Traffic & Transportation

Multi-Step Trajectory Prediction of Port Container Trucks Based on CT-HybridNet Model

Haixiong YE[1], Mingqi GAO[2], Xiliang ZHANG[3], Zhe XU[4]

Original Scientific Paper
Submitted: 9 Oct 2024
Accepted: 7 Mar 2025

[1] hxye@shou.edu.cn, College of Engineering Science and Technology, Shanghai Ocean University, Shanghai, China
[2] m220851438@shou.edu.cn, College of Engineering Science and Technology, Shanghai Ocean University, Shanghai, China
[3] Corresponding author, xlzhang@sspu.edu.cn, School of Intelligent Manufacturing and Control Engineering, Shanghai Polytechnic University, Shanghai, China
[4] xuzhe@shou.edu.cn, College of Engineering Science and Technology, Shanghai Ocean University, Shanghai, China

Publisher:
Faculty of Transport and Traffic Sciences, University of Zagreb

ABSTRACT
In port environments, container stacking at significant heights obstructs satellite signal reception by terminal equipment on container trucks, leading to inaccurate positional tracking data. To address this, it is necessary to predict container truck trajectories to fill in the inaccurate positioning signals. In this study, we collected port container truck trajectory data and compared the predictive performance of long short-term memory (LSTM) networks and transformer models, revealing performance turning points at different prediction steps. Based on these findings, we propose a hybrid model named CT-HybridNet, which integrates the LSTM-based DeepPBM-M model with the transformer-based PatchTST model. Given the independence of the prediction errors of the two models, we assume both errors follow a Gaussian distribution. By performing an affine transformation, the proposed hybrid method's output also follows a Gaussian distribution. Additionally, an adaptive parameter adjustment mechanism optimises performance, enabling CT-HybridNet to achieve dual improvements in trajectory prediction accuracy and stability, with 15% improvement in short-term accuracy and 20% in long-term performance. This study provides a more accurate and stable technical solution for port container truck trajectory prediction, overcoming issues related to positioning inaccuracies and signal obstructions.

KEYWORDS
intelligent vehicle; CT-HybridNet; hybrid model; multi-step trajectory prediction.

1. INTRODUCTION

Ports play a crucial role in the global supply chain network, as their operational efficiency directly influences the cost and effectiveness of logistics. Consequently, enhancing automation in port operations, especially in container vehicle scheduling and management, has become a key industry trend [1]. However, advancing this transformation requires overcoming numerous challenges, including the reliance of port central dispatch systems on Global Navigation Satellite System (GNSS) signals for vehicle monitoring, which are susceptible to signal loss and interference. This vulnerability necessitates the adoption of trajectory prediction technologies to compensate for signal loss, making improved prediction accuracy a primary challenge [2, 3].

The dynamic and complex port environments demand models that can accurately capture the nonlinear behaviour of vehicle movements. Additionally, physical obstructions within ports, such as stacked containers, often interfere with satellite signal reception for terminal devices on container trucks, resulting in inaccuracies in position tracking data. To mitigate these issues, short-term prediction models are employed to correct signal interruptions and ensure precise tracking.

Long-term prediction models, on the other hand, focus on forecasting vehicle positions over extended timeframes, which is crucial for collision prevention and optimising container operations. Given the inherent randomness of vehicle movements over longer periods, these models often face challenges with predictive accuracy, which has made them less common in research. Nevertheless, in dynamic port environments, long-term trajectory prediction is essential for anticipating container truck movements, enabling effective operational planning and minimising the impact of GNSS signal interruptions [4].

Trajectory prediction models typically treat each inference unit as a series of positions. This method aims to comprehensively understand the vehicle's motion patterns. However, most existing studies have only examined the predictive performance of fixed-length trajectory segments, which may not suffice to represent the trajectory of the various vehicle movements in the real world.

With the advent of deep learning, neural network-based trajectory prediction models, especially LSTM networks and transformer models, have increasingly gained attention [5]. LSTM networks, known for their excellent capability in processing time-series data, have been widely applied in trajectory prediction.

This study aims to improve the trajectory prediction performance of port trucks under both short-term and long-term prediction steps. The main contributions are listed as follows:

1) Discovery of performance turning points: This study performs an in-depth analysis of two mainstream trajectory prediction models – the LSTM model and the transformer model – finding performance turning points at various prediction steps.
2) Adaptive fusion strategies for predictive modelling: In this study, the output of prediction models is considered as random variables that follow a normal distribution, and probability theory is integrated with adaptive adjusting techniques to implement a novel model fusion strategy. This method can more effectively integrate the performance advantages of different models under each prediction step size, thereby obtaining better performance in complex prediction tasks. Moreover, this adaptive fusion strategy is not only applicable to the two models integrated in this paper but also suitable for all models exhibiting performance inflection points.
3) CT-HybridNet model: A new hybrid model, CT-HybridNet, is proposed, which combines the LSTM-based DeepPBM-M model with the transformer-based PatchTST model. This innovative structure effectively captures both short-term and long-term temporal dependencies in port container truck trajectory data, offering significant enhancements to the trajectory prediction performance at each step size.

In the subsequent sections of this article, the structure of the paper will be presented. The development status of kinematic models, LSTM networks and transformer models in recent years are reviewed next. A detailed introduction of the integrated DeepPBM-M and PatchTST models, including improvements, and a discussion on the architecture and optimisation strategies of the hybrid model CT-HybridNet is provided thereafter. The dataset is collected, and the predictive performance is described through specific experiments, comparing different models in the following section. The analysis of the experimental results and their theoretical significance will be thoroughly explored later. The findings of the research are summarised, and their significance for future applications in port logistics is discussed in the final section.

2. RELATED WORK

Trajectory prediction is an important research topic that involves the technology of predicting the positional changes of moving entities over a certain period. With the increasing demand for automation transformation applications, research related to trajectory prediction is continuously deepening and expanding. This section will review the relevant work in the field of trajectory prediction, with a particular focus on the development of kinematic models, LSTM and transformer models in this field.

2.1 Kinematic models

Initially, traditional vehicle modelling commonly employed kinematic models. Kinematic models focus on the motion state of vehicles, such as position, velocity and acceleration. Lin et al. [6] enhanced the accuracy of trajectory prediction through the numerical integration of a linearised two-degree-of-freedom car model. Polychronopoulos et al. [7] developed a hierarchical prediction model capable of integrating real traffic conditions with the kinematic model of the ego vehicle, improving the adaptability to different traffic scenarios. Anderson et al. [8] proposed a kinematics-based trajectory prediction model for highway scenarios. By decomposing motions longitudinally and laterally to approximate real driving conditions, they employed Bayesian model averaging to enhance prediction accuracy. Gao et al. [9] introduced an interactive multiple model (IMM) for short-term and long-term trajectory prediction in typical road traffic scenarios for intelligent vehicles, implementing manoeuvre recognition through hidden Markov models to improve accuracy. Zhang et al. [10] utilised the Kalman filtering algorithm for trajectory prediction and its error analysis within the context of trajectory prediction, demonstrating the suitability of Kalman filtering for such research. Zhang utilised the Kalman filter and machine learning algorithms to track the dynamic behaviour of neutrophils, providing valuable reference for trajectory prediction [11]. With time, researchers have studied deep learning methods.

