Traffic & Transportation

Trajectory Prediction of Port Container Trucks Based on DeepPBM-Attention

Haixiong YE[1], Kairong LUAN[2], Mei YANG[3], Xiliang ZHANG[4], Yue ZHOU[5]

Original Scientific Paper
Submitted: 13 Aug. 2023
Accepted: 20 Feb. 2024

[1] hxye@shou.edu.cn, College of Engineering Science and Technology, Shanghai Ocean University
[2] lkr178@126.com, College of Engineering Science and Technology, Shanghai Ocean University
[3] yangmei@portshanghai.com.cn, Shanghai International Port (Group) Co., Ltd.; Academy for Engineering and Technology, Fudan University
[4] Corresponding author, xlzhang@sspu.edu.cn, School of Intelligent Manufacturing and Control Engineering, Shanghai Polytechnic University
[5] Corresponding author, y-zhou@shou.edu.cn, College of Engineering Science and Technology, Shanghai Ocean University

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

ABSTRACT

Existing tracking algorithms mostly rely on model-driven approaches, which can be prone to inaccuracies due to unpredictable human behaviours. This article aims to address the issue of transient errors in tracking port container trucks (PCTrucks) when encountering obstructions. A data-driven algorithm for predicting vehicle trajectories is proposed in this study. The approach involves preprocessing an extensive dataset of GPS information, training a DeepLSTM-Attention model, and integrating the proposed model with the population-based training (PBT) algorithm to optimise network hyperparameters. The objective is to enhance the accuracy of predicting trajectories for vehicles moving horizontally. The trajectory data used are collected from real-world port operations. This research is conducted across nine trajectory segments and benchmarked against traditional approaches like Kalman filtering, machine learning techniques such as support vector regression (SVR) and standard long short-term memory (LSTM) networks. The results demonstrate that the proposed prediction method, that is, DeepPBM-Attention, outperforms other techniques in several evaluation metrics, including root mean square error (RMSE), mean absolute error (MAE), F1 score and trajectory reconstruction error (TRE). Compared to LSTM networks, the performance of DeepPBM-Attention is improved by approximately 40%. The proposed data-driven trajectory prediction algorithm exhibits high accuracy and practicality, which can effectively be applied to the positioning prediction of horizontally moving vehicles in port environments.

KEYWORDS

port container trucks; trajectory prediction; population-based training; deep long short-term memory; attention.

1. INTRODUCTION

With the rapid evolution of the logistics industry, seaports have assumed a pivotal role in the efficient global consolidation and movement of goods. These bustling hubs of commerce facilitate the exchange of cargo between land and sea transportation modes, making precise and optimised vehicle movement within seaport environments essential. However, the horizontal mobility of vehicles within seaports often encounters challenges due to obstacles such as stacked cargo containers and imposing container ships. These obstructions can lead to transient inaccuracies in GPS positioning, potentially compromising the accuracy of goods transloading and overall port operational efficiency.

In this dynamic context, accurately forecasting the trajectories of horizontally moving vehicles has emerged as a pressing concern. Mitigating transient inconsistencies in GPS positioning induced by obstructive elements holds the potential to significantly enhance the precision and integrity of vehicle movements within seaports. By delving into the challenges highlighted in prior research, the critical challenge of accurately forecasting the trajectories of port container trucks (PCTrucks) within the dynamic confines of seaport environments is strategically addressed in this paper.

In confronting the challenge of predicting PCTrucks trajectories within seaports, it is essential to scrutinise the limitations of conventional methods and singular neural network models. Traditional approaches like Kalman filtering, while widely used, prove inadequate in dynamic seaport settings due to their reliance on predefined models, making them susceptible to inaccuracies from unpredictable human behaviours and dynamic obstacles. Similarly, standalone neural network models, exemplified by standard long short-term memory (LSTM) networks, may oversimplify the complexities of seaport dynamics, leading to suboptimal trajectory predictions for horizontally moving vehicles. Recognising these limitations, there is a compelling need for a sophisticated model that embraces diverse data and adapts to the unique challenges of seaport environments. The pursuit of an advanced prediction model stems from the inadequacies of current methods and the goal of achieving heightened accuracy in forecasting PCTrucks trajectories within the intricate and dynamic confines of the seaport. In response to the limitations of traditional methods, there arises a need for innovative approaches. A novel trajectory prediction approach named DeepPBM-Attention is presented here, leveraging the potential of data-driven techniques to meet this demand. The proposed method harnesses the power of deep learning, attention mechanisms and advanced optimisation strategies to accurately forecast the trajectories of PCTrucks within seaport environments.

Furthermore, the incorporation of innovative optimisation techniques fine-tunes the model's hyperparameters, enhancing its adaptability to the unique characteristics of PCTruck trajectories within seaport settings. This holistic approach aims to provide accurate and robust trajectory predictions, mitigating the transient errors often encountered in seaport logistics.

The aim of this study is to leverage DeepBPM-Attention to develop a novel model for executing trajectory prediction tasks for PCTrucks. This paper includes the following significant contributions:

1) DeepPBM-Attention model: A novel model, termed DeepPBM-Attention, is introduced in this work, seamlessly amalgamating the deep long short-term memory (DeepLSTM) framework with an attention mechanism. This innovative structure effectively captures temporal dependencies and spatial relationships in PCTruck trajectory data, offering a significant enhancement over traditional models.
2) Application of population-based training (PBT): This study pioneers the use of the PBT algorithm to optimise the hyperparameters of the trajectory prediction network, thereby improving prediction accuracy. This novel optimisation method significantly enhances the performance of our model.
3) Capitalisation on real-world operational data: Genuine operational data from ports are incorporated to scrutinise the trajectory predictions, effectively bridging the gap between theoretical underpinnings and real-world applicability. This strategy furnishes a more incisive evaluation of the model's performance, grounding the validation in pracsticality.
4) Pronounced enhancement in performance metrics: The proposed DeepPBM-Attention model outperforms traditional methods such as Kalman filtering, SVR and standard LSTM networks across multiple evaluation metrics. Notably, compared to LSTM networks, the performance metrics of DeepPBM-Attention improved by approximately 40%, representing a significant advancement.
5) Elevated practical utility: The proposed DeepPBM-Attention model exhibits high precision and practicality. It is capable of effectively predicting the movement of horizontal vehicles within ports. This novel application significantly enhances the accuracy of trajectory prediction in real-world scenarios.

In the subsequent sections, a structured exploration of the study unfolds. Section 2 is dedicated to a literature review, wherein a comprehensive examination of trajectory prediction methods employed by predecessors and those currently in vogue is undertaken. Section 3 focuses on elucidating the intricacies of data processing and model design. Section 4 details the experimental design tailored specifically for predicting PCTruck trajectories. Subsequently, in Section 5, a thorough discussion of the experimental results is undertaken, concluding the entire study.

2. LITERATURE REVIEW

Over the years, the vehicle trajectory prediction task has received significant attention from scholars, and several methodologies have been proposed. Initially, Park et al. [1] put forward a technique which utilises Kohonen networks and Kalman filters to assess the moving object's trajectory. Hermes et al. [2], on the other hand, established a prediction approach with long-term benefits which integrated particle filtering and trajectory classification frameworks. Chen et al. [3] proposed a groundbreaking approach which adopted real-time and historical traffic data to construct the vehicle trajectories. Kim et al. [4] utilised a particle filter-based localisation method that relied on distance and bearing data extracted from fixed landmarks. Alternatively, Zhang et al. [5] adopted a hybrid technique which combined model-based and data-driven methods, using Kalman filtering to forecast vehicle trajectories. Farahi et al. [6] proposed a probabilistic Kalman filter (PKF) that enhanced tracking estimation by considering stored trajectories, among other factors. Although effective in linear and nonlinear trajectory estimation, these models are limited in complex dynamic environments. They depend on preset physical models and prior knowledge, compromising flexibility and accuracy with atypical patterns. Deep learning methods, especially neural network-based trajectory predictors, are advancing to address these limitations through enhanced feature extraction and pattern recognition.

