

Article

Using Deep Learning to Forecast Maritime Vessel Flows

Xiangyu Zhou ^{1,2,*} , Zhengjiang Liu ¹, Fengwu Wang ¹, Yajuan Xie ^{2,3} and Xuexi Zhang ⁴

¹ Navigation College, Dalian Maritime University, Dalian 116026, China; liuzhengjiang@dlmu.edu.cn (Z.L.); wangfw650105@163.com (F.W.)

² Centre for Maritime Studies, National University of Singapore, Singapore 118414, Singapore; isexy@nus.edu.sg

³ Department of Industrial Systems Engineering and Management, National University of Singapore, Singapore 117576, Singapore

⁴ School of Automation, Guangdong University of Technology, Guangzhou 510006, China; zxxnet@gdut.edu.cn

* Correspondence: zhou.x.y@outlook.com

Received: 13 February 2020; Accepted: 18 March 2020; Published: 22 March 2020



Abstract: Forecasting vessel flows is important to the development of intelligent transportation systems in the maritime field, as real-time and accurate traffic information has favorable potential in helping a maritime authority to alleviate congestion, mitigate emission of GHG (greenhouse gases) and enhance public safety, as well as assisting individual vessel users to plan better routes and reduce additional costs due to delays. In this paper, we propose three deep learning-based solutions to forecast the inflow and outflow of vessels within a given region, including a convolutional neural network (CNN), a long short-term memory (LSTM) network, and the integration of a bidirectional LSTM network with a CNN (BDLSTM-CNN). To apply those solutions, we first divide the given maritime region into $M \times N$ grids, then we forecast the inflow and outflow for all the grids. Experimental results based on the real AIS (Automatic Identification System) data of marine vessels in Singapore demonstrate that the three deep learning-based solutions significantly outperform the conventional method in terms of mean absolute error and root mean square error, with the performance of the BDLSTM-CNN-based hybrid solution being the best.

Keywords: maritime vessel flows; intelligent transportation systems; deep learning

1. Introduction

Forecasting traffic flows of vessels has been recognized as a challenging task in the maritime intelligent transportation system, since it could be affected by various complex factors [1,2]. Accurate and timely traffic information is of significant importance to both maritime managers and individual vessels, as it not only helps the former to better conduct port planning [3,4], alleviate congestion [5,6], mitigate emission of GHG (greenhouse gases) and improve public security [7,8], but also enables the latter to better operate the ship navigation system and plan a route [9,10], so as to avoid collisions and reduce the potential cost due to late arrival [11].

Various solutions have been studied for forecasting maritime traffic flow. Wang et al. [12] developed a vessel flow prediction model using a back propagation (BP) neural network, in which training and data sampling were conducted based on a residual analysis. Haiyan and Youzhen [13] came up with a hybrid scheme for vessel flow forecasting, which integrated an RBF (radial basis function) neural network, grey forecasting and auto-regression into a framework of support vector regression (SVR). The experimental results based on a real dataset demonstrated that the combined

method significantly outperformed each individual. Li et al. [14] conceived a hybrid method centering around a robust support vector regression (RSVR) model. In that method, a chaotic cloud simulated annealing genetic algorithm was first developed to optimize the parameters of RSVR, then the kernel principal component analysis (KPCA) algorithm was utilized to finalize the input vectors. Afterwards, the hybrid framework was established, and the test based on a real dataset justified its advantages over others. Yang et al. [15] developed an enhanced hybrid RBF neural network (EHRBF-NN) to forecast the traffic flow of vessels. The method was robust in that it incorporated a regression tree and a particle swarm optimization (PSO) algorithm into the RBF neural network. More specifically, the regression tree was exploited to calculate the parameters (i.e., centers and radius) of RBF, and the PSO was designated to address the issue of potential over-fitting and update the weights of the neural network. Liu et al. [16] proposed a two-step method to predict vessel flows. Particularly, in the first step, they separated vessel traffic flow into low-rank and sparse components by adopting a non-convex model. Then in the second step, the ARIMA (auto-regressive integrated moving average) and WNN (wavelet neural network) were explored to predict the flows based on the low-rank and sparse components from the first step. He et al. [17] proposed an improved Kalman model for short-term vessel traffic flow prediction. In that model, a preliminary prediction was first done using a regression analysis, which was then adopted to replace the transfer equation of the Kalman filter. Xiao et al. [11] came up with a probabilistic method for maritime traffic forecasting. The method was novel in that it took both waterway pattern knowledge and vessel motion stability into account. In particular, the waterway pattern was learnt via a lattice-based DBSCAN algorithm which also helped narrow down the problem scale, and the motion behavior was modeled through a kernel density estimation, thus, the combination of them was supposed to successfully predict the traffic flow.

Although some successes have been achieved in maritime traffic flow forecasting, most of the methods above are shallow and outdated to some extent, given the fact that the deep learning approach has been unarguably deemed as the state-of-the-art in many areas, such as computer vision, natural language processing, and transportation. The promising advantages brought by deep learning can be summarized as two aspects: (1) deep neural networks can be designed to solve end-to-end tasks, which directly takes the raw data as input without elaborating the intermediate feature extraction; (2) it usually achieves significantly higher accuracy in comparison with conventional methods. Despite the fact that numerous deep learning approaches have been applied in traffic flow or traffic condition forecasting, most of them focus on the land transportation instead of maritime. Lv et al. [18] proposed a deep learning framework to predict the freeway traffic flow, by exploring the big data from road infrastructure sensors. At the heart of that approach was a stacked auto-encoder (SAE) model, which was adopted to learn the latent traffic flow features by inherently taking spatial and temporal correlations into account. Chen et al. [19] implemented a typical stacked LSTM (long short-term memory) network to forecast the traffic congestion, which was applied to a realistic navigation map. The results demonstrated that the proposed approach was able to effectively learn the hidden patterns of traffic flows. Cui et al. [20] proposed a deep stacked bidirectional and unidirectional LSTM (SBU-LSTM) network to predict the traffic speed, which considered both forward and backward dependencies of the traffic data. More specifically, in that approach, a bidirectional LSTM (BDLSTM) layer was developed to learn the spatial features and bidirectional temporal relationships by exploiting the historical data, as such it was able to grasp the dynamic nature of the traffic. Polson and Sokolov [21] came up with a deep learning architecture to predict the short-term traffic flow. Particularly, that architecture integrated a linear model with ℓ_1 regularization and a branch of tanh layers, the first layer of which was designed to identify the spatio-temporal relations among predictors and other layers. As a consequence, it was able to capture the sharp nonlinear patterns of the traffic flow. Yu et al. [22] designed a spatio-temporal deep neural network to predict the short-term traffic speeds. Enlightened by the research outcomes of motion detection in the field of computer vision, they first segmented the road network into grids, then the traffic information inside each grid was input to a convolutional neural network (CNN) to learn the spatial relationship. After that, the output of which was further

taken as the input of a LSTM network, to capture the temporal relationship. Zhang et al. [23] proposed a deep spatio-temporal residual network (ST-ResNet) to forecast the citywide crowd inflow and outflow. In that approach, a residual neural network framework was first adopted to model the temporal closeness, period, and trend properties of the crowd flows, then a sequence of residual convolutional units were designed to model the spatial relationships of crowd flows, and lastly, the output of those residual neural networks were aggregated as the preceding layer to the final output layer of real crowd flows. Guo et al. [24] presented a framework for predictor fusion to forecast the short-term traffic condition. In the framework, three strategies were employed to evaluate the fusion performance, i.e., average fusion, weighted fusion and kNN fusion. Experimental results based on real dataset verified the significant advantages of the fusion method over the stand-alone ones. Liu and Chen [25] developed a novel prediction model for passenger flow using deep learning approach. The core of the model was the combination of a pre-trained unsupervised SAE with a supervised deep neural network, which was supposed to well extract the hierarchical features, so that the passenger flow for any periods from Monday to Sunday could be successfully forecast. Ke et al. [26] devised a deep learning approach to predict the road traffic congestion. The approach was distinguished in that it utilized visual signals to learn the traffic flow, and moving object detection and moving speed calculation were done by integrating a CNN and a Gaussian mixture model (GMM).