2.2 Deep learning methods

Trajectory prediction methods based on deep learning have become one of the mainstream prediction approaches currently. These methods utilise complex data structures and algorithms to simulate and predict vehicle motion trajectories in various environments, especially in complex traffic intersections and other intricate settings. LSTM networks have been extensively studied and applied in trajectory prediction. With their unique gating mechanism, they are particularly suited for handling long-term dependencies, becoming an essential tool in the field of trajectory prediction [12-14]. Against this backdrop, Yao et al. [15] developed a vehicle trajectory prediction network structure based on LSTM neural networks, focusing on frequently changing vehicle operational environments at intersections and better predicting dynamically changing environments. Li et al. [16] proposed a vehicle trajectory prediction model based on clustered convolutional LSTM (CC-LSTM). This model employs a fuzzy clustering method to cluster similar trajectories of surrounding vehicles and density clustering to classify historical trajectory features, identifying similarities during segmentation phases, thereby extracting spatial features of target vehicle trajectories for prediction. Ip et al. [17] introduced an LSTM encoder-decoder model, which also adopted an attention mechanism to manage the significance of the driving stream of target vehicles and adjacent vehicles for trajectory prediction. Yu et al. [18] focused on predicting future vehicle trajectories based on current and past vehicle positions. They proposed a prediction scheme that combines LSTM with recurrent neural networks (RNN), aiding drivers in decision-making by accurately predicting surrounding vehicle trajectories. Gao et al. [19] discussed a driving behaviour intention recognition module using LSTM and an anticipated trajectory prediction module. The former predicts the probability of behaviours such as lane keeping and changing, while the latter employs an encoder-decoder structure with a mixture density network (MDN), predicting future trajectory distributions based on context vectors and driving intentions. This approach significantly improved trajectory prediction accuracy, especially for long-term predictions. Chen et al. [20] proposed a knowledge graph convolutional network long short-term memory (KGCN-LSTM) model to enhance the accuracy and robustness of trajectory prediction. Utilising graph convolutional networks (GCN), points of interest (POI) information is considered as prior knowledge of trajectories, which is significant for optimising urban traffic management and planning. Although the LSTM model performs well in trajectory prediction, researchers are still looking for other algorithms, such as the transformer model.

Since the introduction of the transformer model in 2017, its influence in the field of natural language processing (NLP) has grown rapidly [21]. Its applications have been rapidly expanding across various domains, including vehicle trajectory prediction [22]. The transformer model, through its self-attention mechanism and position encoding, effectively addresses long-distance dependency issues, significantly enhancing the model's capability in processing sequential data. With its successful application in language understanding and generation tasks, researchers began to explore the potential of transformer models in the field of trajectory prediction. Pazho et al. [23] employed a novel graph attention tokenisation (GAT) to capture social interactions among vehicles. The transformer predictor module overcomes the limitations of RNNs and CNNs in handling time-series data, focusing particularly on capturing long-range dependencies. Quintanar et al. [24] used transformer networks with augmented information for trajectory prediction in urban settings. This method is evaluated using metrics like mean average displacement error (MAD/ADE) and final average displacement (FAD/FDE) and shows its unique advantages, especially when dealing with complex scenes such as intersections and roundabouts in urban environments. Geng et al. [25] integrated physical knowledge learning into the transformer model, specifically designed for highway scenarios. This model aims to improve the accuracy and reliability of vehicle trajectory prediction on highways, taking into account physical constraints and interactions. Wang et al. [26] proposed alternative mechanisms instead of commonly used graph convolutional networks (GCN), significantly reducing time costs while maintaining prediction accuracy. Xu et al. [27] utilised transformers and the pNEUMA dataset for predicting vehicle trajectories in urban traffic, showing good performance due to the self-attention mechanism's ability to identify input dependencies. Chen et al. [28] introduced a stochastic non-autoregressive transformer model with multimodal prediction capabilities, where each future trajectory can be inferred in parallel, enhancing the prediction processing speed. Yang et al. [29] proposed a trajectory prediction network with an enhanced graph transformer (TP-EGT) to predict future trajectories of traffic agents, introducing a conflict-aware graph transformer to capture complex social interactions among traffic agents, effectively manage the high dynamics and uncertainty in these scenarios. Gao et al. [30] presented a dual transformer model, comprising a lane-changing intention prediction model and a trajectory prediction model. The anticipated probability vectors are fused, thus obtaining prior knowledge.

The literature introduces a comprehensive range of trajectory prediction methods, particularly those employing deep learning techniques. Despite the plethora of trajectory prediction algorithms proposed over the past decade, the number specifically designed for container port environments is limited. There is still a need for application-focused trajectory prediction algorithms for container trucks in port settings. In the following section, the method used in this article will be introduced in detail.

3. METHODOLOGY

When facing complex time series prediction tasks, traditional single models often struggle to balance the needs of both short-term and long-term forecasting. Based on this, a hypothesis is proposed in this paper: combining the agility of the improved DeepPBM-M model, which is specifically modified and developed in this study based on the DeepPBM-Attention model, in short-term prediction with the efficiency of the PatchTST model in long-term forecasting, can we build a hybrid model framework? To explore this hypothesis, this paper first introduces two core models – DeepPBM-M and PatchTST.

3.1 DeepPBM-M model overview

DeepPBM-M is an enhanced version of the DeepPBM-Attention model, which was initially proposed by Ye et al. [31]. DeepPBM-M, proposed in this paper, is specifically developed to improve upon the original model. The original DeepPBM-Attention model used a three-layer LSTM with a simple additive attention mechanism to predict the single-step trajectory of port container trucks, demonstrating significantly superior performance compared to other LSTM-based models. Despite its effectiveness, it had several shortcomings, including the inability to perform multi-step predictions, slow convergence, limited interpretability of attention weights, susceptibility to overfitting due to its complex structure and lack of robustness when handling noisy or incomplete data.

To better adapt to the requirements of hybrid modelling and enhance the model's flexibility, this paper proposes an improved version, DeepPBM-M, which is specifically developed and modified in this study. An overview of the DeepPBM-M model is illustrated in *Figure 1*, and several key modifications have been made to enhance the model's input-output structure:

Multi-head attention mechanism: In the original model, a simple additive attention mechanism was employed. In this study, it has been adapted into a multi-head attention mechanism suitable for trajectory prediction. Given an input $X \in \mathbb{R}^{n\times d}$, where n is the sequence length and d is the feature dimension, for each head i (out of a total of h heads), define weights $W_i \in \mathbb{R}^{d\times 1}$ and biases $b_i \in \mathbb{R}^{n\times 1}$. Multi-head attention allows the model to simultaneously focus on multiple features of the sequence from different perspectives. This enhances the model's expressiveness and its capability to handle complexity. In the following, the steps involved in implementing this mechanism will be detailed.

1) First, for each head, an attention score e_i is computed by the following operation:

$$e_i = tanh(XW_i + b_i) \tag{1}$$

XW_i is the product of the input and the weights, to which the bias b_i is added, followed by the application of the tanh activation function. The attention score $e_i \in \mathbb{R}^{n\times 1}$ represents the importance of each input element for that head.

2) Then, the attention scores e_i for each head are passed through the Softmax function to obtain attention weights a_i:

$$a_i = Softmax(e_i) \tag{2}$$

Softmax ensures that the sum of weights in each sequence equals 1, representing the relative importance of each element.

3) Each head computes a weighted representation of the input, where each input element is multiplied by its corresponding attention weight:

$$o_i = \sum(X \odot a_i) \tag{3}$$

$\odot$ denotes element-wise multiplication and $o_i \in \mathbb{R}^d$is the weighted sum result computed by that head.

4) Finally, the outputs o_i from all heads are concatenated together to form the final multi-head attention output:

$$O = \ Concatenate\,(o_1, o_2, \ldots, o_h) \tag{4}$$

$X \in \mathbb{R}^{n \times d}$is the final output, containing information from all heads.

Multi-step prediction capability: The original DeepPBM-Attention model was limited to single-step forecasting. This paper enhances the base model by incorporating a fourth LSTM layer. This additional LSTM layer is tasked with maintaining and conveying context information, ensuring that each prediction is predicated on the preceding step's outcomes. Consequently, this modification enables the model to forecast temporal trends over extended horizons with greater accuracy.

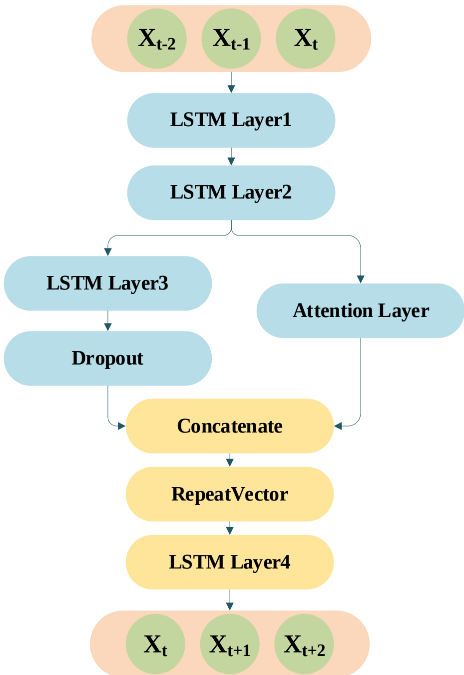

Figure 1 – DeepPBM-M model overview

3.2 PatchTST model overview

Unlike traditional LSTMs, the transformer relies entirely on the self-attention mechanism to capture global dependencies between sequences, thus efficiently processing long sequence data. The transformer network for the hybrid model adopted the PatchTST, an efficiently designed transformer model proposed by Nie et al. [32]. Each time series is divided into overlapping or separate patches of length P, with S as the interval, reducing input tokens from L to $L\,/\,P$. This segmentation significantly reduces computational and memory requirements for attention mechanisms, enabling efficient long-sequence processing and improving forecasting accuracy.

