Journal of
Marine Science and Engineering

MDPI

Article

The Mamba Model: A Novel Approach for Predicting Ship Trajectories

Yongfeng Suo, Zhengnian Ding and Tao Zhang *

Navigation College, Jimei University, Xiamen 361021, China; yfsuo@jmu.edu.cn (Y.S.); dingzhengnian@jmu.edu.cn (Z.D.)
* Correspondence: zhangtony@jmu.edu.cn

Abstract: To address the complexity of ship trajectory prediction, this study explored the efficacy of the Mamba model, a relatively new deep-learning framework. In order to evaluate the performance of the Mamba model relative to traditional models, which often struggle to cope with the dynamic and nonlinear nature of maritime navigation data, we analyzed a dataset consisting of intricate ship trajectory data. The prediction accuracy and inference speed of the model were evaluated using metrics such as the mean absolute error (MAE) and root mean square error (RMSE). The Mamba model not only excelled in terms of the computational efficiency, with inference times of 0.1759 s per batch—approximately 7.84 times faster than the widely used Transformer model—it also processed 3.9052 samples per second, which is higher than the Transformer model's 0.7246 samples per second. Additionally, it demonstrated high prediction accuracy and the lowest loss among the evaluated models. The Mamba model provides a new tool for ship trajectory prediction, which represents an advancement in addressing the challenges of maritime trajectory analysis when compared to existing deep-learning methods.

Keywords: ship trajectory prediction; AIS; mamba; transformer

check for updates

Citation: Suo, Y.; Ding, Z.; Zhang, T. The Mamba Model: A Novel Approach for Predicting Ship Trajectories. *J. Mar. Sci. Eng.* **2024**, *12*, 1321. https://doi.org/10.3390/jmse12081321

Academic Editor: Sergei Chernyi

Received: 15 July 2024
Revised: 2 August 2024
Accepted: 4 August 2024
Published: 5 August 2024

Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Maritime transportation plays a significant role in global trade and logistics, supporting international economic development and contributing to market stability. As globalization progresses, the congestion in shipping lanes and the complexity of the navigational environment increase, raising the risk of maritime accidents, notably ship collisions. This situation highlights the importance of developing advanced ship trajectory prediction techniques. Effective trajectory prediction is crucial for preventing collisions and optimizing routes, as well as for improving the overall efficiency and safety of maritime navigation. Traditional prediction models are gradually improving in managing complex data but often do not meet the required levels of accuracy and computational efficiency. Additionally, these models sometimes struggle to meet the demands of modern navigation systems, which need adaptability to dynamic conditions and the capability to process complex sequence data over extended periods. Therefore, it is vital to develop sophisticated prediction techniques that can adequately address these challenges. Advancements in this area are essential for enhancing navigational performance, improving safety standards, and supporting modernization within the maritime transportation system. This effort not only tackles the immediate challenges of navigation but also aids in achieving the broader objective of ensuring sustainable and safe maritime operations in an increasingly interconnected world.

1.1. Related Work

Currently, research on ship trajectory prediction methods by scholars both domestically and internationally can be broadly divided into two main categories: traditional

method-based research and deep-learning-based research. In traditional method-based research, the filtering algorithm, Markov chain algorithm, Gaussian mixture model, and least squares method are the predominant technical approaches. Filtering algorithms, widely applied in time series prediction, primarily include the Kalman filter algorithm, extended Kalman filter algorithm, and traceless Kalman filter algorithm. For instance, Tong et al. [1] proposed a trajectory prediction algorithm utilizing the Kalman filter, which initially segments AIS trajectories into straight-line and sharply turning trajectories. It then filters the node position information within the ship's trajectory sequence, ultimately enabling trajectory prediction by correlating current with historical trajectory data. In research focused on narrow maritime areas, Mazzarella et al. [2] developed a Bayesian ship trajectory prediction model based on the particle filter, demonstrating enhanced performance in strait waters and inland rivers. Fossen et al. [3] employed the extended Kalman filter algorithm in conjunction with the exogenous Kalman filter to address the nonlinear motion of ships, based on which a visualized ship trajectory prediction system was designed. Conversely, Sun et al. [4] predicted future ship trajectories by matching historical data stored in a spatially encoded database, analyzing and correlating trajectories for future state prediction. Rong et al. [5] considered both horizontal and vertical ship motions, utilizing a Gaussian process for horizontal uncertainty and acceleration for vertical motion estimation, combining these to predict ship movement effectively. Jiang et al. [6] explored the segmented polynomial characteristics of ship trajectories, employing polynomial fitting for state and observation equations and iteratively updating the error covariance matrix for effective trajectory prediction. Finally, Qiao et al. [7] advanced ship trajectory prediction by integrating sensor data and prior knowledge, enhancing ship state monitoring through visualization techniques. Although these traditional methods offer several tools, their reliance on stringent modeling assumptions often restricts their applicability in complex navigational environments.

On the other hand, deep-learning-based trajectory prediction methods predominantly utilize large datasets to train models such as recurrent neural networks, long short-term memory networks, and gated recurrent neural networks. For instance, Mao [8] developed a ship trajectory prediction method for marine environments using an Extreme Learning Machine (ELM), which obviates the need for determining neural network parameters through training. This method has been shown to offer rapid training speeds and high generalization capabilities. Forti et al. [9] and Capobianco et al. [10] employed a recurrent neural network-based encoder–decoder model for sequence-to-sequence trajectory prediction, encoding historical trajectories into fixed vectors for improved prediction accuracy on less complex datasets. Murray et al. [11] leveraged a bilinear self-encoder to learn the spatial and temporal characteristics of historical ship trajectories, employing a multilayer perceptron model to generate predictions, with experiments validating its efficacy across various timescales. Qian et al. [12] introduced a novel ship trajectory prediction method utilizing Genetic Algorithms (GAs) to optimize the hyperparameters of long short-term memory (LSTM) networks, demonstrating high prediction accuracy in inland waters. Suo et al. [13] generated vector representations of historical ship trajectories and temporal information during the encoding phase, applying an attention mechanism to dynamically adjust the vector weights, with the decoder progressively generating predicted trajectories, evidencing superior prediction performance. Gao et al. [14] combined the strengths of TPNet and LSTM to propose an AIS data-based MP-STM trajectory prediction method that is not only straightforward to implement but also highly accurate. Nguyen et al. [15] introduced a scalable sequence-to-sequence ship trajectory prediction model using LSTM, employing a spatial grid to depict ship motion trends, with an encoder and decoder to anticipate future motions toward the destination. Luo et al. [16] applied a reinforcement learning neural network to vectorize ship trajectory data, achieving notable results. Zhang et al. [17] proposed a multi-scale convolutional neural network (MSCNN)-based method for HF radar-based ship trajectory prediction, integrating autoregression, GRU, and an attention mechanism to accurately predict trajectories in cluttered environments. Hoseinzade

et al. [18] utilized a CNN-based framework to extract features for future market predictions, showing good performance across various market datasets. Lastly, Huang et al. [19] combined CNN's feature extraction and parallel computing capabilities to develop a dual self-attentive network (DSANet) for efficient multivariate time series prediction, integrating traditional AR to enhance the model robustness, demonstrating its efficiency in multivariate time series prediction. Tang et al. [20] proposed a long short-term memory network-based model for ship trajectory prediction, utilizing the motion state of the trajectory to learn its nonlinear features, iteratively predicting the next twenty minutes of trajectory points, achieving superior performance over a fully connected neural network.

With the widespread application of long short-term memory networks (LSTMs), various LSTM model variants have increasingly been employed in the field of trajectory prediction. For instance, Xiao et al. [21] introduced a UN-LSTM model based on behavior recognition, which is capable of effectively predicting the horizontal and vertical speeds of vehicles three seconds into the future by processing recognized vehicle behaviors. Conversely, Jaseena et al. [22] developed an EWT-LSTM model that utilizes bidirectional long short-term memory networks to predict low-frequency and high-frequency subsequences separately, combining these predictions to derive the final results. Similarly, Xue et al. [23] proposed the SS-LSTM model, which incorporates three distinct LSTM networks to capture human, social, and scene-scale information, significantly enhancing the prediction accuracy of pedestrian trajectories. While LSTM models excel in handling sequential data, they face challenges with long-term dependencies and parallel computation. To overcome these limitations, the Transformer model has been explored for time series prediction. The Transformer exhibits remarkable potential in various sequence prediction tasks due to its ability to capture long-distance dependencies more effectively through the self-attention mechanism and its superior parallel-processing capabilities. For example, Giuliari et al. [24] developed a Transformer-based model specifically for spatial-temporal prediction tasks, demonstrating enhanced performance over traditional recurrent neural networks. This study underscores the Transformer's effectiveness in managing complex trajectory data. The Trajectory Transformer model, introduced by Yoon et al. [25], focuses on pedestrian trajectory prediction and illustrates how complex social behaviors and diverse interaction patterns can be effectively modeled through its attentional mechanism, further validating the Transformer's suitability for such applications. Liang et al. [26], meanwhile, explored the use of the Transformer in automobile trajectory prediction by designing a specialized attention mechanism to address vehicle–environment interactions, enhancing both the accuracy and utility of the predictions. Although deep-learning-based trajectory prediction methods have advanced significantly in accuracy and efficiency, they often require substantial computational resources and may encounter performance limitations when processing extremely long time series data. Moreover, these models typically depend on extensive training data for optimization, which can restrict their applicability and flexibility in scenarios where the data availability is limited.