The development of machine learning and deep learning has attracted thousands of scholars who have applied these advanced techniques to tackle the issue of vehicle trajectory prediction, with remarkable achievements gained over the past few years. Agrawal et al. [7] introduced a unique trajectory prediction approach that leverages machine learning in combination with a least square curve fitting technique, leading to a novel extended Kalman filter and machine learning-based method. Similarly, Lin et al. [8] proposed a pedestrian trajectory prediction algorithm which employs a dual-mode extended Kalman filter. These innovative approaches enhanced the prediction and localisation accuracy, which also presented favourable results when tested in complex situations.

Deep learning has been extensively employed in traffic flow prediction. For instance, Chen et al. [9] introduced DeepTFP, a deep learning-based time series model, harnessing the capabilities of time series functions and deep learning for feature extraction in traffic flow prediction. Incorporating human factors into trajectory tracking has also been a subject of investigation. Chen et al. [10] presented a strategy that integrates driver behaviour prediction into trajectory tracking control.

Attention mechanisms and spatiotemporal considerations have been significant in trajectory prediction. Dai et al. [11] proposed ST-LSTM, which is based on spatiotemporal considerations. Attention mechanisms were employed by Lin et al. [12], who developed an attention-based LSTM model for predicting trajectories of heavy-duty vehicles in mixed-traffic environments. Furthermore, Yang et al. [13] introduced the spatial-temporal attention (STA) module, capturing the dynamic characteristics of trajectories and employing multiple LSTM networks for different time scales. Subsequently, Desai et al. [14] investigated a novel time series forecasting model, which integrates temporal self-attention mechanisms, convolutional neural networks and convolutional LSTM (ConvLSTM). This model aims to address the challenges of accurately predicting multivariate time series (MTS), but its performance has only been tested on specific datasets and applied to a particular use case (satellite state prediction). Moreover, the model may encounter issues such as lengthy training times and difficulty in learning long-term dependencies in cases where there are gaps in the sequence data, which is why there is consideration for incorporating deep LSTM networks in future work.

In the context of long-distance trajectory predictions, Xin et al. [15] used dual LSTM networks to address prediction accuracy in highly interactive driving environments. Another approach proposed by Sajanraj et al. [16] involved enhancing LSTM input sequences with station information to capture differences between various stations more effectively.

Liu et al. [17] proposed a deep learning-based algorithm, DeepMTT, for manoeuvrable target trajectory prediction, but it showed limitations in prediction accuracy when the estimated position error was substantial. This led Yu et al. [18] to propose the DeepGTT algorithm to improve trajectory prediction in complex scenarios through standardisation and mapping of trajectory data. However, this approach might suffer from estimation delays due to observation noise and nonlinear errors.

Finally, LSTM-based trajectory prediction techniques have been applied to other domains and compared with other models. For example, Siami-Namini et al. [19] compared bidirectional LSTM with ARIMA for financial time series prediction. Yin et al. [20] utilised LAG-LSTM for high-speed railway train trajectories, whereas Yang et al. [21] used Bi-LSTM for ship trajectory prediction. Lin et al. [22] focused on spatial-temporal attention mechanisms and demonstrated superior performance in comparison to convolutional neural networks (CNN) and recurrent neural networks (RNN).

In summary, the literature shows a rich set of methodologies for trajectory prediction, especially employing deep learning. This research builds on these works and specifically addresses the issue of unexpected errors in tracking port container trucks. In light of the foregoing investigation, this study proposes a novel methodology employing a PBT-optimised LSTM-Attention network for predicting PCTruck trajectories.

3. METHODOLOGY

The DeepPBM-Attention model is employed for the prediction of horizontal movement trajectories of PCTrucks within a port environment. *Figure 1* illustrates the comprehensive design process of the DeepPBM-Attention model, which encompasses three primary steps:

1) Gathering and preprocessing raw positional data, including denoising and interpolation, followed by dataset division into training, validation and test sets.
2) Iterative training of the model using the training and validation sets to determine the best trajectory prediction model.
3) Validation of the model using nine real vehicle trajectories from the test set, assessing performance metrics like training duration, prediction time and accuracy.

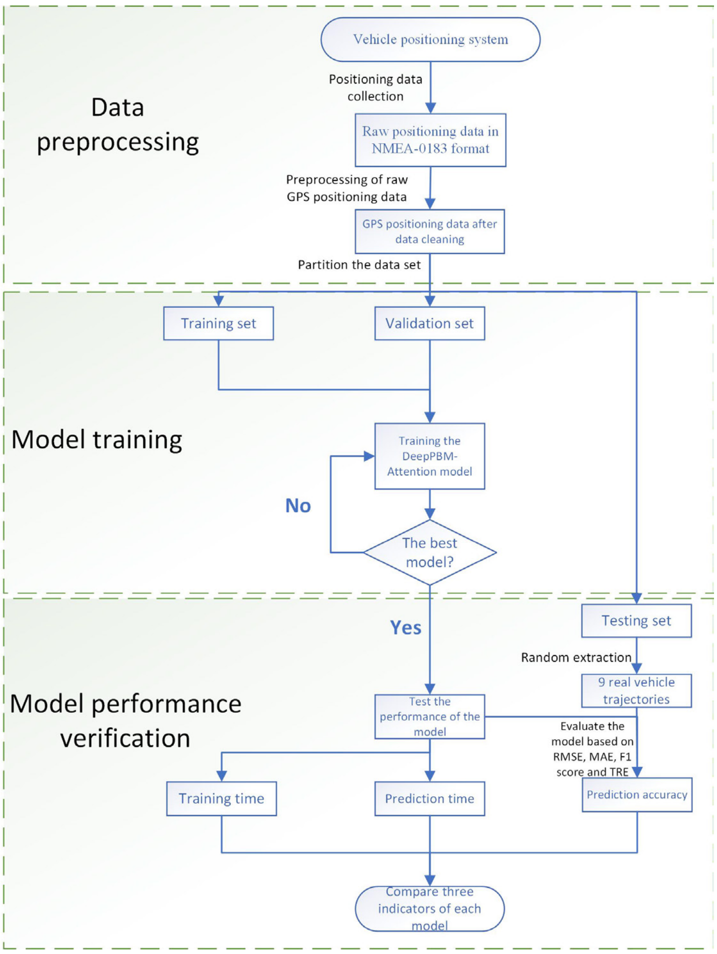


Figure 1 – Overall flow chart of vehicle trajectory prediction

For a more in-depth exposition of the workflow's internal dynamics, including the preprocessing of positioning data, the structure of the DeepPBM-Attention model and the validation steps, *Figure 2* has been developed. It presents a detailed exploration of these processes. In subsequent sections, a more detailed exposition will be presented, focusing specifically on each of these crucial steps, from data cleansing to the sophisticated architecture and validation of the DeepPBM-Attention model.

The content of each step in *Figures 1 and 2* will be introduced in detail in the following sections.