The neural network structure of PatchTST is shown in *Figure 2*. Currently, PatchTST is one of the highest-performing models under the transformer-based architecture for handling complex forecasting problems, as it can capture long-range dependencies in multivariate time series forecasting tasks. This architecture has been proven to be highly effective. However, despite PatchTST's advantages in long-term forecasting, it has certain limitations in short-term forecasting, which may be due to the transformer architecture being less optimised for capturing finer-grained variations over shorter time intervals.

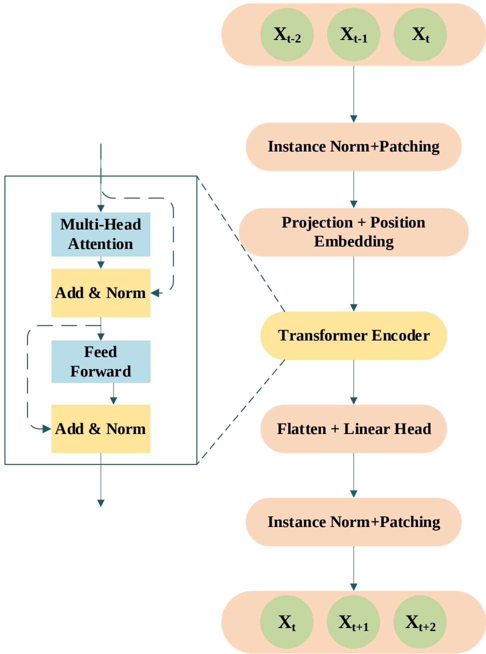

Figure 2 – PatchTST model overview

3.3 Model performance turning point

To systematically evaluate and compare the predictive capabilities of the DeepPBM-M and PatchTST models across varying forecast horizons, we conducted a comprehensive empirical analysis. *Table 1* presents five key evaluation metrics for both models at different prediction steps:

1) Mean absolute error (MAE): Measures the average magnitude of prediction errors.
2) Mean squared error (MSE): Emphasises larger prediction errors by squaring the differences.
3) Mean absolute percentage error (MAPE): Indicates the percentage deviation from actual values.
4) Mean squared percentage error (MSPE): Highlights percentage deviations with greater sensitivity to large errors.
5) Relative squared error (RSE): Provides a normalised measure of prediction accuracy relative to a baseline.

These metrics were chosen to provide a multi-faceted assessment of model performance, capturing both absolute and relative prediction accuracy. The analysis spans prediction horizons from 1 to 96 steps, enabling the evaluation of both short-term and long-term forecasting capabilities.

Table 1 – Evaluation metrics across different steps

Steps	MAE		MSE		MAPE		MSPE		RSE	
	DeepPBM-M	**PatchTST**	**DeepPBM-M**	**PatchTST**	**DeepPBM-M**	**PatchTST**	**DeepPBM-M**	**PatchTST**	**DeepPBM-M**	**PatchTST**
1	***0.029***	0.050	***0.001***	0.007	***0.010***	0.013	***0.005***	0.006	***0.015***	0.039
2	***0.037***	0.052	***0.002***	0.008	***0.012***	0.013	***0.006***	0.007	***0.022***	0.041
4	***0.048***	0.062	***0.004***	0.011	***0.013***	0.015	***0.008***	0.009	***0.029***	0.050
8	***0.067***	0.073	***0.007***	0.016	0.019	***0.018***	***0.013***	0.013	***0.040***	0.059
16	***0.094***	0.096	***0.015***	0.028	0.030	***0.024***	0.044	***0.022***	***0.067***	0.070
32	0.147	***0.120***	0.050	***0.045***	0.041	***0.029***	0.081	***0.038***	0.107	***0.100***
48	0.169	***0.158***	0.062	***0.055***	0.057	***0.038***	0.110	***0.072***	0.138	***0.119***
64	0.224	***0.195***	0.123	***0.119***	0.071	***0.047***	0.138	***0.089***	0.180	***0.151***
80	0.260	***0.229***	0.173	***0.146***	0.076	***0.054***	0.140	***0.118***	0.190	***0.190***
96	0.309	***0.263***	0.232	***0.201***	0.078	***0.063***	0.164	***0.152***	0.224	***0.219***

The experimental results indicate that a significant performance turning point occurs between steps 8 and 16. Before step 16, the DeepPBM-M model outperforms the PatchTST model in all evaluation metrics, whereas after step 16, the PatchTST model gradually surpasses the DeepPBM-M model. This finding provides important guidance for further research.

Based on these preliminary experimental results, this paper proposes the CT-HybridNet model, which aims to combine the strengths of both DeepPBM-M and PatchTST models in order to maintain high prediction accuracy across various prediction horizons.

3.4 CT-HybridNet model architecture

In the preliminary study on container truck trajectory prediction at ports, we observed that LSTM models excel at capturing short-term dependencies, showing outstanding performance in short-term prediction tasks, while transformer models, with their ability to capture long-term dependencies, perform better in long-term prediction tasks. Based on this observation, we hypothesise that LSTM and transformer models have unique advantages in short-term and long-term prediction, respectively. The previous section compared the performance of the LSTM-based DeepPBM-M and the transformer-based PatchTST across different prediction lengths, confirming the existence of performance turning points.

To effectively combine these models and ensure the validity of our hybrid approach, we first conducted a comprehensive residual analysis to validate the normality assumption of the model errors. This analysis is crucial as it provides the theoretical foundation for our subsequent model fusion strategy.

Our residual analysis examined the prediction residuals of both base models across longitude and latitude dimensions. As shown in *Figure 3*, the analysis reveals strong evidence supporting the normality assumption. This normality assumption is further supported by the central limit theorem, as the prediction errors represent the aggregation of multiple independent factors, and the input data has been normalised during preprocessing. For the DeepPBM-M model, the residual distribution in longitude demonstrates clear normal characteristics with mean μ=-0.00108770 and standard deviation σ=0.00326428. The residuals' normality is validated by both statistical measures and graphical analysis: the skewness coefficient of -0.129 (approaching zero) confirms distribution symmetry, while the kurtosis coefficient of 2.834 (close to the theoretical value of 3) supports the normal distribution hypothesis. Similar normal patterns are observed in latitude residuals, with μ=0.00238242, σ=0.00193726, skewness of -0.214 and kurtosis of 2.996.

The residual analysis of the PatchTST model yields comparable results, with both longitude and latitude errors following normal distributions. The longitude residuals show μ=-0.00217192, σ=0.02116637, skewness of -0.058 and kurtosis of 3.460, while latitude residuals exhibit μ=-0.00189542, σ=0.02330102, skewness of -0.054 and kurtosis of 3.208. The histograms of residuals overlaid with theoretical normal distribution curves, as shown in *Figure 3,* provide visual confirmation of these statistical findings.

