones in a vertical array. **Cross-Dataset Validation**: Validation across different shallow-water environments.

A. KAU2 Dataset Description

The KAU2 dataset comes from the Kauai Acomms MURI 2011 (KAM11) experiment [31], conducted off the west coast of Kauai, Hawaii, in shallow water of about 100 m depth. A towed acoustic source transmitted signals in the 4–8 kHz band and was deployed at roughly 45–50 m below the surface.

The receiving system consisted of a vertically suspended 16-element hydrophone array. The first hydrophone (H1) was positioned 7 meters above the seafloor. The communication range between the source and the hydrophone array was 3160 meters. This was one of the more distant signals in the watermark benchmark. Additionally, adjacent hydrophone elements were spaced 3.75 m apart. The second hydrophone element (H2) from the sea floor was non-functional during the measurement and is not used in this paper [32].

Figure 1 illustrates the deployment setup for the KAU2 experiment. The source platform maintained mobility through surface vessel towing operations. The receiving array remained stationary at a fixed underwater location. Moreover, this configuration enabled realistic testing of underwater acoustic propagation channel effects.

The transmitted signal was generated as a linear frequency modulated (LFM) chirp using MATLAB. The chirp swept from 4 kHz to 8 kHz over 0.5 seconds. The original transmitted signal contained 12000 samples at 24 kHz sampling rate. The probe signal utilized these LFM trains transmitted continuously for 32.9 seconds [32].

The signal processing significantly impacted the signal duration. The input chirp signal lasted 0.5 seconds and comprised 12,000 samples. Following the transmission of the KAU2, the received signal had a duration of 0.633 seconds (15,181 samples). This is a 26.6% increase in time, which was caused by multipath propagation and signal dispersion. The sampling rate is constant at 24 kHz, resulting in 15,181 samples per hydrophone in the processed dataset. For this entire prediction study, only the output from the H1 was used for proposed model development and temporal forecasting.

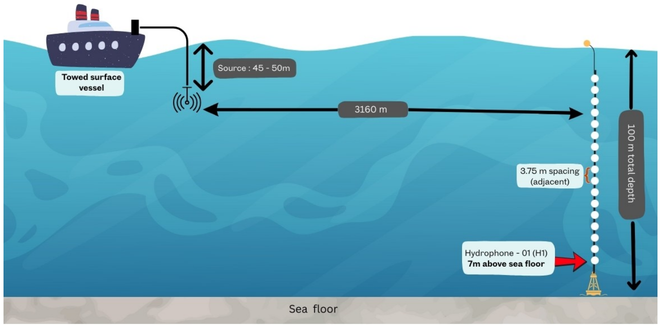


FIGURE 1: Experimental setup for KAU2 dataset showing towed source configuration and vertically suspended 16-element hydrophone array at 100 m water depth with 3160 m communication range. H2 from the seafloor was non-functional.

B. Signal Processing Through Watermark Framework

The watermark benchmark processes signals through a straightforward workflow. The system takes input signals and applies realistic underwater signal effects measured during real experiments. The processing starts with signal preparation, where the input chirp signal is loaded into the watermark framework [32]. The signal fits within the 4-8 kHz frequency band used in KAU2 [31]. The framework then applies the measured underwater signal response to this input signal.

The key operation is a time-varying convolution. In the watermark framework, the transmitted signal is convolved with the channel time-varying impulse response (TVIR) estimated from the KAU2 measurements [32]. This replay can be written as

$$y(t) = \int_{-\infty}^{\infty} \hat{h}(t, \tau) s(t - \tau) d\tau + n(t), \tag{1}$$

where $s(t)$ is the input signal, $\hat{h}(t, \tau)$ denotes the estimated time-varying channel impulse response (TVIR), and $y(t)$ is the output signal. The term $n(t)$ represents additive noise, t denotes the observation time, and τ is the delay variable.

This processing approach reproduces realistic underwater effects, including multipath propagation, Doppler shifts due to source movement, and signal dispersion. The framework preserves the original sampling rate of 24 kHz throughout processing. For KAU2, the system generates outputs for all 16 hydrophones simultaneously. Each hydrophone receives the same input signal but is subject to distinct propagation effects determined by its spatial position.

The processing maintains time synchronization across all hydrophones. The framework saves the convolved signals in standard MATLAB format. The H1 signal depicted in Figure 2 serves as input for the prediction model. This signal retains all relevant characteristics required for evaluating prediction performance.

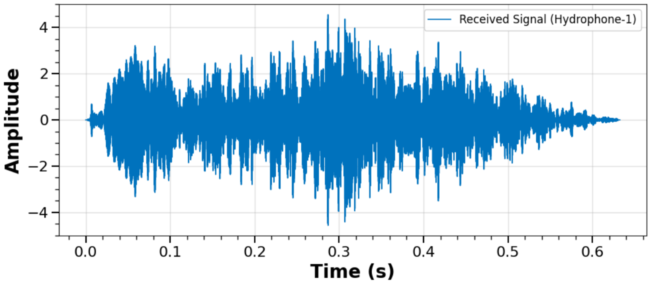


FIGURE 2: Received signal at H1 (KAU2) after underwater signal effects.

C. *Signal Preprocessing and Sequence Preparation*

During this stage, the H1 received signal is prepared for ESN training by converting it into fixed-length, standardized input–output pairs. The procedure preserves temporal structure by forming history windows from past samples and applying the same scaling to the resulting windows. The following three steps describe the preprocessing pipeline. Figure 3 summarizes the workflow used to generate supervised input–output pairs (history windows and targets) from the H1 signal.

1) Signal preparation and splitting: The received signal from the H1 is used as the input time series. The signal is split chronologically into training and testing segments. The training segment contains 12,144 samples (80%) for model training, while the remaining 3,037 samples (20%) are reserved for independent performance evaluation. After sliding-window sequence creation with sequence length (L), these segments yield 12,124 training sequences and 3,017 test sequences, since each length-L input sequence is paired with the next-sample target.

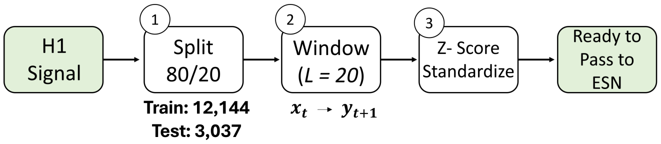


FIGURE 3: Preprocessing pipeline.

2) Sequence generation: Input–output pairs are constructed using a sliding-window approach. Each example consists of L consecutive samples, and the target is the next sample (one step ahead). Multi-step prediction is handled separately via recursive forecasting, as described in Section IV-D (Phase 4). The sequence advances by one sample to form overlapping examples, which improves data utilization while preserving local temporal structure. The parameter L denotes the sequence length of each supervised input sequence as mentioned earlier, and we find $L = 20$ gives good prediction results based on several experiments. Each example is therefore

$$\mathbf{u}_i = [x(i),\, x(i+1),\, \ldots,\, x(i+L-1)]^\top, \tag{2}$$

$$y_i = x(i+L). \tag{3}$$

Here, $x(i)$ denotes the scalar signal sample at discrete-time index i, $\mathbf{u}_i \in \mathbb{R}^L$ is the ith input sequence of length L, and $y_i = x(i+L)$ is the corresponding next-sample (one-step-ahead) target relative to $x(i+L-1)$. This sequence-to-one formulation captures short-term dependencies over the length-L input sequence and defines the supervised input–output pairs used for model training and evaluation.

This Rx-only sequence-to-one formulation differs from the Tx→Rx sequence-to-sequence channel emulation adopted in prior studies [6], [7]. Further, multi-step horizons are obtained via recursive rollout rather than predicting an output vector in a single forward pass.

3) Normalization: Z-score standardization is applied to the input windows and target values. The mean and standard deviation are estimated using the training set only, and the same parameters are then used to standardize both the training and test sets. This ensures that the test data do not influence the normalization parameters. The standardization centers the data to zero mean and unit variance, which keeps the signal scale consistent and facilitates model training. The resulting input–output pairs are then used for ESN training and testing.

IV. Proposed Methodology

In this section, we describe our proposed ESN framework for underwater acoustic signal prediction. Firstly, we review the fundamentals of RC, where the reservoir is randomly initialized and kept fixed, and only the linear readout is trained using ridge regression. Next, we present our ESN architecture and provide details of the training procedure and the key reservoir configuration hyperparameters. Finally, this section outlines the overall experimental workflow and concludes with the performance metrics adopted throughout the study.

A. Reservoir Computing Fundamentals

RC provides an efficient alternative to fully trained RNNs for temporal processing. In conventional RNNs, recurrent weights are optimized through backpropagation through time, which is computationally demanding and may suffer from gradient instability. In RC, the recurrent connections are fixed after random initialization and learning is restricted to the readout layer, which can be estimated in closed form using linear regression [7], [33], [34].

The RC paradigm operates on three foundational principles. First, a randomly connected reservoir projects an input sequence into a high-dimensional non-linear state space. Second, recurrent dynamics provide a fading memory of past inputs, enabling temporal feature extraction. Third, only a linear readout is trained, so model fitting has low computational cost. As such, the overall framework does not require iterative gradient-based optimization but captures temporal relationships between data, which is particularly well-suited to the resource-constrained environment. From a dynamical systems perspective, the reservoir state evolves according to a nonlinear recurrence driven by the input. The reservoir's memory length and stability are primarily controlled by its spectral radius and sparsity; these are selected so as to ensure stable echo-state behavior and effective temporal representation [5], [7].

B. Echo State Network Architecture

An ESN maps the input sequence into a sparsely connected dynamic reservoir, and the resulting reservoir state is used by a linear readout to generate the prediction, as illustrated in Figure 4. This structure enables efficient training while still capturing temporal patterns in underwater acoustic signals.

Notation: In Figure 4, $x(t) \in \mathbb{R}$ denotes the normalized hydrophone sample applied to the reservoir at discrete time t, and $\hat{y}(t)$ is the corresponding ESN prediction. The reservoir state is $\mathbf{r}(t) \in \mathbb{R}^{N_{\text{res}}}$, where N_{res} is the number of reservoir neurons. The fixed input and recurrent weight matrices are $\mathbf{W}_{\text{in}} \in \mathbb{R}^{N_{\text{res}} \times 1}$ and $\mathbf{W}_{\text{res}} \in \mathbb{R}^{N_{\text{res}} \times N_{\text{res}}}$, respectively. The linear readout is parameterized by $\mathbf{w}_{\text{out}} \in \mathbb{R}^{N_{\text{res}}}$ and a bias term $b \in \mathbb{R}$. In the supervised dataset construction, a sequence length L is used to form input sequences $\mathbf{u}_i \in \mathbb{R}^{L}$ for next-sample prediction; the L scalar samples in each $\mathbf{u}_i$ are injected sequentially (one per ESN time step) to obtain the reservoir state used by the readout.