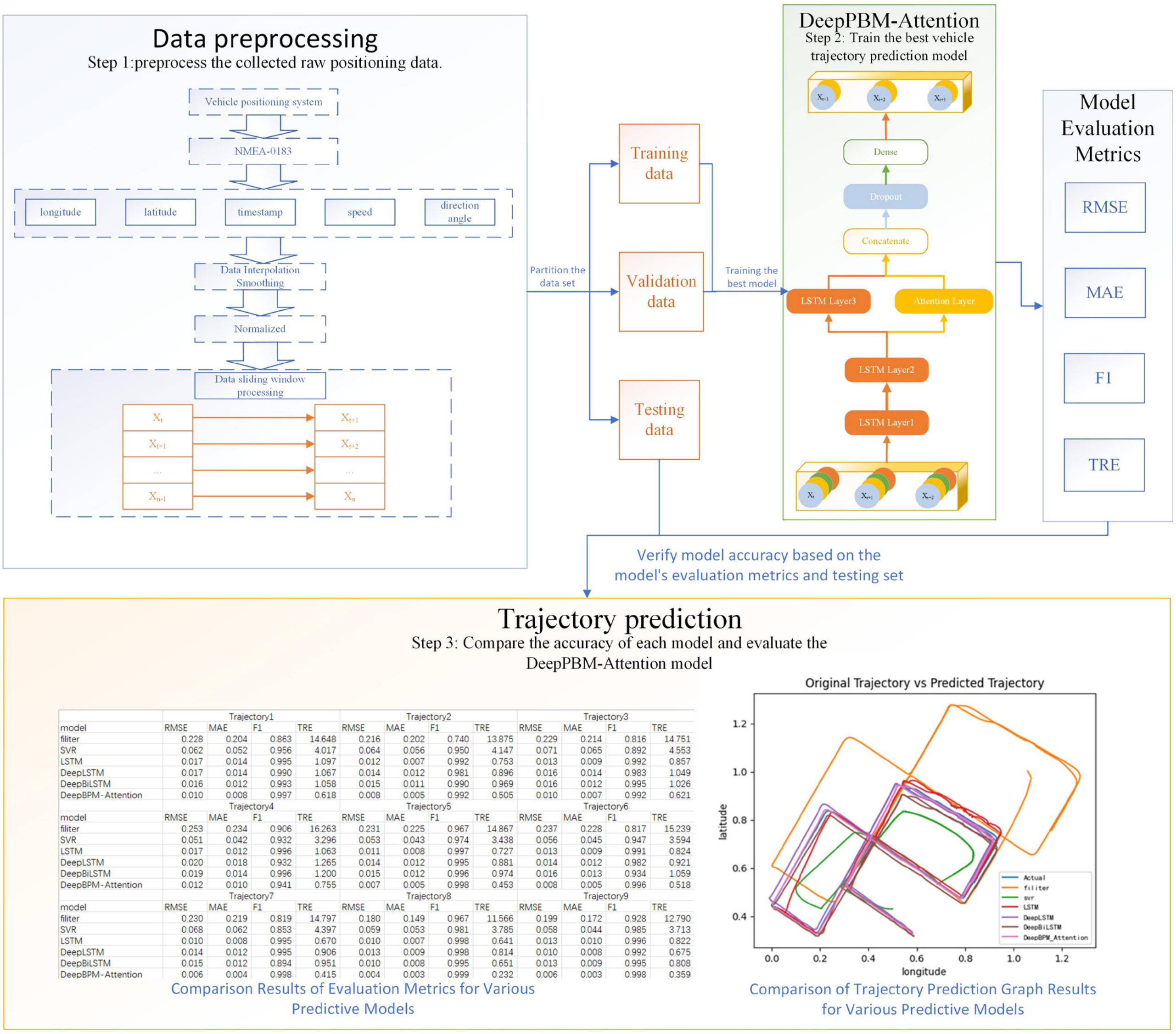


Figure 2 – Design details of the trajectory prediction model

3.1 Data preprocessing

The continuous collection of both temporal and spatial data is constituted by the trajectory data of horizontally moving PCTrucks within the port constitutes. Faster convergence of models and higher prediction accuracy can be accomplished with proper processing. Additionally, computational complexity and the associated training costs can be diminished. Consequently, a meticulous preprocessing of the initial data has been undertaken to refine our prediction model. The detailed procedure of our data processing strategy is depicted in *Figure 3*.

Efficient trajectory prediction is enhanced through the acquisition of GPS data from PCTrucks within the port over 30 days, with a focused analysis conducted on a selected subset of seven days. The collected data are subsequently processed to obtain meaningful findings. This process involved creating a time sequence for the data, extracting features such as timestamp, latitude, longitude, speed and direction, and applying filters to manage outliers. Python 3.7 was utilised for data processing.

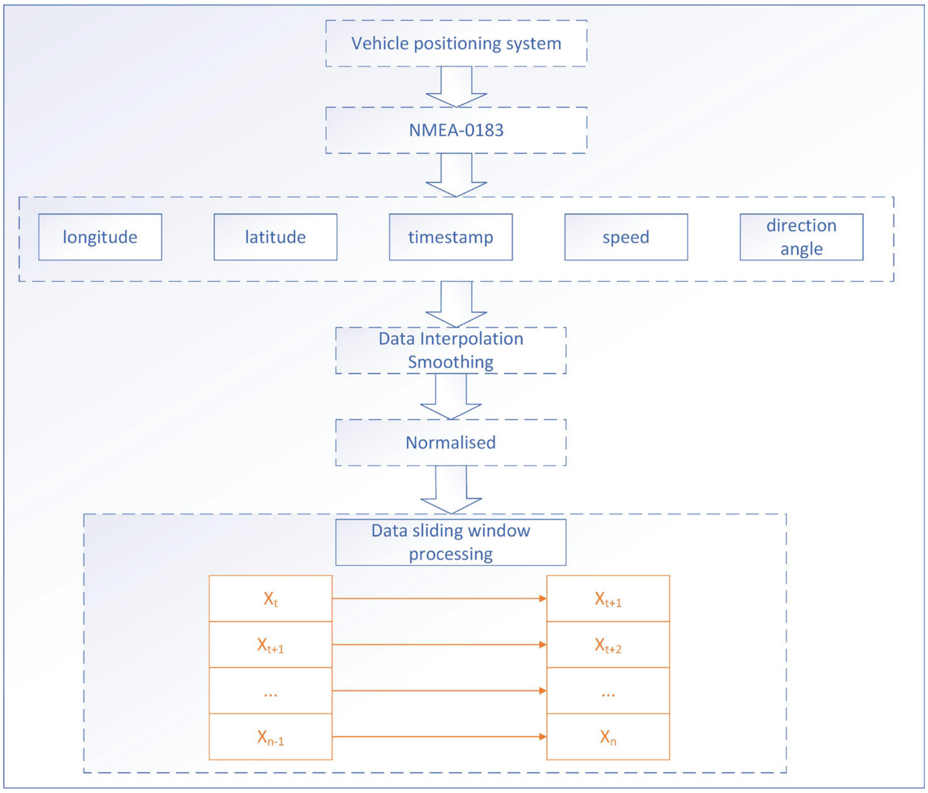


Figure 3 – The pre-processing process of GPS raw data

The study began by parsing the collected GPS data from PCTruck, which were encoded in the NMEA-0183 format. The data were parsed to extract important features such as time, latitude, longitude, speed and heading angle. These features were then stored in a data frame. To reduce the influence of noise and outliers, a filtering technique was applied in this study to remove abnormal and noisy data points.

The challenge of missing GPS coordinates, arising from signal interruptions and resulting in data gaps, was addressed. To maintain data integrity, interpolation was employed to handle these missing values. To estimate missing values, the preceding and succeeding values at the respective time points were used in the linear interpolation. The interpolated results were saved to the data frame, ensuring a continuous and complete dataset.

The GPS data are segmented into time intervals and transformed into a supervised learning format for model training and prediction. The longitude, latitude, speed and direction from the previous time interval are used as input to the model, while the longitude and latitude from the subsequent time interval serve as the predicted output. The prediction accuracy can be enhanced due to this supervised learning structure by enabling the model to learn the relationship between input features and the desired output. Refer to *Figure 4* for an illustration of this data transformation process.

To expedite model training and improve prediction accuracy, it is essential to normalise the positioning data and standardise its data range within the interval of [0, 1].

The preceding paragraphs offer a comprehensive overview of the preprocessing steps involved with GPS data. These steps are contained by data integration, sorting, extraction, filtering, outlier management, missing value treatment, normalisation and the construction of the input structure for the model, which establishes a data foundation for subsequent model training and prediction tasks.

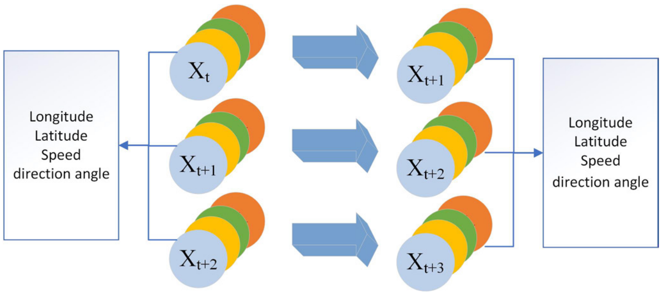


Figure 4 – Data transformation into supervised learning data

3.2 Neural network model design

In the subsequent part of this section, the DeepBPM-Attention model is designed and presented along with an elaboration of the main neural network model employed by this framework, namely the LSTM model. Lastly, the hyperparameter optimisation algorithm will be introduced and adopted in this study.