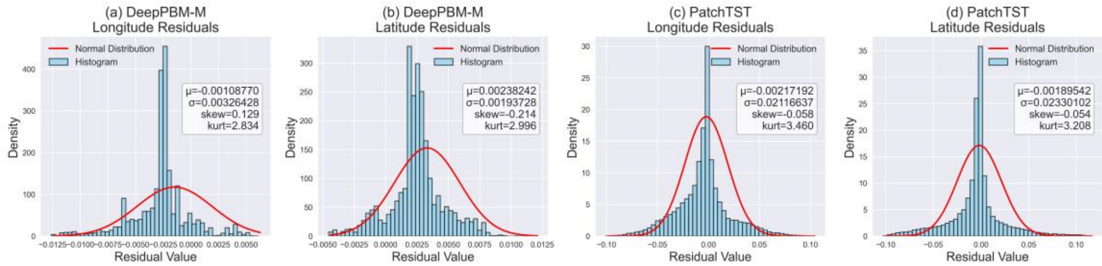

Figure 3 – Residual normality analysis of base models: a) DeepPBM-M longitude residuals; b) DeepPBM-M latitude residuals; c) PatchTST longitude residuals; d) PatchTST latitude residuals

Based on this rigorous residual analysis confirming the normality of prediction errors, we proceeded to develop the mathematical framework for CT-HybridNet. Specifically, this study establishes the prediction error of the DeepPBM-M model at step length l as a random variable P_1, which follows a normal distribution with a mean of $m_{1,l}$ and variance of $\delta_{1,l}^2$, denoted as $P_{1,l} \sim \mathcal{N}\left(m_{1,l}, \delta_{1,l}^2\right)$. Similarly, the error of the PatchTST model at the same step length l follows a normal distribution with a mean of $m_{2,l}$ and variance of $\delta_{2,l}^2$, denoted as $P_{2,l} \sim \mathcal{N}\left(m_{2,l}, \delta_{2,l}^2\right)$. Given the statistical independence between the prediction errors of the DeepPBM-M model and the PatchTST model, the CT-HybridNet model derives a hybrid prediction error by weighting the errors of these two models with a weight parameter α_l, The hybrid prediction error $P_{h,l}$, also follows a normal distribution, denoted as $\alpha_l P_{1,l} + (1-\alpha_l)P_{2,l} \sim \mathcal{N}\left(m_{h,l}, \delta_{h,l}^2\right)$, with its mean, $m_{h,l}$ and variance, $\delta_{h,l}^2$, given by the following equations:

$$m_{h,l} = \alpha_l m_{1,l} + (1-\alpha_l)m_{2,l} \tag{5}$$

$$\delta_{h,l}^2 = \alpha_l^2 \delta_{1,l}^2 + (1-\alpha_l)^2 \delta_{2,l}^2 \tag{6}$$

Given that $m_{1,l}$ and $m_{2,l}$ are both assumed to be 0, the mean of the hybrid prediction error, $m_{h,l}$, is also 0. On this basis, a constraint condition is introduced to ensure that the total variance of the hybrid model does not exceed the variance of any single model:

$$\alpha_l \delta_{1,l}^2 + (1-\alpha_l)\delta_{2,l}^2 \leq min\left(\delta_{1,l}^2, \delta_{2,l}^2\right) \tag{7}$$

In the subsequent derivation, by considering $\delta_{1,l}^2$ as the smaller variance, it is possible to obtain:

$$\frac{1-\alpha_l}{1+\alpha_l} \geq \frac{\delta_{1,l}^2}{\delta_{2,l}^2} \tag{8}$$

Based on this, the range of constraint conditions for α_l can be determined as:

$$\alpha_l \geq \frac{1-\frac{\delta_{1,l}^2}{\delta_{2,l}^2}}{1+\frac{\delta_{1,l}^2}{\delta_{2,l}^2}} \tag{9}$$

Based on these theoretical foundations and constraints established in equations (7)-(9), we formulate the adaptive selection of α_l as a nonlinear constrained optimisation problem, solved using sequential least squares programming (SLSQP). This optimisation method is particularly suitable for handling the nonlinear constraint while maintaining numerical stability. The SLSQP algorithm solves this constrained optimisation problem through iterative approximation. At each iteration k, the algorithm computes:

$$minimize\ \nabla f(\alpha_l{}^k)^{\mathrm{T}} d + \frac{1}{2} d^{\mathrm{T}} B^k d \tag{10}$$

where d is the search direction, B^k is the approximation of the objective function's Hessian matrix. The equality constraint function is defined as:

$$ceq(\alpha_l) = \alpha_{l,1} + \alpha_{l,2} - 1 = 0 \tag{11}$$

The gradients of the objective function and constraints are, respectively:

$$\nabla f(\alpha_l) = [2\alpha_{l,1}\sigma_{l,1}^2, 2\alpha_{l,2}\sigma_{l,2}^2]^{\mathrm{T}} \tag{12}$$

$$\nabla ceq(\alpha_l) = [1, 1]^{\mathrm{T}} \tag{13}$$

The algorithm iteratively updates the search direction and step size to approach the optimal solution:

$$\alpha_l^{k+1} = \alpha_l^k + \lambda^k d^k \tag{14}$$

where λ^k is the step size determined by line search. The iteration continues until the convergence condition is satisfied:

$$\left\|\nabla L(\alpha^k, \mu^k, \lambda^k)\right\|^2 \leq \varepsilon \tag{15}$$

where $L(\alpha_l, \mu, \lambda) = f(\alpha_l) - \mu^{\mathrm{T}} ceq(\alpha_l) - \lambda^{\mathrm{T}} c(\alpha_l)$ is the Lagrangian function.

In the process of hybridising multiple step sizes, the CT-HybridNet model fully leverages the advantages of DeepPBM-M and PatchTST in their respective areas of predictive strength. Further, in the CT-HybridNet model, the scenario of mixing more models is also considered. To ensure the stability and reliability of the system's predictions when mixing multiple models, the Lyapunov energy function is introduced to assess the energy or stability of the hybrid model at each step. This energy function is formulated as follows:

$$V_l = \frac{1}{2} E(||P_{h,l}||^2) > 0 \tag{16}$$

The Lyapunov energy function V_l represents half the expected value of the square of the hybrid model error $P_{h,l}$ at step length l. This expectation emphasises that V_l assesses the expected system energy, crucial for stability analysis. The function's non-negativity ensures that the system's "energy" does not increase indefinitely, avoiding instability. To specifically monitor the hybrid model's stability, we calculate the change in energy ΔV_l, defined step by step as:

$$\Delta V_l = \frac{1}{2}E(\|P_{h,l}\|^2) - \frac{1}{2}E(\|P_{3,l}\|^2) = \frac{1}{2}\delta_{h,l}^2 - \frac{1}{2}\delta_{3,l}^2 < 0 \tag{17}$$

where $\delta_{h,l}^2$ and $\delta_{3,l}^2$ represent the variances of the hybrid model errors at step length l and after the integration of the newly introduced sub-model, respectively. These variances are part of the expectations used to calculate the Lyapunov function and its change. By ensuring that ΔV_l remains negative, i.e. the expected energy at the current step is less than that of the previous step, we can continuously reduce the uncertainty of the system's predictions, thereby enhancing the system's stability. This approach provides a powerful mathematical tool for stability analysis, ensuring that even when multiple models are introduced for prediction, the overall stability and accuracy of the system's predictions are maintained. By continuously monitoring and adjusting the model weights, CT-HybridNet can effectively utilise the strengths of each model, optimising the accuracy of long-term and short-term predictions. The neural network structure of CT-HybridNet is shown in *Figure 4*. In the subsequent sections, the performance evaluation of CT-HybridNet will be elaborated in detail through the experimental results. The following sections will thoroughly introduce the related experimental settings, evaluation metrics and result analysis.

