ISSN: 2472-3703 DOI: 10.6911/WSRJ.202602_12(2).0001

Research on the Impact of Optimizer Design on Generalization Performance in Ship Trajectory Prediction Based on LSTM

Qian Gao, Shaoyi Guo

School of Shipping, Shandong Jiaotong University, Weihai, 264200, China

Abstract

Ship trajectory prediction using the Automatic Identification System (AIS) is a core task in intelligent shipping. Focusing on straight-line navigation segments, this paper constructs a unified multi-target LSTM prediction framework that simultaneously forecasts four key variables: longitude, latitude, speed, and course. Through five-fold cross-validation and under identical network architectures, we compare the performance of three models: standard LSTM, attention-enhanced LSTM, and LSTM optimized with Lookahead-RAdam. Experimental results demonstrate that the LSTM model incorporating the Lookahead-RAdam optimization strategy significantly outperforms the other approaches across all evaluation metrics. Further analysis reveals that optimizer design has a far greater impact on model generalization than network architecture complexity. This study proposes an efficient paradigm for maritime spatiotemporal sequence modeling: "simple architecture + advanced optimization."

Keywords

Ship trajectory prediction; LSTM; RAdam; Lookahead; Cross-validation; Multi-task learning.

1. INTRODUCTION

With the acceleration of global shipping intelligence, high-precision ship trajectory prediction has become a critical technology for collision warning, route planning, and traffic flow management. Traditional dynamic models rely heavily on precise physical parameters and struggle to adapt to complex maritime conditions. In contrast, data-driven methods—particularly recurrent neural networks (RNNs)—have gained widespread attention due to their powerful temporal modeling capabilities [1].

To systematically investigate the combined effects of model architecture complexity and optimizer design on the generalization performance of ship trajectory prediction, this paper designs and implements an experimental framework comprising four distinct yet progressively structured deep learning models. This framework adheres strictly to controlled-variable principles, enabling quantitative assessment of baseline architectural capacity, the added value of attention mechanisms, the benefits of advanced optimizers, and ultimately, the deeper interaction between architecture and optimizer. The three primary models—LSTM [2], Attention-LSTM [3], and Lookahead-RAdam-LSTM—are described in detail below.

Using rigorous five-fold cross-validation, we not only evaluate the average performance of each model but also conduct in-depth analyses of their inter-fold stability, error distribution characteristics, and training dynamics.

ISSN: 2472-3703 DOI: 10.6911/WSRJ.202602_12(2).0001

2. SHIP TRAJECTORY PREDICTION USING THREE LSTM-BASED MODELS

2.1. Data Preprocessing

The experimental AIS data covers a region with longitude from 120.05° to 120.7° and latitude from 35.75° to 36.25°. The dataset comprises 6,844,955 AIS data points from 22 different vessel types and 14,895 unique MMSI identifiers. Raw AIS trajectories undergo systematic preprocessing—including effective filtering, noise removal, missing data repair, redundancy elimination, and trajectory reconstruction—to enhance data usability. Additionally, the Douglas-Peucker (D-P) algorithm [4] is applied to compress the trajectory data.

2.2. Model Architecture

All models share the same backbone: a 2-layer LSTM with 64 hidden units, a dropout rate of 0.3, and an L2 regularization coefficient of 1e-4. The differences lie in their specific enhancements:

Basic LSTM: Ends with a fully connected layer.

Attention-LSTM: Computes attention weights over LSTM outputs, performs weighted summation, and then feeds into a fully connected layer.

Lookahead-RAdam-LSTM: Uses RAdam as the inner optimizer, wrapped by the Lookahead mechanism (with k=5 and α=0.8).

2.3. Experimental Setup and Evaluation Metrics

Experiments were conducted on an NVIDIA RTX 3090 GPU using PyTorch 1.12. All comparative models used identical hyperparameter configurations. Since ship trajectory prediction is a regression task (predicting continuous longitude and latitude coordinates), we employ three complementary and widely accepted regression metrics to comprehensively assess model performance:

Mean Squared Error (MSE): The most common loss function and evaluation metric in regression.