Obviously, the applications of deep learning in land transportation have been investigated far more extensively than that of maritime. On one hand, large-scale data for the land transportation is not that hard to obtain compared with maritime transportation, which is the foundation of applying the deep learning techniques. On the other hand, unlike the vessels in the sea, the mobilities of cars or people on land are mostly restricted to streets or rails, therefore, the traffic pattern for which is comparatively easy to be revealed. Moreover, we would like to note that, a variety of research with respects to the deep learning in maritime have been conducted, however most of them focus on vessel type identification [27], trajectory prediction or reconstruction [28], anomaly detection [29,30] and collision avoidance [31–33]. The problem of how to adopt deep learning to forecast the traffic flows for maritime has not yet been studied.

Therefore, in this paper, centering around the deep learning approach, we propose three different solutions to forecast the inflow and outflow of vessels within a given marine region. More specifically, the three solutions are featured by a CNN, a LSTM network, and the integration of a bidirectional LSTM network with a CNN (BDLSTM-CNN), respectively. In particular, with respects to the BDLSTM-CNN based solution, each input will be first fed into the convolutional layers, then they will go through the forward layer and backward layer of a bidirectional LSTM network. As such, this hybrid solution is supposed to coherently learn the spatial and temporal dependencies pertaining to the vessel flows. Moreover, to apply the three deep learning based solutions into practice, we divide the given marine areas into $M \times N$ grids. Then all the solutions are implemented to forecast the inflow and outflow for each grid. Experimental results based on the real AIS data for a given area in Singapore show that the three deep learning based solutions significantly outperform the conventional method, in which the hybrid solution BDLSTM-CNN achieves the best performance.

The remainder of the paper is organized as follows. Section 2 introduces the preliminaries first and then elaborates the structures and logic of the three deep learning based solutions. Section 3 presents the comprehensive experimental results and analysis. Section 4 concludes the paper and states the future works.

2. Deep Learning Based Solutions

In this section, we first introduce the preliminaries regarding the vessel flow forecasting problem. Then we present three deep learning based solutions, i.e., CNN, RNN, and BDLSTM-CNN, respectively.

2.1. Preliminaries

Our target is to forecast the inflow and outflow of vessels within a given area. To this end, we first divide this area into $M \times N$ grids. Thus, the inflow and outflow at a time step can be represented by two matrix, respectively, and the element of which refers to the counts of vessels entering or leaving the corresponding grid during that time step. In this regard, our task comes down to predict the inflow matrix and outflow matrix using the historical traffic flow data. Formally, given the historical observations $\{X_t | t = 0, 1, \dots, n - 1\}$, our task is to predict $X_{t+\tau}$, where $\tau \geq 1$ and is an integer. In particular, $X_t = [I_t, O_t]$, where I_t and O_t represent the inflow matrix and outflow matrix at time step t , respectively. Taking Figure 1 as an example, at time step t , two vessels will enter grid g_2 , and one vessel will leave, then $I_t(g_2) = 2$, and $O_t(g_2) = 1$, respectively.

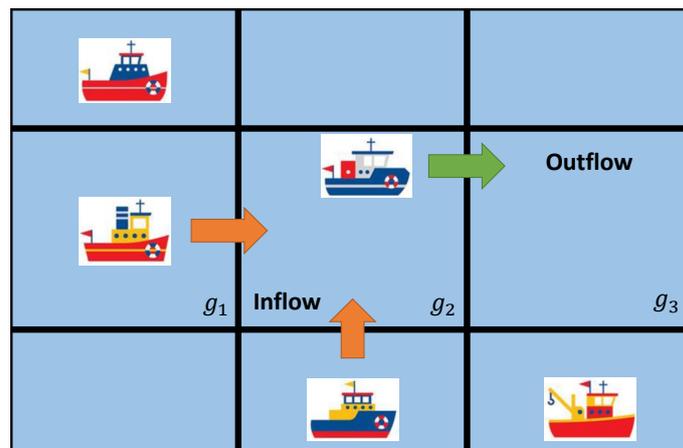


Figure 1. Illustration of the Inflow and Outflow.

2.2. CNN Based Solution

Convolutional neural network has been successfully applied in many areas, such as video analysis [34] and transportation [35], which is characterized by its strong capability of capturing the spacial representation. CNN can be employed here to handle the traffic flow prediction problem for two reasons. On one hand, the flow of vessels in a grid could be affected by the neighbor or distant grids, and CNN has the potential to capture this dependency, as the convolution operation inside CNN somehow can help predict the movement of vessels from the spatial perspective. On the other hand, the inflow and outflow for the whole given area are represented as matrix in our problem, which is the most inherent format of inputs to CNN. And the structure of the proposed CNN-based solution is depicted in Figure 2.

Accordingly, this CNN based solution has m channels, which correspond to the m historical observations as the inputs. Meanwhile, all the channels share the same weights. In particular, each input contains both the inflow and outflow matrix, and each input channel includes two convolutional layers, two ReLU layers, one batch normalization (BN) layer, and one dropout layer. Among them, the convolutional layers play the most important role, which are used to learn spacial features of different levels [22]. The underlying logic of convolutional layer is expressed as follows:

$$o_r^l = \sum_k W_{kr}^l * o_k^{l-1} + b_k^l \quad (1)$$

where o_r^l refers to the output of the r_{th} filter in the l_{th} layer; o_k^{l-1} refers to the output of the k_{th} filter of the preceding layer; W_{kr}^l and b_k^l refers to the weights and bias; $*$ refers to the convolution operation. Besides, the BN layer is used to scale the range of feature values; the ReLU layer acts as the activation function; the Dropout layer is used to select the salient features from the receptive region, so as to avoid redundant features and reduce the scale of the computation; and the output layer generates both

inflow and outflow matrix that we would like to predict. To adopt the CNN based solution to solve the vessel flow forecasting problem, we take X_{t-m+1} to X_t as the inputs at time step t and $X_{t+\tau}$ as the output Y_t .

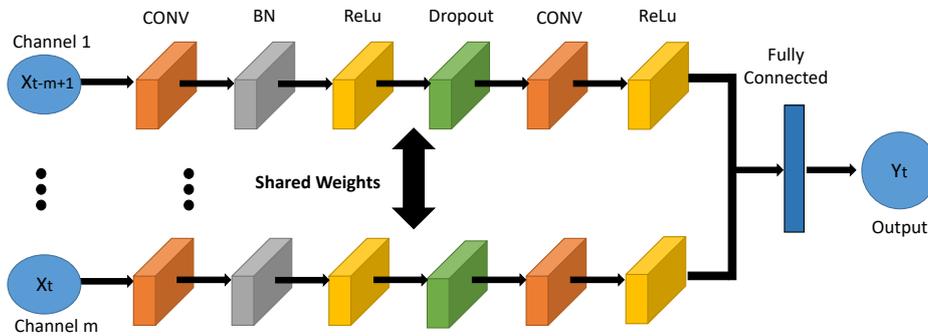


Figure 2. Illustration of the CNN based Solution.