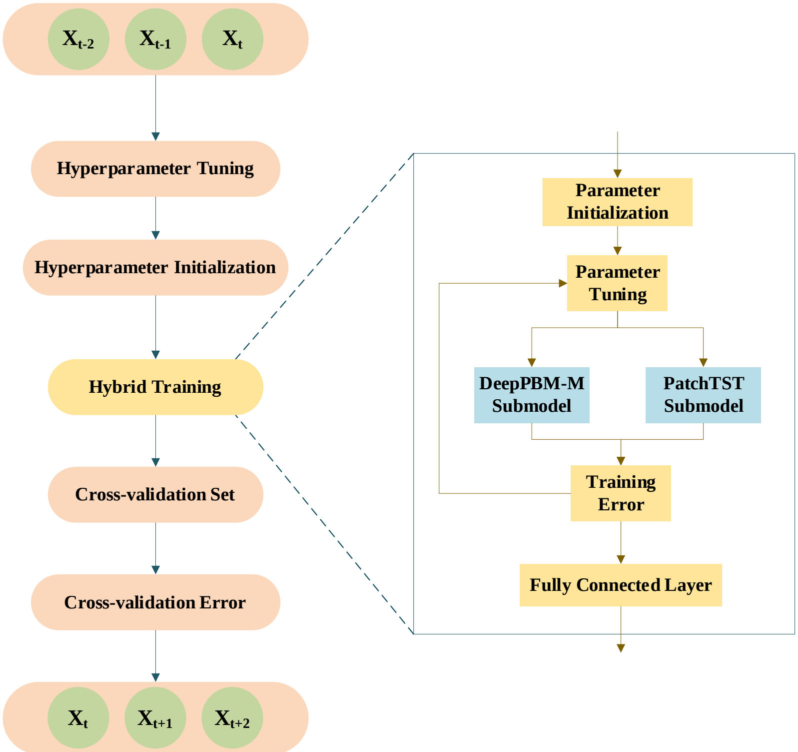

Figure 4 – CT-HybridNet model overview

4. EXPERIMENTS AND RESULTS

The empirical validation of the CT-HybridNet model's performance necessitates a comprehensive experimental framework encompassing both quantitative metrics and qualitative analysis. Through systematic evaluation across multiple prediction horizons and diverse operational scenarios, we aim to demonstrate the model's efficacy in addressing the challenges of port container truck trajectory prediction. The following sections detail our experimental methodology and present the analytical findings.

4.1 Dataset description and data preprocessing

In this section, this study conducts experiments using comprehensive real-world data collected from the terminal. Against the backdrop of port area automation renovation, high-precision positioning devices were installed on all 150 internal container trucks, and their location trajectory data were continuously collected over a month. The positioning devices employed in this study were industrial-grade GNSS receivers, featuring positioning accuracy of 1 cm (CEP) under open sky conditions and an update rate of 5 Hz. These devices recorded latitude and longitude with precision at 7 decimal places, supported multi-constellation reception (GPS, BeiDou), and provided velocity measurements accurate to 0.01 m/s RMS with heading accuracy of 0.1 degrees RMS.

To enhance the performance of predictive models, this study implemented a series of sophisticated data preprocessing steps centred around the temporal and spatial data collected from positioning devices onboard yard trucks within the port. These steps were designed to optimise the efficiency of model training and the accuracy of predictions while minimising computational complexity and training costs. The specific processes included data collection, sorting and feature extraction encompassing timestamps (UTC, millisecond precision), latitude and longitude coordinates, positioning quality indicators, instantaneous speed and heading direction. The preprocessing pipeline incorporated rigorous filtering criteria, removing positions with HDOP exceeding 4.0, data points with fewer than 6 satellites, and physically impossible accelerations greater than 2.5 m/s^2. A Kalman filter was applied to smooth trajectory data and eliminate anomalous readings.

To accommodate the needs of supervised learning, the GNSS data were systematically segmented by time intervals and converted into a supervised learning format. This conversion process facilitates the model's learning of the relationship between input features and predictive outputs, thereby enhancing the accuracy of predictions. To further facilitate efficient model training, the data were normalised to a [0, 1] scale and standardised to ensure that different feature values were comparable, which helped improve the training speed and model convergence. After preprocessing, the dataset, consisting of approximately 10 million trajectory data points, was split into training and validation sets with a 90/10 ratio. The training set was used to fit the model, while the validation set served to evaluate its performance and ensure its generalisation capabilities. All data preprocessing steps were conducted using Python 3.8, as shown in *Figure 5*.

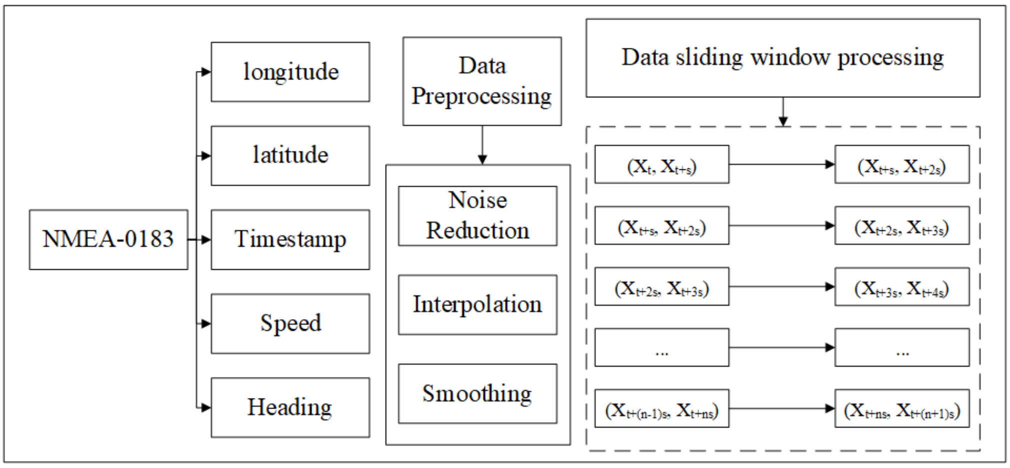

Figure 5 – Data preprocessing workflow for the dataset

After dividing the dataset, special attention was given to the representativeness of the validation set to ensure the model's generalisation capabilities. The validation data were carefully selected to encompass not only common scenarios but also various edge cases within port operations, including regular operation patterns during different times of day, diverse weather conditions affecting GNSS performance, various operational areas within the port, different loading states of container trucks and multiple driving patterns and manoeuvres. As shown in *Figure 6*, the trajectory data in the validation set demonstrate the distribution across various time frames and geographic locations, capturing a wide range of port operation conditions. This comprehensive coverage ensures that the validation process effectively assesses the model's performance across the full spectrum of operational scenarios encountered in real-world port environments.

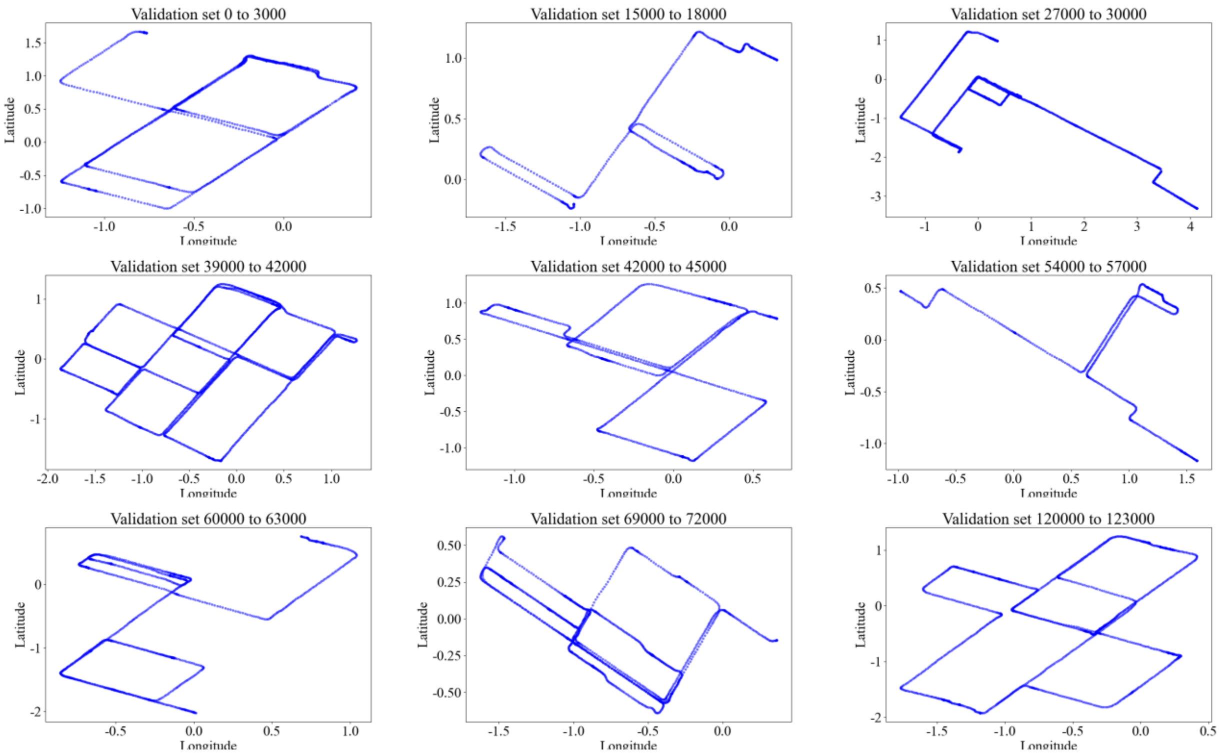

Figure 6 – Validation set trajectory distribution

4.2 Experimental setup

This study finely tuned the hyperparameters of the DeepPBM-M model, the PatchTST model and the CT-HybridNet hybrid model to optimise model performance. The specific hyperparameter settings are shown in *Table 2*. The experiments extensively validated the effectiveness of these models in the task of predicting the trajectories of internal container trucks within the port.