1.2. Contributions

To address the limitations of current deep-learning-based trajectory prediction methods, this paper introduces a trajectory prediction approach utilizing the Mamba model [27]. The Mamba model, an evolution of structured state-space models, is specifically tailored for handling long sequential data and effectively employs deep-learning techniques to address complex trajectory prediction challenges. This study not only demonstrates the Mamba model's advantages in terms of the prediction accuracy, processing speed, and resource efficiency by comparing it with existing prediction models but also explores the model's applicability under specific navigational conditions and scenarios. This research contributes to the modernization of nautical safety management and provides novel solutions for other long series prediction tasks, showcasing the potential of ship management systems that leverage advanced algorithms. The main contributions of this paper are as follows:

1. Utilizing the Mamba model for ship trajectory prediction, this study assesses its prediction accuracy, processing speed, and resource efficiency compared to traditional time series prediction models such as the LSTM, GRU, and Transformer.
2. This paper introduces the Mamba model, which overcomes the limitations of traditional models in handling long time dependencies and resource consumption through a selective state-space model, an optimized parallel algorithm, and a simplified deep-learning architecture, offering an efficient and accurate scheme for ship trajectory prediction.

2. Mode Construction

In this paper, we explore the ship trajectory prediction technique based on the Mamba model, with the study structured into three core sections, as depicted in Figure 1. Initially, the paper focuses on the acquisition and preprocessing of Automatic Identification System (AIS) data, detailing the extraction of key features such as the longitude, latitude, surface speed (SOG), and ground heading (COG) from the AIS data. Through various data preprocessing techniques, including outlier processing and data normalization, we ensure the input data have consistent feature representations and appropriate data distributions, thus laying a solid foundation for efficient model training. Subsequently, we delve into the design specifics of the Mamba model for trajectory prediction, emphasizing how to effectively utilize maritime features for precise trajectory modeling. The model optimizes the time series analysis process through its unique structure, enhancing the learning efficiency and prediction accuracy on complex datasets. Finally, by comparing the performance of the Mamba model with other mainstream neural network models for trajectory prediction, this paper highlights the effectiveness of the Mamba model in ship trajectory prediction tasks. The experimental results demonstrate that the Mamba model not only accelerates model training but also provides high accuracy in trajectory prediction.

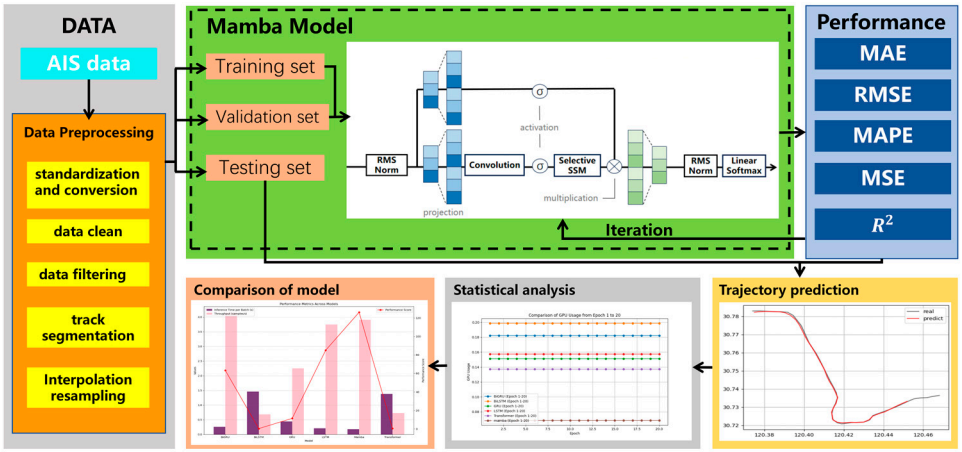


Figure 1. Trajectory prediction based on the Mamba model.

2.1. AIS Data Processing

1. Data Standardization and Conversion: During data transmission, AIS data are often encoded or scaled to reduce the volume and increase the transmission efficiency. For analysis and modeling, these data need to be converted and standardized. For instance, longitude and latitude data are scaled up to a fixed multiple (e.g., 1,000,000), and their true values are recovered by dividing by that multiple during processing.
2. Data Cleansing: Data cleansing is crucial for ensuring data quality and involves processing inaccurate, incomplete, or invalid data points. This includes deleting track point records with missing key values or duplicates, setting reasonable thresholds for detecting and processing outliers. Specifically, we limit the heading values strictly to between 0 and 360 degrees to conform with standard navigational practices, where 0 degrees represents true north and 360 degrees completes the circle, thus ensuring

all the heading values are within this circular range. Additionally, the speeds are ensured to not exceed a reasonable range, relevant to the type of vessel and prevailing conditions. Data can also be smoothed using methods such as moving averages or a Kalman filter to reduce the influence of data noise.

3. Data Filtering: Relevant feature information is extracted from the AIS data based on research needs, and irrelevant data are filtered out. The main features include the timestamp, longitude, latitude, speed, and heading. Specific steps involve retaining only specific types of ship data, such as cargo ships and oil tankers; filtering out data points with irrelevant statuses, such as mooring and anchoring; and filtering out data beyond the research area to ensure only AIS data within the research area are retained.
4. Trajectory Segmentation: To prevent trajectory drift due to factors like round-trip voyages, trajectory segmentation is necessary. The time interval between consecutive AIS signals from the same ship is monitored to determine if segmentation is needed. If the interval exceeds a set threshold (e.g., six minutes), the trajectory is considered to be interrupted, and the segment prior to the interruption is treated as an independent trajectory. Segments containing fewer trajectory points are deleted to ensure data continuity and accuracy.
5. Interpolation and Resampling: Since the AIS data points in each track segment are not evenly spaced, interpolation and resampling are required to ensure data consistency and availability. Common methods include Lagrangian interpolation and cubic spline interpolation, which avoid the issues of high-order polynomial interpolation. The trajectory points are divided into segments, each associated with an interpolation function, to ensure smoothness and continuity. The interpolated trajectory points are then resampled at uniform time intervals (e.g., every three minutes) to enhance the data consistency and improve the model's prediction accuracy.

Through these data-processing steps, the quality of AIS data can be effectively improved, providing a more accurate and reliable database for the subsequent trajectory prediction model. This not only enhances the model's prediction accuracy but also minimizes the impact of anomalous data on model training, thereby improving the overall prediction effect.

2.2. Mamba Model

The Mamba model, as depicted in Figure 2, is a time series forecasting tool based on a state-space model (SSM). It effectively integrates modern deep-learning techniques with classical control theory. At the heart of the Mamba model is a Selective State-Space Model (S6), which enables the model to dynamically adapt its behavior according to different data features and sequence characteristics during sequence processing. This selective mechanism is implemented through parameterization, where the model's state transition matrix and output matrix are dynamically adjusted based on the input data. This flexibility allows the Mamba model to tailor its response to the specific dynamics of the input sequences, enhancing its forecasting accuracy and adaptability.

The Mamba model achieves efficient processing of long sequence data through several key techniques.

2.2.1. Selective State-Space Models

This technique aims to enhance the context-based inference capabilities of the state-space model (SSM) by making its parameters a function of the input. Here, the SSM specifically refers to sequence transitions used in the Structured State-Space Sequence Model (S4), which can be integrated into deep neural networks. This integration allows the model to dynamically adjust its parameters based on the incoming data, significantly improving its ability to model and predict complex sequence behaviors effectively.

The original definition of a state-space model (SSM) is a Linear Time-Invariant (LTI) system that projects input stimulation $x(t) \in \mathbb{R}^L$ to the output response $y(t) \in \mathbb{R}^L$ through

a hidden state $h(t) \in \mathbb{C}^N$. For continuous inputs, the system is formulated by a group of linear ordinary differential equations as follows:

$$h'(t) = Ah(t) + Bx(t) \tag{1}$$

$$y(t) = Ch(t) + Dx(t) \tag{2}$$

where $A \in \mathbb{C}^{N\times N}$, $B \in \mathbb{C}^N$, $C \in \mathbb{C}^N$, and $D \in \mathbb{C}^1$ denote the weight parameters.