2.3. LSTM Based Solution

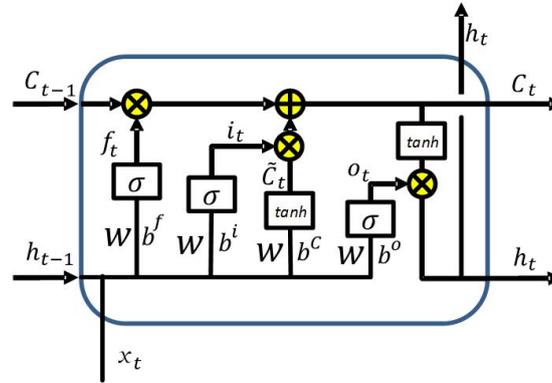
Due to the capability of sequential and temporal modeling, recurrent neural network (RNN) also has been successfully applied to many challenging practices, such as natural language processing, stock forecasting and crowd density prediction [36]. However, traditional RNN sometimes suffers from the issue of vanishing and exploding gradient if the learning sequence is long. To address this issue, a variant of RNN, termed LSTM (Long short-term memory) model, was devised, which utilized memory cells with various gates to preserve useful information for long-term dependencies [37]. In view of the fact that the vessel flow for a given area can be considered as the classical time series with temporal dependency, the LSTM network is supposed to well capture the temporal correlations in the flow forecasting problem.

More specifically, the structure of a single layer LSTM network is depicted in Figure 3a. And the LSTM network updates itself at time step t as follows [38]:

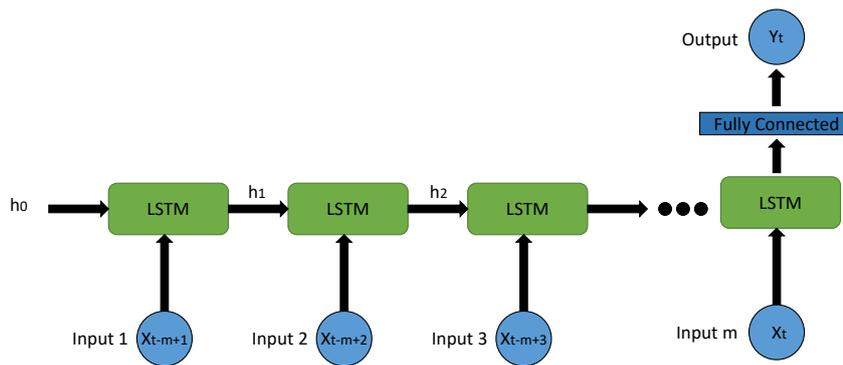
$$\begin{aligned}
 f_t &= \sigma(W^f[h_{t-1}, x_t] + b_f) \\
 i_t &= \sigma(W^i[h_{t-1}, x_t] + b_i) \\
 \tilde{C}_t &= \tanh(W^C[h_{t-1}, x_t] + b_C) \\
 C_t &= f_t * C_{t-1} + i_t * \tilde{C}_t \\
 o_t &= \sigma(W^o[h_{t-1}, x_t] + b_o) \\
 h_t &= o_t * C_t,
 \end{aligned} \tag{2}$$

where h_t is the hidden state; i_t , f_t and o_t refer to the input gate, forget gate and output gate, respectively; \tilde{C}_t and C_t refer to the input modulation gate and memory gate, respectively. $\{W^f, W^i, W^C, W^o\}$ are the weights, and $\{b^f, b^i, b^C, b^o\}$ are the biases for the corresponding gates; $\sigma(\cdot)$ and $\tanh(\cdot)$ are sigmoid and hyperbolic tangent activation functions, respectively. The memory cell unit C_t contains two components, i.e., previous memory cell unit C_{t-1} modulated by f_t and \tilde{C}_t , which is modeled by the current input, and previous hidden state, modulated by the input gate i_t [38]. The essence of sigmoidal operation for i_t and f_t normalizes themselves into the scope of $[0,1]$. Particularly, they could be deemed as knobs that LSTM learns to selectively forget its previous memory or consider its current input.

In a similar way, the output gate o_t models the transfer from memory cells to hidden states. On the basis of these mechanisms, the LSTM network is supposed to learn complex and temporal dynamics that exist in sequential vessel movement measurements, engendering a satisfactory performance for vessel flow forecasting.



(a) Structure of LSTM Network



(b) Structure of LSTM based Solution

Figure 3. Structure of the LSTM network, and the LSTM based Solution.

With the fundamental logic of the LSTM network, we build up the complete structure of the LSTM based solution, which is depicted in Figure 3b. In this solution, there are m inputs at the time step t , i.e., $X_t, X_{t-1}, \dots, X_{t-m+1}$, all of which are fed into the LSTM networks in order. Afterwards, the outputs of the LSTM network are connected to a fully connected network layer, then Y_t is regarded as the final output, which is set as $X_{t+\tau}$.

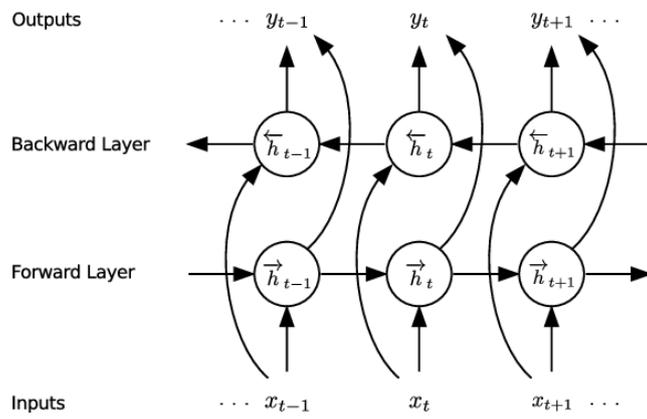
2.4. BDLSTM-CNN Based Hybrid Solution

As we have stated previously, the vessel flow forecasting relies on both spatial and temporal dependencies. Therefore, it would be desirable to integrate the CNN and the LSTM network into a comprehensive framework, to coherently learn the spatiotemporal relationships for the vessel flow forecasting. However, with respect to the temporal feature learning, we exploit a more powerful recurrent neural network, namely, bidirectional LSTM, to replace the traditional unidirectional LSTM.

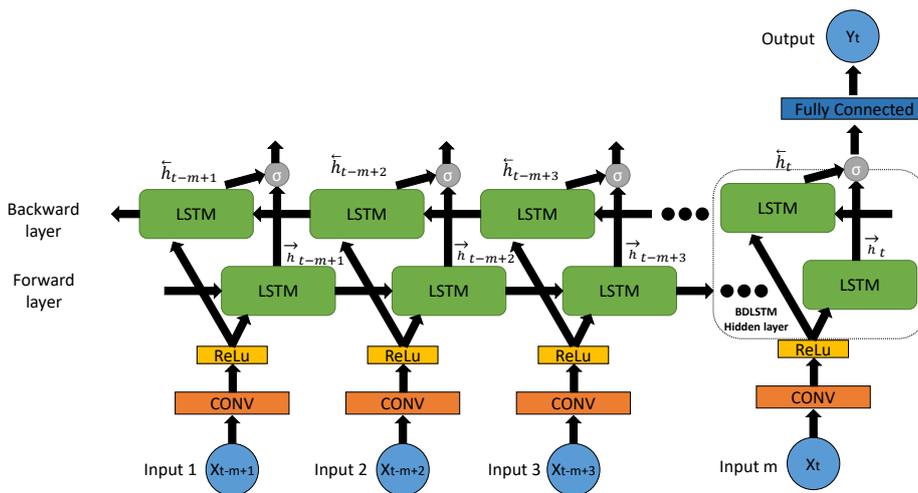
2.4.1. Bidirectional LSTM

The idea of BDLSTMs is derived from the bidirectional RNN, which is used to address a crucial issue that, the conventional RNN is only able to make use of the previous context, thus they only learn representations from previous time steps [39]. However, we might have to learn representations from future time steps to better understand the context and eliminate ambiguity sometimes. As a consequence, the bidirectional RNN was developed to achieve this goal, which processed sequence data in both forward and backward directions with two separate hidden layers. And both of them are connected to the same output layer, as depicted in Figure 4a. More specifically, the bidirectional RNN separates the hidden layer into two parts, forward state sequence \vec{h} and backward state sequence \overleftarrow{h} , and they are computed as follows [40]:

$$\begin{aligned} \vec{h}_t &= \mathcal{H}(W_{x\vec{h}}x_t + W_{\vec{h}\vec{h}}\vec{h}_{t-1} + b_{\vec{h}}) \\ \overleftarrow{h}_t &= \mathcal{H}(W_{x\overleftarrow{h}}x_t + W_{\overleftarrow{h}\overleftarrow{h}}\overleftarrow{h}_{t+1} + b_{\overleftarrow{h}}) \\ y_t &= W_{\vec{h}y}\vec{h}_t + W_{\overleftarrow{h}y}\overleftarrow{h}_t + b_y. \end{aligned} \tag{3}$$