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

Mean Absolute Error (MAE): An intuitive and robust regression metric.

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

Root Mean Squared Error (RMSE): The square root of MSE, offering both sensitivity and interpretability.

$$RMSE=\sqrt{MSE}=\sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i-\hat{y}_i)^2} \quad (3)$$

By reporting MSE, MAE, and RMSE together, this study provides a multi-dimensional and unbiased view of model performance: MSE reveals tolerance to extreme errors, MAE reflects general prediction accuracy, and RMSE offers a balanced composite measure. All metrics are computed under a strict five-fold cross-validation framework, reporting both mean and standard deviation to reflect expected performance and stability across different data subsets.

ISSN: 2472-3703 DOI: 10.6911/WSRJ.202602_12(2).0001

2.4. Experimental Comparison and Results Analysis

2.4.1. Loss Curves

To gain deeper insight into the convergence behavior, stability, and generalization capabilities of different models during training, we systematically analyze the training and testing loss curves. These curves offer crucial insights into the optimization trajectory and serve as key indicators of model health and potential.

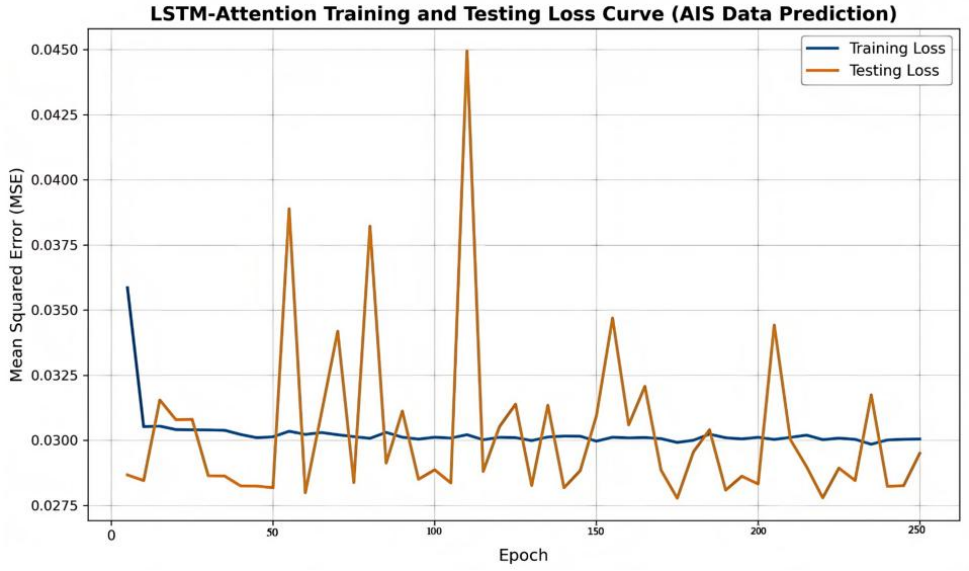