Input projection: The input layer maps the hydrophone sample $x(t)$ to the reservoir through the fixed weight matrix $\mathbf{W}_{\text{in}} \in \mathbb{R}^{N_{\text{res}} \times N_{\text{in}}}$. The entries of $\mathbf{W}_{\text{in}}$ are drawn from a uniform distribution and scaled by an input scale σ_{in} (a tunable hyperparameter), which controls the input magnitude and helps avoid saturation of the reservoir nonlinearity [35]. This random projection produces diverse activations across reservoir neurons without gradient-based optimization, since $\mathbf{W}_{\text{in}}$ is kept fixed after initialization.

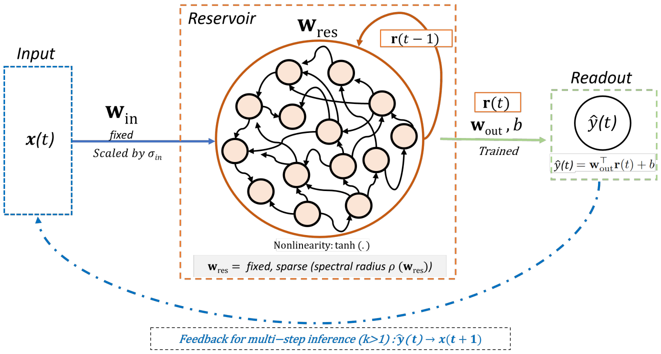


FIGURE 4: Echo state network (ESN) architecture. The input sample $x(t)$ drives the reservoir through the fixed input-weight matrix $\mathbf{W}_{\text{in}}$, while temporal dynamics are generated by the fixed, sparse recurrent matrix $\mathbf{W}_{\text{res}}$ (scaled to a desired spectral radius). The output $\hat{y}(t)$ is produced by a trained linear readout from the reservoir state $\mathbf{r}(t)$.

Reservoir dynamics: The reservoir provides the nonlinear temporal dynamics of the ESN. Its recurrent structure is defined by a fixed weight matrix $\mathbf{W}_{\text{res}} \in \mathbb{R}^{N_{\text{res}} \times N_{\text{res}}}$, initialized from a uniform distribution. To reduce computational cost, $\mathbf{W}_{\text{res}}$ is sparsified by retaining only a fraction of recurrent connections. After initialization, $\mathbf{W}_{\text{res}}$ is rescaled to a desired spectral radius to promote stable echo-state dynamics and to control the memory properties of the network [33]. The $\tanh(\cdot)$ activation introduces nonlinearity and bounds the states to $[-1, 1]$, which keeps the reservoir activations within a limited range.

The reservoir state evolves as

$$\mathbf{r}(t) = \tanh(\mathbf{W}_{\text{in}}\, x(t) + \mathbf{W}_{\text{res}}\, \mathbf{r}(t-1) + \epsilon\, \boldsymbol{\xi}(t)), \tag{4}$$

where $\mathbf{r}(t-1)$ is the previous reservoir state and $\epsilon\,\boldsymbol{\xi}(t)$ is an additive independent and identically distributed Gaussian perturbation with $\boldsymbol{\xi}(t) \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. The perturbation acts as a training-time regularizer. For evaluation and recursive multi-step forecasting, we set $\epsilon = 0$, so all reported test metrics are deterministic. We use no washout (washout length $= 0$), since discarding initial states did not improve validation performance. For multi-step evaluation, the reservoir is initialized by driving it with the last L ground-truth samples before recursive forecasting. We use a non-leaky ESN (leak rate $\lambda = 1$), i.e., no leaky integration is applied in (4). The recurrence provides a fading memory of past inputs, enabling the model to capture temporal dependencies in the hydrophone signal.

Readout layer (linear): The prediction is computed as

$$\hat{y}(t) = \mathbf{w}_{\text{out}}^{\top}\mathbf{r}(t) + b, \tag{5}$$

where $\mathbf{w}_{\text{out}} \in \mathbb{R}^{N_{\text{res}}}$ and $b \in \mathbb{R}$ denote the readout weights and bias. In our implementation, the bias is obtained as the intercept term in ridge regression. Although each training example is formed from an input sequence of length $L = 20$, the ESN input dimension is $N_{\text{in}} = 1$ because the reservoir is driven sequentially by scalar samples. Concretely, for the ith sequence $\mathbf{u}_i = [x(i), \ldots, x(i + L - 1)]^\top$, the samples are injected one-by-one for L consecutive ESN time steps using (4). The resulting feature vector is taken as the final reservoir state after the L updates, denoted by $\mathbf{r}_i \in \mathbb{R}^{N_{\text{res}}}$. Stacking $\mathbf{r}_i$ over all training sequences forms the design matrix $\mathbf{R}$ for ridge regression, with corresponding targets $y_i = x(i + L)$.

The readout is learned using ridge regression:

$$\min_{\mathbf{w}_{\text{out}},\, b} \left\| \mathbf{y} - (\mathbf{R}\mathbf{w}_{\text{out}} + b\mathbf{1}) \right\|_2^2 + \alpha \|\mathbf{w}_{\text{out}}\|_2^2, \tag{6}$$

where $\mathbf{R} \in \mathbb{R}^{N_{\text{train}} \times N_{\text{res}}}$ stacks the collected reservoir states, $\mathbf{y} \in \mathbb{R}^{N_{\text{train}}}$ contains the targets, $\mathbf{1}$ is an all-ones vector, and $\alpha > 0$ is the regularization parameter [35]. Since only the readout is trained, learning is efficient and does not require iterative backpropagation, which also supports the transfer-learning strategy used later.

Echo-state property: The spectral radius of $\mathbf{W}_{\text{res}}$ is

$$\rho(\mathbf{W}_{\text{res}}) = \max_i |\lambda_i|, \tag{7}$$

where $\{\lambda_i\}_{i=1}^{N_{\text{res}}}$ are the eigenvalues of $\mathbf{W}_{\text{res}}$. In practice, selecting $\rho(\mathbf{W}_{\text{res}})$ close to but below unity typically increases the effective memory length, while smaller values yield faster fading dynamics [36].

Reuse across hydrophones and datasets: After training on H1, $\mathbf{W}_{\text{in}}$ and $\mathbf{W}_{\text{res}}$ are kept fixed. For a new hydrophone or dataset, only $(\mathbf{w}_{\text{out}}, b)$ is re-estimated via ridge regression using the corresponding training data, enabling lightweight adaptation.

Training procedure: Training consists of a single state-collection pass through the data. The matrices $\mathbf{W}_{\text{in}}$ and $\mathbf{W}_{\text{res}}$ are initialized, sparsified, and then fixed. The readout parameters $(\mathbf{w}_{\text{out}}, b)$ are obtained by solving (6), which admits a closed-form solution and does not require iterative backpropagation.

C. Hyperparameter Configuration

Table 2 lists the ESN hyperparameters used in this study. These values were selected through experimentation to balance R^2 scores with computational efficiency.

Note that in Table 2, $N_{\text{in}} = 1$ denotes the scalar input applied to the ESN at each internal time step, whereas $L = 20$ is the sequence length used to form each supervised training example. For each example, the L samples are injected sequentially (one sample per ESN time step), and the final reservoir state is used as the feature vector for the readout. The number of sequences is smaller than the number of raw samples due to sliding-window construction. With next-sample targets, N raw samples (where N is the number of samples in the split) yield $N - L$ supervised pairs, resulting in 12124 training sequences and 3017 test sequences. Moreover, the complexity-reduced ESN employs $N_{\text{res}} = 200$ reservoir neurons with a connectivity of 0.27, meaning that approximately 27% of the entries in $\mathbf{W}_{\text{res}}$ are nonzero. Here, sparsity refers to the ESN reservoir architecture itself, and the recurrent matrix $\mathbf{W}_{\text{res}}$ is intentionally sparsified to reduce computation while preserving rich reservoir dynamics. It should not be confused with possible sparsity of the underwater channel impulse response. The reservoir's recurrent weights are scaled to have a spectral radius of $\rho(\mathbf{W}_{\text{res}}) = 0.9$, which helps retain temporal information while keeping the internal dynamics stable. A near-unity spectral radius is particularly appropriate for underwater acoustic signals, which exhibit strong short-term temporal correlation due to multipath propagation and slow channel variation. A larger spectral radius extends the effective memory of the reservoir, allowing it to capture these correlated temporal structures, while remaining below unity ensures the echo-state property is satisfied and the reservoir dynamics remain asymptotically stable. Each supervised input sequence has length $L = 20$ samples (0.833 ms at 24 kHz), and the samples are applied sequentially to the reservoir as a scalar input $x(t)$ at each ESN time step. Regularization is introduced through additive reservoir noise ($\epsilon = 1.19 \times 10^{-4}$) and ridge regression ($\alpha = 1.74 \times 10^{-5}$). During training, the reservoir state is propagated sequentially within each length-L sequence, enabling the readout to exploit temporal dependencies contained in the L-sample history. Further, multi-step forecasts for horizons of 1–6 steps ahead (approximately 0.042–0.25 ms at 24 kHz) are produced recursively by feeding previous predictions back as subsequent inputs.

D. Overall Architecture Workflow

Figure 5 summarizes the end-to-end workflow used in this study. The pipeline is organized into sequential phases, where the output of each phase serves as the input to the next.

Phase 1: Data preprocessing and sequence creation – The received H1 signal is converted into normalized supervised input–target pairs using the preprocessing procedure described in Section III-C.

Phase 2: ESN architecture – The preprocessed sequences are applied to the proposed ESN architecture described in Section IV-B and illustrated in Figure 4.

Phase 3: Training on H1 – The ESN is configured using the hyperparameters in Table 2 and trained on H1, as detailed in Section IV-C.