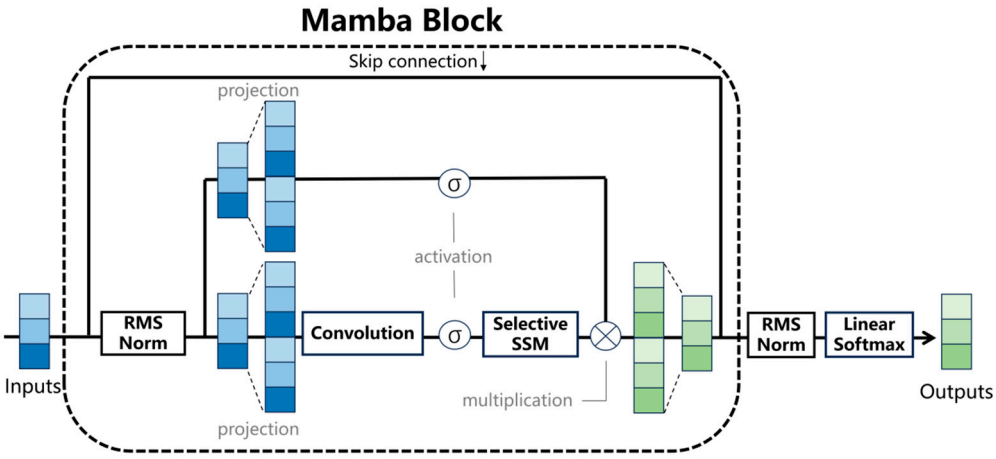


Figure 2. Mamba block.

By discretizing the group of ordinary differential equations, continuous-time state-space models (SSMs) can be adapted to process discrete inputs, such as language, speech, and image pixels. This discretization allows the continuous-time model to effectively manage the discrete nature of these types of data. The process involves converting the continuous system equations into a format suitable for discrete operations, thereby enabling the model to handle inputs that are naturally segmented or episodic. As a result of this discretization, the model can be solved using an analytic solution, which facilitates the efficient processing of input data by directly applying mathematical formulas to derive the state transitions and outputs based on the given discrete inputs.

$$h_t = e^{A\Delta t}[h(t-\Delta t) + \int_{t-\Delta t}^{t} e^{A(t-\tau)}B(\tau)x(\tau)d\tau] \tag{3}$$

The continuous-time state-space model (SSM) is discretized using a Zero-Order Hold (ZOH) approach, resulting in an effective transition to a discrete model. This method holds the input constant over the sampling interval, which simplifies the translation of continuous systems into discrete equivalents. By maintaining the last known value until the next input is sampled, the Zero-Order Hold ensures that the system can be represented in discrete time without losing critical information during the conversion process. This leads to the following discrete model formulation:

$$h_t = \overline{A}h_{t-1} + \overline{B}x_t \tag{4}$$

$$y_t = Ch_t + Dx_t \tag{5}$$

In the discretized state-space model (SSM), the parameters are transformed for handling discrete inputs, where $\overline{A} = \exp(\Delta A)$, $\overline{B} = \Delta A^{-1}(\exp(\Delta A) - I)\cdot\Delta B$ are the results of these transformations. Here, Δ is a learnable parameter that estimates the discrete intervals, crucial for the precise timing of the input samples. This structured SSM (S4) utilizes a

convolution process for efficient computation, distinctly different from basic recurrent inference methods. The convolution kernel is given by:

$$\overline{K} = \left(C\overline{B}, C\overline{AB}, \ldots, C\overline{A}^{M-1}\overline{B}\right) \tag{6}$$

$$y = x * \overline{K} \tag{7}$$

The prediction is computed as $y = x * \overline{K}$, where x is the input and $\overline{K}$ serves as the convolution kernel. This convolutional approach not only accelerates the computations but also enhances the model's ability to capture and leverage the inherent spatial-temporal relationships in the data, leading to more accurate and efficient predictions.

The earlier linear time-invariant state-space model could not perform selective tasks based on the input content due to its lack of content awareness. This limitation caused the S4 model to equally focus on all tokens. However, in practice, the importance of tokens varies and their significance changes dynamically during the training process. To address this, it is more effective to allocate more attention to crucial content and dynamically adjust importance levels to align with complex input content. To achieve this, the Mamba framework integrates a selectivity mechanism into the state-space model, resulting in the development of Selective State-Space Models (S6). Specifically designed to handle an input sequence with batch size B, length L, and D channels, Mamba applies the SSM independently to each channel. In Mamba, the matrices B, C, and Δ in S4 are functions of the inputs, allowing adaptive adjustment of the model's behavior based on the inputs. The discretization process incorporating the selection mechanism is outlined as follows:

$$\overline{B} = S_B(x) \tag{8}$$

$$\overline{C} = S_C(x) \tag{9}$$

$$\Delta = \tau_A(Parameter + S_A(x)) \tag{10}$$

where $B \in \mathbb{R}^{B\times L\times N}$, $C \in \mathbb{R}^{B\times L\times N}$ and $\Delta \in \mathbb{R}^{B\times L\times N}$. The functions $S_B(x)$ and $S_C(x)$ linearly project the input x into an N-dimensional space, while $S_A(x)$ projects the hidden state dimension D linearly into the required dimension, linking to the RNN gating mechanism. These calculations transform the parameters Δ, B, C into input-dependent functions with length L, converting the time-invariant model into a time-varying model, thereby achieving selectivity.

The dimensions of Δ have been modified from D to (B, L, D), indicating that each token in a batch (totaling $B \times L$) has a distinct Δ for the input data dependency and enhanced control functions. A larger step size of Δ shifts the model's focus more toward the inputs rather than the stored state. Conversely, a smaller step size results in the model de-emphasizing specific inputs and focusing more on the stored state. Parameters B and C have become dependent on the input data as functions within the S4 framework, which facilitates more nuanced control over whether the input x influences the state h or if the state h impacts the output y. Although parameter A does not inherently depend on the data, the discretization process of the state-space model (SSM) allows for A to be influenced by the input via Δ's data dependency. Moreover, given that parameter A has dimension N, it serves varied roles across each SSM dimension, enabling a precise generalization of all the previous content rather than simple compression.

2.2.2. Hardware-Aware State Expansion

The selectivity mechanism in the Mamba model addresses the constraints of previous linear time-invariant state-space models but introduces computational complexity due to its time-varying nature. In Mamba, the relationship between input and output is not a static mapping, which precludes the use of fixed kernels in efficient convolutional computation. To overcome this, researchers developed a hardware-aware parallel algorithm that operates in cyclic mode. Unlike traditional models that rely on convolution, Mamba's computations

are performed through scanning. This scanning algorithm overcomes the limitations of recurrent neural networks (RNNs), which cannot execute in parallel due to their sequential dependency. In the scan operation, each new state requires the previous state for computation, making direct parallel scanning through traditional loop computation infeasible. Researchers found that each state effectively represents a complete compression of all the preceding states, which allows previous states to be directly utilized in computing the new state, indicating that the execution order is independent of the sequence's attributes. Mamba implements the selective scan algorithm by segmenting sequences, computing them iteratively, and integrating them with the parameters influenced by the input data. Furthermore, leveraging the capabilities of GPUs, which have numerous processors for parallel computations, Mamba utilizes the GPU's High-Bandwidth Memory (HBM) and rapid Static Random-Access Memory (SRAM) IOs to minimize frequent SRAM writes to the HBM through kernel fusion. Specifically, Mamba conducts discretization and recursive operations within the higher-speed SRAM, then writes the output back to the HBM. When inputs are loaded from the HBM to the SRAM, the intermediate state is not preserved but is recomputed during backpropagation.

2.2.3. Simplified Deep-Learning Architecture

Compared to traditional Transformer-based models, the Mamba model diverges by eliminating the complex self-attention mechanism and instead addressing sequence dependencies through a more efficient state-space approach. This method not only simplifies the model structure but also significantly reduces the number of parameters. Mamba innovatively combines the fundamental blocks of state-space models (SSM) with Multi-Layer Perceptron (MLP) blocks, which are prevalent in modern neural networks. These hybrid blocks are then superimposed and integrated with normalization and residual connectivity techniques to create the Mamba network architecture (refer to Figure 3). The design of Mamba enhances the computational efficiency and scalability, making it suitable for a variety of complex sequence modeling tasks that require both precision and speed.

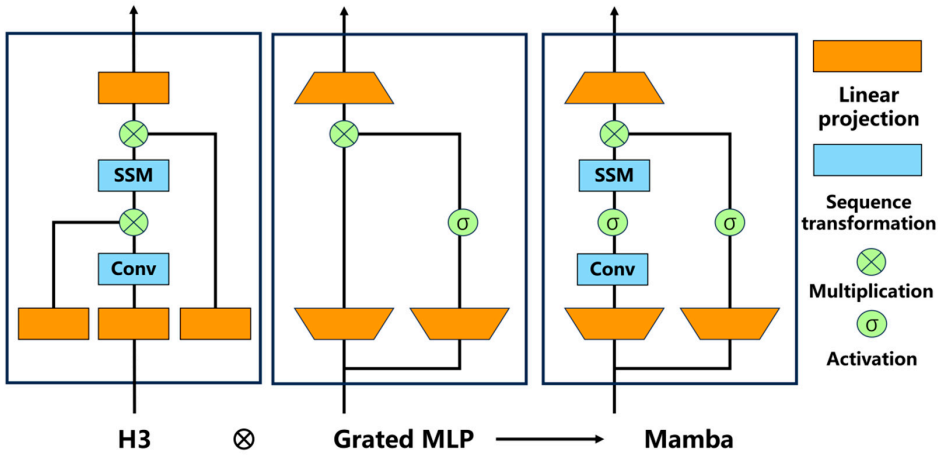


Figure 3. Simplified Mamba architecture.