(a) Structure of Bidirectional RNN



(b) Structure of the BDLSTM-CNN based Hybrid Solution

Figure 4. Structure of Bidirectional RNN, and the Hybrid Solution.

Accordingly, the deep bidirectional RNN can be established by stacking multiple RNN hidden layers on top of each other. Each hidden state sequence, h^n , is replaced by the forward and backward, \overrightarrow{h}^n and \overleftarrow{h}^n . On the other hand, deep bidirectional LSTM (BDLSTM) can be achieved based on the integration of deep bidirectional RNN and the LSTM network. Thus, the BDLSTM is supposed to have the ability to model the deep representation of long-span features [39].

2.4.2. The Hybrid Solution

With the framework of bidirectional LSTM, CNN is further integrated to build up a hybrid solution, which is depicted in Figure 4b. In this framework, each input first goes through a convolution layer and a Relu layer, to learn the spatial dependency. Then the output will be fed into the bidirectional LSTM network (i.e, the dotted box in Figure 4b), as the input for both forward sequence and backward sequence. Afterwards, the outputs of the bidirectional LSTM hidden layers are further connected to a sigmoid activation function, which is followed by a fully connected layer. Lastly, the final output is regarded as the predicted vessel flow.

To summarize from a high level perspective, at each time step t , there are m inputs that will be fed into the synthesized networks, i.e., X_{t-m+1} , X_{t-m+2} , ..., and X_t , each of which consists of both inflow matrix and outflow matrix. At the same time, the output Y_t is set as $X_{t+\tau}$. Normally, τ is equal to 1, however it can also take much larger integer values. On the other hand, since the vessel flow forecasting is a regression task in nature, root mean square error or mean absolute error are usually adopted as the loss function. However, in this hybrid solution, we consider a new loss function for final objective optimization, namely, smooth ℓ_1 loss [41], which is expressed as follows:

$$\text{loss}(x, y) = \begin{cases} 0.5(x - y)^2, & \text{if } |x - y| < 1; \\ |x| - 0.5, & \text{otherwise.} \end{cases} \quad (4)$$

Smooth ℓ_1 loss, also called Huber loss, is usually less sensitive to abnormal inputs, and also helps the networks prevent gradient exploding to some extent [41].

3. Experimentation and Evaluation

In this section, we conduct experimentation in different settings to test the proposed solutions, and demonstrate their advantages over the baseline. Particularly, we first introduce the data processing and experimental settings, then we compare and evaluate the three deep learning based solutions. Finally, we compare the three solutions with a conventional method, i.e., support vector regression (SVR).

3.1. Data Processing and Experimentation Setup

We use the AIS (Automatic Identification System) data of maritime vessels to perform the forecasting task, which is an automatic tracking system that uses transponders on ships to generate trajectory-related information [30]. The AIS data for a vessel contains many useful attributes regarding its movement and mobility, which can be obtained from <https://www.marinetraffic.com>. Here we mainly exploit the information of vessel ID, coordinates (i.e., longitude and latitude) and time stamp. The testing filed we selected is an rectangle marine area southeast to Singapore, which is shown as the red rectangle in Figure 5. The location of the left upper point is (1.287979°, 103.892723°), and that of the right lower point is (1.235027°, 103.996817°). In all the experimentation we conducted, we uniformly divided this area into 7×7 grids (i.e., $M = N = 7$), which means that both inflow matrix I_t and outflow matrix O_t have a size of 7×7 .



Figure 5. The Selected Marine Area: 7×7 Grids.

Then we process and analyze the AIS data, and identify all the vessels entering and leaving those grids accordingly. Particularly, the AIS data we collected for the given area lasts about 31 days, from 01-10-2013 to 31-10-2013. We set the duration for each time step as 5 min. Then we have 8525 samples for each grid (with some data missed), which include the amount of vessels entering and leaving the given grid. We divide them into training dataset and testing dataset according to the time order, i.e., from 01-10-2013 to 25-10-2013, and from 26-10-2013 to 31-10-2013, which include 6305 and 2220 samples, respectively. In addition, we implement all the experimentation using pytorch, in a laptop with Intel i7 CPU, 8G RAM, and Nvidia GTX 1060.

3.2. Error Performance for the Deep Learning Based Solutions

We set $Y_t = X_{t+1}$ and $Y_t = X_{t+2}$, respectively, and conduct experimentation according to the above configurations. We use two types of errors to measure the performance of the proposed solutions, i.e., mean absolute error (MAE) and root mean square error (RMSE), since they are the two most important metrics for regression problems. We also use their normalized forms to further evaluate the performance, i.e., NMAE and NRMSE. All the results for the three deep learning based solutions are shown in Figures 6 and 7. Before looking into them, we would like to note that the errors in those figures are calculated by considering inflow and outflow together. Additionally, all the results below are obtained based on the testing data.

From Figure 6a,b we can see that, as the training iteration increases, the MAE for testing data of the three deep learning based solutions drop quickly. With respect to both $Y_t = X_{t+1}$ and $Y_t = X_{t+2}$, all the solutions seem to converge after about 85 iterations. Pertaining to the two different Y_t settings, the BDLSTM-CNN based solution always achieves the lowest MAE, which are around 1.11 and 1.14, respectively. In contrast, the LSTM based solution achieves the second lowest MAE of 1.15 and 1.25, and the CNN based solution achieves the third lowest MAE of 1.29 and 1.35. It makes sense that the BDLSTM-CNN based hybrid solution engenders the best results as it coherently explores the spatial and temporal dependencies regarding the vessel flows, by integrating both CNN and LSTM. Moreover, with respect to the temporal expression, it exploits the bidirectional LSTM to take advantage of both past and future information, which is supposed to get better results than the unidirectional LSTM. Comparing the CNN based solution and the LSTM based solution, we can observe that, the latter presents a better performance, which is probably due to that the temporal relationship is much relatively important in comparison with the spatial relationship in this case. Although that the former solution takes every five preceding flow matrix to generate the succeeding one, and the convolution layer also has the potential to predict the movement of vessels from the spatial perspective, the structure of CNN itself does not have a scheme to capture the temporal correlation. This also might be able to explain the oscillation in the curve of CNN based solution, because it does not have any mechanism to capture the seasonality pattern in the vessel flows. On the other hand, there are

only slight oscillations for LSTM based solution and BDLSTM-CNN based solution, which justifies their favorable capability of handling the seasonality in sequence. Comparing Figure 6a,b, we can observe that, the performance of the three solutions slightly deteriorates for $Y_t = X_{t+2}$. It is reasonable as one can always make a more accurate forecast for a near future than for a far future. However, the hybrid solution still achieves the lowest error. Additionally, we would like to highlight that, the difference between the LSTM based solution and the hybrid solution is larger for $Y_t = X_{t+2}$ than that of $Y_t = X_{t+1}$. It happened because in the case of $Y_t = X_{t+2}$, the bidirectional LSTM may bring more useful information from the backward layer, in comparison with the solo forward layer in the LSTM based solution.

