

Comparative Evaluation of AI Models (CNN, GRU, SimpleRNN) for Bathymetric Depth Prediction from Multibeam Echo Sounder Data

PHAM Ngoc Quang^{1,2}, NGUYEN Gia Trong^{1,2}, HOANG Ngoc Ha¹,
NGUYEN Dinh Hai³, TRAN Duc Vinh³, LE Duc Tinh^{1,*}

¹ Hanoi University of Mining and Geology, Hanoi 10000, Vietnam

² Geodesy and Environment research group, Hanoi University of Mining and Geology, Hanoi 10000, Vietnam

³ Viet Nam's people Naval Hydrographic and Oceanographic Department

* Corresponding author: leductinh@hmg.edu.vn

Abstract: This study evaluates the effectiveness of three deep learning models - Convolutional Neural Network (CNN), Gated Recurrent Unit (GRU), and Simple Recurrent Neural Network (SimpleRNN) - in analyzing multibeam echo sounding (MBES) data collected from the southern coastal waters of Vietnam. The objective is to assess the potential of these models in standardizing and improving the accuracy of bathymetric data processing, with a focus on modeling depth variations. Input data were extracted from MBES surveys conducted in 2023, and structured into standardized CSV formats containing longitude, latitude, and depth. Each model was trained and validated using the same dataset, and performance was evaluated using key metrics such as Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and Mean Squared Error (MSE). Results indicate that the GRU model achieved the highest prediction accuracy, with $RMSE = 0.1961$, $MAE = 0.1485$, and $MSE = 0.0384$. In comparison, the CNN model yielded $RMSE = 0.2072$, $MAE = 0.1537$, and $MSE = 0.0429$, while SimpleRNN performed the least effectively. Among the three, the GRU model demonstrated faster convergence and better capability in capturing complex depth variations. These findings suggest that deep learning - particularly GRU - offers a promising alternative to traditional MBES data processing techniques. The approach proposed in this study may help enhance the quality, consistency, and efficiency of seabed mapping, particularly in areas where post-processing resources are limited. Further improvements could be achieved by incorporating environmental variables and uncertainty estimation into the modeling framework. The models were applied to standardize MBES measurements into regular grids, predict seabed depths, and compare the consistency of depth information derived from different sources. Their comparative performance provides practical insights into the suitability of AI models for bathymetric applications.

Keywords: CNN(Conv1D); GRU; SimpleRNN; Multibeam Echo-sounder data

1. Introduction

Multibeam Echo Sounding (MBES) systems play a crucial role in accurately mapping the seafloor by providing high-resolution bathymetric data across wide swaths. These data are fundamental not only for marine navigation safety, coastal engineering, and environmental monitoring, but also for military applications such as submarine navigation, mine detection, and strategic seabed surveillance (Calder & Mayer, 2003), (Lurton, 2002). In naval operations, precise seafloor characterization enhances mission planning, ensures safe maneuvering in shallow waters, and supports undersea warfare systems. Moreover, MBES data are increasingly integrated with geospatial analytics and AI-driven interpretation to provide actionable insights for defense and marine spatial planning (Clark, 2020). The ability to continuously update seabed maps using MBES enhances both civilian and naval situational awareness, particularly in complex coastal zones and disputed maritime regions.

Multibeam Echo Sounding data processing is essential for accurate bathymetric mapping, involving steps such as sound velocity correction, beamforming, outlier detection, and surface generation (Mayer, 2006a). Traditional approaches rely on manual editing and rule-based filters to remove noise and correct artifacts caused by refraction or motion compensation errors (Lurton, 2002). However, these methods are labor-intensive and often subjective. Recently, artificial intelligence (AI) techniques, particularly machine learning and deep learning, have been introduced to automate data cleaning, seafloor classification, and anomaly detection (Wan et al., 2022). For example, convolutional neural networks (CNNs) and recurrent models like GRU or LSTM can learn spatial-temporal features directly from raw MBES data, improving

accuracy and reducing manual workload. The integration of AI enhances the scalability, consistency, and efficiency of seafloor mapping processes, especially in large-scale and real-time applications (Manjur et al., 2025).

The increasing volume, complexity, and strategic relevance of MBES datasets have intensified the demand for automated and scalable data processing solutions. A typical MBES data processing pipeline involves several stages: sound velocity profile (SVP) correction, motion compensation, beamforming, noise filtering, outlier detection, surface interpolation, and final bathymetric modeling (Calder & Mayer, 2003; Mayer, 2006b). While commercial software offers semi-automated solutions, most workflows still rely heavily on manual editing, threshold tuning, and visual inspection by domain experts. Such methods are not only time-consuming and operator-dependent but also ill-suited for high-tempo operational environments requiring near-real-time data turnaround.

In response to these challenges, recent years have witnessed a significant shift toward leveraging Artificial Intelligence (AI) - particularly Deep Learning (DL) - to automate MBES data interpretation. Unlike traditional algorithms, DL models can learn complex spatial and temporal features from raw acoustic data, enabling robust inference across heterogeneous seafloor types. For example, Wan et al (Wan et al., 2022) proposed a decision-level fusion architecture using deep neural networks for seabed sediment classification, achieving higher classification accuracy than rule-based or statistical methods. Similarly, Manjur et al (Manjur et al., 2025) introduced an attention-guided Convolutional Neural Network (CNN) to detect gas seeps from MBES backscatter data, improving mean average precision (mAP) by 12.4% over baseline models. These developments have shown that DL can not only match but also outperform conventional approaches in accuracy and consistency.

However, despite these promising results, a critical review of prior work reveals several gaps and limitations that motivate the present study.

First, most studies focus on task-specific applications - such as gas seep detection (Manjur et al., 2025), SVP estimation (Wei et al., 2024), or sediment classification (Wan et al., 2022) - without providing a broader comparative evaluation of different DL architectures across generalized MBES tasks. While these works validate the use of CNNs for spatial feature learning, they often ignore the sequential nature of MBES data that arises from the ping-based acquisition process. Temporal dynamics, especially relevant in tracking geological continuity or acoustic propagation patterns, are better modeled by Recurrent Neural Networks (RNNs) such as Gated Recurrent Units (GRUs) and SimpleRNNs. Yet, few studies have systematically assessed how temporal models perform relative to spatial ones (e.g., CNNs) on MBES datasets under real-world conditions.