3. Experiments

3.1. Experimental Data Selection

In this study, we selected the Automatic Identification System (AIS) data of vessels on the Beijing–Hangzhou Canal for the period from September to November 2021 as our research sample, amounting to a total of 157,114 data entries, as depicted in Figure 4. Specifically, the experimental waterway stretches from Tangjiawan through Beilianli to Nanniwan, an area known for its suitability for the navigation of various vessel types, owing to its ample width and depth. However, the curvilinear segments of the waterway present significant challenges to ship navigation efficiency and safety. To effectively predict

and understand vessel behavior in these complex waterways, this paper employs the Mamba model to forecast ship trajectories. The model is adept at capturing the spatial attributes of trajectory points and effectively handles the prediction of both straight and curved sections of the waterway.



Figure 4. Selected areas and trajectory display.

In our experiments, we utilized data from the past five time points to predict the sailing state at the next moment, as based on the ship's longitude (lon), latitude (lat), heading angle (cog), and speed (sog). The application of the Mamba model not only enhances the prediction capabilities for vessels sailing directly but also optimizes the handling of dynamic changes in curved waterways. To ensure the generality and efficiency of the model training, we standardized the data before training.

We employed the Mamba model to predict ship trajectories in specific sections of the Beijing–Hangzhou Canal. To comprehensively evaluate the performance of the Mamba model and compare it with other models widely used in ship trajectory prediction, we selected several benchmark models for comparative testing. These included the traditional long short-term memory network (LSTM), bidirectional long short-term memory network (Bi-LSTM), Gated Recurrent Units (GRUs), Bidirectional Gated Recurrent Units (Bi-GRUs), and the Transformer model, which has garnered significant attention in recent years.

The primary objective of this study is to explore the effectiveness of these different models in predicting complex fairways. Uniform parameter settings were applied across all the models to ensure a fair comparison. The model training process adhered to a strict data partitioning principle: 80% of the data were used for training and optimization, while 20% served as a validation set to monitor and adjust the models' hyperparameters. All the models employed the mean square error (MSE) as the loss function, which aids in quantifying the deviation between the predicted and true values. The optimization algorithm used was Adam, noted for its stable performance, with an initial learning rate set at 0.01. This was coupled with a dynamic learning rate adjustment strategy (ReduceLROnPlateau), aiming for faster convergence and higher learning efficiency without compromising the model's responsiveness.

3.2. Evaluation Indicators

In this study, we utilized five key metrics to assess the performance of the Mamba model in ship trajectory prediction: mean absolute error (MAE), mean absolute percentage

error (MAPE), coefficient of determination (R^2), root mean square error (RMSE), and mean square error (MSE). These metrics collectively provide a comprehensive view of the predictive performance of all the models, facilitating an in-depth understanding of each model's effectiveness and potential in real-world applications. By simultaneously analyzing these metrics, we comprehensively evaluated the performance of each model across different training cycles and datasets. The findings suggest that the Mamba model offers notable improvements in accuracy and efficiency for long series predictions. Through the comparison of these metrics, the enhanced performance of the Mamba model in predicting ship trajectories is confirmed.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \widehat{y_i}| \tag{11}$$

$$MAPE = \frac{100\%}{n}\sum_{i=1}^{n}\left|\frac{y_i - \widehat{y_i}}{y_i}\right| \tag{12}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \widehat{y_i})^2}{\sum_{i=1}^{n}(y_i - \overline{y})^2} \tag{13}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \widehat{y_i})^2} \tag{14}$$

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \widehat{y_i})^2 \tag{15}$$

where y_i is the true value, $\widehat{y_i}$ is the predicted value, $\overline{y}$ is the mean of the true values of the samples in the test set, and n is the total number of samples.

3.3. Experimental Results and Analysis

To effectively mitigate overfitting, the model training cycles were capped at a maximum of 50. Training was halted if the validation loss did not improve significantly over five consecutive cycles. It was observed that the Mamba model achieved its best fit at 20 training cycles. To objectively assess the performance of the different models in relation to the ship trajectory prediction problem, we compared the prediction performance of each model at 5, 10, and 20 training cycles. The models used the latitude, longitude, heading, and speed from the first five trajectory points as inputs to predict the values for the sixth trajectory point. Relative to other models, the Mamba model performed better across all the evaluated metrics, particularly after 20 training cycles. Compared to the other high-accuracy models, the Mamba model demonstrated notable improvements in the fitting effectiveness and error minimization, and it also excelled in terms of the inference speed and resource utilization.

According to the trends of the four evaluation indicators—mean absolute error (MAE), mean absolute percentage error (MAPE), coefficient of determination (R^2), and root mean square error (RMSE)—displayed in Figure 5, it is evident how the models' performance evolves over different training cycles. At the onset of the training period (the first five cycles), the MAE and MAPE metrics for the models are high, indicating initial inaccuracies. These metrics gradually decrease as the training progresses, suggesting an improvement in model accuracy, while the performance of the R^2 and RMSE tends to stabilize, indicating a reliable model fit. The BiGRU and Transformer models exhibit notable performance within the first five cycles, demonstrating a rapid decrease in the MAE and MAPE, alongside better fitting capabilities, as shown by the R^2 and RMSE improvements. However, extending the training period to the 10th cycle, the performance of the Mamba model shows marked improvements, with significant reductions in the MAE and MAPE, an R^2 approaching 1, and strong RMSE results. This highlights the Mamba model's advantages in terms of the prediction accuracy and model-fitting ability, particularly in later training stages, where it effectively controls prediction errors and showcases an excellent trajectory prediction

capability. In contrast, the LSTM and GRU models, while performing well initially, exhibit slower improvements in later stages and their performance tends to stabilize.

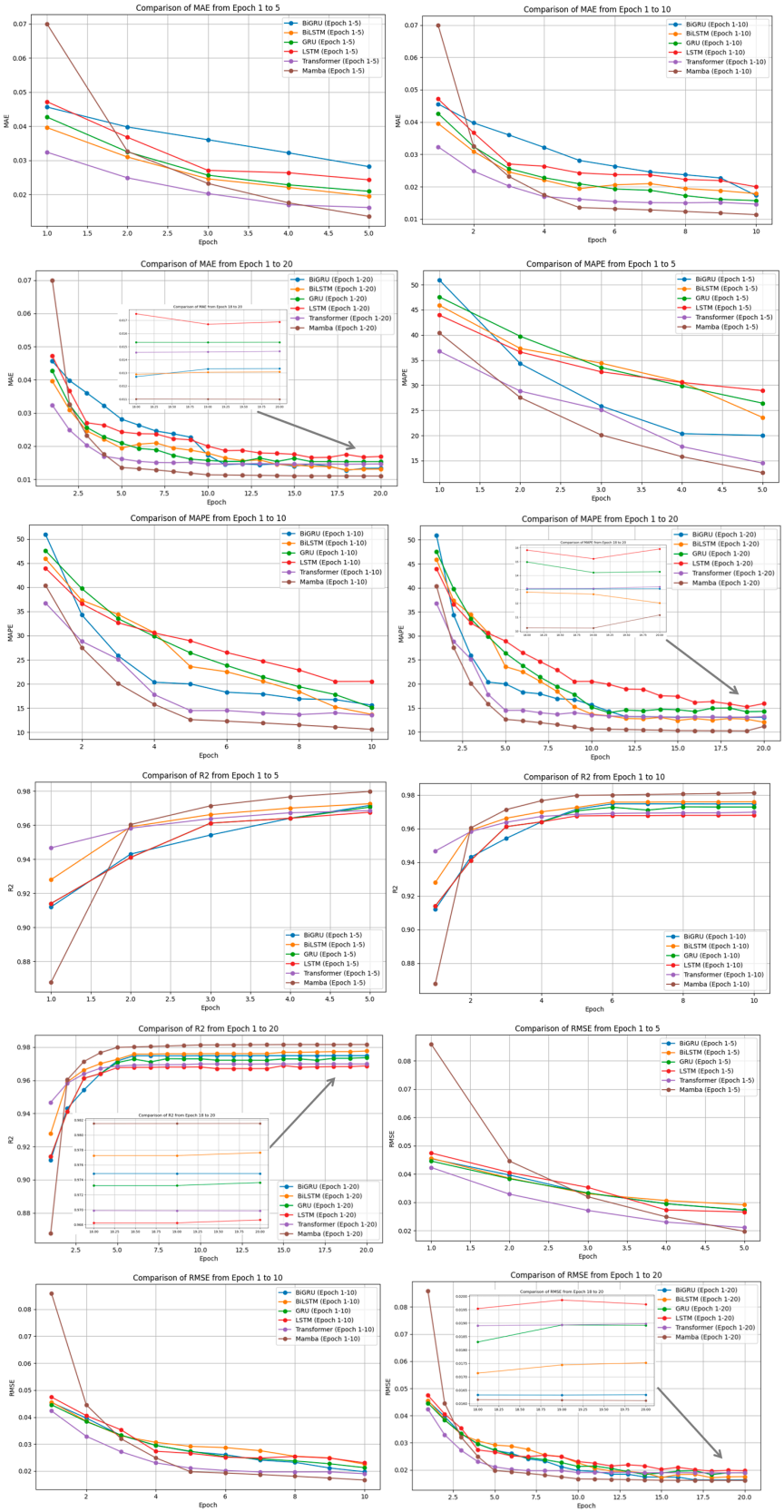


Figure 5. Indicator evaluation comparison chart.