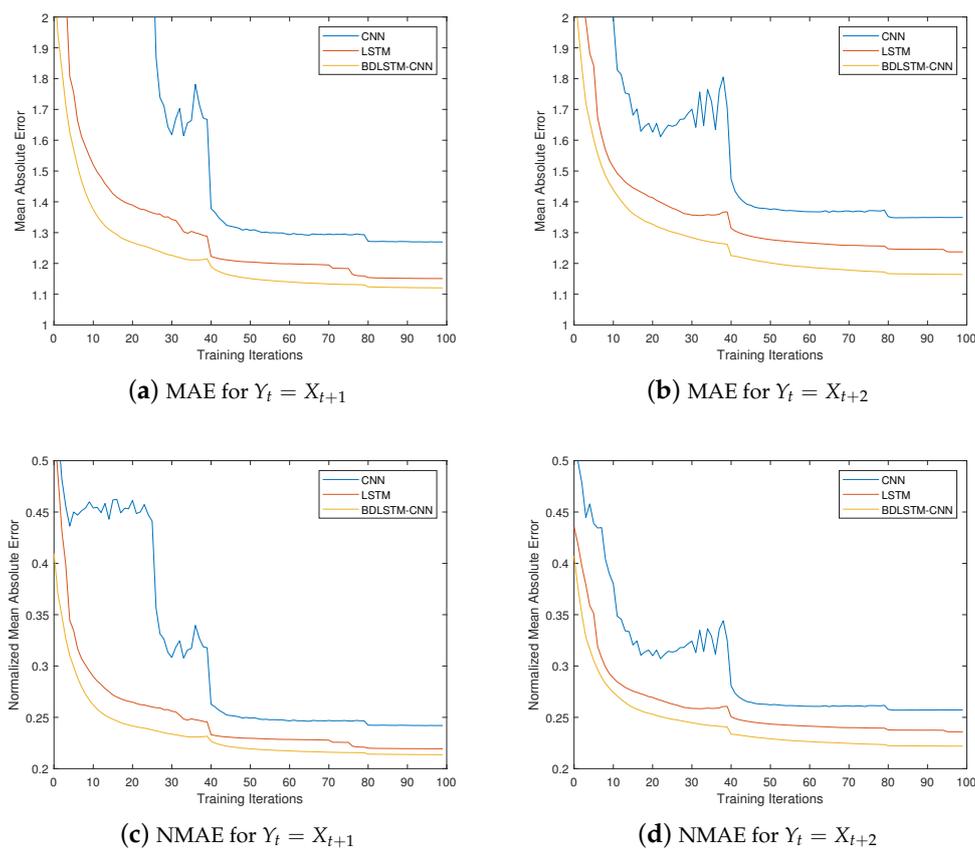


Figure 6. Performance Regarding MAE and NMAE.

From Figure 6c,d, we can see that they share similar pattern with Figure 6a,b. This is normal because they are simply the normalized version of the latter, which can be regarded as a kind of error rate. The difference in the curve shape comes from the fact that we only show part of errors in Figure 6a,b, in order to highlight the region of interests. Looking into Figure 6c,d, we can observe that, the three solutions achieve error rates (in terms of MAE) of around 24.5%, 22.5% and 22.0%, respectively, for $Y_t = X_{t+1}$, and around 26%, 24% and 22.5%, respectively, for $Y_t = X_{t+2}$, with the hybrid solution being the lowest. The error rates in terms of MAE would be decreased about 1.5%, 1.5% and 0.5%. So it can be concluded that the results of three deep learning based solutions considerably are better than the traditional approach expressed in MAE, with the performance of the BDLSTM-CNN based hybrid solution being the smallest.

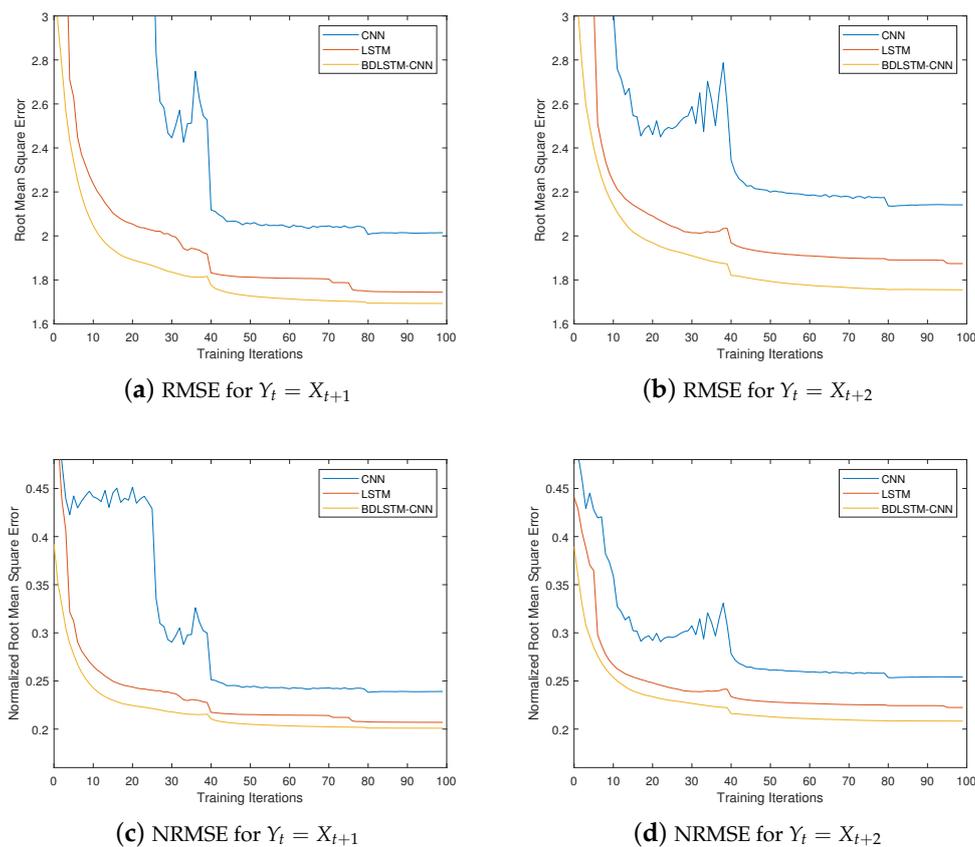


Figure 7. Performance Regarding RMSE and NRMSE.

Likewise, we also use the RMSE (root mean square error) to further evaluate the proposed solutions, and the results are depicted in Figure 7, which share similar pattern with that of Figure 6. From Figure 7a,b, we can observe that, the three solutions achieve a RMSE of around 2.01, 1.75 and 1.68 for $Y_t = X_{t+1}$, respectively, and around 2.18, 1.88 and 1.75 for $Y_t = X_{t+2}$, respectively, with the hybrid solution being the lowest for both cases. From Figure 7c,d, we can see that, the three solutions achieve error rates (in terms of RMSE) of around 24%, 21% and 20%, respectively, for $Y_t = X_{t+1}$, and around 26%, 23.5% and 21%, respectively, for $Y_t = X_{t+2}$, with the hybrid solution being the lowest for both cases. Similarly, the error rates in terms of RMSE will be lowered by 2%, 2.5% and 1%. Consequently, it can be found that the outcomes of three deep learning based solutions noticeably outperformed than the traditional approach in terms of RMSE, with the performance of the BDLSTM-CNN based hybrid solution being the lowest.