Second, the lack of unified benchmarking criteria across studies makes it difficult to evaluate model effectiveness holistically. For example, while Zhang et al (Zhang, Zhao, Li, & Cao, 2025) developed a CNN-LSTM model for Doppler Velocity Log (DVL) correction in underwater navigation, the insights are limited to motion estimation rather than bathymetric prediction. Furthermore, existing studies often prioritize accuracy metrics alone, neglecting important considerations such as training efficiency, computational cost, and generalizability across diverse seabed morphologies.

Third, recent progress in multi-sensor fusion, underwater SLAM, and autonomous survey systems increasingly calls for lightweight and robust models that can operate on embedded hardware in AUVs or real-time sonar acquisition platforms. In this context, understanding the trade-offs between different DL architectures is critical for operational deployment, especially in defense or resource-constrained field conditions (Heshmat et al., 2025).

From a model architecture standpoint, 1D Convolutional Neural Networks (Conv1D CNNs) are particularly well-suited for extracting localized features from sonar amplitude profiles, offering fast training and effective generalization for spatial classification tasks. On the other hand, GRUs and SimpleRNNs are more adept at capturing longer-term temporal dependencies, making them ideal for sequences of MBES pings. While each architecture offers unique strengths, the field still lacks a comprehensive comparison under standardized conditions using real MBES datasets - an essential step to guide model selection for practical use cases.

This study presents a comparative evaluation of three prominent deep learning architectures - Conv1D CNN, GRU, and SimpleRNN - with the aim of assessing their suitability and performance in analyzing bathymetric data obtained from MBES measurements. The models are evaluated based on multiple performance criteria, including prediction accuracy, training time, and robustness to seabed

variation. Our goal is not only to benchmark the effectiveness of each model but also to explore their operational implications for automated seabed mapping, underwater navigation, and AI-enabled sonar data pipelines. The findings from this study contribute to the growing body of work on intelligent acoustic sensing and offer practical insights for future deployment in both civilian and defense-oriented marine environments.

The main objective of this study is to evaluate the potential of applying artificial intelligence (AI) models in the analysis of multibeam echo sounder (MBES) bathymetric data. Specifically, the study examines how such models could be used for three major purposes: (i) standardizing bathymetric measurements into regular grids, (ii) predicting sea depths, and (iii) reconciling depth information derived from different data sources. This evaluation provides a basis for understanding the suitability and performance of AI-based approaches in seabed mapping applications.

2. Data and Methodology

2.1. Dataset

The bathymetric data employed in this research were obtained from Viet Nam's People Naval Hydrographic and Oceanographic Department. Depth measurements were collected using a multibeam echo sounder system, allowing for detailed and accurate seabed topography mapping. The surveyed site lies within a coastal region and is geographically delineated by a rectangular area formed by four corner coordinates, as listed in Table 1. The study area extends from approximately 8.80276° to 8.85481° North in latitude and from 115.49339° to 115.54154° East in longitude. This clearly defined spatial extent provides a reliable foundation for subsequent spatial assessments and seabed modeling.

Tab. 1. Geographic coordinates of the four corner points outlining the study area

No.	Name of point	B (degree)	L (degree)
1	North west	8.85481	115.49339
2	North east	8.85481	115.54154
3	South west	8.80276	115.49339
4	South east	8.80276	115.54154

The study area is illustrated in Figure 1, where panel (a) shows the general map of Vietnam and panel (b) indicates the location of the study area.

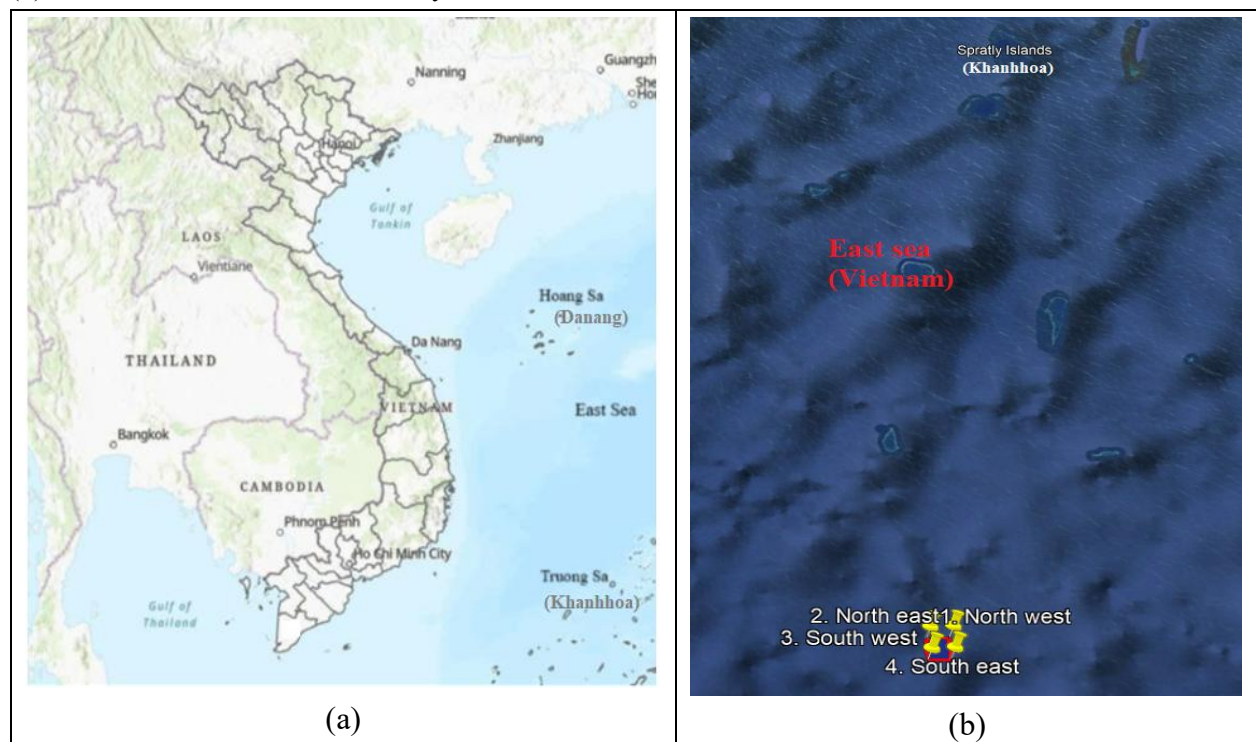


Fig. 1. Location of the study area: (a) general map of Vietnam; (b) the study area