This analysis explores the optimization process of the six models by evaluating their training and testing losses, specifically using the mean square error (MSE), as depicted in Figure 6. The figure illustrates the changes in the losses for each model at 5, 10, and 20 training cycles. The initial rapid decrease in both the training loss and testing loss across the models indicates that they are effectively learning and optimizing from the early stages. Notably, the BiLSTM and GRU models demonstrate a strong learning capability by significantly reducing their loss values within the first five cycles. However, as the training progresses, the Mamba model distinguishes itself by exhibiting a particularly strong performance in terms of both the training loss and testing loss. By the end of 20 training cycles, the loss values for the Mamba model are substantially lower than those of the other models, with a minimal gap between the training and testing losses. This suggests that the Mamba model possesses robust fitting capabilities on the training set and excellent generalization on the test set. Such performance underscores the Mamba model's superior ability to capture data features and provide more accurate predictions in ship trajectory prediction tasks. While the Transformer and BiGRU models also show commendable initial training performance, they slightly lag in later-stage optimization. Detailed values for the relevant metrics during training are provided in Table 1.

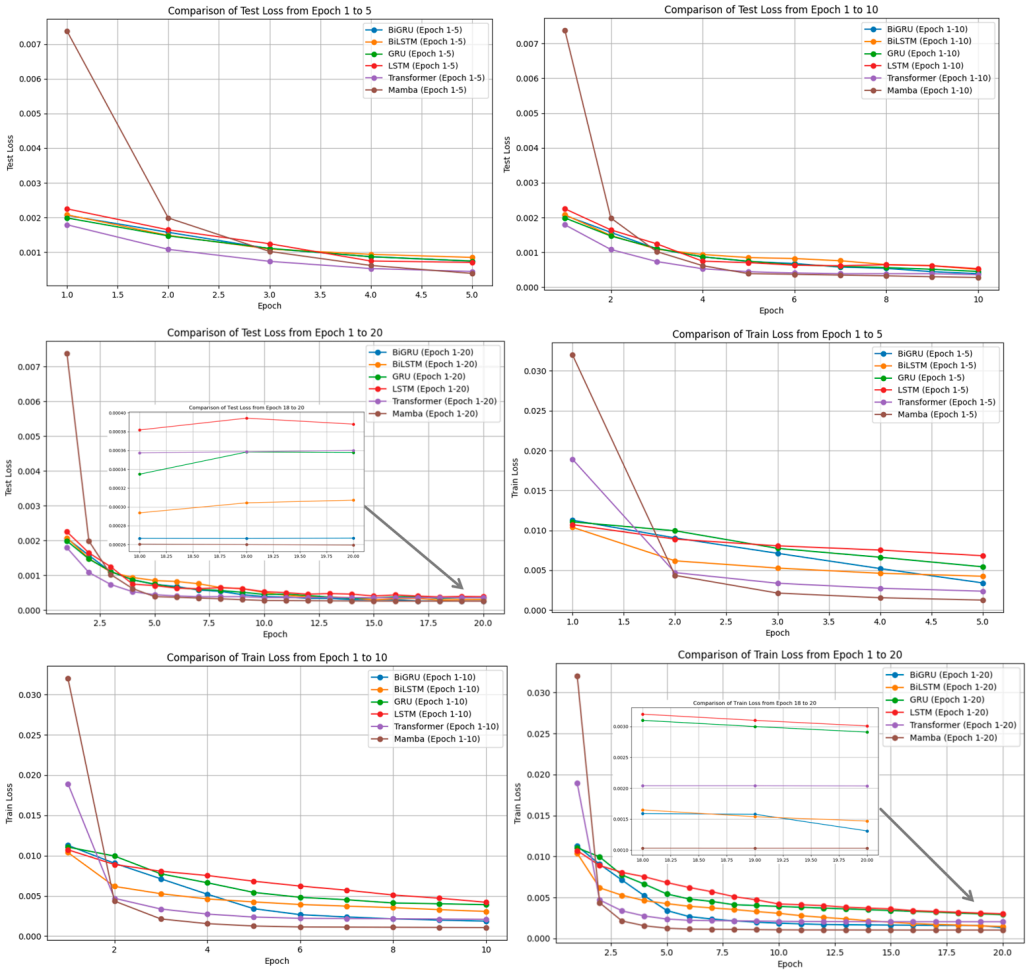


Figure 6. Training set validation set loss comparison plot.

Table 1. Parameters of indicators.

Model	Epochs	MSE ($\times 10^{-3}$)	MAE ($\times 10^{-2}$)	MAPE	RMSE ($\times 10^{-2}$)	R^2
LSTM	5	0.7071	2.4259	28.9257	2.6594	0.9676
Bi-LSTM	5	0.8520	1.9424	23.5968	2.9171	0.9726
GRU	5	0.7475	2.0908	26.4280	2.7341	0.9706
Bi-GRU	5	0.7406	2.8135	19.9996	2.7214	0.9716
Transformer	5	0.4458	1.6149	14.4905	2.1115	0.9685
Mamba	5	0.3901	1.3557	12.6175	1.9762	0.9798
LSTM	10	0.5325	2.0010	20.5223	2.3075	0.9680
Bi-LSTM	10	0.5091	1.7872	13.7067	2.2582	0.9760
GRU	10	0.4528	1.5702	15.1582	2.1279	0.9729
Bi-GRU	10	0.3900	1.7297	15.6274	1.9750	0.9748
Transformer	10	0.3624	1.4620	13.5549	1.9032	0.9698
Mamba	10	0.2783	1.1363	10.6065	1.6693	0.9813
LSTM	20	0.3880	1.6880	15.9107	1.9698	0.9686
Bi-LSTM	20	0.3072	1.3080	12.0198	1.7522	0.9776
GRU	20	0.3579	1.5330	14.2759	1.8917	0.9736
Bi-GRU	20	0.2668	1.3330	13.0470	1.6332	0.9748
Transformer	20	0.3601	1.4642	13.1895	1.8973	0.9699
Mamba	20	0.2594	1.0994	11.1567	1.6115	0.9815

To comprehensively assess the prediction accuracy of each model, this study employs scatter plots for the accuracy analysis. Scatter plots provide a visual representation of the differences between predicted and actual values, enabling quick identification of the prediction accuracy and errors through the distribution and color changes of the points. Figure 7 displays the scatter plots for the six models based on their predicted values, where each point represents the difference between predicted and actual values. A warmer color in the plot indicates a more accurate prediction, while a cooler color signifies a larger error. Although the LSTM and BiLSTM models demonstrate smaller errors at some data points, the Mamba model consistently shows a more concentrated distribution of prediction errors across the majority of points, indicating that its predictions are generally closer to the actual values. This enhanced error control by the Mamba model may be attributed to its effective integration of dynamic temporal features with the nonlinear features present in the serial data, resulting in highly accurate predictions under various conditions. In contrast, while the GRU and Transformer models exhibit high accuracy at specific data points, their overall error distribution is more dispersed and exhibits some volatility, suggesting less consistency in their predictive performance across different data scenarios.

Figure 8 presents a comparative analysis of the inference speed and throughput for the six different models, along with a combined performance metric calculated based on these two factors, with the specific values detailed in Table 2. The inference speed refers to the duration required for a model to make predictions on a batch of data, whereas the throughput indicates the volume of data processed by the model per unit of time. The combined performance metric is quantified by dividing the throughput by the square of the inference time, providing a measure of the overall efficiency. According to the figure, the Mamba model achieves a shorter inference time and higher throughput, outperforming the other models in terms of the overall performance metrics. Notably, the advantages of the Mamba model in terms of the inference speed and throughput are especially significant when compared with the BiLSTM and GRU models. Additionally, Figure 9 compares the GPU utilization rates of the models, revealing that the Mamba model has the lowest GPU utilization rate. This indicates that the Mamba model is capable of handling more tasks with the same resources and exhibits high resource utilization efficiency.