Figure 1. Training and validation loss curves of the LSTM model

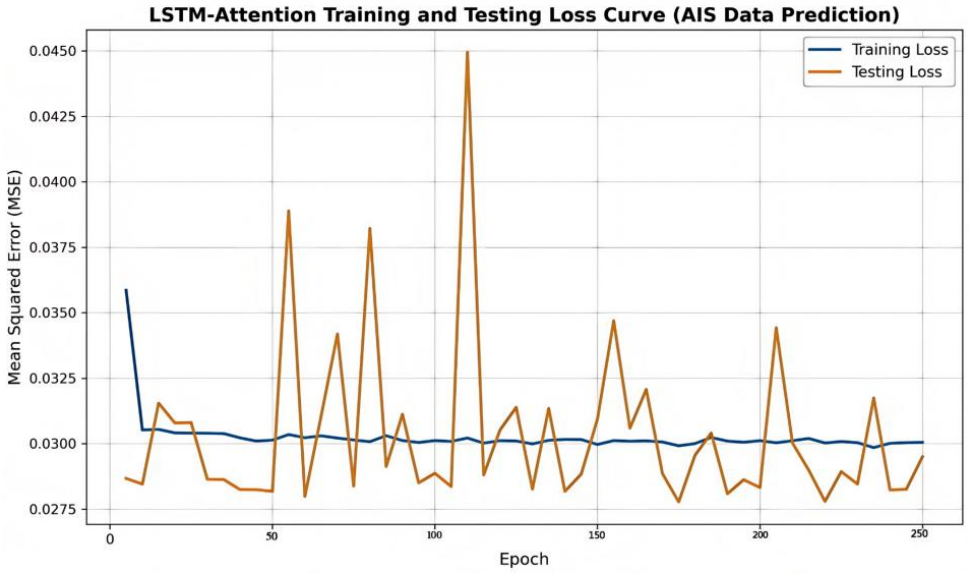


Figure 2. Training and validation loss curves of Attention-LSTM model

ISSN: 2472-3703 DOI: 10.6911/WSRJ.202602_12(2).0001

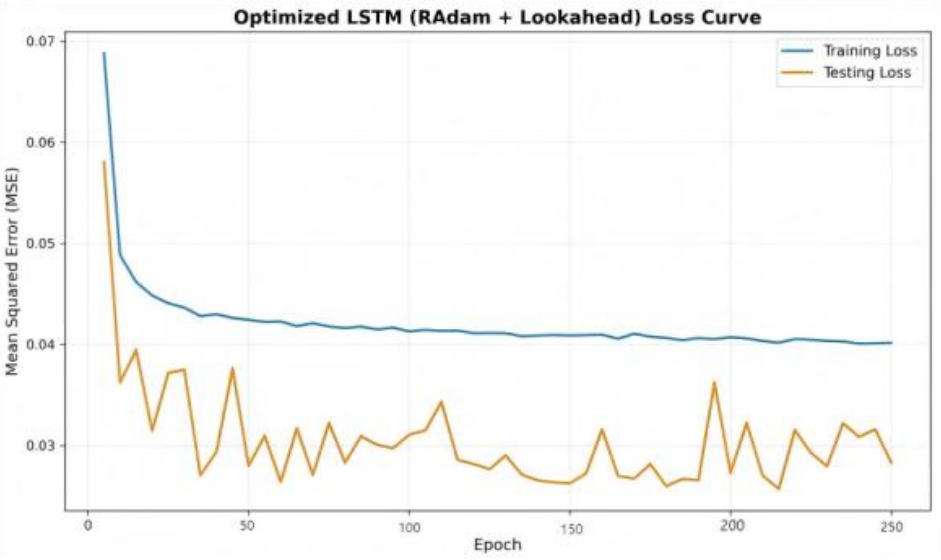


Figure 3. Training and validation loss curves of radam-lookahea-LSTM model

Basic LSTM (Fig. 1): Training loss drops rapidly in early epochs and stabilizes around 0.030, indicating effective fitting to training data. However, test loss exhibits significant high-variance fluctuations, with a sharp spike (~0.06) around epoch 200.

Attention-LSTM(Fig. 2): Shows a similar pattern—training loss stabilizes near 0.030, while test loss remains volatile, though its overall mean is slightly lower than that of Basic LSTM.

Lookahead-RAdam-LSTM (Fig. 3): Displays markedly different dynamics. Training loss declines more gradually from a higher initial value, eventually stabilizing around 0.040. Crucially, test loss shows a smooth, continuous decline throughout training and maintains a stable low level in later stages. Although its final test loss is slightly higher than Basic LSTM's, its extremely low variance and consistent convergence demonstrate the superior ability of the RAdam+Lookahead combination to stabilize the optimization path and suppress gradient noise.