2.2. Deep learning models

2.2.1. CNN(conv1D) model

One-dimensional Convolutional Neural Networks (Conv1D) have been proven effective for analyzing structured sequential data, especially in domains involving time-series or spatial profiles such as sensor signals and geophysical measurements (Kiranyaz, Ince, & Gabbouj, 2015), (Zheng, Liu, Chen, Ge, & Zhao, 2014). In this study, we adopt a Conv1D architecture to extract local spatial features from multibeam echo sounding (MBES) data, which contains depth values recorded along swaths of the seafloor.

Given an input sequence $x = [x_1 \ x_2 \ \dots \ x_T]$, the Conv1D layer applies a kernel $w = [w_0 \ w_1 \ \dots \ w_{k-1}]$ to compute local features via:

$$y_t = \sum_{i=0}^{k-1} w_i x_{t+1} \quad (1)$$

This operation is followed by a non-linear activation function, typically ReLU, defined as:

$$z_t = \text{ReLU}(y_t) = \max(0, y_t) \quad (2)$$

ReLU activation introduces non-linearity while preserving the positive portions of the convolved features, enhancing the model's representational power (Nair & Hinton, 2010). To further abstract spatial characteristics and reduce dimensionality, max pooling is applied:

$$p_t = \max(z_t, z_{t+1}, \dots, z_{t+m-1}) \quad (3)$$

The resulting feature maps are flattened and passed through one or more fully connected layers. For depth prediction tasks, a linear activation function is used in the output layer:

$$\hat{y} = W^T h + b \quad (4)$$

The model is trained using the Mean Squared Error (MSE) loss function:

$$L_{MSE} = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2 \quad (5)$$

This architecture enables efficient learning of spatial patterns embedded in MBES data sequences, offering a robust solution for seafloor topography analysis and related geospatial tasks (LeCun, Bengio, & Hinton, 2015).

2.2.2. GRU model

The Gated Recurrent Unit (GRU) is a recurrent neural network (RNN) variant designed to efficiently model sequential data while reducing computational complexity compared to traditional LSTM models (Cho et al., 2014). GRUs utilize two gates - update gate z_t and reset gate r_t - to control the flow of information through time:

$$z_t = \sigma(W_z x_t + U_z h_{t-1}) \quad (6)$$

$$r_t = \sigma(W_r x_t + U_r h_{t-1}) \quad (7)$$

$$\tilde{h}_t = \tanh(W_h x_t + U_h (r_t \square h_{t-1})) \quad (8)$$

$$h_t = (1 - z_t) \square h_{t-1} + z_t \square \tilde{h}_t \quad (9)$$

These mechanisms enable GRUs to maintain long-term dependencies while avoiding the vanishing gradient problem common in standard RNNs. In this study, GRU is applied to multibeam echo sounding (MBES) data to model spatial dependencies in seafloor topography. Its efficiency and ability to learn temporal structures with fewer parameters make GRU particularly suitable for bathymetric time series analysis (Chung, Gulcehre, Cho, & Bengio, 2014; Siami-Namini, Tavakoli, & Namin, 2018).

2.2.3. SimpleRNN model

The Simple Recurrent Neural Network (SimpleRNN) is a foundational architecture for modeling sequential data, particularly effective in capturing short-term dependencies (Schmidhuber & Hochreiter, 1997). Each unit in a SimpleRNN maintains a hidden state h_t updated at every time step based on the current input x_t and the previous state h_{t-1} :

$$h_t = \tanh(Wx_t + Uh_{t-1} + b) \quad (10)$$

Where W , U , and b are trainable parameters, and $\tanh$ is the non-linear activation function. While SimpleRNN is conceptually simple and computationally efficient, it often struggles with long-term dependency learning due to vanishing gradients during training (Bengio, Simard, & Frasconi, 1994).

In this study, SimpleRNN is applied to multibeam echo sounding (MBES) data to model the sequential patterns in seafloor depth measurements. Despite its limitations compared to gated architectures like GRU and LSTM, SimpleRNN provides a useful baseline for evaluating temporal modeling performance in bathymetric data analysis (Gers, Schmidhuber, & Cummins, 2000).

2.3. Calculation process

In this study, the models were trained to predict depth values on standardized grids derived from MBES data, thereby enabling direct comparison of their predictive accuracy.

The research methodology presented in the paper is illustrated in Figure 2.