Phase 4: Multi-step prediction on H1 – Multi-step forecasts are generated for horizons $k = 1, \ldots, 6$ using recursive feedback forecasting. The special case $k = 1$ corresponds to standard one-step-ahead prediction, while for $k > 1$ predicted samples are fed back as subsequent inputs, which can lead to error accumulation as the horizon increases. Before starting the recursive rollout, the reservoir state is

TABLE 2: ESN Hyperparameter Configuration

Category	Parameter	Value
Reservoir	Number of neurons (N_{res})	200
	Spectral radius, $\rho(\mathbf{W}_{\text{res}})$	0.9
	Connectivity level	0.27 (27% nonzero)
	Leak rate (λ)	1.0 (non-leaky ESN)
	Activation function	$\tanh(\cdot)$
	Washout period	0 timesteps
	Initial state	$\mathbf{r}(0) = \mathbf{0}$
Input	Input dimension (N_{in})	1 (scalar)
	Scaling factor (σ_{in})	0.4
	Sequence length (L)	20 samples (0.833 ms at 24 kHz)
Regularization	Noise magnitude (ϵ)	1.19×10^{-4} (training); 0 (testing)
	Ridge parameter (α)	1.74×10^{-5}
Data split	Training	80% (12,144 raw samples → 12,124 seq.)
	Testing	20% (3,037 raw samples → 3,017 seq.)
Prediction	Horizon range	1–6 steps ahead
	Method	Recursive forecasting
Reproducibility	Random seed	42

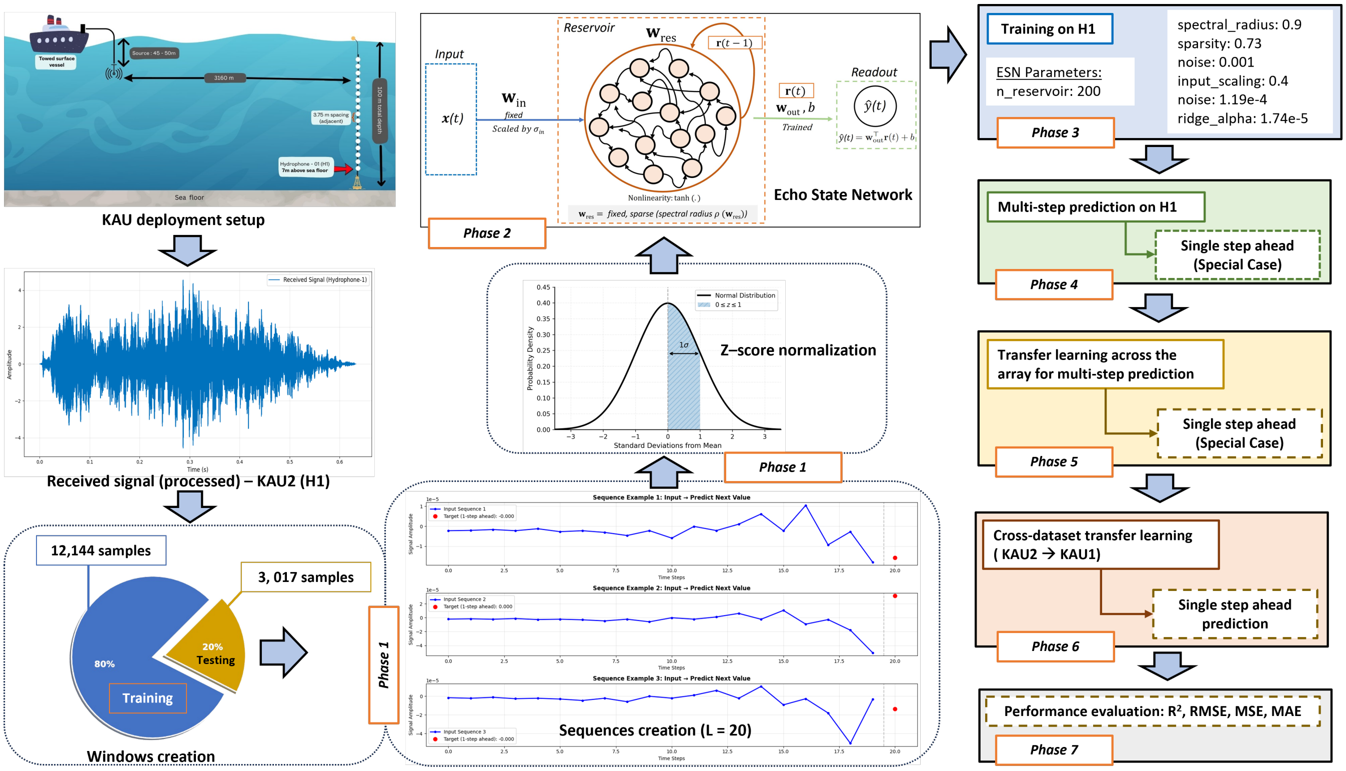


FIGURE 5: Overall workflow of the proposed ESN framework: Phase 1) preprocessing and sequence creation (80/20 split, sliding sequence length with $L = 20$, and Z-score normalization), Phase 2) ESN architecture, Phase 3) training on H1 (source model), Phase 4) multi-step prediction on H1 for $k = 1, \ldots, 6$ with $k = 1$ as the single-step special case, Phase 5) transfer learning across the array for multi-step prediction (with $k = 1$ reported as the special case), Phase 6) cross-dataset transfer learning (KAU2 → KAU1), and Phase 7) performance evaluation using R^2, MSE, RMSE, and MAE.

initialized (warm-up) by driving the reservoir with the last L ground-truth samples $\{x(t-L+1), \ldots, x(t)\}$ using (4) (one sample per ESN time step) to obtain the current state $\mathbf{r}(t)$. This warm-up is used only to initialize the state for forecasting and is separate from the training washout (set to 0); it uses the same sequence length L as in Phase 1.

Starting from $\mathbf{r}(t)$, the k-step-ahead prediction is obtained by recursively applying the trained readout and updating the reservoir state using the predicted sample as the next input. For $i = 0, \ldots, k-1$, the $(i+1)$-step prediction is computed as

$$\hat{y}(t+i+1) = \mathbf{w}_{\text{out}}^{\top}\mathbf{r}(t+i) + b, \tag{8}$$

and the reservoir state is updated as

$$\mathbf{r}(t+i+1) = \tanh\Big(\mathbf{W}_{\text{in}}\,\hat{y}(t+i+1) + \mathbf{W}_{\text{res}}\,\mathbf{r}(t+i)\Big), \tag{9}$$

where the noise term is disabled during recursive forecasting (i.e., $\epsilon = 0$), making (9) deterministic.

Phase 5: Transfer learning across the array – The H1-initialized reservoir is reused for the remaining hydrophones by keeping $\mathbf{W}_{\text{in}}$ and $\mathbf{W}_{\text{res}}$ fixed. For each hydrophone, only the readout $(\mathbf{w}_{\text{out}}, b)$ is re-estimated using the corresponding local training data. Multi-step performance is evaluated using the same recursive forecasting procedure for horizons $k = 1, \ldots, 6$, with $k = 1$ reported as the single-step special case.

Phase 6: Cross-dataset transfer learning – To assess generalization across different underwater environments, the ESN trained on KAU2 is transferred to another environment with a different setting by retaining $\mathbf{W}_{\text{in}}$ and $\mathbf{W}_{\text{res}}$ while re-estimating only the readout $(\mathbf{w}_{\text{out}}, b)$ using data from the target environment. The adapted model is then evaluated under the same preprocessing and prediction settings to validate cross-dataset generalization capability.

Phase 7: Performance evaluation – Performance is quantified using R^2, MSE, RMSE, and MAE. When normalization is applied, metrics are computed after applying the inverse transform to return predictions and targets to the original signal scale. The same evaluation protocol is used across all phases to enable fair comparison.

E. Performance Evaluation Metrics

Prediction performance is evaluated using complementary error metrics computed on the test set. When normalization is applied, all metrics are reported on the original signal scale by inverse-transforming the predicted and ground-truth values prior to evaluation [6], [5]. Let y_i and $\hat{y}_i$ denote the ground-truth and predicted samples, respectively, for $i = 1, \ldots, n$, and let $\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$.

1) Coefficient of Determination (R^2): R^2 measures the fraction of variance in y_i explained by the predictor. Values closer to 1 indicate a better fit, while negative values imply performance worse than the mean predictor (defined when $\sum_{i=1}^{n}(y_i - \bar{y})^2 > 0$):

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}. \tag{10}$$

2) Mean Squared Error (MSE): MSE penalizes large errors more strongly due to the squared term and reflects average error energy:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2. \tag{11}$$

3) Root Mean Squared Error (RMSE): RMSE has the same units as the signal, which supports direct interpretation of the typical error magnitude:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}. \tag{12}$$

4) Mean Absolute Error (MAE): MAE weights all deviations linearly and is less sensitive to outliers than MSE/RMSE:

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|. \tag{13}$$

V. Results and Analysis

This section presents an overview of the experimentally determined results for predicting the underwater acoustic signals from our proposed model. We categorize our findings into three parts. First, we assess the variability of multi-step prediction performance in relation to the forecasting horizon (i.e., from 1 step to 6 steps ahead). Next, we investigate how well the spatial transfer learning can be applied across the 14 functional hydrophones that make up the vertical array. Third, we evaluate cross-dataset transfer learning by reusing the ESN reservoir configured on KAU2 and retraining only the linear readout on data from a different environment setting to assess generalization capability.

A. Multi-Step Prediction Performance on H1

This section reports the prediction performance of the proposed ESN on H1 using the KAU2 dataset. We first present single-step prediction results (the special case of the recursive framework) and then discuss multi-step recursive forecasting performance up to six steps ahead.

1) Single-step Prediction Performance (Special Case)

Single-step prediction is the special case ($k = 1$) of the recursive multi-step framework. On the test set, our proposed ESN achieves $R^2 = 0.9978$, RMSE=0.0152, and MAE=0.0113 (Table 3).

For one-step-ahead prediction on H1, the reservoir is driven by the most recent $L = 20$ true samples to obtain $\mathbf{r}(t)$, and the next sample is then estimated using the linear readout in (5).

Figure 6 compares the actual signal and the ESN output over an approximately 0.12 s segment of the H1 test data.

TABLE 3: Multi-Step Prediction Performance: ESN Model ($N_{\text{res}} = 200$, $L = 20$) on H1 (KAU2 Dataset)

Prediction Step	R^2 Score Test	RMSE Test	MSE Test	MAE Test	N Samples Test
1	0.9978	0.0152	0.0002	0.0113	3017
2	0.9979	0.0151	0.0002	0.0113	3016
3	0.9656	0.0609	0.0037	0.0454	3015
4	0.9660	0.0603	0.0036	0.0450	3014
5	0.8289	0.1352	0.0183	0.1018	3013
6	0.8232	0.1372	0.0188	0.1029	3012

The black rectangle marks a region of interest that is expanded in the inset (0.020–0.030 s). The inset shows that the ESN follows the waveform and peak timing closely, with only small residual deviations in amplitude.