Combining the performance of both MAE and RMSE in Figures 6 and 7, it can be possibly proving that the BDLSTM-CNN based hybrid solution outperforms the CNN based solution and the LSTM based solution. However, considering that, (1) all the solutions only adopt five historical data to predict a new one, (2) all the errors are derived based on integrating inflow and outflow together, we believe that the performance of all the solutions are sufficiently satisfactory, although superiority and inferiority exist among them.

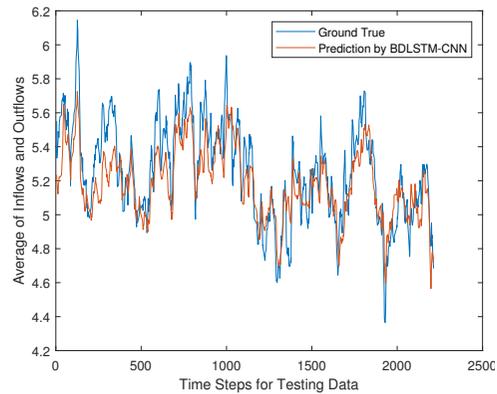
3.3. Breakdown Performance for the Hybrid Solution

Since the BDLSTM-CNN based solution has the best overall performance, we look into this hybrid solution and analyze its prediction capability in a breakdown perspective. To this end, we plot the curves of the average forecasting value of the inflow and outflow for the whole given region, and the forecasting value of inflow and outflow for a given grid, respectively. We would like to note that, all the results below are obtained based on the testing data.

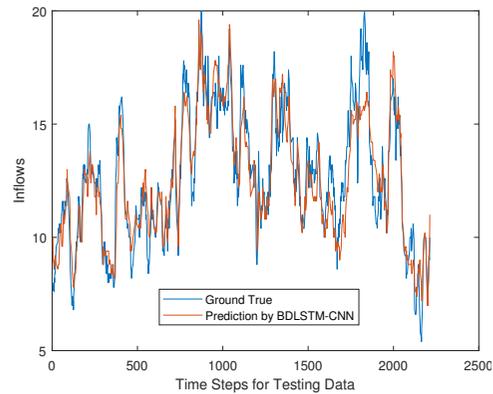
We first plot the above forecasting values against the ground truth for $Y_t = X_{t+1}$ in Figure 8. From Figure 8a we can see that, the ground truth value of the average by considering inflow and outflow together, changes dramatically as time goes on. However, the BDLSTM-CNN based solution can well capture those trends, such as the sharp changes at time step 100, 1300, 1700 and 2200, respectively, because this hybrid solution combines the advantages of the capability of CNN and BDLSTM, to learn the spatial and temporal features in a unified way. Nevertheless, we observe some imperfect tracking, such as time step 250, which does not catch the crest. However, considering that the ground truth is the average of both inflow and outflow for the whole given area, some minor errors are tolerable. We also plot the forecasting curves of a single grid for inflow and outflow in Figure 8b,c, respectively, by taking grid (6,6) as an example. From Figure 8b, we observe that, the hybrid solution can catch most of the crests and troughs for the inflow curve, no matter how sharp or smooth they are, such as the crest at time step 400 and the trough at time step 1250. An unsatisfactory forecasting is found as well at time step 1800, where a sharp crest is missed. From Figure 8c, we observe that the hybrid solution can also successfully follow both sharp and smooth crests and troughs for the outflow curve, such as the crest at time step 1050, and trough at time step 1450. Although the hybrid solution mismatched the ground truth at some points, such as time step 1800, most of the failures are tolerable.

We then continue to plot the forecasting values against the ground truth for $Y_t = X_{t+2}$ in Figure 9. From Figure 9a we can see that, the hybrid solution is still able to basically track the ground truth value, although slight deterioration is observed in comparison with the performance in Figure 8a. This can be explained by the fact that, a near future is comparatively easier to be forecast than a distant future. Despite the slight deterioration in the average forecasting, the hybrid solution still shows competitive performance for inflow and outflow forecasting pertaining to grid (6,6), which mostly well captures all the sharp or smooth crests and troughs.

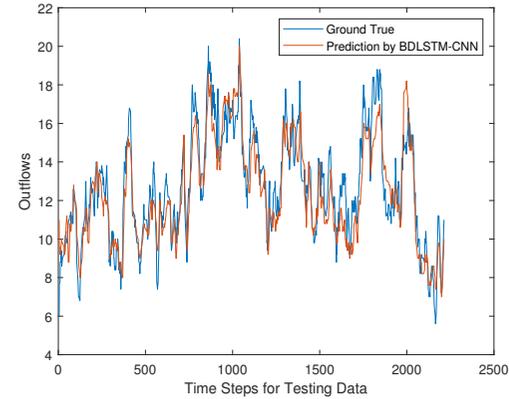
Combining the results in both Figures 8 and 9, we can conclude that the hybrid solution has strong capability of forecasting the inflow and the outflow of vessels. Though the inflow and outflow changed dramatically, the trend can be well captured by the hybrid solution.



(a) Average of Inflow and Outflow

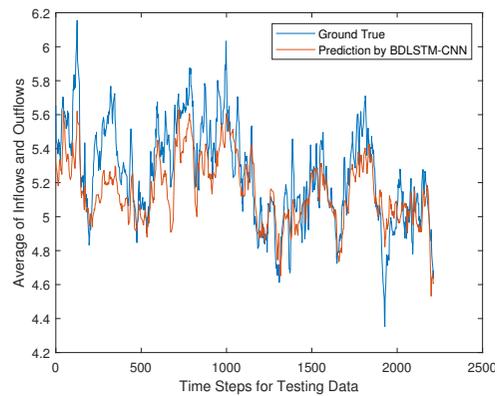


(b) Inflow for Grid (6,6)

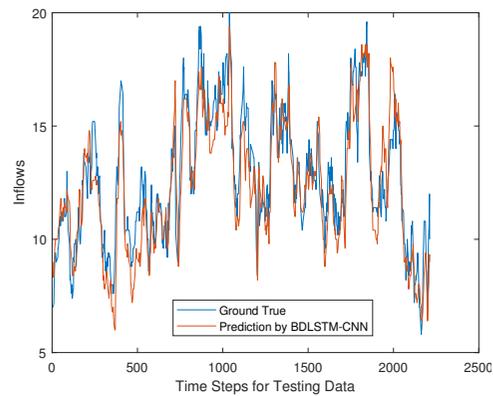


(c) Outflow for Grid (6,6)

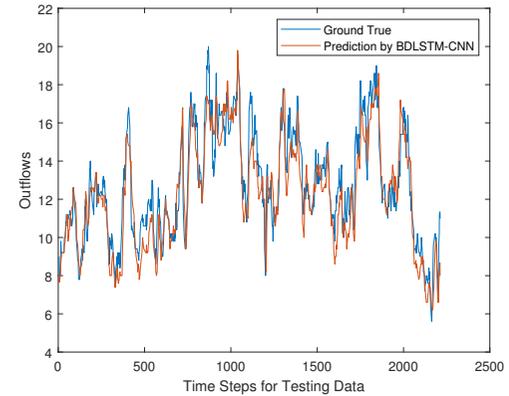
Figure 8. Prediction for $Y_t = X_{t+1}$.



(a) Average of Inflow and Outflow



(b) Inflow for Grid (6,6)



(c) Outflow for Grid (6,6)

Figure 9. Prediction for $Y_t = X_{t+2}$.

3.4. Comparison with the Conventional Method

In this subsection, we compare the performance of the three deep learning based solutions with a conventional method, namely the support vector regression (SVR) based approach. In particular, we use the python sklearn package to implement this function, and utilize the same training dataset to optimize its parameters, and then do the evaluation using the same testing dataset, the results of which are recorded in Table 1. Since the normalized errors reflect a kind of error rate in forecasting, we mainly concentrate on the normalized forms of the two errors, i.e., NMAE and NRMSE.