LSTM

The LSTM is a variant of recurrent neural network (RNN) that employs gating mechanisms to regulate the flow of information and update its memory. For an individual LSTM cell, internal computations are performed as follows:

$$f_t = \sigma(W_{xf}x_t + W_{hf}x_{t-1} + W_{cf} \cdot C_{t-1} + b_f) \quad (1)$$

$$i_t = \sigma(W_{xi}x_t + W_{hi}x_{t-1} + W_{ci} \cdot C_{t-1} + b_i) \quad (2)$$

$$o_t = \sigma(W_{xo}x_t + W_{ho}x_{t-1} + W_{co}C_{t-1} + b_o) \quad (3)$$

$$C_t = f_t \cdot C_{t-1} + i_t \tanh(W_{xc}x_t + W_{hc} \cdot h_{t-1} + b_c) \quad (4)$$

$$h_t = o_t \tanh(C_t) \quad (5)$$

In *Equations 1–3,* the σ function (sigmoid function) and the *tanh* function are defined as follows:

$$\sigma(z) = \frac{1}{1+e^{-z}} = \frac{1+\tanh\left(\frac{z}{2}\right)}{2} \quad (6)$$

$$\tanh(z) = \frac{e^{z} - e^{-z}}{e^{z} + e^{-z}} \quad (7)$$

Exceptional capabilities in the realm of general time series modelling and prediction have been exhibited in LSTM, outperforming traditional recurrent neural networks (RNNs). This is accomplished by incorporating multiple threshold gates that balance memory retention and forgetting. As a result, LSTM is considered an effective approach for solving time series prediction problems. Each LSTM block is comprised of a memory cell and three gates: the input gate (h_t), the forget gate (f_t) and the output gate (o_t), as depicted in *Figure 5*. The critical aspect of LSTM lies in the storage of the memory cell state (C_t), represented by the horizontal line at the top of *Figure 5*. Additionally, the input and output of the LSTM are respectively denoted by the x_t and h_t.

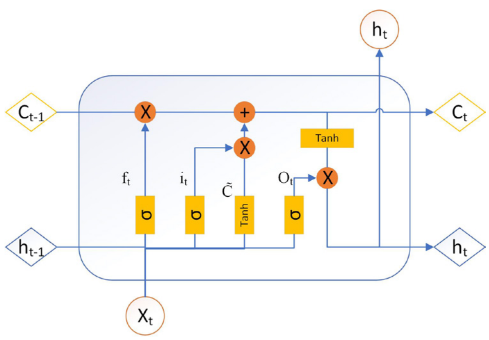


Figure 5 – LSTM internal structure

Given the effectiveness of LSTM in time series prediction tasks, the use of the stored memory cell state (C_t) essentially acts as an accumulator of state information. Upon receiving new input, the forget gate (f_t) determines which information should be discarded from the memory cell state. Subsequently, the input gate decides which values need to be updated, and a new candidate value vector, denoted as C_t, is generated by the Tanh layer. This candidate value can be added to the memory cell state. These results are amalgamated for state updates. Finally, the output gate (o_t) and C_t contribute to the generation of the final state, h_t.

DeepLSTM-Attention

The challenge of predicting long sequence trajectories was addressed by employing a DeepLSTM-Attention model, which effectively captures complex temporal relationships and essential characteristics within vehicle trajectory sequences. By stacking multiple LSTM layers, our model was equipped to perform hierarchical representation and modelling of vehicle trajectory sequences, effectively extracting higher-level abstract features and better addressing the long-term dependencies and intricate patterns in sequential data. The hidden state from the previous layer was performed as input for each LSTM layer with computations of gated units within it. As depicted in *Figure 6*, our DeepLSTM-Attention model initially processes the input data through two LSTM layers, facilitating the extraction of temporal features from the input data.

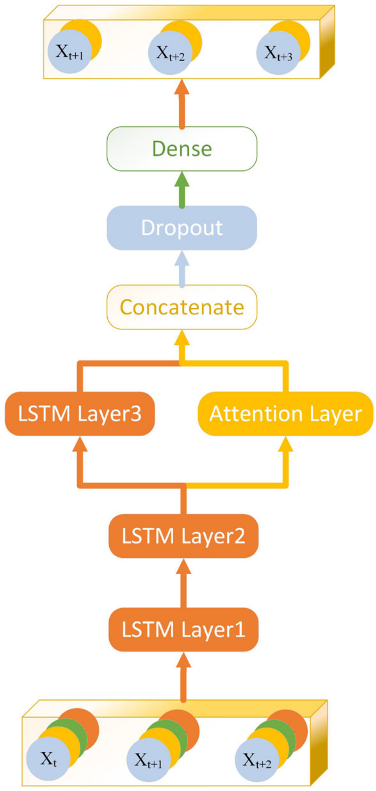


Figure 6 – DeepLSTM-Attention

An attention mechanism was incorporated to enhance the prediction of future vehicle trajectories. The attention layer computes a weighted sum between each time step of the input and the model's learned weights, resulting in an attention weight for each time step. This weight vector can be regarded as the attention assigned to different time steps in the input data, reflecting the importance assigned by the model to each time step. The computation of the attention mechanism in the DeepLSTM-Attention model consisted of two steps: score calculation and normalisation. Suppose the input sequence's length was denoted by T, and the hidden state at each time step was represented as h_t.

Score calculation. Assuming the length of the input sequence is denoted as T, and the hidden state at each time step is represented as h_t. The attention mechanism utilises a multilayer perceptron (MLP) to map the hidden state h_t into attention scores e_t:

$$e_t = MLP(h_t) \tag{8}$$

where e_t represents the attention score at the t time step. By performing a nonlinear transformation and weighted summation of the hidden states, the attention mechanism enables the network to automatically focus on the relevant parts of the input sequence that are pertinent to the current task.

Normalisation. To obtain the attention weights h_t, it is necessary to normalise the scores. Initially, the scores are transformed into positive values using the exponential function. Subsequently, the scores of all time steps are summed and normalised, resulting in the attention weights α_t:

$$\alpha_t = \frac{\exp(e_t)}{\sum_{i=1}^{T} \exp(e_i)} \tag{9}$$

In the DeepLSTM-Attention model, the output of the attention layer is concatenated with the output of the third LSTM layer, effectively amalgamating the information from these two key components. The third LSTM layer captures temporal patterns and relevant features in the vehicle trajectory data, while the attention layer enhances focus on different time steps. Due to concatenation, the input data can be grasped through this model comprehensively and holistically by considering all the diverse information these components provide. Moreover, the attention layer can better utilise the output of the third LSTM layer through this operation, preserving more information and details for subsequent processing. Without this concatenation, the attention layer would only operate on the LSTM's output, limiting its use over the entire trajectory data and preventing it from accessing deeper-level information. Multiple experiments have been conducted to support this finding.

A dropout layer was added after the attention layer to address potential overfitting that may occur when randomly omitting a fraction of the neuron outputs.

After conducting extensive experimentation and result comparisons, it was recognised that incorporating the dropout layer at this point enhances the model's ability to manage overfitting issues and improve its generalisation performance on the unobserved dataset.

The feature fusion strategy in this model combines the representational power of LSTM and attention layers to effectively capture input sequence features, enhancing overall performance.

In summary, the DeepLSTM-Attention model assimilates a multi-layer LSTM architecture, feature fusion, an attention mechanism and a dropout layer, consequently delivering exceptional performance. The model's superior function hinges on hierarchical representation and modelling, automatic prioritisation of vital sequence components, feature amalgamation and regularisation, all of which serve to effectively seize long-term dependencies and crucial information in sequential data. This, in turn, bolsters the model's performance and generalisation abilities. Future refinement of this model will entail the exploration of hyperparameter optimisation techniques, forming the subsequent progression of this investigation.

Figure 7 illustrates the four-dimensional input to the model. After data preprocessing, the input sequence is time-ordered. The four variables – longitude, latitude, speed and heading angle – are subsequently admitted into the model as inputs.