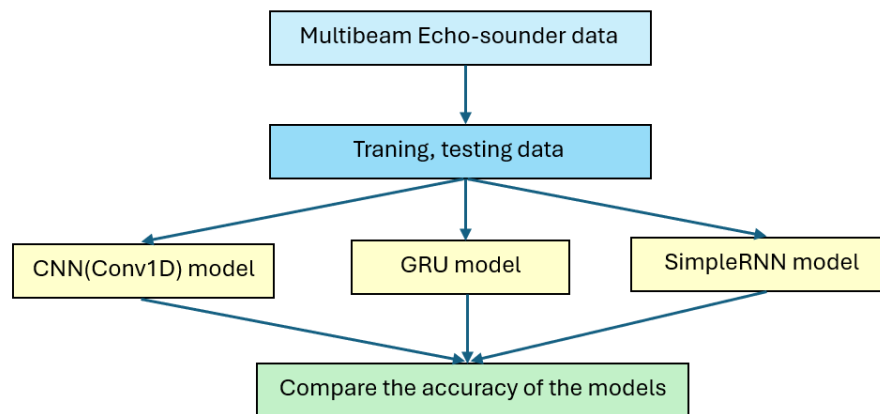


Fig. 2. Research Methodology

The research methodology involves a structured process of data preparation, model implementation, and performance evaluation. Initially, raw multibeam echo sounding data is collected and converted into .CSV format to facilitate preprocessing and integration with deep learning frameworks. The dataset is then divided into two subsets: 70% is allocated for training the models, while the remaining 30% is reserved for testing. This ensures the models can learn from the majority of the data while also being evaluated on unseen data to measure their generalization capabilities.

Three deep learning models - CNN (Conv1D), GRU, and SimpleRNN - are employed to analyze the multibeam echo sounding data. Each model is trained separately on the same dataset, allowing for a fair comparison of their performance under identical conditions. Once the training process is completed, the models are tested and their prediction accuracy is assessed.

To evaluate and compare the models, a set of standard performance metrics is used, including Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), R-squared (R^2), Cohen's Kappa, and F1-score. MSE and RMSE provide insight into the overall magnitude of prediction errors, with RMSE offering a more interpretable scale. MAE reflects the average error magnitude without considering direction, making it easier to understand. R-squared measures how well the model captures the variability in the data. Cohen's Kappa evaluates agreement between the predicted and actual values, adjusting for chance agreement. F1-score, commonly used in classification tasks, balances precision and recall to give a single measure of model performance. These metrics collectively offer a robust foundation for comparing the effectiveness of the three models and drawing meaningful conclusions about their relative accuracy and reliability.

In addition to the structured model training process, it is important to note that data quality plays a

critical role in model performance. While this study focuses primarily on depth values, future iterations could benefit from including other MBES-derived features such as backscatter intensity or uncertainty values. Moreover, evaluating the influence of different environmental conditions (e.g., sea state, tide level, sediment type) on model behavior would provide a more comprehensive understanding of the models' generalization capabilities. Data augmentation and noise injection techniques can also be explored to improve model robustness, particularly in operational settings where raw MBES data are often incomplete or noisy.

3. Result and Discussion

Adam (Adaptive Moment Estimation) is an advanced optimization algorithm designed for training deep learning models. It combines the advantages of two other popular methods: AdaGrad and RMSProp, by computing adaptive learning rates for each parameter based on the estimates of first and second moments of the gradients. This makes Adam particularly effective in dealing with sparse gradients and non-stationary objectives. It requires minimal memory and is well-suited for problems with large data and parameters. Additionally, Adam's ability to adjust learning rates individually contributes to its fast convergence and wide adoption in neural network training (Kingma & Ba, 2014).

The Mean Squared Error (MSE) loss function is commonly used in regression tasks due to its straightforward formulation and sensitivity to large errors. It calculates the average of the squared differences between predicted and actual values, penalizing larger errors more significantly than smaller ones. This characteristic encourages the model to prioritize minimizing substantial deviations, which is crucial for accurate predictions. Despite its simplicity, MSE remains a fundamental tool in evaluating model performance, especially when output precision is critical (Goodfellow, Bengio, Courville, & Bengio, 2016).

The evaluation directly addresses the study's objective of determining how well different AI models can predict sea depths from MBES data on standardized grids.

3.1. GRU model results

The GRU (Gated Recurrent Unit) model was constructed to learn temporal patterns in the bathymetric dataset. Prior to training, the input data were reshaped into three-dimensional arrays to match the expected input format for recurrent models, with shape (samples, timesteps, features). The model architecture consists of a GRU layer with 200 units followed by a fully connected Dense layer with a single output neuron. The model was compiled using the Adam optimizer and mean squared error (MSE) as the loss function. It was trained for 200 epochs with a batch size of 32, using a validation split consisting of 40% of the dataset. This configuration demonstrated strong convergence and high prediction accuracy.

The performance metrics of the model in this case are presented in Table 2.

Tab. 1. Performance statistics of the GRU model

MSE (m)	RMSE (m)	MAE (m)	R ²	F1-score	Kappa
3.56×10^{-3}	0.0596	0.0548	1.0	1.0	1.0

The evaluation results indicate exceptionally high predictive accuracy across all performance measures. The Mean Squared Error (MSE) is extremely low at 3.56×10^{-3} meters, suggesting minimal average deviation between predicted and actual values. Correspondingly, the Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) are also low, at 0.0596 m and 0.0548 m respectively, highlighting a high level of precision in the model's output. Notably, the coefficient of determination (R²) equals 1.0, which reflects a perfect fit of the predictions to the true data. Furthermore, the F1-score and Cohen's Kappa both reaching 1.0 confirm complete agreement between predicted and actual classifications, indicating not only accuracy but also strong model robustness and consistency. Overall, these metrics strongly affirm the model's effectiveness and reliability.

Figures 3, 4, and 5 below illustrate the loss values, predicted values, and actual values on the training and testing datasets using the GRU model.