The box plots in Figure 7 summarize the test-set prediction error distributions across consecutive time segments within the same ~0.12 s test segment. Median errors remain close to zero across the segments, indicating minimal systematic bias. The spread is larger in the earliest segments and becomes tighter afterward, while most errors remain concentrated in a narrow band around zero (roughly within ±0.02). A few outliers (red crosses) appear occasionally, representing isolated larger deviations beyond the whisker range.

2) Multi-step Recursive Forecasting Results (Up to 6 Steps)

Multi-step prediction is obtained using a recursive strategy. The ESN first generates a one-step estimate, and that prediction is fed back into the input window to produce the next horizon. As the prediction step increases, errors can accumulate because later forecasts depend increasingly on earlier predicted samples.

As shown in Table 3, very short-horizon forecasts remain highly accurate. Steps 1 and 2 achieve test $R^2 \geq 0.9978$ with RMSE ≤ 0.0152, confirming that the model captures the short-term temporal structure of the underwater acoustic signal. The nearly identical performance at steps 1 and 2 indicates stable behavior over the first recursive iterations.

Figure 8 illustrates the reduction in test R^2 as the prediction step increases from 1 to 6. Performance remains strong up to step 4 ($R^2 > 0.96$), with the first clear drop appearing at step 3 ($R^2 = 0.9656$), where RMSE increases to 0.0609 compared to 0.0152 at step 1. A more noticeable degradation occurs at steps 5–6, where R^2 decreases to 0.8289 and 0.8232, respectively, and RMSE rises to 0.1352–0.1372 with MAE increasing to 0.1018–0.1029. This trend is consistent with recursive error propagation becoming dominant at longer horizons.

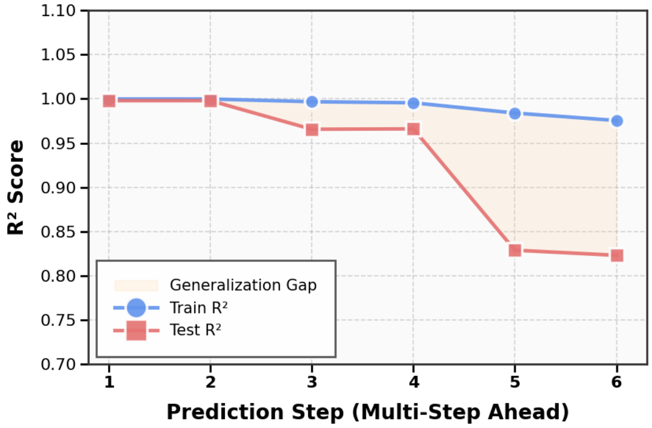


FIGURE 8: Multi-step prediction performance on H1: train and test R^2 across prediction steps 1–6.

Across the evaluated horizons, the error metrics increase monotonically (RMSE: 0.0152 → 0.1372, MAE: 0.0113 → 0.1029, MSE: 0.0002 → 0.0188), reflecting the compounding effect of feeding predictions back into the input window. The number of test samples decreases slightly with the horizon (from 3017 at step 1 to 3012 at step 6) due to the recursive forecasting setup.

This trend is also consistent with the statistical behavior of the received signal under recursive forecasting. As the prediction horizon increases, each forecast is fed back into the next input window, so small one-step errors can accumulate over time. This effect becomes more obvious when the signal exhibits stronger variability or weaker short-term temporal regularity, which naturally makes longer-horizon prediction more difficult.

In summary, our proposed ESN provides accurate forecasts for horizons up to six steps with a minimum R^2 of 82% under the considered channel conditions.

B. Transfer Learning Performance Across Multi-Hydrophone Array

This section evaluates how well the H1-trained ESN generalizes across the vertical hydrophone array in the KAU2 dataset. We reuse the same pre-trained reservoir as a common backbone and adapt the model to each hydrophone by retraining only the linear readout. We first report the single-

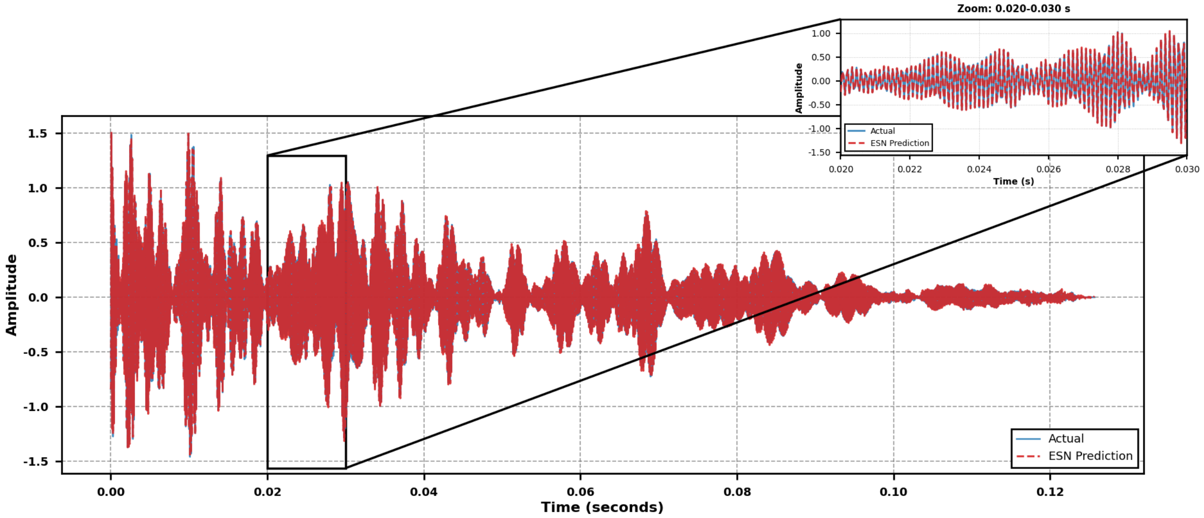


FIGURE 6: ESN single-step prediction on H1 test segment: full signal comparison with a magnified inset over 0.020–0.030 s.

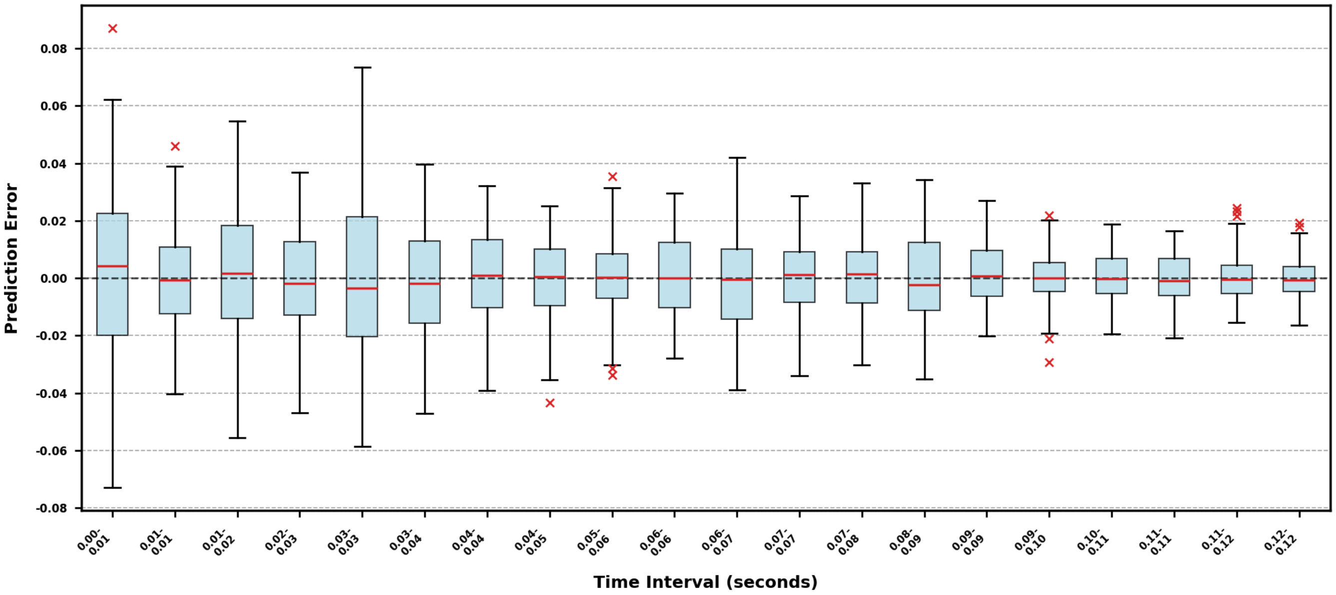


FIGURE 7: Single-step prediction errors on an H1 test segment (about 0.12 s), grouped into consecutive time intervals.

step transfer results, corresponding to the case $k = 1$, and then present multi-step transfer learning results for horizons up to six steps.

1) Single-step Transfer Learning Performance

In the single-step setting ($k = 1$), we keep the reservoir and input weights fixed and retrain only the linear readout for each target hydrophone. Table 4 reports the test performance for hydrophones H3–H16. Note that H2 was non-functional during the experiment, so its results are not reported. All single-step metrics are computed using the same test length of 3017 samples for each hydrophone, which allows a fair comparison across depths.

The transferred model maintains consistently strong accuracy across the array. The single-step test R^2 values range from 0.9976 to 0.9989 across hydrophones H3–H16, indicating that the reservoir trained on H1 captures temporal dynamics that generalize well to other depths. The best single-step transfer performance is observed for H6 and H11 ($R^2 = 0.9989$), while the lowest value is obtained for H14 ($R^2 = 0.9976$). Overall, the variation across the array is small, which suggests that readout-only adaptation is effective despite depth-dependent differences in the received signal characteristics.