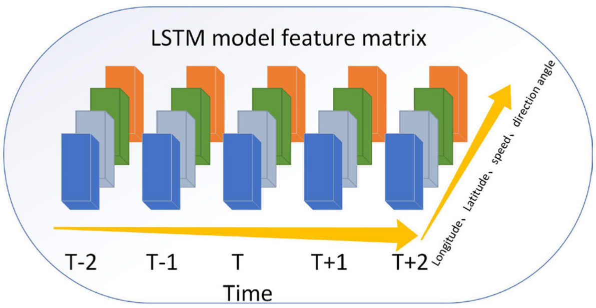


Figure 7 – Input and output of the model

PBT optimised DeepLSTM-Attention

Population-based training (PBT) [23] operates as an evolutionary optimisation technique for hyperparameters. It cultivates a population of models, each endowed with distinct hyperparameters and weights. PBT interprets the training process as an evolutionary progression of a population, commencing with an array of neural networks alongside their associated hyperparameters. In every iteration, individuals within the population are processed utilising their allocated computing resources. Simultaneously, they "replicate" to yield new individuals possessing modified hyperparameters, thereby updating the population.

This process bears a resemblance to genetic algorithms where individuals learn and compete with each other. Presented below is the principal computational procedure of PBT:

$$f_i = (1 - P) \cdot f_i \tag{10}$$

$$f_j = P \cdot f_j \tag{11}$$

where f_i and f_j denote the fitness values of the i-th and j-th individuals in the population, respectively, whereas P represents the probability of pruning an individual. According to *Equation 10*, an individual with a higher fitness value, i, will have a probability of (1-P) to be retained. *Equation 11* implies that an individual with a lower fitness value, j, will be pruned with a probability of P. Below is a detailed description of the implementation procedure:

Step 1: Population initialisation. Randomly selecting hyperparameters from the hyperparameter space to initialise neural networks and thus create an initial population.

Step 2: Individual training. The neural network corresponding to each individual in the initial population is trained using its hyperparameters and performance metrics are obtained.

Step 3: Individual replication and training. Each replicates itself and generates new individuals by slightly modifying their hyperparameters. Respective hyperparameters are utilised to train neural networks and compete with each other within these new individuals.

Step 4: Inter-individual competition. In each iteration, the individuals in the population compete with each other to determine the individuals to be retained. This process presents an opportunity for individuals to learn from the successes of their peers and potentially get replaced by others.

Step 5: Population update. The next generation population is formed by removing the individuals with poorer fitness and adding the individuals with better fitness.

Due to the significant effect hyperparameters have on neural network performance, the BPT algorithm was employed to optimise the hyperparameters of the proposed DeepLSTM-Attention network (DeepBPM -Attention). This allowed us to find the optimal hyperparameters and create the best model.

4. EXPERIMENTAL DESIGN AND RESULTS

In this segment, an empirical assessment of the DeepPBM-Attention algorithm is presented using authentic container terminal data. Initially, the experimental scenario and data source are elucidated, followed by formulating experiments for horizontal trajectory prediction of PCTruck. Several methodologies, including Kalman filtering, support vector regression (SVR), LSTM, DeepLSTM, DeepBiLSTM and the proposed DeepPBM-Attention are deployed and juxtaposed. In conclusion, a thorough assessment and comparison of various algorithms are conducted to highlight the advantages of DeepPBM-Attention in addressing horizontal positioning errors for PCTrucks at container terminals in practical scenarios.

The data from daily activities at Shanghai Port forms the foundation for PCTrucks' trajectory prediction research. To accomplish the automation revamp and conversion of traditional container terminals, installing GPS positioning systems on a total of 117 PCTrucks at the terminal is essential. Consequently, the GPS positioning data for a month were collected without interruption and seven days of these are analysed deeply. A substantial amount of NMEA-0183 formatted data were provided for this study. Following this, the harvested data were subjected to preprocessing, resulting in the generation of the experimental dataset requisite for this study. The integration of this dataset enhances our understanding of the complex behaviours exhibited by PCTrucks as they navigate the intricacies of seaport settings.

4.1 Model and parameter design for trajectory prediction

Kalman filter

The Kalman filter, a recognised probabilistic method, is employed for estimating state variable values in dynamic systems. In the context of this article's comparative analysis, the algorithm operates in a cyclical process of measurement updates and prediction steps. In the measurement update phase, the state estimate integrates new measurements (y) with the current state (x), heavily relying on the measurement noise covariance matrix R. Set as a diagonal matrix with 1 element, R reflects the estimated uncertainty in measurement noise.

In the ensuing prediction phase, the future state is forecasted using the state transition matrix (A), where the process noise covariance matrix Q is of crucial importance. Configured as a diagonal matrix with each element valued at 0.1, Q denotes the assumed uncertainty in the system's process noise. The configuration of Q and R critically affects the balance between reliance on the system's dynamics and the measurement data, thus influencing the Kalman filter's overall performance and accuracy in tracking and predicting state variables.

SVR

Support vector regression (SVR) is a machine learning algorithm applied for predicting continuous variables, acting as an extension of the support vector machine (SVM) algorithm in the realm of regression analysis. SVR seeks to find an optimal hyperplane that minimises the average distance between the training data points and the hyperplane, thereby aiming to yield accurate predictions. Within this study, SVR is employed to predict the latitude and longitude of vehicles by evaluating the provided features and latitude-longitude data. Once the model is trained, it is applied to forecast the vehicle's latitude and longitude at new locations.

LSTM and its variants

The hyperparameter settings of the neural network are shown in *Table 1*. Extensive experiments were conducted on the task of predicting the trajectories of horizontally moving vehicles, with a special focus on deep learning methodologies. LSTM and its variant models were leveraged to fulfil this task, and several adjustments were made to the hyperparameters of each model. Additionally, the BPT algorithm was leveraged to optimise the hyperparameters of our proposed model, DeepBPM-Attention. Through this process, four superior models were devised, along with their respective parameters, as depicted in *Table 1*. In the subsequent section, the experimental evaluations and comparative analyses of the aforementioned designed models are presented.

Table 1 – 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_{LSTM2}**	**optimiser**	**epochs**	**batch_size**
LSTM	729,000	81000	128	\	\	Adam	200	3600
DeepLSTM	729,000	81000	128	256	256	Adam	200	3600
DeepBiL-STM	729,000	81000	128	256	256	Adam	200	3600
DeepBPM-Attention	729,000	81000	128	256	256	Adam	200	3600

4.2 Result and analysis

Experiment and analysis of model training efficiency

The efficiency of training deep learning models is often regarded as a crucial evaluation metric. In light of our advancement from a solitary LSTM neural network to the integration of multiple layers of LSTM and additional neural network components, such as attention, it is essential to devise experiments that meticulously assess and compare the training efficiency of each model. Such comparative analyses are essential for a more comprehensive assessment of the predictive models.

In this experiment, the hyperparameter configurations for each model are outlined in *Table 1*. Under equivalent hyperparameter settings and an equal number of datasets, the training efficiency of each model is presented in *Table 2*.

Table 2 – Comparison of training efficiency of various models

Model	Training time (s)
Kalman filter	\
SVR	\
LSTM	65.4090625
DeepLSTM	227.2934141
DeepBiLSTM	550.7698176
DeepBPM_Attention	245.1058742

Due to the increased depth of the LSTM neural network and the incorporation of an attention layer, our model becomes more intricate, enabling a better understanding of vehicle trajectory data within the port context. Consequently, the training time for our designed model is approximately four times that of a single LSTM neural network. In future investigations, the training efficiency of complex models will be focused on studying.

Analysis of prediction results of the DeepPBM-Attention model

The proposed DeepPBM-Attention network is utilised in our study to predict the trajectories of PCTrucks within a harbour region. Individual predictions for longitude, latitude and the PCTruck trajectories are executed by leveraging processed data. The predicted outcomes are subsequently juxtaposed with the actual PCTruck trajectories. A comparison of the model's forecasted longitude values with the real data is graphically exhibited in *Figure 8*.

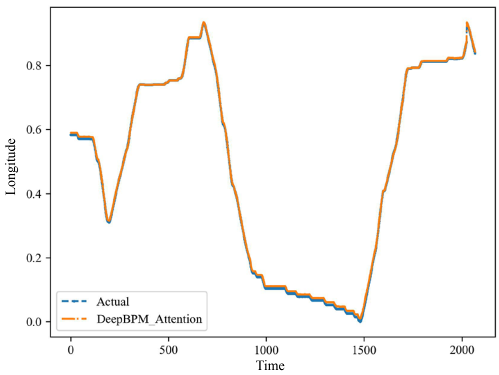


Figure 8 – Comparison chart of true longitude and predicted longitude

Likewise, *Figure 9* shows the predicted latitude values juxtaposed against the actual data.