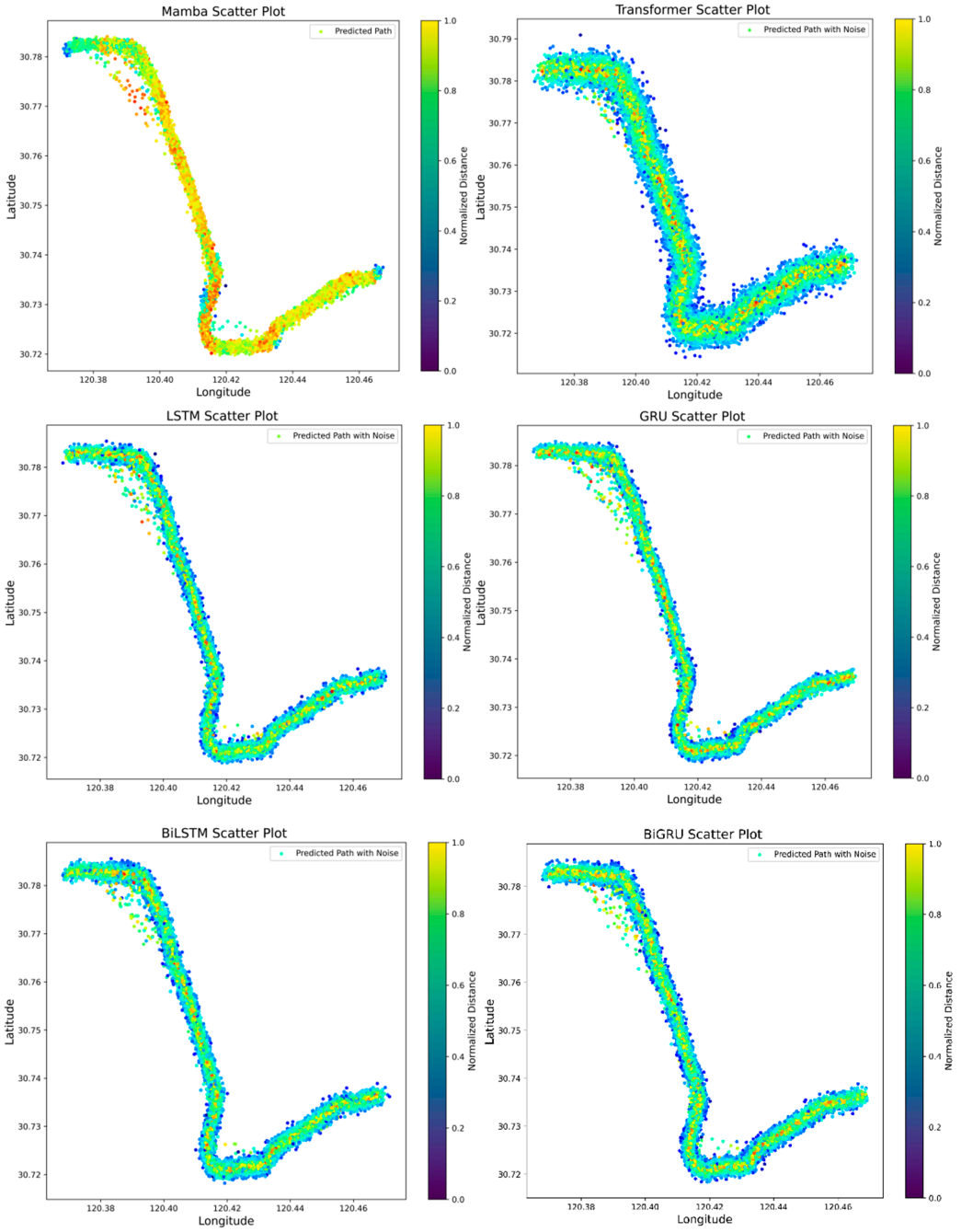


Figure 7. Scatterplot of predictions by model.

Table 2. Data for the indicators.

Model	Inference Time per Batch (s)	Throughput (Samples/s)	GPU Usage
LSTM	0.2099	3.7459	0.1573
Bi-LSTM	1.4622	0.6839	0.1985
GRU	0.4446	2.2490	0.1510
Bi-GRU	0.2561	4.1532	0.1820
Transformer	1.3800	0.7246	0.1371
Mamba	0.1760	3.9052	0.0685

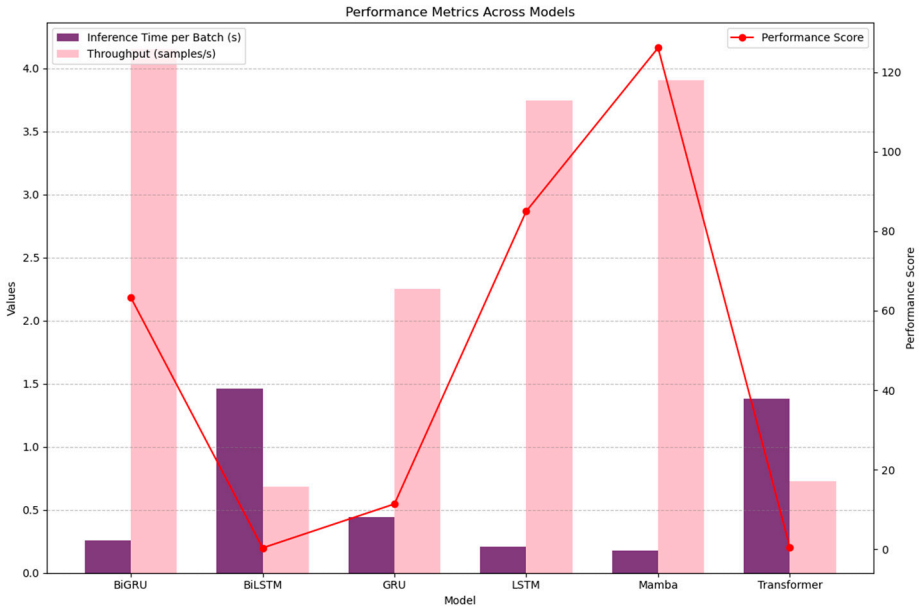


Figure 8. Throughput and inference speed evaluation metrics.