2.4.2. Detailed Result Analysis

To further explore the robustness differences among architectures and optimization strategies, we conduct a longitudinal comparison across training stages using the trajectory of a representative vessel (MMSI: 106603304_1) with a 20-step input window. Predictions and evaluation metrics for longitude, latitude, speed, and course are recorded at Epochs 100, 150, 200, and 250.

ISSN: 2472-3703 DOI: 10.6911/WSRJ.202602_12(2).0001

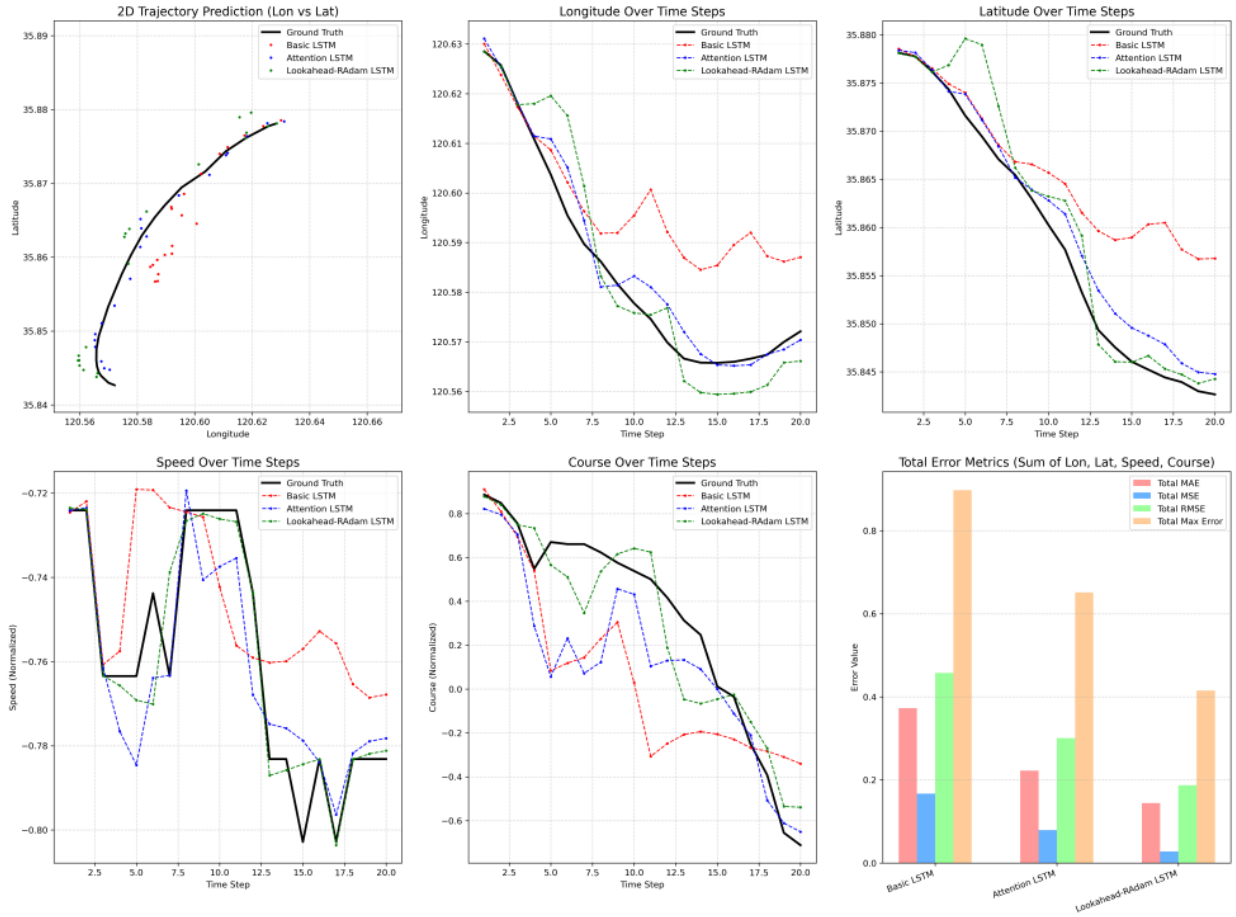


Figure 4. Trajectory prediction, error metrics, comparisons of longitude/latitude and speed/course at Epoch = 100