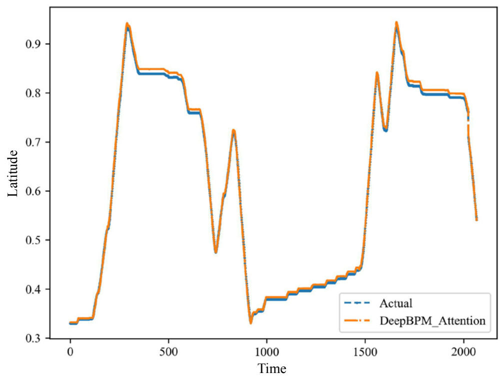


Figure 9 – Comparison chart of real latitude and predicted latitude

Lastly, in *Figure 10*, the model's overall trajectory predictions contrasted with the real trajectories are exhibited.

From *Figures 8–10*, it becomes clear that our model demonstrates substantial accuracy in predicting the horizontal motion of PCTrucks within the port vicinity, whether considering longitude, latitude or trajectory prediction. In the following sections, the DeepPBM-Attention model with other trajectory prediction models is juxtaposed to conduct a thorough evaluation of our proposed methodology.

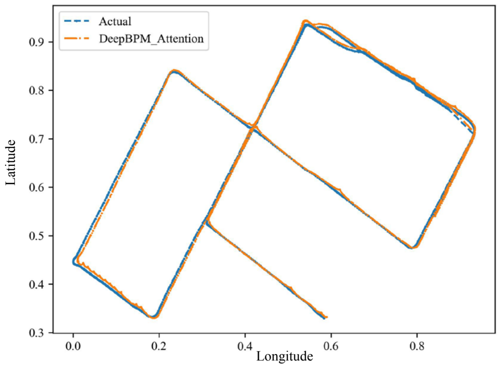


Figure 10 – Comparison chart of real trajectory and predicted trajectory

Comparison of the experimental results of each model

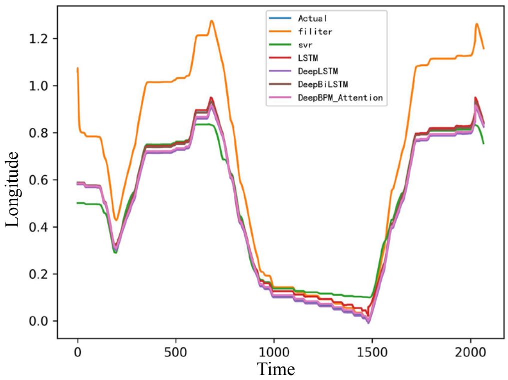
The goal of this research was to perform a comparative analysis of six trajectory prediction methods to verify their predictive capabilities. RMSE, MAE, F1 score and TRE are utilised as evaluation metrics. Data-driven neural network algorithms showcased superior predictive performance compared to the model-driven Kalman filter approach, as per the analysis results. These algorithms excelled over the latter method in terms of RMSE, MAE, F1 score and TRE. Moreover, data-driven neural network algorithms more precisely grasped the shape and curves of the predicted trajectories, thereby aligning them more accurately with real-world scenarios. The specific comparison of the efficacy of different methods is undertaken in predicting PCTruck trajectories. The evaluated methods included Kalman filtering, SVR, LSTM, DeepLSTM, DeepBiLSTM and DeepPBM-Attention. The predictive performance and errors of these methods are scrutinised to thoroughly assess their applicability. The subsequent section introduces the experimental findings.

Figures 11a and 11b respectively display the predictions of longitude and latitude for the PCTruck made by various models, while *Figure 11c* shows a comparison of the predicted trajectories composed of the predicted longitudes and latitudes. From *Figure 10*, it can be observed that regardless of whether it is longitude or latitude, the results predicted by our proposed DeepPBM-Attention model are superior to those of other models. The findings imply that the DeepPBM-Attention method is proficient at predicting PCTruck trajectories in harbour settings.

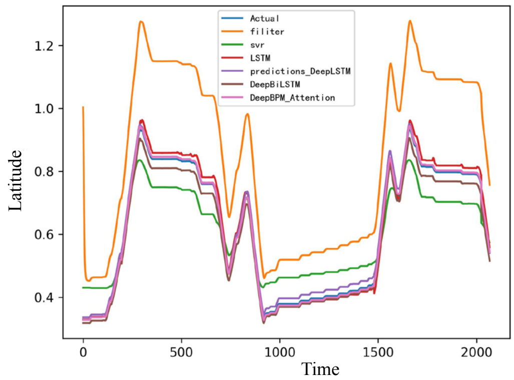
Nine trajectories, depicted in *Figure 12*, were utilised to evaluate the model's generalisation capacity. The model's competence to predict PCTruck movements beyond the scope of the training and testing datasets was validated through these trajectories.

In the task of vehicle trajectory prediction, the prediction times vary among different models when using the same test set. Furthermore, the time taken to predict trajectories of the same length is a crucial metric for evaluating model performance. A robust trajectory prediction model should strive to deliver more accurate predictions within shorter time frames. To assess this aspect, the tests on the prediction times for various models on nine trajectory segments are conducted, as depicted in *Figure 12*. The comparative results are presented in *Table 2*. It is noteworthy that the experiments were conducted on a computer with an Intel Core i7 CPU, an RTX 1050 GPU and 16 GB of RAM.

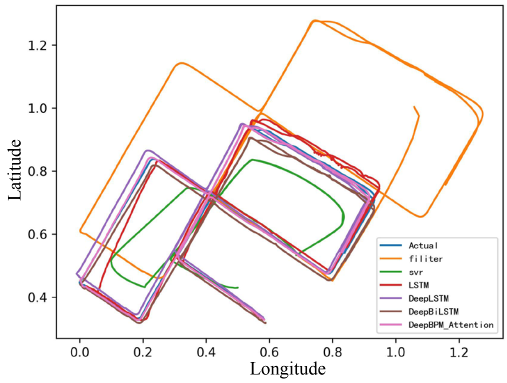
The experimental results indicate that model-driven approaches exhibit significantly lower prediction times compared to data-driven methods. Within the realm of data-driven methods, our designed DeepBPM-Attention model outperforms other models in 5 out of the 9 test segments. Despite the increased complexity of our model structure, it demonstrates competitive prediction speeds compared to alternative methods.



a) Comparison of longitude prediction results



b) Comparison of latitude prediction results



c) Comparison of trajectory prediction results

Figure 11 – Comparison chart of PCTruck trajectory prediction results of each model