The next section will provide a detailed experimental evaluation and comparative analysis of the models designed above, thereby verifying each model's predictive performance and practicality.

Table 2 – Hyperparameter settings

Models	**The number of training set**	**The number of validation set**	**Number of neurons in h_{LSTM1}**	**Number of neurons in h_{LSTM2}**	**Number of neurons in h_{LSTM3}**	**optimiser**	**epochs**	**batch_size**
DeepPBM-M	8900,000	810000	128	256	256	Adam	200	3600
PatchTST	8900,000	810000	N/A	N/A	N/A	Adam	100	3600
CT-HybridNet	8900,000	810000	128	256	256	Adam	200	3600

4.3 Model training time

The training time is a key metric for evaluating the efficiency of models in the development and optimisation process. By systematically comparing the training duration of various models, it is possible not only to reveal their efficiency differences in practical applications but also to gain insights into their performance under large-scale data scenarios.

To comprehensively evaluate the training efficiency of each model, this study conducted a detailed record and analysis of the training times for DeepPBM-M, PatchTST and CT-HybridNet under identical datasets and computing environments. All models were trained on a Windows 11 system equipped with an NVIDIA RTX 4090 GPU. The Python version used was 3.8, and PyTorch was employed as the deep learning framework. The training times, early stopping rounds and MSE results are shown in *Table 3*.

Table 3 – Model training time comparison

Models	Training time [min]	Early stopping rounds	MSE
DeepPBM-M	129.1048648	72	0.003290
PatchTST	215.1638892	168	0.003755
CT-HybridNet	225.2612862	166	***0.002797***

From *Table 3*, it can be observed that DeepPBM-M achieved the fastest training time, requiring 129.10 minutes and stopping after 72 rounds with an MSE of 0.003290. This demonstrates a relatively efficient training process compared to the other models. On the other hand, PatchTST and CT-HybridNet had significantly longer training times, 215.16 minutes and 225.26 minutes, respectively. Notably, CT-HybridNet had the lowest MSE value of 0.002797, suggesting better accuracy in predictions.

4.4 Model prediction time

In addition to training time, prediction time is another critical factor for evaluating model performance, especially in real-world applications where quick inference is often essential. By comparing the prediction time of different models, we can better understand their efficiency during the inference phase.

To thoroughly assess the prediction efficiency, this study measured the prediction times for DeepPBM-M, PatchTST and CT-HybridNet under identical validation conditions. The models were evaluated using 62,000 validation sets across different prediction steps (1, 8 and 96). The prediction times were recorded in a consistent computing environment (Windows 11 system with NVIDIA RTX 4090 GPU, Python 3.8, PyTorch). The results are summarised in *Table 4*.

Table 4 – Model prediction time comparison

Models	Number of validation sets	Steps	Prediction time [s]
DeepPBM-M	62,000	1	20.6088280
PatchTST			***8.8169904***
CT-HybridNet			21.3077526
DeepPBM-M	62,000	8	***58.1089582***
PatchTST			70.6860214
CT-HybridNet			66.9997851
DeepPBM-M	62,000	96	***452.2246989***
PatchTST			562.0177092
CT-HybridNet			542.004211

Table 4 presents a detailed breakdown of the prediction times for each model across different step sizes. The prediction time indicates the duration each model took to generate predictions for the given number of validation sets. It can be observed that CT-HybridNet demonstrated competitive performance, particularly for larger step sizes. For 8 steps, CT-HybridNet had a prediction time of 66.9997851 seconds, which was faster than PatchTST and only slightly longer than DeepPBM-M, indicating an efficient inference capability while maintaining high accuracy.

For the most computationally intensive scenario with 96 steps, CT-HybridNet required 542.004211 seconds, which was comparable to PatchTST and DeepPBM-M. Importantly, despite being a hybrid model, CT-HybridNet did not exhibit increased prediction time or complexity compared to other models. This shows that the model's hybrid nature effectively integrates different components without compromising its efficiency, which is a significant advantage over traditional single-architecture models.

4.5 Evaluation of prediction performance

In this study, multiple evaluation metrics were employed to comprehensively assess the performance of the DeepPBM-M model, the PatchTST model and the CT-HybridNet model proposed in this research, in the task of predicting the trajectories of port container trucks. These evaluation metrics include MAE, MSE, MAPE, MSPE and RSE. These indicators thoroughly reflect the accuracy and practicality of the model predictions.

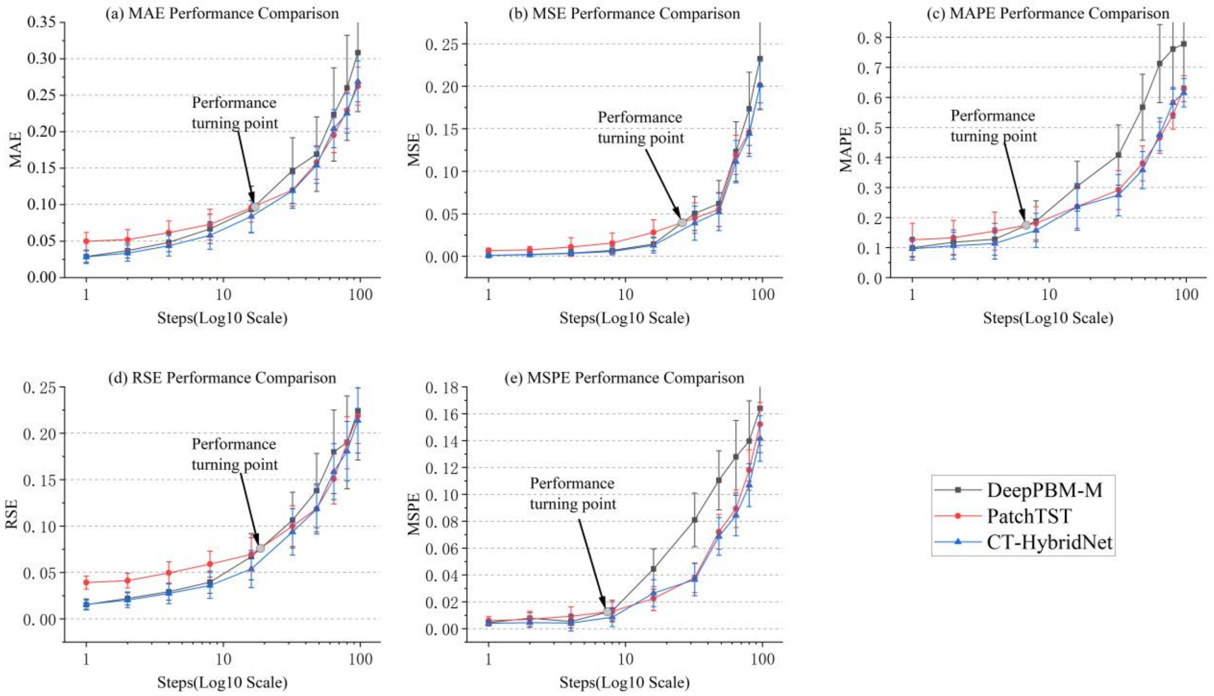

Figure 7 – Evaluation metrics for DeepPBM-M, PatchTST and CT-HybridNet: a) MAE performance comparison; b) MSE performance comparison; c) MAPE performance comparison; d) RSE performance comparison; e) MSPE performance comparison

Figure 7 highlights the performance turning points of both DeepPBM-M and PatchTST in multi-step prediction, particularly between steps 8 and 16. These turning points are crucial as they mark significant changes in model performance, with DeepPBM-M showing reduced accuracy and PatchTST beginning to excel. Using the validation set described in Section 4.1, which consists of 810,000 trajectory data points carefully selected to encompass various operational scenarios, we performed nine independent evaluations and included error bars representing mean ± standard deviation for each evaluation metric. The error bars demonstrate the consistency and stability of our results across different test scenarios. Notably, the CT-HybridNet model shows smaller error bars compared to both base models, particularly in the critical transition period (steps 8-16), indicating more stable and reliable predictions. The reduced standard deviations of CT-HybridNet (average of ±0.022 for MAE compared to ±0.032 for DeepPBM-M and ±0.023 for PatchTST) suggest that our hybrid approach not only improves accuracy but also reduces prediction variability. This enhanced stability is particularly evident in long-term predictions (beyond step 32), where the CT-HybridNet maintains consistently smaller standard deviations despite the increasing prediction horizon. *Figure 7* clearly shows these transitions and stability patterns, providing visual support to the detailed metrics in *Table 5*, which captures how the performance metrics evolve across different prediction steps.