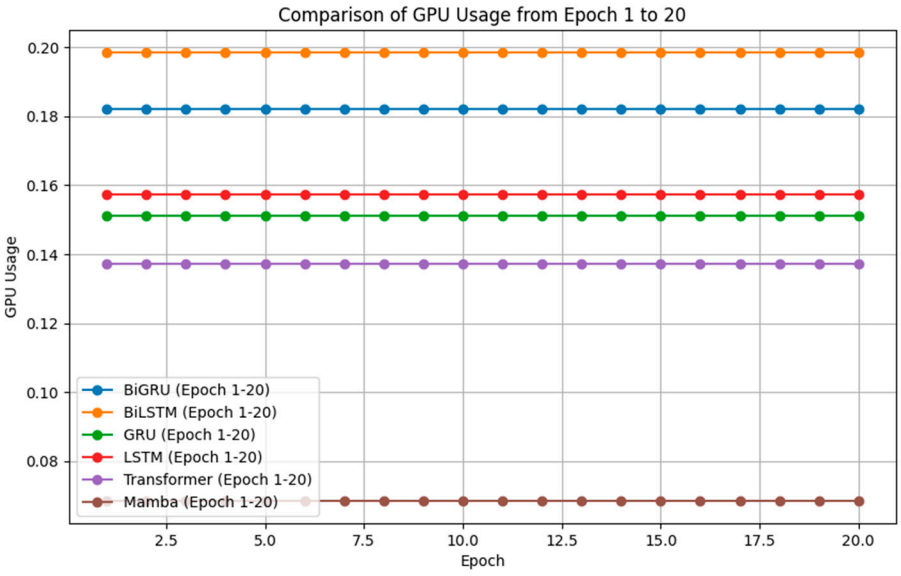
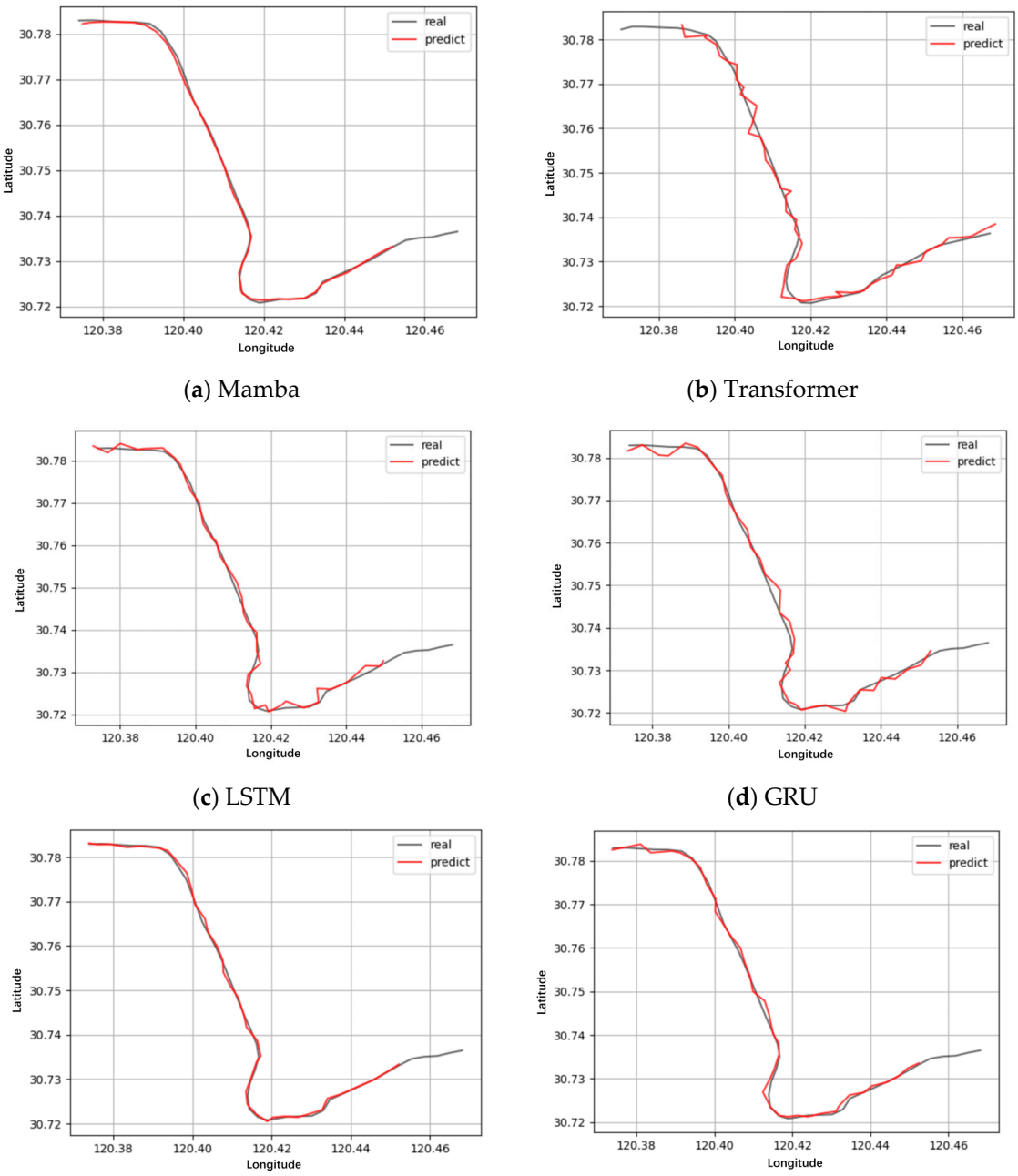


Figure 9. GPU usage by model.

To measure the inference speed, we timed how long each model took to process a predefined batch of data from input to output. The throughput was assessed by counting how many data points each model could process per second. The GPU utilization was measured using system-monitoring tools that track the percentage of GPU resources used by each model during the inference process. This approach not only provided a measure of raw processing power and speed but also highlighted the efficiency of the resource use by each model, particularly showcasing the Mamba model's capability to deliver high performance while consuming fewer GPU resources.

Figure 10 presents the trajectories of the real and predicted values for the six models using identical ship data for comparison. The results show that the predicted trajectories of the Mamba model are consistent with the actual trajectories, exhibiting small errors and high prediction accuracy. While the trajectories predicted by the other models generally follow the actual trajectories, they exhibit larger deviations at certain nodes, with the overall error being higher compared to the Mamba model. Notably, the Transformer models displayed significant deviations in their predictions at crucial nodes, impacting the

overall accuracy of the predictions. These deviations may stem from the models' inherent limitations in handling complex nonlinear trajectory changes. Despite the Transformer model's exemplary performance in capturing long-distance dependencies, it may struggle with intricate trajectory shifts. Similarly, the LSTM and GRU models face performance bottlenecks when addressing long time dependencies, which curtails their effectiveness in extended time series predictions. Furthermore, the limitations of traditional recurrent neural networks in parallel computation result in low training efficiency and hinder the full utilization of large datasets for optimization. Collectively, these factors contribute to the observed shortcomings in the trajectory prediction accuracy and stability for the baseline models.



(**a**) Mamba (**b**) Transformer

(**c**) LSTM (**d**) GRU

(**e**) BiLSTM (**f**) BiGRU

Figure 10. Effectiveness of the six model predictions for ship MMS1:413760205.

Figure 11 illustrates the comparison of the training times for the six models. The results indicate that the Mamba model requires the most time to train, performing less efficiently in terms of the training duration compared to the other models. In contrast, the BiGRU and Transformer models exhibit shorter training times, demonstrating higher training efficiency. Despite its longer training times, the Mamba model outperforms the other models in key performance areas. By the 20th training cycle, it achieves a mean square error (MSE) of 0.2594×10^{-3} and a coefficient of determination (R^2) of 0.9815, indicating high accuracy. It also records an inference speed of 0.1760 s per batch and a throughput of 3.9052 samples per second, showcasing its computational efficiency and effective resource usage. The model's robust performance and adaptability make it well suited for practical scenarios involving ship trajectory prediction. This evaluation highlights the Mamba model's ability to deliver accurate results and manage resources efficiently, making it a good choice for complex real-world applications where precision and resource management are crucial.

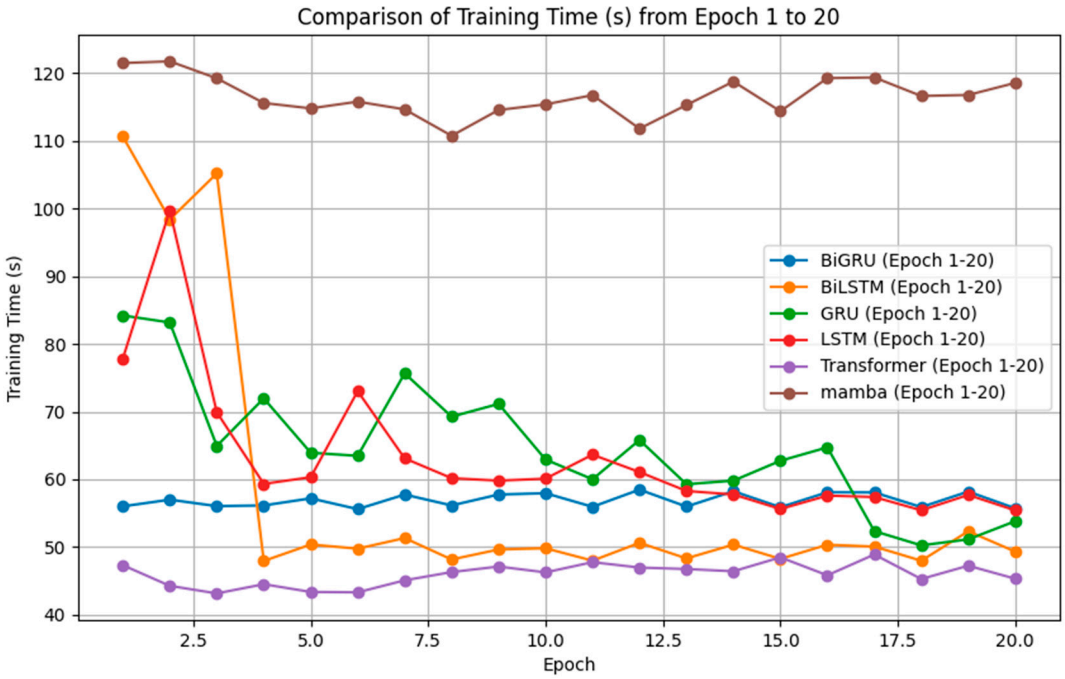


Figure 11. Training time for each model.