Epoch = 100 (Fig4): In 2D trajectory plots, Lookahead-RAdam-LSTM shows the best alignment with ground truth, with minimal deviation. Attention-LSTM performs moderately, while Basic LSTM deviates significantly, especially at curve points. In time-series predictions, Lookahead-RAdam-LSTM exhibits the lowest volatility in speed/course forecasting. Overall error metrics (Total MAE, MSE, Max Error) are lowest for Lookahead-RAdam-LSTM, followed by Attention-LSTM, with Basic LSTM performing worst.

ISSN: 2472-3703 DOI: 10.6911/WSRJ.202602_12(2).0001

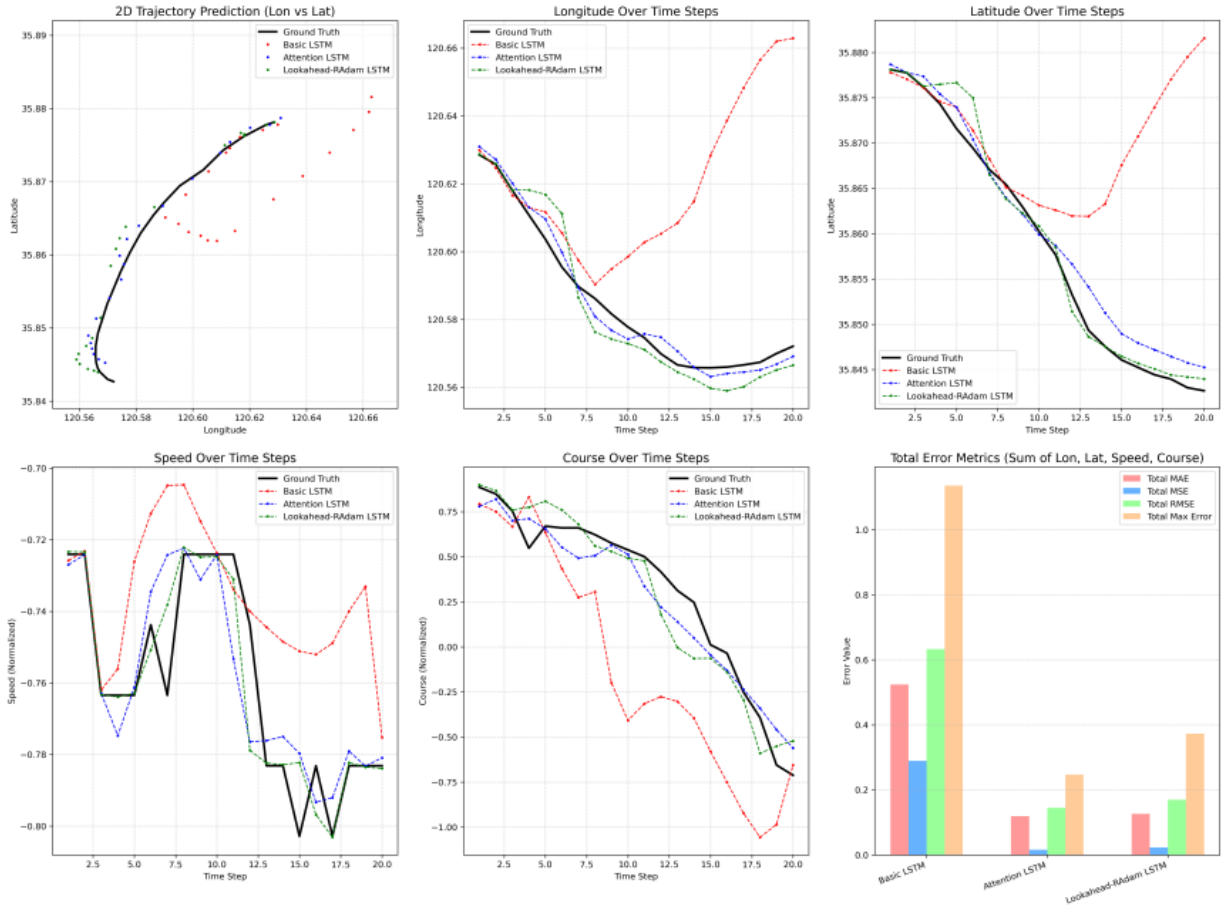


Figure 5. Trajectory prediction, error metrics, comparisons of longitude/latitude and speed/course at Epoch = 150

Epoch = 150 (Fig.5): Attention-LSTM temporarily achieves the best spatial fit, while Lookahead-RAdam-LSTM shows slight lag in the latter segment. Basic LSTM diverges severely at the trajectory end. Error metrics confirm Attention-LSTM's temporary superiority, though Lookahead-RAdam-LSTM continues to improve steadily.