Table 5 – Model evaluation metrics across different steps

Model	Steps	MAE	MSE	MAPE	MSPE	RSE
DeepPBM-M	1	0.029	0.001	0.010	0.005	***0.015***
PatchTST		0.050	0.007	0.013	0.006	0.039
CT-HybridNet		***0.028***	***0.001***	***0.010***	***0.004***	0.016

Model	Steps	MAE	MSE	MAPE	MSPE	RSE
DeepPBM-M	2	0.037	0.002	0.012	0.006	0.022
PatchTST		0.052	0.008	0.013	0.007	0.041
CT-HybridNet		***0.033***	***0.002***	***0.011***	***0.006***	***0.020***
DeepPBM-M	4	0.048	0.004	0.013	0.008	0.029
PatchTST		0.062	0.011	0.015	0.009	0.050
CT-HybridNet		***0.044***	***0.003***	***0.011***	***0.004***	***0.028***
DeepPBM-M	8	0.067	0.007	0.019	0.013	0.040
PatchTST		0.073	0.016	0.018	0.013	0.059
CT-HybridNet		***0.058***	***0.006***	***0.016***	***0.009***	***0.036***
DeepPBM-M	16	0.094	0.015	0.030	0.044	0.067
PatchTST		0.096	0.028	***0.024***	***0.022***	0.070
CT-HybridNet		***0.084***	***0.013***	0.026	0.026	***0.054***
DeepPBM-M	32	0.147	0.050	0.041	0.081	0.107
PatchTST		0.120	0.045	0.029	0.038	0.100
CT-HybridNet		***0.119***	***0.039***	***0.027***	***0.037***	***0.094***
DeepPBM-M	48	0.169	0.062	0.057	0.110	0.138
PatchTST		0.158	0.055	0.038	0.072	0.119
CT-HybridNet		***0.154***	***0.052***	***0.036***	***0.069***	***0.118***
DeepPBM-M	64	0.224	0.123	0.071	0.138	0.180
PatchTST		***0.195***	0.119	***0.047***	0.089	***0.151***
CT-HybridNet		0.204	***0.111***	0.048	***0.084***	0.159
DeepPBM-M	80	0.260	0.173	0.076	0.140	0.190
PatchTST		0.229	0.146	***0.054***	0.118	0.190
CT-HybridNet		***0.225***	***0.144***	0.058	***0.107***	***0.181***
DeepPBM-M	96	0.309	0.232	0.078	0.164	0.224
PatchTST		***0.263***	***0.201***	0.063	0.152	0.219
CT-HybridNet		0.269	0.202	***0.062***	***0.142***	***0.214***

As shown in the results of *Table 5*, the DeepPBM-M and PatchTST models exhibit a significant performance turning point in multi-step prediction of port vehicle trajectories in the prediction range of 8 to 16 steps. The CT-HybridNet model proposed in this paper adopts a weighted fusion strategy, which takes full advantage of DeepPBM-M's ability to capture temporal dependencies in short-term prediction, as well as PatchTST's superior performance in long-term prediction of long sequence data. It performs well not only in short-term prediction but also shows significant performance advantages in long-term prediction.

As can be seen from the table, the DeepPBM-M model performs well in short-term prediction, especially in the 1 to 4 step prediction range, where its MAE and MSE are significantly lower than those of the PatchTST model. For example, for 1-step prediction, the MAE is 0.029 and the MSE is 0.001, demonstrating the advantage of LSTM in capturing short-term dependencies. In contrast, PatchTST performs slightly worse in short-term prediction, such as in the 4-step prediction, where the MAE is 0.062 and the MSE is 0.011. The transformer architecture is not as sensitive as LSTM in capturing local features, resulting in lower accuracy in short-term prediction compared to DeepPBM-M.

In long-term prediction, the PatchTST model exhibits superior performance. For example, in the 16-step prediction, the MAE of PatchTST is 0.096 and the MSE is 0.028, which is about 34.7% better than the MAE of 0.147 for DeepPBM-M. This performance improvement is attributed to the self-attention mechanism of the transformer, which allows PatchTST to more effectively capture long-term dependencies. DeepPBM-M's performance gradually declines in long-term prediction, with an MAE of 0.147 and MSE of 0.050 in the 32-step prediction, indicating limitations in LSTM's ability to capture long-term dependencies, leading to cumulative prediction errors.

CT-HybridNet achieves dual performance improvement in both short-term and long-term predictions by adaptively fusing the prediction results of DeepPBM-M and PatchTST. In short-term prediction, for example, in the 1-step prediction, CT-HybridNet's MAE is 0.028, close to and slightly better than DeepPBM-M, and compared to PatchTST's 0.050, the MAE is reduced by about 44%. In long-term prediction, such as in the 32-step prediction, the MAE of CT-HybridNet is 0.119, which is lower than DeepPBM-M's 0.147, representing an improvement of about 20%. The adaptive fusion strategy enables CT-HybridNet to combine the advantages of LSTM and transformer, maintaining high prediction accuracy and stability in both short-term and long-term predictions, especially with a significant accuracy improvement of about 15% in long-term predictions beyond 16 steps.

5. CONCLUSIONS

This study analyses the performance turning points of DeepPBM-M and PatchTST at different prediction steps, introducing an adaptive training mechanism through a weighted fusion strategy to optimise performance before and after these breakpoints. Building on this research, a novel hybrid trajectory prediction model, CT-HybridNet, is proposed. This model integrates the strengths of DeepPBM-M and PatchTST, specifically tailored for multi-step trajectory prediction of port container trucks. Comparative experiments on trajectory prediction technology have verified that CT-HybridNet exhibits superior predictive accuracy and stability compared to single models.

The implementation of CT-HybridNet in real port operations, integrated with the PathSync Collision Avoidance system, has demonstrated significant economic and operational benefits. Based on a six-month pilot deployment at a major container port, the model achieved a 15% reduction in positioning errors during container stacking operations, enabling the central control server to implement more precise safety management. The multi-step prediction data are transmitted to the central control server and integrated with the PathSync Collision Avoidance system, which analyses trajectory overlaps to provide early collision warnings. This integration enables dynamic path planning for manned vehicles and establishes a comprehensive collision prevention network. Through this optimised routing system and reduced correction manoeuvres, waiting times at key logistics nodes decreased by 10%, while fuel consumption dropped by 8%, contributing to lower carbon emissions in port operations. These comprehensive operational improvements collectively resulted in an estimated annual cost saving of $1.2 million for the test port, demonstrating the substantial economic value of implementing the CT-HybridNet model in port operations.

Looking forward, despite the significant progress made with the CT-HybridNet model, it is believed that several directions merit exploration in future research. Firstly, further optimisation of the model structure and parameters, such as the introduction of more advanced attention mechanisms like the multi-layer self-attention structure of transformers, could improve performance when processing more complex datasets. Secondly, considering that different port environments may have unique features and challenges, conducting research on model adjustments and optimisations tailored to specific port environments would be of significant importance. Additionally, with the development of Internet of Things technology and high-precision positioning technology, collecting and integrating more types of data (e.g. weather conditions, traffic flow, types of cargo) into the prediction model could further enhance the accuracy and robustness of predictions.