TABLE 4: Single-step Transfer Learning Performance Metrics for Hydrophones H3–H16 Using Pre-Trained ESN Model from H1

Hydro-phone	**Depth from Surface (m)**	R^2 **Score Test**	**RMSE Test**	**MSE Test**	**MAE Test**
H3	85.50	0.9986	0.0160	2.6×10^{-4}	0.0124
H4	81.75	0.9985	0.0153	2.3×10^{-4}	0.0112
H5	78.00	0.9987	0.0124	1.5×10^{-4}	0.0094
H6	74.25	0.9989	0.0159	2.5×10^{-4}	0.0121
H7	70.50	0.9984	0.0193	3.7×10^{-4}	0.0152
H8	66.75	0.9985	0.0170	2.9×10^{-4}	0.0133
H9	63.00	0.9981	0.0227	5.2×10^{-4}	0.0166
H10	59.25	0.9986	0.0208	4.3×10^{-4}	0.0161
H11	55.50	0.9989	0.0164	2.7×10^{-4}	0.0127
H12	51.75	0.9980	0.0190	3.6×10^{-4}	0.0146
H13	48.00	0.9984	0.0067	4.5×10^{-5}	0.0052
H14	44.25	0.9976	0.0185	3.4×10^{-4}	0.0135
H15	40.50	0.9987	0.0040	1.6×10^{-5}	0.0031
H16	36.75	0.9979	0.0022	4.8×10^{-6}	0.0016

*H2 was non-functional during the entire experiment, so results are not reported.

Note: All single-step results use 3017 test samples per hydrophone.

The remaining variation across hydrophones is physically reasonable because each hydrophone observes a different depth-dependent propagation channel. Although the same transmitted signal is received across the array, changes in hydrophone depth alter the local multipath structure, delay spread, and interference pattern. Hydrophones whose received signals remain more temporally regular and statistically closer to H1 are more easily adapted through readout-only transfer, whereas hydrophones with more distinct depth-dependent channel responses tend to show slightly larger prediction errors. In addition, the low RMSE values observed for some shallow hydrophones may indicate that their received signals exhibit lower temporal variability or are more closely aligned with the H1 reference pattern, although confirming the exact physical cause would require further channel-level analysis.

The main efficiency benefit comes from updating only the readout. In our ESN, the readout contains $N_{\text{res}} + 1$ trainable parameters (the reservoir state plus a bias term). With $N_{\text{res}} = 200$, this corresponds to 201 updated parameters. The reservoir matrix has N_{res}^2 total connections (here 200×200), but due to sparsity, only 27% of these connections are nonzero. Using the measured connectivity of 0.27, the number of active reservoir connections is $0.27\,N_{\text{res}}^2 = 0.27\times 40{,}000 = 10{,}800$, and these remain fixed during transfer learning. Therefore, the fraction of parameters updated during transfer is $(N_{\text{res}} + 1)/(0.27\,N_{\text{res}}^2 + (N_{\text{res}} + 1))$, which equals $201/(10{,}800 + 201) \approx 1.83\%$ for our settings. This confirms that transfer learning adjusts only the final linear mapping, while the reservoir dynamics remain unchanged.

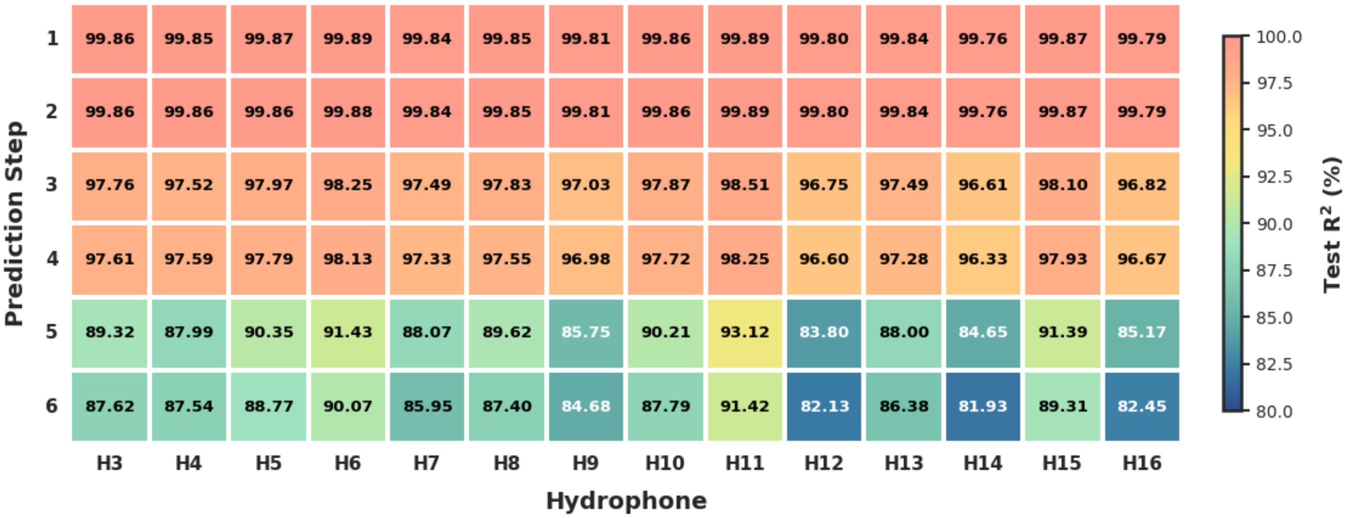
2) Multi-step Transfer Learning Performance Analysis

We next evaluate multi-step transfer learning under the same setting, where the reservoir trained on H1 is kept fixed, and only the readout is retrained for each hydrophone. Figure 9 summarizes the test R^2 scores (in %) across hydrophones H3–H16 and prediction steps 1–6. This heatmap highlights how performance changes with both prediction horizon and hydrophone location.

Two clear trends can be observed. First, short-horizon transfer remains very strong. At steps 1 and 2, all hydrophones achieve $R^2 \geq 99.76\%$, showing that the H1-trained reservoir supports accurate immediate prediction across depths. Although the first-step results are all very high, a slight variation is still visible across hydrophones, with step-1 R^2 values ranging from 99.76% to 99.89%. Second, performance decreases as the horizon increases, which is expected under recursive forecasting because prediction errors can accumulate over steps.

At steps 3 and 4, performance remains high, although differences across hydrophones become more visible. At steps 5 and 6, the degradation is more noticeable, with R^2 further decreasing depending on the hydrophone location. Overall, the results confirm that the transferred reservoir generalizes well across the array for horizons up to six steps, while longer horizons are more sensitive to depth-dependent signal variations and recursive error growth.

This behavior can be interpreted in terms of both propagation physics and signal statistics. Depth-dependent changes in multipath characteristics and delay spread affect the temporal structure observed at each hydrophone, while recursive prediction becomes more sensitive to these differences as the horizon increases. Consequently, hydrophones whose



Prediction Step	H3	H4	H5	H6	H7	H8	H9	H10	H11	H12	H13	H14	H15	H16
1	99.86	99.85	99.87	99.89	99.84	99.85	99.81	99.86	99.89	99.80	99.84	99.76	99.87	99.79
2	99.86	99.86	99.86	99.88	99.84	99.85	99.81	99.86	99.89	99.80	99.84	99.76	99.87	99.79
3	97.76	97.52	97.97	98.25	97.49	97.83	97.03	97.87	98.51	96.75	97.49	96.61	98.10	96.82
4	97.61	97.59	97.79	98.13	97.33	97.55	96.98	97.72	98.25	96.60	97.28	96.33	97.93	96.67
5	89.32	87.99	90.35	91.43	88.07	89.62	85.75	90.21	93.12	83.80	88.00	84.65	91.39	85.17
6	87.62	87.54	88.77	90.07	85.95	87.40	84.68	87.79	91.42	82.13	86.38	81.93	89.31	82.45

FIGURE 9: R^2 performance heatmap: multi-step transfer learning results across hydrophones (H3–H16) and prediction steps (1–6) on the test set.

received signals deviate more strongly from the H1 temporal pattern tend to show slightly faster degradation at longer horizons.

C. Cross-Dataset Transfer Learning Results

This section evaluates the cross-dataset generalization of our proposed ESN by transferring the reservoir trained on KAU2 to a different operating condition. We test whether the reservoir dynamics learned from KAU2 remain useful when the propagation geometry and shallow-water conditions change.

We adapt the model to a different dataset, KAU1, by retraining only the readout. The KAU1 dataset was collected during the same KAM11 experimental campaign as KAU2 [31]. However, it represents a different deployment and propagation scenario. Generalization to KAU1 matters because real underwater systems face changes in geometry and environment. In many cases, retraining a full model for every new condition is not practical. Since the KAU2 setup is already described in detail, we highlight only the factors that make KAU1 different in this cross-dataset test. Compared to KAU2, KAU1 was recorded in slightly deeper water (106 m). The vertical array position also changes, with H1 located 9 m above the sea floor. The transmitter–receiver range is shorter (1080 m). These changes can modify the multipath structure and the temporal dynamics at each hydrophone. They can also shift the signal statistics observed by the ESN. Therefore, KAU1 provides a direct test of whether a reservoir trained on KAU2 can be reused without changing its internal dynamics. We report single-step prediction performance across the vertical array on KAU1 and discuss how the dataset shift affects accuracy.

In this cross-dataset experiment, we evaluate whether a reservoir trained on KAU2 can generalize to the KAU1 environment by freezing the reservoir and retraining only the linear readout using KAU1 training data. Table 5 reports the single-step prediction results across the vertical array for H1 and H3—H16. H2 is excluded because it was non-functional during KAU1 measurements as well. Overall, the transfer remains effective across the functional hydrophones in KAU1. Most R^2 values fall between 0.93 and 0.97, with the highest value observed at H11 ($R^2 = 0.9706$). The RMSE ranges from 0.0995 to 0.1975, and the MAE ranges from 0.0702 to 0.1301. The smallest errors occur at H12, while H3 shows the largest error values, which suggests stronger sensitivity to the changed geometry and multipath conditions at that depth.

The cross-dataset performance variation is also physically reasonable because KAU1 and KAU2 differ in Tx-Rx range, array position, and shallow-water propagation conditions. These differences modify the multipath structure and the temporal statistics of the received signal seen by each hydrophone. As a result, hydrophones whose target responses differ more strongly from those in KAU2 are more difficult to adapt using readout-only transfer.