Table 1. Comparison for Error Rates (%)

	SVR	CNN	LSTM	BDLSTM-CNN
$Y_t = X_{t+1}$, NMAE	50.1	24.5	22.5	22.0
$Y_t = X_{t+2}$, NMAE	50.5	26.0	24.0	22.5
$Y_t = X_{t+1}$, NRMSE	51.2	24.0	21.0	20.0
$Y_t = X_{t+2}$, NRMSE	51.4	26.0	23.5	21.0

From Table 1 we can see that, the error rates of the SVR method are around 51% for both measurements of MAE and RMSE, almost twice as much as the three deep learning based solutions. The inferiority comes from the fact that, the SVR method does not have a sophisticated scheme to learn the underlying spatial dependency, or the long-term temporal dependency. In contrast, CNN has a convolution layer, and LSTM or BDLSTM has a memory and gate to handle the two situations [42,43], respectively. On the other hand, the BDLSTM-CNN based hybrid solution always achieves the lowest error rates of 20% to 22.5%, which implies a forecasting accuracy of 77.5% to 80%. Considering that we only use five historical data to predict a new one, and both inflow and outflow of vessels change dramatically, the performance achieved by the hybrid solution is sufficiently satisfactory.

4. Conclusions and Future Work

In this paper, we propose three deep learning based solutions to forecast the flow of maritime vessels. To apply the deep learning approach, we first divide the given marine area into $M \times N$ grids, then we predict both the inflow and outflow of vessels for each grid. In particular, the three solutions are characterized by a CNN, a LSTM, and the integration of BDLSTM and CNN, respectively. When testing them with the real AIS data of vessels, the hybrid solution based on BDLSTM-CNN achieves the best performance in terms of mean absolute error (MAE) and root mean square error (RMSE), both error rates will be decreased by 1–2% compared with other methods, as it is able to coherently learn the spatial and temporal representations for both the inflow and outflow. On the other hand, when further comparing them with a conventional approach, the three deep learning based solutions significantly outperform the SVR method. We would like to note that, $N = 7$ is only an empirical value for the given area in this paper, and it can vary with different scenarios. Moreover, the number of columns does not need to be the same with the rows.

However, the proposed approach still needs to be improved further and tested more extensively. In future, we will work on the following directions: (1) we will consider an attention model in the BDLSTM-CNN based solution to further improve the performance; (2) we will explore more relevant features, such as weather, date, kinematics and kinetics of ship, variable speeds of ship movement, collision avoidance maneuvers; (3) we will use more than five data points to predict a new one, and also forecast the flow longer-time away, such as $Y_t = X_{t+3}, X_{t+4}, \dots$; (4) we will investigate multi-agent based methods and reinforcement learning based methods to solve the route planning for maritime vessel flows; (5) we will try to apply integral of multiplied by absolute error (ITAE) and integral of time multiplied by square error (ITSE) performance criteria to assess the quality of vessel traffic forecasting.

Author Contributions: Conceptualization, X.Z. (Xiangyu Zhou), Z.L.; methodology, X.Z. (Xiangyu Zhou), F.W.; software, X.Z. (Xiangyu Zhou), Y.X., X.Z. (Xuexi Zhang), writing—original draft preparation, X.Z. (Xiangyu Zhou); writing—review and editing, Z.L., F.W., Y.X. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by China-APEC Cooperation Fund/Research on integration of education, training and employment of seafarers in APEC region (grant number 01871901), National Key Research and Development Program of China (grant number 2018YFC0810402).

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

References

1. Zissis, D.; Xidias, E.K.; Lekkas, D. Real-time vessel behavior prediction. *Evol. Syst.* **2016**, *7*, 29–40. [[CrossRef](#)]
2. Xiao, Z.; Fu, X.; Zhang, L.; Ponnambalam, L.; Goh, R.S.M. Data-driven multi-agent system for maritime traffic safety management. In Proceedings of the IEEE 20th International Conference on Intelligent Transportation Systems (ITSC), Yokohama, Japan, 16–19 October 2017; pp. 1–6.
3. Fazi, S.; Roodbergen, K.J. Effects of demurrage and detention regimes on dry-port-based inland container transport. *Transp. Res. A* **2018**, *89*, 1–18. [[CrossRef](#)]
4. Zhou, C.; Li, H.; Lee, B.K.; Qiu, Z. A simulation-based vessel-truck coordination strategy for lighterage terminals. *Transp. Res. A* **2018**, *95*, 149–164. [[CrossRef](#)]
5. Chen, L.; Hopman, H.; Negenborn, R.R. Distributed model predictive control for vessel train formations of cooperative multi-vessel systems. *Transp. Res. A* **2018**, *92*, 101–118. [[CrossRef](#)]
6. Kazemi, S.; Abghari, S.; Lavesson, N.; Johnson, H.; Ryman, P. Open data for anomaly detection in maritime surveillance. *Expert Syst. Appl.* **2013**, *40*, 5719–5729. [[CrossRef](#)]
7. Ruiz-Aguilar, J.J.; Turias, I.; Moscoso-López, J.A.; Jiménez-Come, M.J.; Cerbán, M. Forecasting of short-term flow freight congestion: A study case of Algeciras Bay Port (Spain). *DYNA* **2016**, *83*, 163–172. [[CrossRef](#)]
8. Hänninen, M.; Banda, O.A.V.; Kujala, P. Bayesian network model of maritime safety management. *Expert Syst. Appl.* **2014**, *41*, 7837–7846. [[CrossRef](#)]
9. Chen, C.H.; Khoo, L.P.; Chong, Y.T.; Yin, X.F. Knowledge discovery using genetic algorithm for maritime situational awareness. *Expert Syst. Appl.* **2014**, *41*, 2742–2753. [[CrossRef](#)]
10. Kisialiou, Y.; Gribkovskaia, I.; Laporte, G. Robust supply vessel routing and scheduling. *Transp. Res. A* **2018**, *90*, 366–378. [[CrossRef](#)]
11. Xiao, Z.; Ponnambalam, L.; Fu, X.; Zhang, W. Maritime traffic probabilistic forecasting based on vessels' waterway patterns and motion behaviors. *IEEE Trans. Intell. Transp. Syst.* **2017**, *18*, 3122–3134. [[CrossRef](#)]
12. Wang, C.; Zhang, X.; Chen, X.; Li, R.; Li, G. Vessel traffic flow forecasting based on BP neural network and residual analysis. In Proceedings of the 4th International Conference on Information, Cybernetics and Computational Social Systems (ICCS), Dalian, China, 24–26 July 2017; pp. 350–354.
13. Haiyan, W.; Youzhen, W. Vessel traffic flow forecasting with the combined model based on support vector machine. In Proceedings of the International Conference on Transportation Information and Safety (ICTIS), Wuhan, China, 25–28 June 2015; pp. 695–698.
14. Li, M.W.; Han, D.F.; Wang, W.L. Vessel traffic flow forecasting by RSVR with chaotic cloud simulated annealing genetic algorithm and KPCA. *Neurocomputing* **2015**, *157*, 243–255. [[CrossRef](#)]
15. Yang, L.; Hao, Y.; Liu, Q.; Zhu, X. Ship traffic volume forecast in bridge area based on enhanced hybrid radial basis function neural networks. In Proceedings of the International Conference on Transportation Information and Safety (ICTIS), Wuhan, China, 25–28 June 2015; pp. 38–43.
16. Liu, R.W.; Chen, J.; Liu, Z.; Li, Y.; Liu, Y.; Liu, J. Vessel traffic flow separation-prediction using low-rank and sparse decomposition. In Proceedings of the IEEE 20th International Conference on Intelligent Transportation Systems (ITSC), Yokohama, Japan, 16–19 October 2017; pp. 1–6.
17. He, W.; Zhong, C.; Sotelo, M.A.; Chu, X.; Liu, X.; Li, Z. Short-term vessel traffic flow forecasting by using an improved Kalman model. *Clust. Comput.* **2017**; pp. 1–10. [[CrossRef](#)]
18. Lv, Y.; Duan, Y.; Kang, W.; Li, Z.; Wang, F.Y.; others. Traffic flow prediction with big data: A deep learning approach. *IEEE Trans. Intell. Transp. Syst.* **2015**, *16*, 865–873. [[CrossRef](#)]
19. Chen, Y.Y.; Lv, Y.; Li, Z.; Wang, F.Y. Long short-term memory model for traffic congestion prediction with online open data. In Proceedings of the IEEE 19th International Conference on Intelligent Transportation Systems (ITSC), Rio de Janeiro, Brazil, 1–4 November 2016; pp. 132–137.