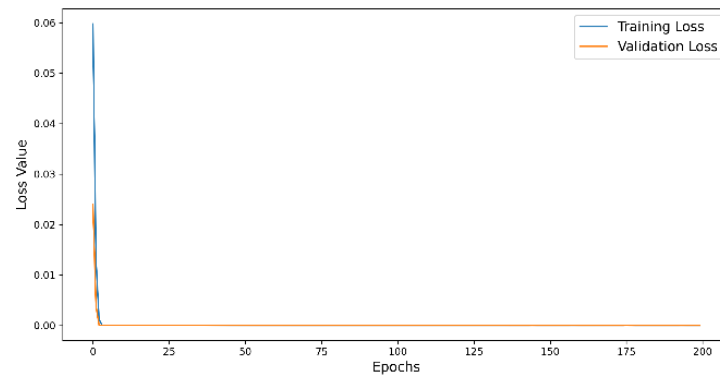


Fig. 3. Loss curve of the GRU model during training

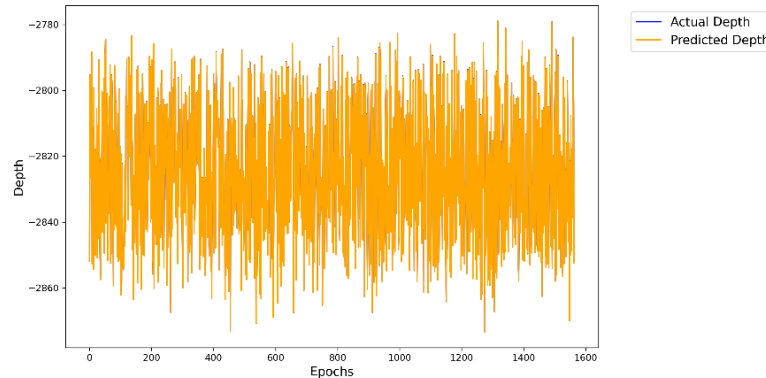


Fig. 4. Comparison of predicted and actual values on the training dataset with GRU model

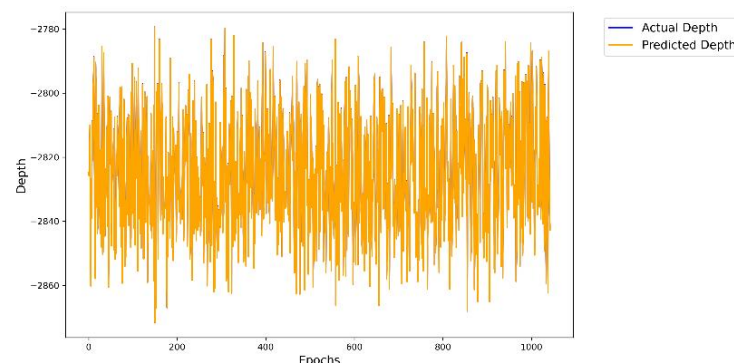


Fig. 5. Comparison of predicted and actual values on the training dataset with GRU model

Although the GRU model consistently outperformed the CNN and SimpleRNN architectures in terms of numerical accuracy, each model has unique strengths that may suit different operational contexts. For instance, the CNN model's superior ability to extract localized spatial patterns makes it advantageous in tasks involving abrupt terrain changes or feature recognition. The SimpleRNN model, while computationally lightweight, can be considered a good starting point for systems with limited processing capacity. These results suggest that model selection should consider not only accuracy but also computational constraints and the nature of the bathymetric features to be analyzed.

3.2. SimpleRNN model results

The SimpleRNN model was built as a baseline recurrent architecture to compare against more complex variants. The input data were normalized and reshaped into three-dimensional arrays with shape (samples, timesteps = 1, features). The model comprises a SimpleRNN layer with 200 hidden units using the tanh activation function, followed by a single Dense output layer. It was compiled using the Adam optimizer and mean squared error as the loss function. Training was performed for 200 epochs with a batch size of 32, and model performance was validated on 40% of the dataset. While simpler in structure, the model provides insight into the benefits and limitations of basic recurrent neural networks for depth prediction tasks.

The performance metrics of the model in this case are presented in Table 3.

Tab. 3. Performance statistics of the SimpleRNN model

MSE (m)	RMSE (m)	MAE (m)	R ²	F1-score	Kappa
5.39×10^{-3}	0.0734	0.0685	1.0	1.0	1.0

The reported metrics indicate that the model demonstrates outstanding predictive performance. With a Mean Squared Error (MSE) of 5.39×10^{-3} meters, the average squared difference between predicted and actual values remains exceptionally low, pointing to high accuracy in numerical predictions. The Root Mean Squared Error (RMSE) of 0.0734 meters and the Mean Absolute Error (MAE) of 0.0685 meters further confirm the model's reliability, as both values suggest only minor discrepancies from actual measurements. The coefficient of determination (R²) reaching 1.0 signifies a perfect match between the model's predictions and the true data, indicating that the model explains 100% of the observed variance. Moreover, the F1-score and Cohen's Kappa, both at 1.0, validate flawless classification capability and strong agreement between predicted and actual outcomes. These consistent and high-performing results across multiple metrics clearly establish the model's robustness and suitability for real-world predictive applications.

When comparing the performance of the SimpleRNN model (Tab. 3) and the GRU model (Tab. 2), it is evident that both models perform exceptionally well, with near-identical results across several metrics. However, the GRU model demonstrates a slight edge in terms of accuracy. The GRU model has a lower Mean Squared Error (MSE) of 3.56×10^{-3} meters, compared to 5.39×10^{-3} meters for the SimpleRNN model, suggesting that the GRU model achieves finer predictions on average. Additionally, the GRU model outperforms the SimpleRNN with a lower Root Mean Squared Error (RMSE) of 0.0596 meters versus 0.0734 meters. Similarly, the Mean Absolute Error (MAE) for the GRU model is also lower at 0.0548 meters, compared to 0.0685 meters for the SimpleRNN. Despite these minor differences, both models achieve perfect R², F1-score, and Kappa values of 1.0, reflecting excellent model fit and classification performance.

Figures 6, 7, and 8 below illustrate the loss values, predicted values, and actual values on the training and testing datasets using the SimpleRNN model.