D. Comparative Analysis and Model Complexity

To assess our proposed model against competitive baselines, we compare it to four other models with the identical experimental protocol. They are a classical autoregressive (AR) model, a GRU network, a temporal convolutional network (TCN), and an LSTM network. To perform a fair comparison, each model uses the same sequence length ($L = 20$), an 80% / 20% split of the data for training and testing, the same Z-Score normalization trained only on the training dataset, and the same recursive feedback mechanism for multiple-step forecasting. In this recursive setup, each one-step forecast is fed back as input to generate the subsequent forecast. By using the same experimental

TABLE 5: Cross-dataset transfer learning (KAU2→KAU1) single-step prediction performance on KAU1 (H1 and H3–H16)

Hydro-phone	Depth from surface (m)	R^2 Test	RMSE Test	MSE Test	MAE Test
H1	97.00	0.9697	0.1339	0.0179	0.0994
H3	89.50	0.9338	0.1975	0.0390	0.1301
H4	85.75	0.9600	0.1426	0.0203	0.1025
H5	82.00	0.9454	0.1600	0.0256	0.1128
H6	78.25	0.9413	0.1714	0.0294	0.1293
H7	74.50	0.9503	0.1479	0.0219	0.1026
H8	70.75	0.9575	0.1352	0.0183	0.0949
H9	67.00	0.9661	0.1121	0.0126	0.0824
H10	63.25	0.9503	0.1389	0.0193	0.0974
H11	59.50	0.9706	0.1308	0.0171	0.0921
H12	55.75	0.9502	0.0995	0.0099	0.0702
H13	52.00	0.9460	0.1394	0.0194	0.1019
H14	48.25	0.9499	0.1472	0.0217	0.1067
H15	44.50	0.9593	0.1170	0.0137	0.0873
H16	40.75	0.9357	0.1150	0.0132	0.0811

Metrics are computed on the KAU1 test split using sequence length $L = 20$, giving 10,206 test sequences per hydrophone (from 51,130 samples with an 80/20 split).

*Hydrophone H2 was non-functional during KAU1 measurements and is excluded.

Depths computed for KAU1 using water depth 106 m, H1 height 9 m above sea floor, and 3.75 m spacing.

protocol for all models, differences in model performance are due to model characteristics rather than experimental conditions.

The AR model fits a linear autoregression with $L = 20$ lagged terms plus an intercept via ordinary least squares on the training signal in original scale. No normalization is applied since the AR model is scale-invariant under ordinary least squares (OLS). The GRU and LSTM networks each comprise a single recurrent layer with 64 hidden units, a dropout rate of 0.1, and a linear dense output layer, trained with the Adam optimizer (initial learning rate 10^{-3}, reduced on plateau), a batch size of 256, and a maximum of 60 epochs with early stopping on a 10% validation split. The TCN uses two dilated causal convolutional blocks with 64 filters, kernel size 3, and exponentially growing dilation rates $(1, 2)$, matching the 64-unit capacity of the recurrent baselines, with the same optimizer and training schedule. This matched-capacity design isolates the effect of architecture rather than raw model size. Table 6 summarizes the key configuration parameters for all models.

With the configurations of all five models established in Table 6, Table 7 presents the recursive multi-step forecasting results on H1 (KAU2) for all five models at prediction steps $k = 1, \ldots, 6$.

TABLE 6: Baseline model configurations used for comparison with the proposed ESN (see Table 2 for ESN configuration)

Parameter	LSTM	GRU	TCN	AR
Hidden units / filters	64 units	64 units	64 filters	–
Hidden layers	1	1	2 dilated blocks	–
Dropout	0.1	0.1	0.1	–
Optimizer	Adam	Adam	Adam	OLS
Learning rate	10^{-3}	10^{-3}	10^{-3}	–
Batch size	256	256	256	–
Max epochs	60	60	60	1 pass

All deep models use early stopping (patience 10), validation split 0.1, and MSE loss. AR: 20 lag coefficients + 1 intercept via OLS.

As shown in Table 7, the proposed ESN consistently outperforms all baseline models for recursive multi-step forecasting on the KAU2 H1 test set. Specifically, it yields the highest R^2 and the lowest RMSE, MAE, and MSE at every prediction horizon from step 1 to step 6. Although AR remains competitive at the shortest horizons, its accuracy declines more rapidly as the forecasting step increases. A similar, but clearer, degradation is observed for LSTM, while GRU and TCN exhibit substantially weaker performance, including negative R^2 values at several horizons. Overall, these results indicate that the proposed ESN is not only more accurate at short horizons but also more robust to recursive error accumulation at longer horizons, thereby providing the strongest overall multi-step prediction performance among the evaluated models.

The greater multi-step stability of the ESN under recursive forecasting can be attributed to two structural properties. First, the fixed reservoir provides stable nonlinear projections of the input regardless of input quality. Since the recurrent weights are never trained, they cannot overfit the training distribution or amplify the distribution shift that occurs when predicted samples replace ground-truth inputs during recursive rollout. Second, the linear readout produces bounded, smooth outputs that do not compound errors as aggressively as the gating mechanisms in LSTM and GRU, which were optimized for ground-truth inputs and may respond unpredictably when fed back their own predictions. In contrast, AR degrades gracefully at short horizons due to its linear structure but lacks the nonlinear representational capacity needed for longer horizons.

Table 8 reports the complexity metrics for all five models measured under identical hardware and experimental conditions, including training time, inference latency (mean ± standard deviation over 200 single-sample runs), peak memory usage, and trainable parameter count.

Several observations are identified from Table 8. First, our proposed ESN trains in 8.42 s and provides an average inference time for each sample of 0.49 ± 0.01 ms. As such,

TABLE 7: Recursive multi-step forecasting performance on H1 (KAU2), $L = 20$ (test set). Best value per row in **bold**

	R^2					**RMSE**				
Step	ESN	LSTM	GRU	TCN	AR	ESN	LSTM	GRU	TCN	AR
1	**0.9978**	0.9893	0.2227	0.6653	0.9974	**0.0152**	0.0341	0.2897	0.1901	0.0167
2	**0.9979**	0.9842	0.0813	0.6171	0.9977	**0.0151**	0.0413	0.3147	0.2031	0.0156
3	**0.9656**	0.8606	-0.4092	-0.2832	0.9275	**0.0609**	0.1225	0.3894	0.3716	0.0883
4	**0.9660**	0.7087	-0.1972	-0.5079	0.9405	**0.0603**	0.1765	0.3577	0.4015	0.0798
5	**0.8289**	0.4963	-0.2138	-1.3732	0.3768	**0.1352**	0.2320	0.3602	0.5036	0.2581
6	**0.8232**	0.0300	0.0339	-1.8792	0.5171	**0.1372**	0.3213	0.3207	0.5536	0.2267

	MAE					**MSE**				
Step	ESN	LSTM	GRU	TCN	AR	ESN	LSTM	GRU	TCN	AR
1	**0.0113**	0.0242	0.1967	0.1448	0.0124	**0.0002**	0.0012	0.0839	0.0361	0.0003
2	**0.0113**	0.0288	0.2134	0.1495	0.0115	**0.0002**	0.0017	0.0989	0.0413	0.0002
3	**0.0454**	0.0860	0.2610	0.2853	0.0661	**0.0037**	0.0150	0.1516	0.1381	0.0078
4	**0.0450**	0.1237	0.2395	0.2966	0.0595	**0.0036**	0.0311	0.1279	0.1612	0.0064
5	**0.1018**	0.1645	0.2401	0.3842	0.1934	**0.0183**	0.0538	0.1298	0.2537	0.0666
6	**0.1029**	0.2266	0.2144	0.4085	0.1693	**0.0188**	0.1033	0.1028	0.3065	0.0514

TABLE 8: Complexity evaluation across all five models on KAU2 H1 ($L = 20$)

Model	**Parameters**	**Train (s)**	**Latency (ms)**	**Memory (MB)**
Proposed ESN	201	8.42	0.49 ± 0.01	38.06
LSTM	16,961	48.16	90.18 ± 8.24	5.51
GRU	12,929	12.03	88.20 ± 5.05	5.23
TCN	49,601	13.54	88.03 ± 6.41	6.86
AR	21	0.03	0.009 ± 0.001	10.18

ESN trainable parameters include only the readout layer ($N_{\text{res}}{+}1 = 201$). All reservoir weights are fixed and therefore do not need to be updated during training. The higher memory footprint reported for ESN and AR reflects full process-level memory tracking, whereas LSTM, GRU, and TCN use internal memory management that can allocate tensor buffers outside the tracked scope.

the proposed ESN inference is approximately $184\times$ faster than the LSTM (90.18 ms), GRU (88.20 ms), and the TCN (88.03 ms). This gap in the inference time is expected because ESN inference involves a deterministic sequence of matrix–vector operations followed by a single linear readout. However, LSTM, GRU, and TCN incur additional runtime overhead per call due to framework-level execution and bookkeeping. The AR model attains the fastest training (0.03 s) and inference (0.009 ms) because it is purely linear, admits a closed-form solution, and has only 21 parameters.

For peak memory, the ESN reports 38.06 MB and AR reports 10.18 MB compared with 5 to 7 MB for the neural baselines. This gap is primarily due to measurement scope rather than an inherent memory disadvantage. During training, the ESN explicitly forms the full state matrix $\mathbf{R} \in \mathbb{R}^{N_{\text{train}} \times N_{\text{res}}}$ as a dense array (approximately 19 MB), and all of these memory allocations are fully captured in the reported peak memory. Similarly, AR stores the complete data matrix for its closed-form solution, and its allocations are fully tracked. In contrast, LSTM, GRU, and TCN employ internal memory management that allocates tensor buffers outside the scope of this tracking method and thus do not fully reflect their actual memory footprints. The ESN and AR figures therefore represent complete and accurate memory measurements, and the associated overhead remains modest and bounded.

The computational advantage of the proposed ESN stems from its training mechanism. During training, the reservoir is driven sequentially through all N_{train} input sequences, each of length L, and the final reservoir state is collected for each sequence. With sparse recurrent connectivity $c = 0.27$, the number of active recurrent connections in $\mathbf{W}_{\text{res}}$ is $cN_{\text{res}}^2 = 0.27 \times 200^2 = 10{,}800$. Therefore, the dominant cost of reservoir state collection is $\mathcal{O}(N_{\text{train}} \cdot L \cdot cN_{\text{res}}^2)$. After state collection, the readout is estimated in closed form by ridge regression on the collected state matrix $\mathbf{R} \in \mathbb{R}^{N_{\text{train}} \times N_{\text{res}}}$, with complexity $\mathcal{O}(N_{\text{train}} N_{\text{res}}^2 + N_{\text{res}}^3)$. Hence, the overall ESN training complexity is

$$\mathcal{O}(N_{\text{train}} \cdot L \cdot cN_{\text{res}}^2 + N_{\text{train}} N_{\text{res}}^2 + N_{\text{res}}^3).$$

Since only the linear readout is trained, this computation is performed in a single pass without iterative backpropagation. For window-based inference over a sequence of length L, the cost is $\mathcal{O}(L \cdot cN_{\text{res}}^2)$, which is independent of the training set size.