20. Cui, Z.; Ke, R.; Wang, Y. Deep Stacked Bidirectional and Unidirectional LSTM Recurrent Neural Network for Network-wide Traffic Speed Prediction. In Proceedings of the 6th International Workshop on Urban Computing (UrbComp), Halifax, Nova Scotia, Canada, 14 August 2016.
21. Polson, N.G.; Sokolov, V.O. Deep learning for short-term traffic flow prediction. *Transp. Res. A* **2017**, *79*, 1–17. [[CrossRef](#)]
22. Yu, H.; Wu, Z.; Wang, S.; Wang, Y.; Ma, X. Spatiotemporal recurrent convolutional networks for traffic prediction in transportation networks. *Sensors* **2017**, *17*, 1501. [[CrossRef](#)] [[PubMed](#)]
23. Zhang, J.; Zheng, Y.; Qi, D. Deep Spatio-Temporal Residual Networks for Citywide Crowd Flows Prediction. In Proceedings of the 31st AAAI Conference on Artificial Intelligence (AAAI), San Francisco, CA, USA, 4–9 February 2017; pp. 1655–1661.
24. Guo, F.; Polak, J.W.; Krishnan, R. Predictor fusion for short-term traffic forecasting. *Transp. Res. A* **2018**, *92*, 90–100. [[CrossRef](#)]
25. Liu, L.; Chen, R.C. A novel passenger flow prediction model using deep learning methods. *Transp. Res. A* **2017**, *84*, 74–91. [[CrossRef](#)]
26. Ke, X.; Shi, L.; Guo, W.; Chen, D. Multi-Dimensional Traffic Congestion Detection Based on Fusion of Visual Features and Convolutional Neural Network. *IEEE Trans. Intell. Transp. Syst.* **2018**, *20*, 2157–2170. [[CrossRef](#)]
27. Nguyen, D.; Vadaine, R.; Hajduch, G.; Garello, R.; Fablet, R. A Multi-task Deep Learning Architecture for Maritime Surveillance using AIS Data Streams. In Proceedings of the 2018 IEEE International Conference on Data Science and Advanced Analytics (DSAA), Turin, Italy, 1–3 October 2018.
28. Jiang, X.; Liu, X.; de Souza, E.N.; Hu, B.; Silver, D.L.; Matwin, S. Improving point-based AIS trajectory classification with partition-wise gated recurrent units. In Proceedings of the International Joint Conference on Neural Networks (IJCNN), Anchorage, AK, USA, 14–19 May 2017, pp. 4044–4051.
29. Mazzarella, F.; Vespe, M.; Alessandrini, A.; Tarchi, D.; Aulicino, G.; Vollero, A. A novel anomaly detection approach to identify intentional AIS on-off switching. *Expert Syst. Appl.* **2017**, *78*, 110–123. [[CrossRef](#)]
30. Tu, E.; Zhang, G.; Rachmawati, L.; Rajabally, E.; Huang, G.B. Exploiting AIS data for intelligent maritime navigation: A comprehensive survey from data to methodology. *IEEE Trans. Intell. Transp. Syst.* **2018**, *19*, 1559–1582. [[CrossRef](#)]
31. Liu, X.; Jin, Y. Design of Transfer Reinforcement Learning Mechanisms for Autonomous Collision Avoidance. In Proceedings of the 8th International Conference on Design Computing and Cognition, Politecnico di Milano, Italy, 2–4 July 2018; pp. 339–358.
32. Perera, L.P. Autonomous Ship Navigation Under Deep Learning and the Challenges in COLREGs. In Proceedings of the ASME 37th International Conference on Ocean, Offshore and Arctic Engineering, Madrid, Spain, 17–22 June 2018; pp. 1–10.
33. Cheng, Y.; Zhang, W. Concise deep reinforcement learning obstacle avoidance for underactuated unmanned marine vessels. *Neurocomputing* **2018**, *272*, 63–73. [[CrossRef](#)]
34. Xu, Z.; Yang, Y.; Hauptmann, A.G. A discriminative CNN video representation for event detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 1798–1807.
35. Yu, B.; Yin, H.; Zhu, Z. Spatio-Temporal Graph Convolutional Networks: A Deep Learning Framework for Traffic Forecasting. In Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI-18), Stockholm, Sweden, 13–19 July 2018, pp. 3634–3640.
36. Zonoozi, A.; Kim, J.J.; Li, X.L.; Cong, G. Periodic-CRN: A Convolutional Recurrent Model for Crowd Density Prediction with Recurring Periodic Patterns. In Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI-18), Stockholm, Sweden, 13–19 July 2018; pp. 3732–3738.
37. Greff, K.; Srivastava, R.K.; Koutník, J.; Steunebrink, B.R.; Schmidhuber, J. LSTM: A search space odyssey. *IEEE Trans. Neural Netw. Learn. Syst.* **2017**, *28*, 2222–2232. [[CrossRef](#)] [[PubMed](#)]
38. Chen, Z.; Zhang, L.; Jiang, C.; Cao, Z.; Cui, W. WiFi CSI Based Passive Human Activity Recognition Using Attention Based BLSTM. *IEEE Trans. Mob. Comput.* **2018**, *18*, 2714–2724. [[CrossRef](#)]
39. Yu, Z.; Ramnarayanan, V.; Suendermann-Oeft, D.; Wang, X.; Zechner, K.; Chen, L.; Tao, J.; Ivanou, A.; Qian, Y. Using bidirectional LSTM recurrent neural networks to learn high-level abstractions of sequential features for automated scoring of non-native spontaneous speech. In Proceedings of the IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU), Scottsdale, AZ, USA, 13–17 December 2015; pp. 338–345.

40. Graves, A.; Jaitly, N.; Mohamed, A.R. Hybrid speech recognition with deep bidirectional LSTM. In Proceedings of the IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU), Olomouc, Czech Republic, 8–12 December 2013; pp. 273–278.
41. Berrada, L.; Zisserman, A.; Kumar, M.P. Smooth Loss Functions for Deep Top-k Classification. In Proceedings of the International Conference on Learning Representations, Vancouver, BC, Canada, 30 April–3 May 2018; pp. 1–25.
42. Liu, Y.; Zheng, H.; Feng, X.; Chen, Z. Short-term traffic flow prediction with Conv-LSTM. In Proceedings of the 2017 9th International Conference on Wireless Communications and Signal Processing (WCSP), Nanjing, China, 11–13 October 2017; pp. 1–6.
43. Duan, Z.; Yang, Y.; Zhang, K.; Ni, Y.; Bajgain, S. Improved deep hybrid networks for urban traffic flow prediction using trajectory data. *IEEE Access* **2018**, *6*, 31820–31827. [[CrossRef](#)]



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).