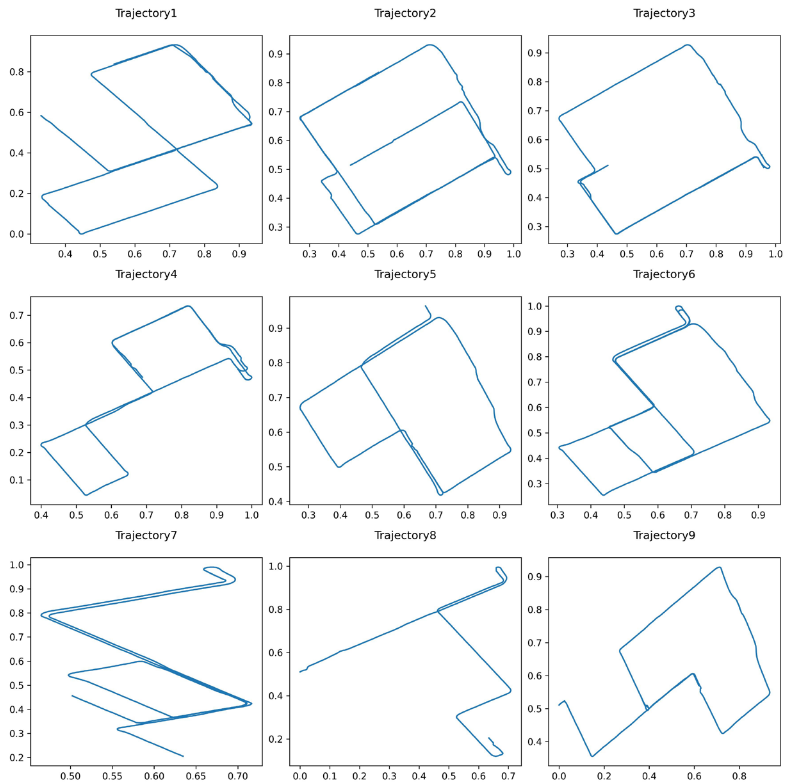


Figure 12 – Nine trajectory segments for testing

Table 3 – Comparison table of prediction time of each modelå

Model	The number of validation set	Test time (s)	The number of validation set	Test time (s)	The number of validation set	Test time (s)
	Trajectory1		**Trajectory2**		**Trajectory3**	
Kalman filter	2068	0.268	2069	0.286	2069	0.268
SVR		0.009		0.014		0.008
LSTM		1.328		1.344		1.297
DeepLSTM		1.073		1.108		1.072
DeepBiLSTM		2.173		2.153		2.360
DeepBPM_Attention		1.138		1.158		1.310
	Trajectory4		**Trajectory5**		**Trajectory6**	
Kalman filter	2069	0.255	2069	0.253	2069	0.263
SVR		0.007		0.009		0.008
LSTM		1.325		1.437		1.379
DeepLSTM		1.223		1.442		1.193
DeepBiLSTM		2.215		2.472		2.243
DeepBPM_Attention		1.149		1.344		1.131
	Trajectory7		**Trajectory8**		**Trajectory9**	
Kalman filter	2069	0.243	2069	0.254	2069	0.273
SVR		0.007		0.008		0.009
LSTM		1.302		1.355		1.426
DeepLSTM		1.366		1.149		1.209
DeepBiLSTM		2.696		2.228		2.335
DeepBPM-Attention		1.307		1.129		1.191

A comparative analysis was performed on the prediction accuracy for the nine trajectories. By juxtaposing the forecasted values with the actual ones, the RMSE, MAE, F1 score and TRE for each method were computed.

RMSE is defined as follows:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\hat{y}_t^i - y_t^i)^2} \tag{12}$$

MAE is defined as follows:

$$MAE = \frac{1}{N}\sum_{i=1}^{N}\left|\hat{y}_i^i - y_i^i\right| \tag{13}$$

F1 Score is defined as follows:

$$F1\ Score = \frac{2 \cdot Precision}{Precision + Recall} \tag{14}$$

Within the *Equation 14*, the computation methods for precision and recall are as follows:

$$Precision = \frac{TP}{TP+FP} \tag{15}$$

$$Recall = \frac{TP}{TP+FN} \tag{16}$$

True positives (TP) signifies the count of samples that the model accurately predicts as positive examples. False positives (FP) denote the count of samples that the model erroneously predicts as positive when they are negative examples. False negatives (FN) represent the number of samples that the model mistakenly predicts as negative when they are positive examples.

TRE is defined as follows:

$$TRE = \sqrt{\sum_{i=1}^{n} (y_i - \hat{y})^2} \tag{17}$$

where y_i represents the real value, $\hat{y}$ represents the predicted value, and n represents the number of samples.

Table 4 showcases the experimental outcomes, illustrating that the DeepPBM-Attention accomplished the highest accuracy in the trajectory prediction task.

Table 4 – Experimental results of RMSE, MAE, F1 and TRE for different models predicting trajectories

Model	**Trajectory1**				**Trajectory2**				**Trajectory3**			
Evaluation	RMSE	MAE	F1	TRE	RMSE	MAE	F1	TRE	RMSE	MAE	F1	TRE
Kalman filter	0.228	0.204	0.863	14.648	0.216	0.202	0.740	13.875	0.229	0.214	0.816	14.751
SVR	0.062	0.052	0.956	4.017	0.064	0.056	0.950	4.147	0.071	0.065	0.892	4.553
LSTM	0.017	0.014	0.995	1.097	0.012	0.007	0.992	0.753	0.013	0.009	0.992	0.857
DeepLSTM	0.017	0.014	0.990	1.067	0.014	0.012	0.981	0.896	0.016	0.014	0.983	1.049
DeepBiLSTM	0.016	0.012	0.993	1.058	0.015	0.011	0.990	0.969	0.016	0.012	0.995	1.026
DeepBPM-Attention	0.010	0.008	0.997	0.618	0.008	0.005	0.992	0.505	0.010	0.007	0.992	0.621
	Trajectory4				**Trajectory5**				**Trajectory6**			
Evaluation	RMSE	MAE	F1	TRE	RMSE	MAE	F1	TRE	RMSE	MAE	F1	TRE
Kalman filter	0.253	0.234	0.906	16.263	0.231	0.225	0.967	14.867	0.237	0.228	0.817	15.239
SVR	0.051	0.042	0.932	3.296	0.053	0.043	0.974	3.438	0.056	0.045	0.947	3.594
LSTM	0.017	0.012	0.996	1.063	0.011	0.008	0.997	0.727	0.013	0.009	0.991	0.824
DeepLSTM	0.020	0.018	0.932	1.265	0.014	0.012	0.995	0.881	0.014	0.012	0.982	0.921
DeepBiLSTM	0.019	0.014	0.996	1.200	0.015	0.012	0.996	0.974	0.016	0.013	0.934	1.059
DeepBPM-Attention	0.012	0.010	0.941	0.755	0.007	0.005	0.998	0.453	0.008	0.005	0.996	0.518
	Trajectory7				**Trajectory8**				**Trajectory9**			
Evaluation	RMSE	MAE	F1	TRE	RMSE	MAE	F1	TRE	RMSE	MAE	F1	TRE
Kalman filter	0.230	0.219	0.819	14.797	0.180	0.149	0.967	11.566	0.199	0.172	0.928	12.790
SVR	0.068	0.062	0.853	4.397	0.059	0.053	0.981	3.785	0.058	0.044	0.985	3.713
LSTM	0.010	0.008	0.995	0.670	0.010	0.007	0.998	0.641	0.013	0.010	0.996	0.822
DeepLSTM	0.014	0.012	0.995	0.906	0.013	0.009	0.998	0.814	0.010	0.008	0.992	0.675
DeepBiLSTM	0.015	0.012	0.894	0.951	0.010	0.008	0.995	0.651	0.013	0.009	0.995	0.808
DeepBPM-Attention	0.006	0.004	0.998	0.415	0.004	0.003	0.999	0.232	0.006	0.003	0.998	0.359

5. DISCUSSION AND CONCLUSIONS

The proposed method is superior to other recent methods, including model-driven methods, such as the Kalman filter, and data-driven methods, such as LSTM, as shown in the last section. Different from other model-driven methods [1–8] mentioned in the literature review section, the proposed method is a fully data-driven algorithm. Model-driven methods are based on physical models and statistical principles, facilitating ease of comprehension and interpretation, eliminating the need for extensive data collection and delivering high inference efficiency. However, their efficacy dwindles in the face of complex and nonlinear scenarios, where adapting and implementing physical models becomes a formidable challenge. Our experiments further corroborate this, revealing model-driven methods' relatively inferior performance in forecasting PCTruck trajectories. In contrast, data-driven approaches excel in extracting a broader spectrum of features in complex settings, thereby ensuring superior accuracy. Similarly, other researches, for example [12, 15–18, 20–22], also utilised data-driven algorithms, such as LSTM or its variants with/without an attention mechanism, for higher accuracy. Although the scenarios and datasets in the works of literature differ from the proposed algorithm, most of these algorithms, such as LSTM, etc., were compared with the proposed one in the previous section. From the results of the last section, it is clear that the proposed method achieves a lower error in this scenario and dataset, and it does so without incurring significant additional training time as a trade-off. Additionally, the proposed method not only exhibits good computational efficiency but also achieves this efficiency without the cost of excessive training time, positioning it as the state of the art.