In contrast, the recurrent baselines require iterative gradient-based optimization via backpropagation through

TABLE 9: Training complexity order for all five models

Model	Training Complexity
Proposed ESN	$\mathcal{O}(N_{\text{train}} \cdot L \cdot cN_{\text{res}}^2 + N_{\text{train}}N_{\text{res}}^2 + N_{\text{res}}^3)$
LSTM	$\mathcal{O}(E \cdot N_{\text{train}} \cdot L \cdot (DN_{\text{units}} + N_{\text{units}}^2))$
GRU	$\mathcal{O}(E \cdot N_{\text{train}} \cdot L \cdot (DN_{\text{units}} + N_{\text{units}}^2))$
TCN	$\mathcal{O}(E \cdot N_{\text{train}} \cdot L \cdot k_{\text{tcn}} \cdot F^2)$
AR	$\mathcal{O}(N_{\text{train}} \cdot L^2 + L^3)$

E: number of training epochs, D: input dimension, k_{tcn}: TCN kernel size, F: number of filters.

time (BPTT). The training cost of an LSTM with N_{units} hidden units is

$$\mathcal{O}\big(E \cdot N_{\text{train}} \cdot L \cdot (DN_{\text{units}} + N_{\text{units}}^2)\big),$$

where E denotes the number of training epochs and D is the input dimension. When $D \ll N_{\text{units}}$, the dominant term reduces to $\mathcal{O}(E \cdot N_{\text{train}} \cdot L \cdot N_{\text{units}}^2)$. The GRU has the same asymptotic training complexity as the LSTM, but with a smaller constant factor due to its simpler gating structure. For a fixed-depth temporal convolutional network (TCN), the training cost scales as

$$\mathcal{O}\big(E \cdot N_{\text{train}} \cdot L \cdot k_{\text{tcn}} \cdot F^2\big),$$

where k_{tcn} is the TCN kernel size and F is the number of filters. The AR model has training complexity

$$\mathcal{O}(N_{\text{train}} \cdot L^2 + L^3),$$

assuming the coefficients are estimated by ordinary least squares through the normal equations on a lagged design matrix. Therefore, among the considered baselines, AR is computationally efficient only when the lag order remains small. However, its representational capacity is limited to linear temporal dependencies, and it cannot capture the nonlinear dynamics of underwater acoustic channels. In this context, the proposed ESN provides a favorable trade-off: it preserves single-pass training without iterative optimization while benefiting from a high-dimensional nonlinear reservoir state space for accurate multi-step forecasting. The training complexity orders are summarized in Table 9 and are consistent with the measured training times reported in Table 8.

To position the proposed framework relative to the closest ESN-based UWAC baseline, we benchmark against the configuration reported in [7]. Unlike [7], which follows a Tx→Rx sequence-to-sequence formulation with a larger reservoir, our approach uses an Rx-only sequence-to-one forecasting formulation with recursive multi-step rollout, a compact 200-neuron reservoir, and readout-only transfer across hydrophones–yielding strong accuracy with substantially reduced computational cost.

We compare the proposed complexity-reduced ESN against the ESN configuration reported as optimal for Data 3 in [7], by re-running that configuration on the same KAU2 H1 signal and evaluation pipeline. In [7], Data 3 corresponds to the Lake Tahoe experiment under mild disturbance. The Onasami Data-3 ESN employs Xavier initialization, a reservoir size of $N_{\text{res}} = 578$, $\rho(\mathbf{W}_{\text{res}}) = 0.5$, tanh activation, and a ridge-regression linear readout. Since other training hyperparameters (e.g., connectivity, input scaling, training noise, and ridge regularization strength) are not explicitly reported for Data 3, we set them to the best-matching values used in our model to ensure a fair and reproducible comparison on KAU2.

The proposed model uses $N_{\text{res}} = 200$ neurons, compared with $N_{\text{res}} = 578$ in [7], corresponding to a 65.4% reduction in reservoir size. Since the recurrent matrix scales with N_{res}^2, this reduction also yields a substantial decrease in recurrent parameters (approximately $(200/578)^2 \approx 0.12$, i.e., about 88% fewer recurrent weights before accounting for sparsity). Using the same sequence length ($L = 20$), the proposed model also reduces runtime: the measured training time is 20.264 seconds for the proposed ESN versus 87.859 seconds for the Onasami-style ESN on the same KAU2 H1 setup.

Table 10 reports the recursive multi-step forecasting performance on H1 (KAU2) up to six steps ahead using R^2, RMSE, MSE, and MAE. The proposed ESN maintains a higher R^2 and lower error values across the entire 1–6 step horizon under recursive feedback. At short horizons (steps 1–2), both models perform strongly. However, the proposed model yields ($R^2 \approx 0.9980$) with substantially smaller RMSE/MAE. The gap increases at longer horizons, for example, at step 3, the proposed ESN retains $R^2 = 0.9656$, whereas the Onasami-style configuration drops to $R^2 = 0.3906$. From step 5 onward, the Onasami-style model yields negative R^2, indicating performance worse than a mean predictor at those horizons, while the proposed ESN remains stable and positive up to step 6.

Overall, these results show that the proposed hyperparameter configuration achieves a favorable balance between predictive accuracy and recursive forecasting stability. Further, the proposed model achieves good computational efficiency while substantially reducing model complexity, thereby making it well-suited for deployment on resource-constrained underwater communication platforms.

E. Limitations and Challenges

Several limitations and challenges affect the current ESN implementation for underwater acoustic signal prediction. One major issue is the lack of publicly available real at-sea datasets. Collecting high-quality underwater data is costly and difficult to capture, and even when data are available, the transmitter–receiver distance is often small due to practical deployment limits. This makes it hard to test the model under long-range conditions that are common in real operations. Another limitation appears in multi-step forecasting. Since we use recursive prediction, each new step depends on the previous predicted output, so small errors build up over time. Because of this, prediction accuracy drops as the horizon increases, and long-horizon forecasting is still not reliable.

TABLE 10: Recursive multi-step forecasting performance comparison on H1 (KAU2), $L = 20$ (test set)

Step	R^2		**RMSE**		**MSE**		**MAE**	
	Proposed	**Onasami (Data 3)**	**Proposed**	**Onasami (Data 3)**	**Proposed**	**Onasami (Data 3)**	**Proposed**	**Onasami (Data 3)**
1	0.9978	0.9551	0.0152	0.0696	0.0002	0.0048	0.0113	0.0505
2	0.9979	0.9619	0.0151	0.0641	0.0002	0.0041	0.0113	0.0465
3	0.9656	0.3906	0.0609	0.2561	0.0037	0.0656	0.0454	0.1859
4	0.9660	0.5131	0.0603	0.2281	0.0036	0.0520	0.0450	0.1652
5	0.8289	-1.4766	0.1352	0.5145	0.0183	0.2647	0.1018	0.3732
6	0.8232	-0.8588	0.1372	0.4448	0.0188	0.1979	0.1029	0.3222

Cross-dataset transfer learning also shows that the model is sensitive to environmental changes. When transferring from one dataset to another, performance is usually lower than within-dataset testing, which is expected because underwater channels can change with water depth, range, and multipath structure. This means the method should be validated on more environments before drawing strong conclusions about generalization. In addition, hydrophone reliability can strongly affect results. The non-functional H2 hydrophone shows that faulty or poor-quality channels can cause large performance degradation, so real deployments may need basic fault detection or signal-quality checks before applying transfer learning across the entire array. The experiments are also limited to shallow-water conditions and a narrow set of signal types and frequency bands, so performance in deep water or under different waveforms has not yet been confirmed. Finally, although the 200-neuron ESN reduces complexity, real-time use on underwater platforms has not been demonstrated, and factors such as power usage, processing delay, and onboard memory limits should be tested in realistic conditions.

VI. Conclusion

This study demonstrates that a complexity-reduced ESN can predict underwater acoustic signals accurately while keeping computational complexity low. By reducing the reservoir to 200 neurons, the proposed model substantially lowers the number of parameters compared to larger ESN designs, yet it still produces highly accurate predictions on real experimental data. For single-step prediction, the model achieves a test R^2 of 0.9978, confirming that strong performance is possible with a small reservoir and simple training of the linear readout. The same model also performs multi-step forecasting using a recursive strategy. It remains reliable over short horizons, with R^2 staying above 0.96 for steps 1–4, while performance drops for longer horizons as errors accumulate. Therefore, prediction horizons up to around 6 steps appear most practical for real-time use, where stable performance and fast processing are important.

Another main contribution is the transfer learning strategy used to extend the predictor across the hydrophones in different depths. Instead of retraining the full architecture, the reservoir is kept fixed, and only the linear readout is retrained for each hydrophone. Under the nonzero-parameter accounting used in this work, this adaptation updates only about 1.83% of the network parameters when moving to a new hydrophone location. This leads to fast retraining while maintaining strong performance across the 15 functional hydrophones, covering depths from 40.5 m to 89.5 m. In addition, cross-dataset transfer from KAU2 to KAU1 demonstrates that the approach can adapt across different shallow-water environments, with most hydrophones achieving R^2 values above 0.93 despite changes in geometry and propagation conditions. Overall, the proposed framework provides a practical trade-off between prediction accuracy, model compactness, and quick adaptation, making it a strong candidate for resource-limited underwater platforms.

Future work will focus on reducing the drop in long-horizon forecasting, validating the method on more real at-sea datasets with larger transmitter–receiver ranges, and extending evaluation to deep-water conditions, different frequency bands, and other waveform types. Finally, real-time deployment on onboard underwater devices will be explored to evaluate processing delay, power consumption, and long-term robustness under realistic operating constraints.

Acknowledgments

This work was supported in part by the Canada Excellence Research Chair (CERC) Program CERC-2022-00109, in part by the Natural Sciences and Engineering Research Council of Canada (NSERC) Discovery Grant Program RGPIN-2025-04941, in part by Canada First Research Excellence Fund Transforming Climate Action, Grant TCA-LRP-20241-MUN-1.3-03, and in part by Canada Research Chairs Program CRC-2022-00187. The work of D.-B. Ha is partially funded by the Vietnam National Foundation for Science and Technology Development (NAFOSTED) under grant number 102.04-2025.72. The work of B. Canberk is partially supported by The Scientific and Technological Research Council of Türkiye (TÜBİTAK) 1515 Frontier R&D Laboratories Support Program for Türk Telekom 6G R&D Lab under project number 5249902.