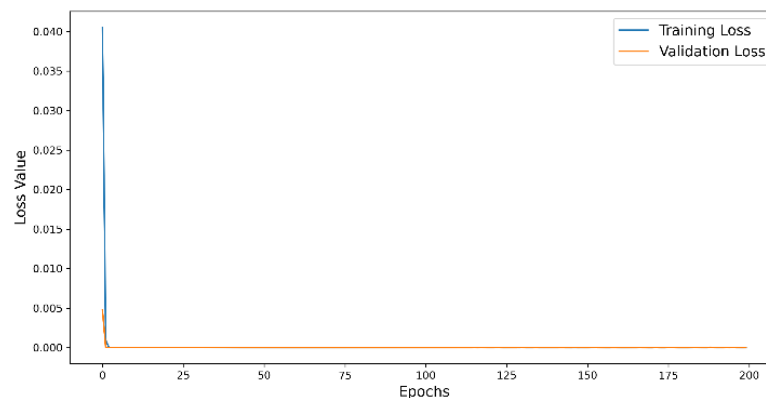


Fig. 6. Loss curve of the SimpleRNN model during training

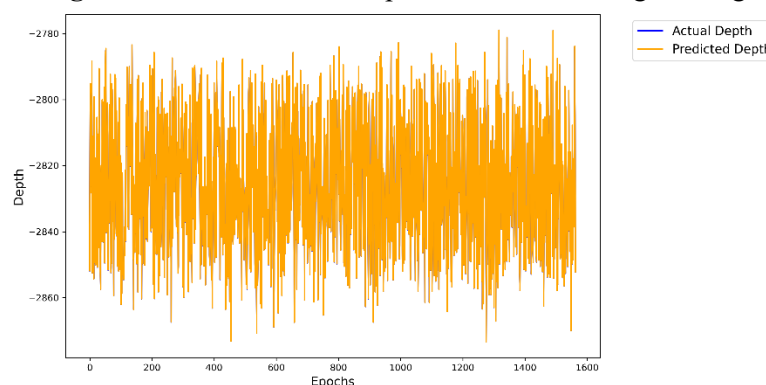


Fig. 7. Comparison of predicted and actual values on the training dataset with SimpleRNN model

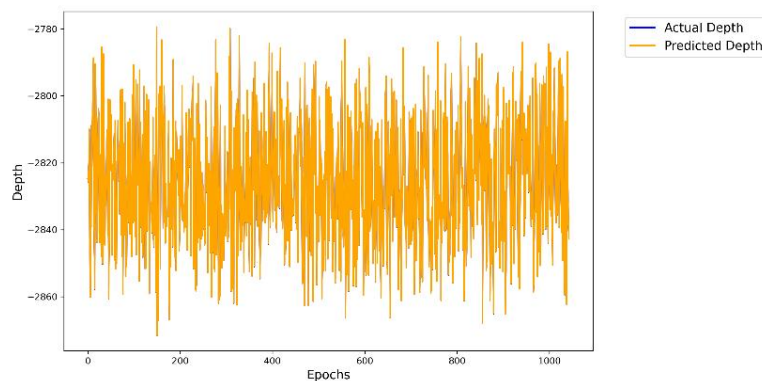


Fig. 8. Comparison of predicted and actual values on the testing dataset with SimpleRNN model

3.3. CNN(Conv1D) model results

The CNN model, based on the Conv1D architecture, was designed to capture local patterns in the input features. The data were reshaped to three-dimensional arrays in the form (samples, features, 1) to match the requirements of the Conv1D layer. The model architecture includes a convolutional layer with 64 filters and a kernel size of 2, followed by a GlobalAveragePooling1D layer to reduce dimensionality. A Dense layer with one output neuron completes the network. The model was compiled with the Adam optimizer and MSE loss. Training was conducted over 200 epochs with a batch size of 32, and validation was performed using 40% of the dataset. The model showed reliable performance in learning from spatially structured input.

The performance metrics of the model in this case are presented in Table 4.

Tab. 3. Performance statistics of the CNN(conv1D) model

MSE (m)	RMSE (m)	MAE (m)	R ²	F1-score	Kappa
9.76×10^{-3}	0.0988	0.0866	1.0	1.0	1.0

The results presented reflect a highly accurate model performance. The Mean Squared Error (MSE) of 9.76×10^{-3} meters indicates that prediction errors are minimal and well-controlled. The Root Mean Squared Error (RMSE) value of 0.0988 m, alongside a Mean Absolute Error (MAE) of 0.0866 m, confirms that the model produces estimates that are very close to the true values on average. The R² value of 1.0 demonstrates that the model accounts for the entire variance in the dataset. Additionally, perfect scores in both F1 and Cohen's Kappa further underscore the model's consistency and classification precision.

When comparing the performance of the CNN (conv1D) model (Table 4) to the GRU model (Table 2), it is clear that the GRU model outperforms the CNN model in terms of accuracy. The GRU model exhibits a significantly lower Mean Squared Error (MSE) of 3.56×10^{-3} meters, compared to the CNN model's MSE of 9.76×10^{-3} meters, indicating more precise predictions on average. Additionally, the GRU model achieves a lower Root Mean Squared Error (RMSE) of 0.0596 meters, while the CNN model has an RMSE of 0.0988 meters. Similarly, the GRU model has a lower Mean Absolute Error (MAE) of 0.0548 meters versus the CNN model's MAE of 0.0866 meters. Both models, however, achieve perfect R², F1-score, and Kappa values of 1.0, confirming exceptional classification performance. Despite the CNN model's slightly higher error values, it still demonstrates excellent predictive capability, though the GRU model remains the more accurate choice.

The CNN (conv1D) model (Tab. 4) and the SimpleRNN model (Tab. 3) display similar performance characteristics, but the CNN model shows slightly higher error rates. The SimpleRNN model achieves a lower Mean Squared Error (MSE) of 5.39×10^{-3} meters, compared to the CNN model's 9.76×10^{-3} meters. This suggests that the SimpleRNN model is more accurate in predicting values on average. Additionally, the SimpleRNN model outperforms the CNN model in terms of Root Mean Squared Error (RMSE), with an RMSE of 0.0734 meters, compared to 0.0988 meters for the CNN model. The Mean Absolute Error (MAE) is also lower for the SimpleRNN at 0.0685 meters, in contrast to the CNN model's 0.0866 meters. Despite these differences, both models achieve perfect R², F1-score, and Kappa values of 1.0, reflecting their strong classification and model fit performance. However, the SimpleRNN model demonstrates better overall accuracy compared to the CNN(Conv1D) model.