4. Conclusions

The experimental results demonstrate that the Mamba model has greater advantages across all the assessment indicators. It performs well in terms of the mean absolute error (MAE), mean percentage error (MAPE), coefficient of determination (R^2), and root mean square error (RMSE) compared to other deep-learning-based ship trajectory prediction models. After 20 training cycles, the MAE and MAPE of the Mamba model decrease, the R^2 approaches 1, and the RMSE also shows strong performance, indicating high prediction accuracy and model-fitting ability. In terms of the inference speed and throughput, the Mamba model also performs well. Specifically, the inference time for the Mamba model is 0.1759 s per batch, which is approximately 7.84 times faster than the Transformer model's 1.38 s per batch. Moreover, the throughput of the Mamba model is 3.9052 samples per second, about 5.39 times higher than the Transformer model's 0.7246 samples per second. These results underline the Mamba model's advantages not only in prediction accuracy but also in computational efficiency. Regarding the training and testing loss, the Mamba model exhibits strong learning and generalization capabilities. The training and testing losses decrease rapidly in the initial training cycles and remain stable and significantly lower than those of other models throughout the subsequent training process, indicating balanced performance across different datasets. To further enhance the model, future work will focus

on reducing the training times by optimizing the model's architecture and improving its efficiency on large-scale datasets. Additionally, we plan to expand the model's application to more diverse maritime environments and integrate real-time data-processing capabilities to enhance its practicality in operational settings. In summary, the Mamba model performs well in relation to ship trajectory prediction, outperforming the other models in prediction accuracy and computational efficiency, providing high practical value and potential for real-world applications. Despite its advantages, the Mamba model also has limitations, such as the longer training times due to its complex design. This characteristic may necessitate more computational resources and time when handling large-scale datasets. Compared to the widely used Transformer-based large models, the Mamba model offers a new approach, showcasing effective predictive capabilities and resource utilization efficiency.

Author Contributions: Conceptualization, Y.S., T.Z. and Z.D.; methodology, Z.D. and T.Z.; validation, Y.S. and Z.D.; writing—review and editing, T.Z., Z.D. and Y.S.; funding acquisition, Y.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China (No. 52371369), the Key Projects of National Key R&D Program (No. 2021YFB390150), the Natural Science Project of Fujian Province (Nos. 2022J01323, 2021J01822, 2020J01660, 20230019), and the Fuzhou-Xiamen-Quanzhou Independent Innovation Region Cooperated Special Foundation (No. 3502ZCQXT2021007).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author. The data are not publicly available due to confidentiality.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Tong, X.; Chen, X.; Sang, L.; Mao, Z.; Wu, Q. Vessel Trajectory Prediction in Curving Channel of Inland River. In Proceedings of the International Conference on Transportation Information and Safety, Wuhan, China, 25–28 June 2015; IEEE: Piscataway, NJ, USA, 2015; pp. 706–714.
2. Mazzarella, F.; Arguedas, V.F.; Vespe, M. Knowledge-based Vessel Position Prediction using Historical AIS Data. In Proceedings of the Sensor Data Fusion: Trends, Solutions, Applications, Bonn, Germany, 6–8 October 2015; pp. 1–6.
3. Fossen, S.; Fossen, T.I. Xogenous Kalman Filter (XKF) for Visualization and Motion Prediction of Ships using Live Automatic Identification System (AIS) Data. *Model. Identif. Control. (MIC)* **2018**, *39*, 233–244. [CrossRef]
4. Sun, L.; Zhou, W. Vessel Motion Statistical Learning based on Stored AIS Data and Its Application to Trajectory Prediction. In Proceedings of the International Conference on Machinery, Materials and Computing Technology (ICMMCT 2017), Beijing, China, 25–26 March 2017; pp. 1183–1189.
5. Rong, H.; Teixeira, A.; Soares, C.G. Ship Trajectory Uncertainty Prediction Based on a Gaussian Process Model. *Ocean. Eng.* **2019**, *182*, 499–511. [CrossRef]
6. Jiang, B.; Guan, J.; Zhou, W.; Chen, X. Vessel Trajectory Prediction Algorithm Based on Polynomial Fitting Kalman Filtering. *J. Signal Process.* **2019**, *5*, 741–746.
7. Qiao, S.-J.; Han, N.; Zhu, X.-W.; Shu, H.-P.; Zheng, J.-L.; Yuan, C.-A. A Dynamic Trajectory Prediction Algorithm Based on Kalman Filter. *Acta Electron. Sin.* **2018**, *46*, 418.
8. Mao, S.; Tu, E.; Zhang, G.; Rachmawati, L.; Rajabally, E.; Huang, G.B. An Automatic Identification System (AIS) Database for Maritime Trajectory Prediction and Data Mining. In *Proceedings of the ELM-2016*; Springer: Berlin/Heidelberg, Germany, 2018; pp. 241–257.
9. Fortin, M.; Millefiori, L.M.; Braca, P.; Willett, P. Prediction of Vessel Trajectories from AIS Data via Sequence-to-Sequence Recurrent Neural Networks. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Barcelona, Spain, 4–8 May 2020; pp. 8936–8940.
10. Capobianco, S.; Millefiori, L.M.; Forti, N.; Braca, P.; Willett, P. Deep Learning Methods for Vessel Trajectory Prediction Based on Recurrent Neural Networks. *IEEE Trans. Aerosp. Electron. Syst.* **2021**, *57*, 4329–4346. [CrossRef]
11. Murray, B.; Perera, L.P. A Dual Linear Autoencoder Approach for Vessel Trajectory Prediction Using Historical AIS Data. *Ocean Eng.* **2020**, *209*, 107478. [CrossRef]
12. Qian, L.; Zheng, Y.; Li, L.; Ma, Y.; Zhou, C.; Zhang, D. A New Method of Inland Water Ship Trajectory Prediction Based on Long Short-term Memory Network Optimized by Genetic Algorithm. *Appl. Sci.* **2022**, *12*, 4073. [CrossRef]

13. Suo, Y.; Chen, W.; Claramunt, C.; Yang, S. A Ship Trajectory Prediction Framework Based on a Recurrent Neural Network. *Sensors* **2020**, *20*, 5133. [CrossRef] [PubMed]
14. Gao, D.; Zhu, Y.; Zhang, J.; He, Y.; Yan, K.; Yan, B. A Novel MP-LSTM Method for Ship Trajectory Prediction Based on AIS Data. *Ocean Eng.* **2021**, *228*, 108956. [CrossRef]
15. Nguyen, D.D.; Le Van, C.; Ali, M.I. Vessel Trajectory Prediction Using Sequence-to-Sequence Models over Spatial Grid. In Proceedings of the 12th ACM International Conference on Distributed and Event-Based Systems, Hamilton, New Zealand, 25–29 June 2018; pp. 258–261.
16. Luo, W.; Zhang, G. Ship Motion Trajectory and Prediction Based on Vector Analysis. *J. Coast. Res.* **2020**, *95*, 1181–1183. [CrossRef]
17. Zhang, L.; Zhang, J.; Niu, J.; Wu, Q.J.; Li, G. Track Prediction for HF Radar Vessels Submerged in Strong Clutter Based on MSCNN Fusion with GRU-AM and AR Model. *Remote Sens.* **2021**, *13*, 2164. [CrossRef]
18. Hoseinzade, E.; Haratizadeh, S. CNNpred: CNN-based Stock Market Prediction Using a Diverse Set of Variables. *Expert Syst. Appl.* **2019**, *129*, 273–285. [CrossRef]
19. Huang, S.; Wang, D.; Wu, X.; Tang, A. Dsanet: Dual Self-attention Network for Multivariate Time Series Forecasting. In Proceedings of the 28th ACM International Conference on Information and Knowledge Management, Beijing, China, 3–7 November 2019; pp. 2129–2132.
20. Tang, H.; Yin, Y.; Shen, H. A Model for Vessel Trajectory Prediction Based on Long Short-term Memory Neural Network. *J. Mar. Eng. Technol.* **2019**, *21*, 136–145. [CrossRef]
21. Xiao, H.; Wang, C.; Li, Z.; Wang, R.; Bo, C.; Sotelo, M.A.; Xu, Y. UB-LSTM: A Trajectory Prediction Method Combined with Vehicle Behavior Recognition. *J. Adv. Transp.* **2020**, *2020*, 8859689. [CrossRef]
22. Jaseena, K.U.; Kovoor, B.C. Decomposition-based Hybrid Wind Speed Forecasting Model Using Deep Bidirectional LSTM Networks. *Energy Convers. Manag.* **2021**, *234*, 113–144. [CrossRef]
23. Xue, H.; Huynh, D.Q.; Reynolds, M. Ss-lstm: A Hierarchical LSTM Model for Pedestrian Trajectory Prediction. In Proceedings of the 2018 IEEE Winter Conference on Applications of Computer Vision (WACV), Lake Tahoe, NV, USA, 12–15 March 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 1186–1194.
24. Giuliari, F.; Hasan, I.; Cristani, M.; Galasso, F. Transformer networks for trajectory forecasting. In Proceedings of the International Conference on Pattern Recognition (ICPR), Milan, Italy, 10–15 January 2021; pp. 10335–10342.
25. Janner, M.; Li, Q.; Levine, S. Offline reinforcement learning as one big sequence modeling problem. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 1273–1286.
26. Liang, J.; Jiang, L.; Niebles, C.; Hauptmann, A.G.; Li, F.-F. Peeking into the future: Predicting future person activities and locations in videos. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 5725–5734.
27. Gu, A.; Dao, T. Mamba: Linear-Time Sequence Modeling with Selective State Spaces. *arXiv* **2023**, arXiv:2312.00752.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.