REFERENCES

[1] A. V. Gadagkar and B. R. Chandavarkar, "A Comprehensive Review on Wireless Technologies and their Issues with Underwater Communications," in *Proc. 12th Int. Conf. Comput. Commun. Netw. Technol. (ICCCNT)*, Kharagpur, India, 06-08 Jul. 2021, pp. 1–6.

[2] C. M. G. Gussen, P. S. R. Diniz, M. L. R. de Campos, W. A. Martins, F. M. Costa, and J. N. Gois, "A Survey of Underwater Wireless Communication Technologies," *Journal of Communication and Information Systems*, vol. 31, pp. 242–255, Oct. 2016.

[3] Y. Chang, "A Comprehensive Review of Underwater Optical, Acoustic, and Electromagnetic Communication," in *Proc. 3rd Int. Conf. Softw. Eng. Mach. Learn.*, Apr. 2025, pp. 197–202.

[4] L. Huang, Y. Wang, Q. Zhang, J. Han, W. Tan, and Z. Tian, "Machine Learning for Underwater Acoustic Communications," *IEEE Wireless Communications*, vol. 29, pp. 102–108, May 2022.

[5] R. V. S. S. B. R., R. Hossain, S. Read, S. Divya, M. Natesan, and R. Dhilipkumar, "Channel Modeling for Underwater Acoustic Communication Systems Using Liquid State Machines with Reservoir Computing," in *Proc. 3rd Int. Conf. Integr. Circuits Commun. Syst. (ICICACS)*, Raichur, India, 21-22 Feb. 2025, pp. 1–6.

[6] O. Onasami, D. Adesina, and L. Qian, "Underwater Acoustic Communication Channel Modeling using Deep Learning," in *Proc. 15th Int. Conf. Underwater Netw. Syst. (WUWNet)*, Shenzhen, China, Nov. 2021, pp. 1–8.

[7] O. Onasami, M. Feng, H. Xu, M. Haile, and L. Qian, "Underwater Acoustic Communication Channel Modeling Using Reservoir Computing," *IEEE Access*, vol. 10, pp. 56 550–56 563, May 2022.

[8] H. P. Yoong, K. B. Yeo, K. T. K. Teo, and W. L. Wong, "Modeling of Acoustic Channel for Underwater Wireless Communication System in AUV Application," in *Proc. 14th UKSim Int. Conf. Comput. Modelling Simul.*, Cambridge, UK, 28-30 Mar. 2012, pp. 603–607.

[9] M. Chitre, S. Shahabudeen, and M. Stojanovic, "Underwater Acoustic Communications and Networking: Recent Advances and Future Challenges," *Marine Technology Society Journal*, vol. 42, no. 1, pp. 103–116, May 2008.

[10] K. Rehan and G. Qiao, "A Survey of Underwater Acoustic Communication and Networking Techniques," *Research Journal of Applied Sciences, Engineering and Technology*, vol. 5, pp. 778–789, Jul. 2013.

[11] R. Diamant and L. H.-J. Lampe, "Low Probability of Detection for Underwater Acoustic Communication: A Review," *IEEE Access*, vol. 6, pp. 19 099–19 112, Dec. 2018.

[12] M. C. Domingo, "Overview of channel models for underwater wireless communication networks," *Physical Communication*, vol. 1, no. 3, pp. 163–182, Sep. 2008.

[13] M. Stojanovic and J. Preisig, "Underwater acoustic communication channels: Propagation models and statistical characterization," *IEEE Communications Magazine*, vol. 47, no. 1, pp. 84–89, Feb. 2009.

[14] Y. Mahmutoglu, A. Yazgan, E. Tugcu, and I. H. Cavdar, "The Effects of Sea Environmental Conditions on the Underwater Acoustic Communication Systems," in *Proc. 36th International Conference on Telecommunications and Signal Processing (TSP)*, Jul. 2013, pp. 133–137.

[15] S. I. Siddiqui, M. Ludvigsen, and H. Dong, "Analysis, verification and optimization of AUV navigation using underwater acoustic models," in *Proc. OCEANS 2015 – MTS/IEEE Washington*, Oct. 2015, pp. 1–6.

[16] M. M. Khan, R. H. Ashique, B. N. Liya, M. Sajjad, M. A. Rahman, and M. T. H. Amin, "New Wavelet Thresholding Algorithm in Dropping Ambient Noise from Underwater Acoustic Signals," *Journal of Electromagnetic Analysis and Applications*, vol. 7, pp. 53–60, Mar. 2015.

[17] N. Morozs, W. Gorma, B. T. Henson, L. Shen, P. D. Mitchell, and Y. V. Zakharov, "Channel Modeling for Underwater Acoustic Network Simulation," *IEEE Access*, vol. 8, pp. 136 151–136 175, Jul. 2020.

[18] R. Jiang, S. Cao, C. Xue, and L. Tang, "Modeling and Analyzing of Underwater Acoustic Channels with Curvilinear Boundaries in Shallow Ocean," in *Proc. IEEE Int. Conf. Signal Process., Commun. Comput. (ICSPCC)*, Xiamen, China, 22-25 Oct. 2017, pp. 1–6.

[19] R. Jiang, X. Wang, S. Cao, J. Zhao, and X. Li, "Deep Neural Networks for Channel Estimation in Underwater Acoustic OFDM Systems," *IEEE Access*, vol. 7, pp. 23 579–23 594, Feb. 2019.

[20] Y. Zhang, J. Li, Y. V. Zakharov, J. Li, Y. Li, C. Lin, and X. Li, "Deep Learning Based Single Carrier Communications Over Time-Varying Underwater Acoustic Channel," *IEEE Access*, vol. 7, pp. 38 420–38 430, Mar. 2019.

[21] Y. Zhang, R. Venkatesan, O. A. Dobre, and C. Li, "Efficient Estimation and Prediction for Sparse Time-Varying Underwater Acoustic Channels," *IEEE Journal of Oceanic Engineering*, vol. 45, no. 3, pp. 1112–1125, May 2019.

[22] C. S. Sai Ganesh, V. B. N. Jyothi, S. R. Y. Aoutithiye Barathwaj, and R. Azhagumurugan, "Machine Learning Based Classification and Modelling of Underwater Acoustic Communication," in *Proc. OCEANS 2022 – Chennai*, Feb. 2022, pp. 1–7.

[23] N. Q. T. Thoong, A. A. Cheema, B. Canberk, D. T. Tran, O. A. Dobre, and T. Q. Duong, "Channel Estimation for Reconfigurable Intelligent Surface-Aided 6G NOMA Systems: A Quantum Machine Learning Approach," *IEEE Transactions on Network Science and Engineering*, vol. 13, pp. 2197–2218, Sep. 2025.

[24] C. Nguyen, T. M. Hoang, and A. A. Cheema, "Channel Estimation Using CNN-LSTM in RIS-NOMA Assisted 6G Network," *IEEE Transactions on Machine Learning in Communications and Networking*, vol. 1, pp. 43–60, May 2023.

[25] C. Nguyen, A. A. Cheema, C. Kurnaz, A. Rahimian, C. Brennan, and T. Q. Duong, "Deep Learning Models for Time-Series Forecasting of RF-EMF in Wireless Networks," *IEEE Open Journal of the Communications Society*, vol. 5, pp. 1399–1414, Feb. 2024.

[26] S. Hussain, S. F. N. Bacha, A. A. Cheema, B. Canberk, and T. Q. Duong, "Geometrical Features Based-mmWave UAV Path Loss Prediction Using Machine Learning for 5G and Beyond," *IEEE Open Journal of the Communications Society*, vol. 5, pp. 5667–5679, Aug. 2024.

[27] J. Wang, S. Ma, Y. Cui, H. Sun, M. Zhou, B. Wang, J. Li, and L. Liu, "Signal Detection for Full-Duplex Cognitive Underwater Acoustic Communications with SIC Using Model-Driven Deep Learning Network," in *Proc. IEEE Int. Conf. Signal Process., Commun. Comput. (ICSPCC)*, Dalian, China, 20-22 Sep. 2019, pp. 1–6.

[28] K. Zheng, B. Qian, S. Li, Y. Xiao, W. Zhuang, and Q. Ma, "Long-Short Term Echo State Network for Time Series Prediction," *IEEE Access*, vol. 8, pp. 91 961–91 974, May 2020.

[29] T. Akiyama and G. Tanaka, "Computational Efficiency of Multi-Step Learning Echo State Networks for Nonlinear Time Series Prediction," *IEEE Access*, vol. 10, pp. 28 535–28 544, Mar. 2022.

[30] Z. Li and G. Tanaka, "Multi-reservoir echo state networks with sequence resampling for nonlinear time-series prediction," *Neurocomputing*, vol. 467, pp. 115–129, Sep. 2021.

[31] W. S. Hodgkiss and J. C. Preisig, "Kauai Acomms MURI 2011 (KAM11) Experiment," in *Proc. Eur. Conf. Underwater Acoust.*, Edinburgh, UK, Jul. 2012, pp. 993–1000.

[32] P. A. V. Walree, F.-X. Socheleau, R. Otnes, and T. Jenserud, "The Watermark Benchmark for Underwater Acoustic Modulation Schemes," *IEEE Journal of Oceanic Engineering*, vol. 42, no. 4, pp. 1007–1018, Oct. 2017.

[33] H. Jaeger, "The "Echo State" Approach to Analysing and Training Recurrent Neural Networks," German National Research Center for Information Technology (GMD), Bonn, Germany, Tech. Rep. 148, 2001.

[34] S. Prasanth and C. B. Phung, "A Hybrid Ensemble Deep Learning and Reservoir Computing Approach for Exchange Rate Prediction," *EAI Endorsed Transactions on Tourism, Technology and Intelligence*, vol. 2, no. 4, Dec. 2025.

[35] M. Lukoševičius, "A Practical Guide to Applying Echo State Networks," in *Neural Networks: Tricks of the Trade*, G. Montavon, G. B. Orr, and K.-R. Müller, Eds. Berlin, Heidelberg: Springer, 2012, pp. 659–686.

[36] I. B. Yildiz, H. Jaeger, and S. J. Kiebel, "Re-visiting the echo state property," *Neural Networks*, vol. 35, pp. 1–9, Nov. 2012.