ISSN: 2472-3703 DOI: 10.6911/WSRJ.202602_12(2).0001

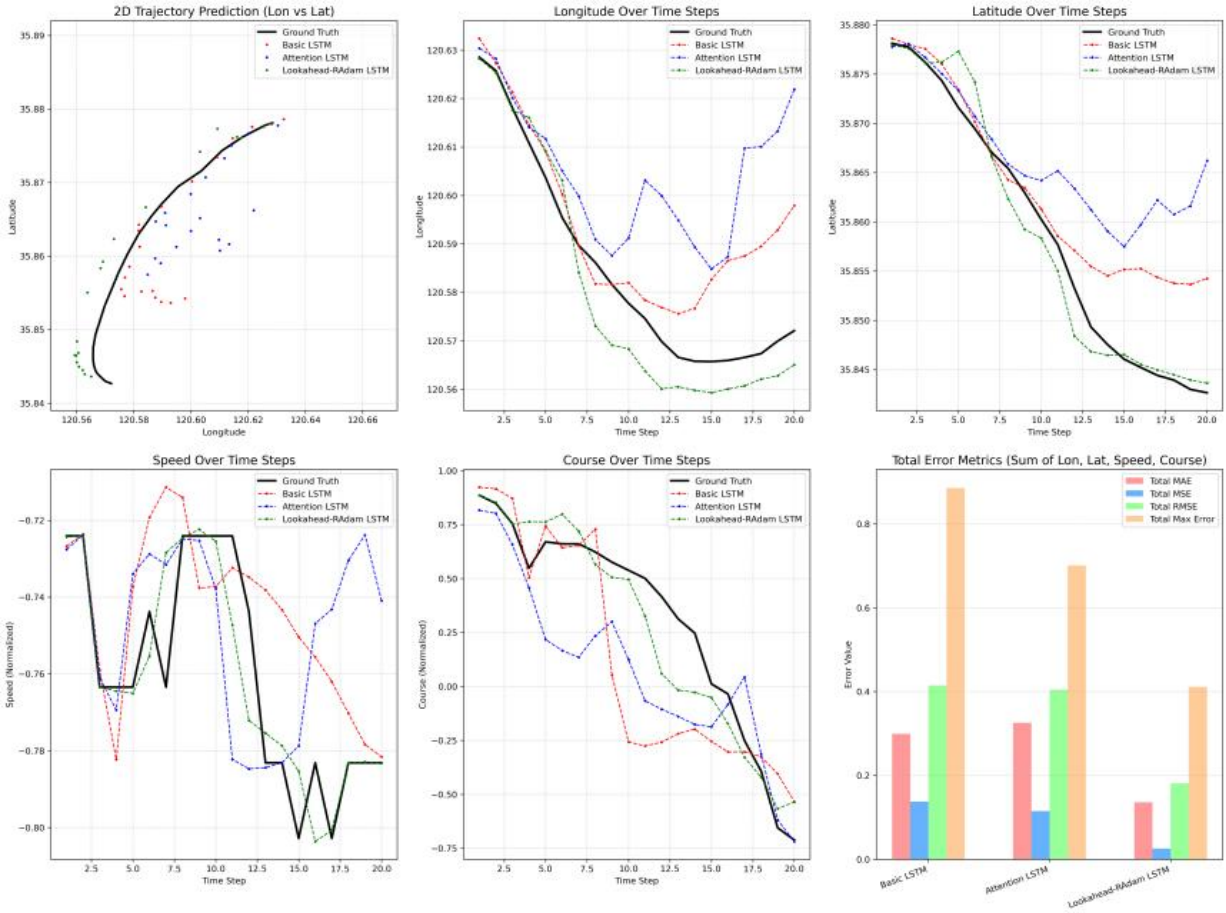


Figure 6. Trajectory prediction, error metrics, comparisons of longitude/latitude and speed/course at Epoch = 200

Epoch = 200 (Fig.6): Lookahead-RAdam-LSTM regains clear dominance in both spatial and temporal predictions. Attention-LSTM begins to show signs of instability and increased error, suggesting emerging overfitting. Basic LSTM remains highly erratic.