In the comparative experiments of this study, which are also based on data-driven methods, utilising the same dataset and computational resources, the DeepPBM-Attention model requires a significantly longer training duration compared to a standard LSTM model, about four times as lengthy. This indicates its higher complexity, attributed to its advanced ability to decode intricate data patterns and dependencies. Though the extended training time might limit practicality in some scenarios, the model's sophisticated approach effectively handles complex tasks. In terms of post-training prediction time performance, the DeepPBM-Attention model outshines other deep learning models in PCTrucks trajectory prediction, achieving the lowest prediction times in the majority of nine tests, reflecting its efficiency and consistent reliability. Moreover, it excels in predictive accuracy, outperforming other models across four key metrics: RMSE, MAE, F1 score and TRE. The model's application in GPS trajectory prediction within container terminals, particularly in mitigating positioning errors from obstructions like stacked containers and cranes, demonstrates its superiority over traditional methods such as Kalman filtering, machine learning methods like SVR, and other deep learning approaches, including LSTM. It shows a notable 40% improvement over LSTM in PCTruck trajectory prediction, emphasising its high accuracy and effectiveness.

This study not only underscores the DeepPBM-Attention model's proficiency in harbour positioning but also highlights potential research areas, including its application in complex environments like urban and mountainous terrains. The exploration of integrating additional sensors such as radar and LiDAR could enhance GPS reliance, improving accuracy and robustness. Additionally, merging this model with advanced navigation and path planning algorithms may significantly advance automation and efficiency in PCTruck transport, fostering the development of smarter port logistics. Overall, our research places the DeepPBM-Attention model at the vanguard of PCTruck trajectory prediction, offering innovative solutions for specific situational challenges.

ACKNOWLEDGEMENT

The authors would like to thank the SIPG for providing the data for this research.

REFERENCES

[1] Park J, Yun J, Lee J. Trajectory estimation of a moving object using Kalman filter and Kohonen networks. *Robotica*. 2007;25(5):567-574. DOI: 10.1017/s0263574707003451.

[2] Hermes C, et al. Long-term vehicle motion prediction. *2009 IEEE Intelligent Vehicles Symposium*. 2009;652-657. DOI: 10.1109/ivs.2009.5164354.

[3] Chen H, Rakha HA. Prediction of dynamic freeway travel times based on vehicle trajectory construction. *2012 15th International IEEE Conference on Intelligent Transportation Systems*. 2012. p. 576-581. DOI: 10.1109/itsc.2012.6338825.

[4] Kim TG, Choi HT, Ko NY. Localization of a robot using particle filter with range and bearing information. *2013 10th International conference on ubiquitous robots and ambient intelligence (URAI).* 2013. p. 368-370. DOI: 10.1109/urai.2013.6677389.

[5] Zhang B, et al. Predicting vehicle trajectory via combination of model-based and data-driven methods using Kalman filter. *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering.* 2023;09544070231161846. DOI: 10.1177/09544070231161846.

[6] Farahi F, Yazdi HS. Probabilistic Kalman filter for moving object tracking. *Signal Processing: Image Communication.* 2020;82:115751. DOI: 10.1016/j.image.2019.115751.

[7] Agrawal A, et al. Accurate prediction and estimation of 3d-repetitive-trajectories using Kalman filter, machine learning and curve-fitting method for high-speed target interception. *Artificial Intelligence for Robotics and*

Autonomous Systems Applications. 2023;93-122. DOI: 10.1007/978-3-031-28715-2_4.

[8] Lin CY, Kau LJ, Chan CY. Bimodal extended Kalman filter-based pedestrian trajectory prediction. *Sensors*. 2022;22(21):8231. DOI: 10.3390/s22218231.

[9] Chen Y, Shu L, Wang L. Traffic flow prediction with big data: A deep learning based time series model. *2017 IEEE conference on Computer Communications Workshops (INFOCOM WKSHPS)*. 2017;1010-1011. DOI: 10.1109/infcomw.2017.8116535.

[10] Chen Y, Hu C, Wang J. Human-centered trajectory tracking control for autonomous vehicles with driver cut-in behavior prediction. *IEEE Transactions on Vehicular Technology.* 2019;68(9):8461-8471. DOI: 10.1109/tvt.2019.2927242.

[11] Dai S, Li L, Li Z. Modeling vehicle interactions via modified LSTM models for trajectory prediction. *IEEE Access.* 2019;7:38287-38296. DOI: 10.1109/access.2019.2907000.

[12] Lin L, et al. Long short-term memory-based human-driven vehicle longitudinal trajectory prediction in a connected and autonomous vehicle environment. *Transportation Research Record.* 2021;2675(6):380-390. DOI: 10.1177/0361198121993471.

[13] Yang C, Pei Z. Long-short term spatio-temporal aggregation for trajectory prediction. *IEEE Transactions on Intelligent Transportation Systems.* 2023;24(4):4114-4126. DOI: 10.1109/tits.2023.3234962.

[14] Fu E, et al. Temporal self-attention-based convlstm network for multivariate time series prediction. *Neurocomputing.* 2022;501:162-173. DOI: 10.1016/j.neucom.2022.06.014.

[15] Xin L, et al. Intention-aware long horizon trajectory prediction of surrounding vehicles using dual LSTM networks. *2018 21st International Conference on Intelligent Transportation Systems (ITSC).* 2018. p. 1441-1446. DOI: 10.1109/itsc.2018.8569595.

[16] Sajanraj T, et al. Passenger flow prediction from AFC data using station memorizing LSTM for metro rail systems. *Neural Network World.* 2021;(3). DOI: 10.14311/nnw.2021.31.009.

[17] Liu J, Wang Z, Xu M. Deepmtt: A deep learning maneuvering targettracking algorithm based on bidirectional LSTM network. *Information Fusion*. 2020;53:289-304. DOI: 10.1016/j.inffus.2019.06.012.

[18] Yu W, et al. Deepgtt: A general trajectory tracking deep learning algorithm based on dynamic law learning. *IET Radar, Sonar & Navigation*. 2021;15(9):1125-1150. DOI: 10.1049/rsn2.12111.

[19] Siami-Namini S, Tavakoli N, Namin AS. The performance of LSTM and BiLSTM in forecasting time series. *2019 IEEE International conference on big data (Big Data).* 2019. p. 3285-3292. DOI: 10.1109/bigdata47090.2019.9005997.

[20] Yin J, Ning C, Tang T. Data-driven models for train control dynamics in high-speed railways: Lag-LSTM for train trajectory prediction. *Information Sciences.* 2022;600:377-400. DOI: 10.1016/j.ins.2022.04.004.

[21] Yang CH, et al. Ais-based intelligent vessel trajectory prediction using Bi-LSTM. *IEEE Access*. 2022;10:24302-24315. DOI: 10.1109/access.2022.3154812.

[22] Lin L, et al. Vehicle trajectory prediction using LSTMs with spatial-temporal attention mechanisms. *IEEE Intelligent Transportation Systems Magazine*. 2021;14(2):197-208. DOI: 10.1109/mits.2021.3049404.

[23] Jaderberg M, et al. Population based training of neural networks. *arXiv preprint arXiv:1711.09846*. 2017. DOI: 10.48550/arXiv.1711.09846.

叶海雄，栾开荣，杨楣，张希靓，周悦

基于DeepPBM-Attention的港口集装箱卡车的轨迹预测

摘要：

大多数目标跟踪算法均基于模型驱动的方法，受制于人为动机未知而影响算法准确性。本研究旨在解决港口集装箱卡车的定位在被遮挡时存在短暂误差的问题，提出一种基于数据驱动方法的车辆轨迹预测算法。该方法通过预处理海量GPS数据和训练DeepLSTM-Attention模型，并结合PBT（Population Based Training)算法优化网络超参数，从而提高预测水平运转集卡车辆轨迹的准确性。轨迹数据为真实港口采集的定位数据，这项研究分别在9条不同的轨迹段上对比了传统方法如卡尔曼滤波、机器

学习方法如支持向量回归(SVR)和常见的LSTM网络，结果表明基于PBT优化的Deep-LSTM-Attention模型（DeepPBM-Attention）的轨迹预测方法在均方根误差（RMSE），平均绝对误差（MAE），F1得分和轨迹重建误差（tre）上的表现均优于其他方法，性能相较于LSTM网络提升了约40%。本研究所提出的基于数据驱动方法的目标轨迹预测算法具有高精度和实用性，可有效应用于港口水平运转车辆的定位预测。

关键字：
港口集卡车；轨迹预测；基于群体的神经网络训练；深层长短期记忆神经网络；注意力机制