REFERENCES

[1] Almeida F. Challenges in the digital transformation of ports. *Businesses*. 2023;3(4):548–568. DOI: 10.3390/businesses3040034.

[2] Sislian L, Jaegler A, Cariou P. A literature review on port sustainability and ocean carrier network problem. *Research in Transportation Business & Management*. 2016;19:19–26. DOI: 10.1016/j.rtbm.2016.03.003.

[3] Hua C, et al. Evaluation and governance of green development practice of port: A sea port case of China. *Journal of Cleaner Production*. 2020;249:119434. DOI: 10.1016/j.jclepro.2019.119434.

[4] Wasesa M, Stam A, van Heck E. The seaport service rate prediction system: Using drayage truck trajectory data to predict seaport service rates. *Decision Support Systems*. 2017;95:37–48. DOI: 10.1016/j.dss.2016.12.002.

[5] Singh A. Trajectory-prediction with vision: A survey. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. Paris, France; 1-6 Oct. 2023. p. 3318–3323. DOI: 10.48550/arXiv.2303.13354.

[6] Lin CF, Ulsoy AG, LeBlanc DJ. Vehicle dynamics and external disturbance estimation for vehicle path prediction. *IEEE Transactions on Control Systems Technology*. 2000;8(3):508–518. DOI: 10.1109/87.845881.

[7] Polychronopoulos A, Tsogas M, Amditis AJ, Andreone L. Sensor fusion for predicting vehicles' path for collision avoidance systems. *IEEE Transactions on Intelligent Transportation Systems*. 2007;8(3):549–562. DOI: 10.1109/TITS.2007.903439.

[8] Anderson C, Vasudevan R, Johnson-Roberson M. A kinematic model for trajectory prediction in general highway scenarios. *IEEE Robotics and Automation Letters*. 2021;6(4):6757–6764. DOI: 10.1109/LRA.2021.3094491.

[9] Gao H, et al. An interacting multiple model for trajectory prediction of intelligent vehicles in typical road traffic scenario. *IEEE Transactions on Neural Networks and Learning Systems*. 2021;34(9):6468–6479. DOI: 10.1109/TNNLS.2021.3136866.

[10] Zhang X, Yu W. Research on the application of Kalman filter algorithm in aircraft trajectory analysis. In: *2022 7th International Conference on Intelligent Computing and Signal Processing (ICSP)*. Xi'an, China; 15-17 Apr. 2022. p. 196–199. DOI: 10.1109/ICSP54964.2022.9778746.

[11] Zhang X. *Modelling and Identification of Neutrophil Cell Dynamic Behaviour*. PhD thesis. University of Sheffield; 2016.

[12] Van Houdt G, Mosquera C, Nápoles G. A review on the long short-term memory model. *Artificial Intelligence Review*. 2020;53(8):5929–5955. DOI: 10.1007/S10462-020-09838-1.

[13] Ni R, Guo Z, Jiang Y, Liu S. Research on port truck trajectory completion based on long short-term memory model and speed distribution characteristics. In: *2021 7th International Conference on Hydraulic and Civil Engineering & Smart Water Conservancy and Intelligent Disaster Reduction Forum (ICHCE & SWIDR)*, 7-10 Dec. 2021, Nanjing, China. 2021. p. 1281–1286. DOI: 10.1109/ICHCESWIDR54323.2021.9656235.

[14] Liu S, et al. A spline-LSTM for autonomous truck trajectory prediction based on curve feature extraction. In: *2021 6th International Conference on Transportation Information and Safety (ICTIS)*. Wuhan, China: IEEE; 2021. p. 1280–1285. DOI: 10.1109/ICTIS54573.2021.9798492.

[15] Yao B, et al. LSTM-based vehicle trajectory prediction using UAV aerial data. In: *Proceedings of KES-STS International Symposium*. Rome, Italy: Springer; 2023. p. 13–21. DOI: 10.1007/978-981-99-3284-9.

[16] Li R, Zhong Z, Chai J, Wang J. Autonomous vehicle trajectory combined prediction model based on CC-LSTM. *International Journal of Fuzzy Systems*. 2022;24(8):3798–3811. DOI: 10.1007/s40815-022-01288-x.

[17] Ip A, Irio L, Oliveira R. Vehicle trajectory prediction based on LSTM recurrent neural networks. In: *2021 IEEE 93rd Vehicular Technology Conference (VTC2021-Spring)*. Helsinki, Finland: IEEE; 2021. p. 1–5. DOI: 10.1109/VTC2021-Spring51267.2021.9449038.

[18] Yu D, Lee H, Kim T, Hwang SH. Vehicle trajectory prediction with lane stream attention-based LSTMs and road geometry linearization. *Sensors*. 2021;21(23):8152. DOI: 10.3390/s21238152.

[19] Gao Z, Bao M, Gao F, Tang M. Probabilistic multi-modal expected trajectory prediction based on LSTM for autonomous driving. *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*. 2023. p. 1-12. DOI: 10.1177/09544070231167906.

[20] Chen J, Fan D, Qian X, Mei L. KGCN-LSTM: A graph convolutional network considering knowledge fusion of point of interest for vehicle trajectory prediction. *IET Intelligent Transport Systems*. 2023;17(6):1087–1103. DOI: 10.1049/itr2.12341.

[21] Vaswani A, et al. Attention is all you need. *Advances in Neural Information Processing Systems*. 2017;30:5998–6008. DOI: 10.48550/arXiv.1706.03762.

[22] Chen X, et al. Deep reinforcement learning assisted genetic programming ensemble hyper-heuristics for dynamic scheduling of container port trucks. *IEEE Transactions on Evolutionary Computation*. 2024;28(3):456–472. DOI: 10.1109/TEVC.2024.3381042.

[23] Danesh Pazho A, Katariya V, Alinezhad Noghre G, Tabkhi H. VT-Former: A transformer-based vehicle trajectory prediction approach for intelligent highway transportation systems. *arXiv e-prints*. 2023. arXiv:2311.06623. DOI: 10.48550/arXiv.2311.06623.

[24] Quintanar Á, Izquierdo R, Parra I, Fernández-Llorca D. Goal-oriented transformer to predict context-aware trajectories in urban scenarios. *Engineering Proceedings*. 2023;39(1):57. DOI: 10.3390/engproc2023039057.

[25] Geng M, Li J, Xia Y, Chen XM. A physics-informed transformer model for vehicle trajectory prediction on highways. *Transportation Research Part C: Emerging Technologies*. 2023;154:104272. DOI: 10.1016/j.trc.2023.104272.

[26] Wang Z, et al. Lane transformer: A high-efficiency trajectory prediction model. *IEEE Open Journal of Intelligent Transportation Systems*. 2023;4:2–13. DOI: 10.1109/OJITS.2023.3233952.

[27] Xu Y, Wang Y, Peeta S. Leveraging transformer model to predict vehicle trajectories in congested urban traffic. *Transportation Research Record*. 2023;2677(2):898–909. DOI: 10.1177/03611981221109594.

[28] Chen X, et al. Stochastic non-autoregressive transformer-based multi-modal pedestrian trajectory prediction for intelligent vehicles. *IEEE Transactions on Intelligent Transportation Systems*. 2023;24(12):14567–14579. DOI: 10.1109/TITS.2023.3342040.

[29] Yang B, et al. A multi-task learning network with a collision-aware graph transformer for traffic-agents trajectory prediction. *IEEE Transactions on Intelligent Transportation Systems*. 2024;25(3):1234–1245. DOI: 10.1109/TITS.2023.3345296.

[30] Gao K, et al. Dual transformer based prediction for lane change intentions and trajectories in mixed traffic environment. *IEEE Transactions on Intelligent Transportation Systems*. 2023;24(6):6203–6216. DOI: 10.1109/TITS.2023.3248842.

[31] Ye H, et al. Trajectory prediction of port container trucks based on DeepPBM-Attention. *Promet-Traffic & Transportation*. 2024;36(3):525–543. DOI: 10.7307/ptt.v36i3.420.

[32] Nie Y, Nguyen NH, Sinthong P, Kalagnanam J. A time series is worth 64 words: Long-term forecasting with transformers. *arXiv preprint arXiv:2211.14730*. 2022. DOI: 10.48550/arXiv.2211.14730.