ISSN: 2472-3703 DOI: 10.6911/WSRJ.202602_12(2).0001

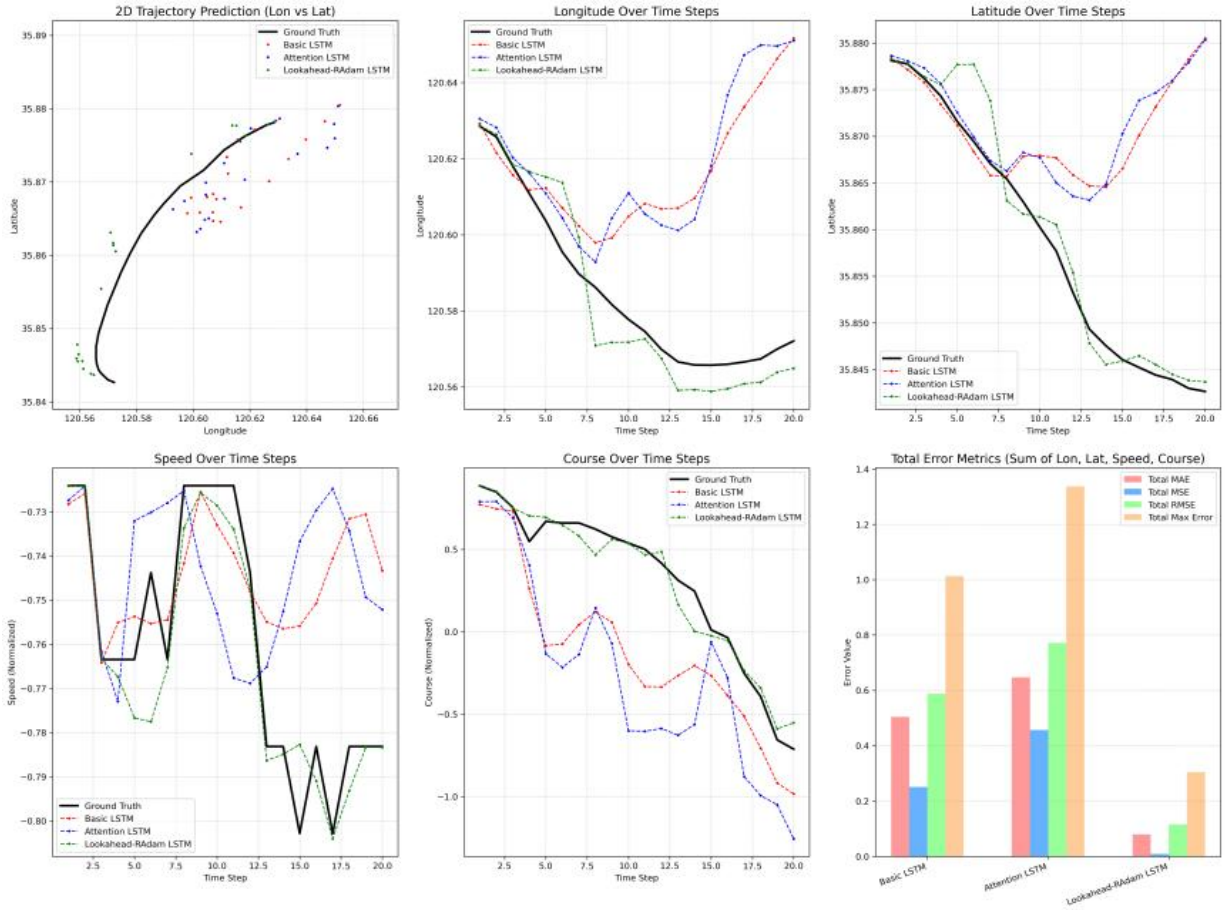


Figure 7. Trajectory prediction, error metrics, comparisons of longitude/latitude and speed/course at Epoch = 250

Epoch = 250 (Fig.7): Lookahead-RAdam-LSTM maintains consistently high accuracy with minimal fluctuation. Both Attention-LSTM and Basic LSTM exhibit severe divergence in later time steps. Error metrics reveal that Attention-LSTM's performance degrades significantly due to overfitting from its additional attention parameters, while Basic LSTM shows no meaningful improvement. In contrast, Lookahead-RAdam-LSTM achieves an MSE of 0.01 and a maximum error of only 0.35, demonstrating exceptional long-term stability.

Analysis Summary:

Basic LSTM suffers from persistent performance degradation due to the instability of the standard Adam optimizer, which relies solely on first-moment estimates and is prone to gradient oscillations.

Attention-LSTM initially benefits from selective feature weighting but later overfits due to its increased parameter count, leading to unstable predictions after prolonged training.

Lookahead-RAdam-LSTM excels throughout training. RAdam dynamically adjusts the learning rate—behaving like SGD in early stages for stability and activating adaptive learning later—while Lookahead's "fast-slow weight" update mechanism smooths the optimization path, suppresses gradient noise, and enhances generalization.