Figures 9, 10, and 11 below illustrate the loss values, predicted values, and actual values on the

training and testing datasets using the CNN(Conv1D) model.

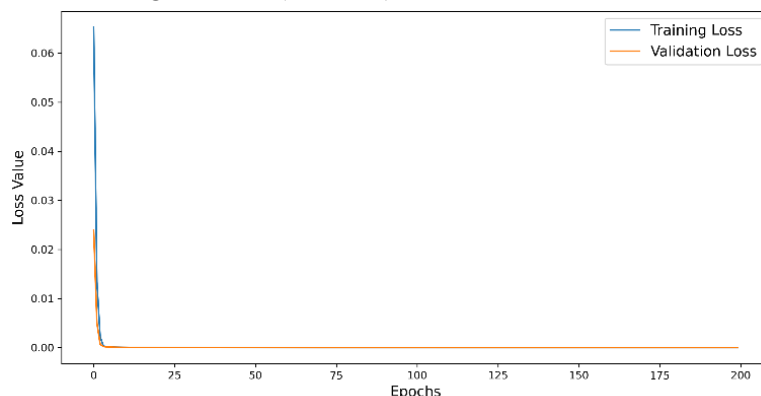


Fig. 9. Loss curve of the CNN(Conv1D) model during training

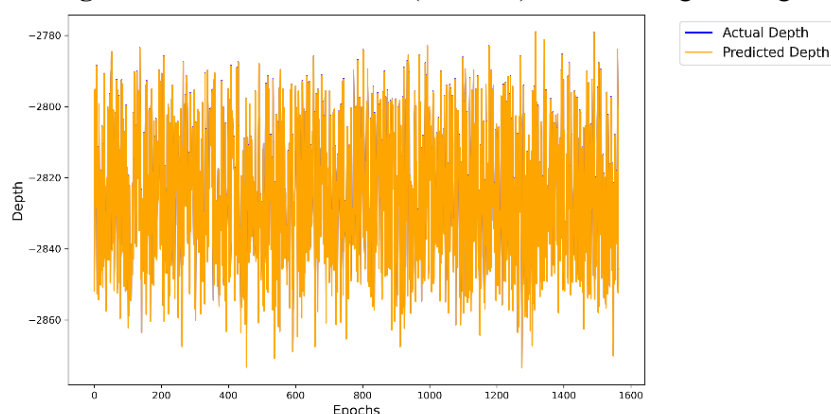


Fig. 10. Comparison of predicted and actual values on the training dataset with CNN(conv1D) model

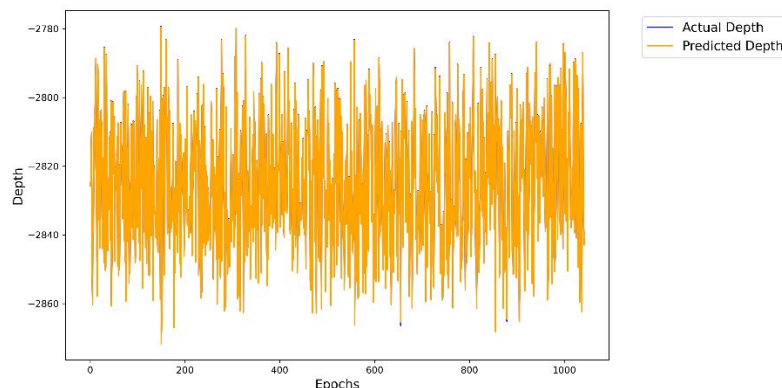


Fig. 11. Comparison of predicted and actual values on the testing dataset with CNN(Conv1D) model

Despite these differences, both models achieve perfect R^2 , F1-score, and Kappa values of 1.0, reflecting their strong classification and model fit performance. However, the SimpleRNN model demonstrates better overall accuracy compared to the CNN(Conv1D) model.

Taken together, the results confirm that all three AI models are capable of predicting sea depths from MBES data on standardized grids, with clear differences in accuracy and robustness depending on the chosen architecture.

4. Conclusion

This study presents a comprehensive analysis of three deep learning models - CNN (Conv1D), GRU, and SimpleRNN - applied to bathymetric data obtained from the People Naval Hydrographic and Oceanographic Department of Vietnam. The goal was to model seabed topography using multibeam echo sounding (MBES) data, leveraging each model's ability to capture spatial and temporal patterns within the dataset. The findings demonstrate that AI models can effectively standardize MBES bathymetric data into regular grids, predict seabed depths, and reconcile information across sources.

The performance results reveal that all three models achieved perfect scores in key classification

metrics, including R^2 , F1-score, and Cohen's Kappa, underscoring their robust predictive capabilities. However, when comparing the models' regression performance, some interesting distinctions arise. The GRU model demonstrated the best performance overall, with the lowest MSE (3.56×10^{-3} meters), RMSE (0.0596 m), and MAE (0.0548 m), indicating the highest accuracy in depth prediction. Its superior ability to model temporal dependencies with fewer parameters and its efficient learning mechanism likely contributed to its top performance. Moreover, the GRU model achieved exceptional consistency, with perfect agreement between predicted and actual outcomes across multiple evaluation metrics.

In contrast, the SimpleRNN model, although slightly less accurate than the GRU model, still exhibited impressive results. It achieved a very low MSE of 5.39×10^{-3} meters, RMSE of 0.0734 meters, and MAE of 0.0685 meters. While the SimpleRNN was less adept at capturing long-term dependencies compared to the GRU, its performance was still highly reliable for this task. These results highlight the SimpleRNN's potential as a baseline model for sequential data analysis, particularly in cases where computational efficiency is critical.

The CNN (Conv1D) model, while still performing well, lagged behind the GRU and SimpleRNN models in terms of prediction accuracy. With an MSE of 9.76×10^{-3} meters, RMSE of 0.0988 meters, and MAE of 0.0866 meters, the CNN model showed slightly higher errors. However, it maintained perfect classification performance, achieving an R^2 , F1-score, and Kappa value of 1.0. The CNN model's ability to effectively learn local spatial features in the data makes it a strong contender for tasks requiring the extraction of intricate spatial patterns, even though it fell short in comparison to the GRU and SimpleRNN models in this specific regression task.