3. CONCLUSION

Integrating Lookahead and RAdam into the LSTM framework significantly improves both prediction accuracy and training stability. This optimization strategy effectively addresses the high variance and instability commonly seen in traditional LSTM training. Experimental results confirm that the improved LSTM model outperforms both Basic LSTM and Attention-LSTM

ISSN: 2472-3703 DOI: 10.6911/WSRJ.202602_12(2).0001

across all metrics (MSE, MAE, RMSE), with smoother loss curves and no performance collapse during extended training. This work validates the efficacy of the "simple architecture + advanced optimization" paradigm for maritime spatiotemporal modeling.

REFERENCES

[1] You L, Xiao S, Peng Q, et al. St-seq2seq: A Spatio-Temporal Feature-optimized Seq2seq Model for Short-Term Vessel Trajectory Prediction [J]. IEEE Access, 2020, 8: 218565–21857.

[2] Hochreiter S, Schmidhuber J. Long Short-Term Memory [J]. Neural Computation, 1997, 9(8): 1735-1780. DOI:10.1162/neco.1997.9.8.1735.

[3] Bahdanau D, Cho K, Bengio Y. Neural Machine Translation by Jointly Learning to Align and Translate [C]//Proceedings of the 3rd International Conference on Learning Representations. San Diego: ICLR, 2015.

[4] Bellman R. On the theory of dynamic programming [J]. Proceedings of the National Academy of Sciences of the United States of America, 1952, 38(8): 716-719.