Overall, the results from this research underscore the power and versatility of deep learning models in geospatial analysis. The GRU model emerges as the most accurate for predicting bathymetric depths, while the SimpleRNN and CNN models offer useful alternatives depending on the specific requirements of the task. Future research could explore hybrid models or additional techniques to further improve prediction accuracy and model robustness.

Furthermore, incorporating uncertainty quantification techniques such as Monte Carlo dropout or confidence interval estimation would enhance the interpretability and trustworthiness of model predictions, especially in safety-critical applications like nautical charting or subsea infrastructure planning. It is also recommended to experiment with hybrid models, such as CNN-GRU or attention-based architectures, which could harness both spatial and temporal features more effectively. Lastly, expanding the study to include cross-validation with datasets from different survey areas would help assess model scalability and applicability across diverse marine environments.

5. Recommendations

Based on the results of this study, several suggestions can be made for future work in bathymetric data analysis. Given the GRU model's high performance, it is advisable to focus on this model for tasks requiring precise predictions, particularly where long-term dependencies in sequential data play a significant role. Although the SimpleRNN model demonstrated solid performance, further research could explore its enhancement by combining it with more advanced models, such as LSTMs or GRUs, to better capture intricate temporal dependencies.

While the CNN (Conv1D) model showed slightly higher error rates, it remains a useful tool for extracting detailed spatial features. Future work could investigate hybrid approaches that leverage the strengths of both convolutional and recurrent models, potentially enhancing overall accuracy and model robustness.

Acknowledgment

The authors would like to express their sincere gratitude to the Naval Hydrographic and Oceanographic Department of Vietnam for kindly providing the bathymetric data used in this study. Their support was essential for the completion of this research.

Funding

This work was financially supported by the Ministry of Natural Research and Environment in Vietnam under grant number TNMT.DL.2025.06.03.

References

- Bengio, Y.; Simard, P.; Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5(2), 157-166.
- Calder, B.R.; Mayer, L.A. (2003). Automatic processing of high-rate, high-density multibeam echosounder data. *Geochemistry, Geophysics, Geosystems*, 4(6).
- Clark, R.M. (2020). *Geospatial intelligence: Origins and evolution*: Georgetown University Press.
- Cho, K.; Van Merriënboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv*.
- Chung, J.; Gulcehre, C.; Cho, K.; Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv*.
- Gers, F.A.; Schmidhuber, J.; Cummins, F. (2000). Learning to forget: Continual prediction with LSTM. *Neural computation*, 12(10), 2451-2471.
- Goodfellow, I.; Bengio, Y.; Courville, A.; Bengio, Y. (2016). *Deep learning* (Vol. 1): MIT press Cambridge.
- Heshmat, M.; Saad Saoud, L.; Abujabal, M.; Sultan, A.; Elmezain, M.; Seneviratne, L.; Hussain, I. (2025). Underwater SLAM Meets Deep Learning: Challenges, Multi-Sensor Integration, and Future Directions. *Sensors*, 25(11), 3258.
- Kingma, D.P.; Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint*. doi:<https://doi.org/10.48550/arXiv.1412.6980>
- Kiranyaz, S.; Ince, T.; Gabbouj, M. (2015). Real-time patient-specific ECG classification by 1-D convolutional neural networks. *IEEE transactions on biomedical engineering*, 63(3), 664-675.
- LeCun, Y.; Bengio, Y.; Hinton, G. (2015). Deep learning. *Nature*, 521(7553), 436-444. doi:10.1038/nature14539
- Lurton, X. (2002). *An introduction to underwater acoustics: principles and applications*: Springer Science & Business Media.
- Manjur, S.M.; Senyurek, V.; Kalski, R.; Gupta, S.; Skarke, A.; Gurbuz, A.C. (2025). Automated Detection of Seafloor Gas Seeps in Multibeam Echosounder Data With an Attention-Guided Convolutional Neural Network. *IEEE Journal of Selected Topics in Applied Earth Observations Remote Sensing*.
- Mayer, L.A. (2006a). Frontiers in seafloor mapping and visualization. *Marine Geophysical Researches*, 27, 7-17.
- Mayer, L.A. (2006b). Frontiers in seafloor mapping and visualization. *Marine Geophysical Researches*, 27(1), 7-17.
- Nair, V.; Hinton, G.E. (2010). Rectified linear units improve restricted boltzmann machines. Paper presented at the Proceedings of the 27th international conference on machine learning (ICML-10).
- Schmidhuber, J.; Hochreiter, S. (1997). Long short-term memory. *Neural Computer*, 9(8), 1735-1780.
- Siami-Namini, S.; Tavakoli, N.; Namin, A.S. (2018). A comparison of ARIMA and LSTM in forecasting time series. Paper presented at the 2018 17th IEEE international conference on machine learning and applications (ICMLA).
- Wan, J.; Qin, Z.; Cui, X.; Yang, F.; Yasir, M.; Ma, B.; Liu, X. (2022). MBES seabed sediment classification based on a decision fusion method using deep learning model. *Remote Sensing*, 14(15), 3708.
- Wei, Z.; Shaohua, J.; Gang, B.; Yang, C.; Chengyang, P.; Haixing, X. (2024). A method for sound speed profile prediction based on CNN-BiLSTM-Attention network. *Journal of Marine Science Engineering*, 12(3), 414.
- Zhang, F.; Zhao, S.; Li, L.; Cao, C. (2025). Underwater DVL Optimization Network (UDON): A Learning-Based DVL Velocity Optimizing Method for Underwater Navigation. *Drones*, 9(1), 56.
- Zheng, Y.; Liu, Q.; Chen, E.; Ge, Y.; Zhao, J.L. (2014). Time series classification using multi-channels deep convolutional neural networks. Paper presented at the International conference on web-age information management.