

Article

# Deep Learning-Based Caution Area Traffic Prediction with Automatic Identification System Sensor Data

Kwang-Il Kim  and Keon Myung Lee \* 

Department of Computer Science, Chungbuk National University, Cheongju 28644, Korea; kikim82@cbnu.ac.kr

\* Correspondence: kmlee@cbnu.ac.kr; Tel.: +82-43-261-2263

Received: 30 July 2018; Accepted: 17 September 2018; Published: 19 September 2018



**Abstract:** In a crowded harbor water area, it is a major concern to control ship traffic for assuring safety and maximizing the efficiency of port operations. Vessel Traffic Service (VTS) operators pay much attention to caution areas like ship route intersections or traffic congestion area in which there are some risks of ship collision. They want to control the traffic of the caution area at a proper level to lessen risk. Inertial ship movement makes swift changes in direction and speed difficult. It is hence important to predict future traffic of the caution area earlier on so as to get enough time for control actions on ship movements. In the harbor area, VTS stations collect a large volume of Automatic Identification Service (AIS) sensor data, which contain information about ship movement and ship attributes. This paper proposes a new deep neural network model called Ship Traffic Extraction Network (STENet) to predict the medium-term traffic and long-term traffic of the caution area. The STENet model is trained with AIS sensor data. The STENet model is organized into a hierarchical architecture in which the outputs of the movement and contextual feature extraction modules are concatenated and fed into a prediction module. The movement module extracts the features of overall ship movements with a convolutional neural network. The contextual modules consist of five separated fully-connected neural networks, each of which receives an associated attribute. The separation of feature extraction modules at the front phase helps extract the effective features by preventing unrelated attributes from crosstalking. To evaluate the performance of the proposed model, the developed model is applied to a real AIS sensor dataset, which has been collected over two years at a Korean port called Yeosu. In the experiments, four methods have been compared including two new methods: STENet and VGGNet-based models. For the real AIS sensor dataset, the proposed model has shown 50.65% relative performance improvement on average for the medium-term predictions and 57.65% improvement on average for the long-term predictions over the benchmark method, i.e., the SVR-based method.

**Keywords:** sensor data; deep learning; traffic prediction; automatic identification data sensor; convolution neural network; VTS

## 1. Introduction

Maritime traffic has been increasing over the past decades with economic growth, and the scale of ports has been accordingly increased. Ship traffic routes, which are the sea lanes regularly used by ships to travel, are crowded with many inbound and outbound ships, especially in harbor sea areas. Ships can neither swiftly change their course, nor swiftly change their speed. Hence, careful monitoring and proactive control for ship traffic are important to avoid maritime accidents such as ship collisions, stranding, capsizing, and so on.

Ship traffic monitoring and control are difficult due to the following characteristics of maritime traffic in a harbor water area. First, no visible lanes are available on ship traffic routes. Second, ship traffic routes sometimes are merged, split or cross each other [1]. Third, inertial ship movement makes

swift changes in direction and speed hard. Fourth, ship movement is affected not only by voyage schedule, but also by neighboring traffic and the environment [2].

For voyage and safety, many ships are equipped with various sensors like radar, GPS, Doppler log, Gyro compass, anemogram, and so on. In the water area, it is important for ships to broadcast their identifier and state information like position and movement to avoid maritime accidents and getting assistance. Those ships send out their AIS messages through an Automatic Identification System (AIS) transponder, which collects data from the installed onboard sensors [3]. A ship collects AIS messages aired by other ships to use them for safe navigation. Such collected data are called AIS sensor data, which contain various information about ship movement and attributes. The shore-side monitoring and controlling service stations, which are usually called VTSs (Vessel Traffic Service systems), collect AIS sensor data from their coverage and visualize the received data on the electronic navigational chart system for VTS operators. In addition, the VTSs provide ships with information about estimated traffic congestion and collision risks and enforce the regulations on navigation such as speed reduction, course change, and so on.

The on-duty VTS operators pay much attention to the caution area. A caution area means the crowded area at which many ship routes are merged, split or cross each other. In the caution area, there are some potential risks of ship collisions. To reduce collision risk, it is important to control the ship traffic at a proper level in the caution area [4]. To predict the traffic in the caution area at a future time point, we need some method to estimate the arrival time of a ship at the caution area. Experienced VTS operators have been trained in various similar situations for a long time and can predict future traffic in the caution area with some tools measuring the distance between two points. They take some control actions to maintain the caution area traffic at a proper level based on their estimation. This human-based approach is not good at medium-term and long-term predictions [5].

Real-time AIS sensor data are used firsthand for traffic surveillance at VTS stations. Once they are collected, they can be used later for some statistical analysis and machine learning tasks [6,7]. AIS sensor data are valuable assets with which new problem-solving methods might be developed.

Deep learning is a branch of machine learning that uses artificial neural networks with many layers, also known as deep neural networks, for approximating some function that fits well with the training data. The deep neural networks have the capability of automatically extracting and using relevant features appropriate for the target problem while being trained with the training data. There are various deep neural network models such as the Convolutional Neural Network (CNN), LSTM recurrent neural network, generative adversarial networks and capsule network [8]. Deep neural network models demand many training data because they have many parameters to be trained. Over the last ten years, deep learning techniques have demonstrated impressive performance improvement for difficult problems such as image and pattern classification, speech recognition, natural language processing tasks and signal processing tasks [9].

Various maritime traffic prediction methods have been developed [10–21]. Some methods are based on mathematical modeling [10,11]. It is difficult to develop satisfactory mathematical model-based traffic prediction because the traffic is not just governed by physical laws, but also by various regulations enforced in the harbor sea area. There are various machine learning-based prediction methods [12–21]. Deep neural networks are a powerful tool to attack some difficult problems like traffic prediction in a caution area. There are some works that have applied deep learning techniques in maritime applications. Nguyen et al. [22] proposed a recurrent neural network-based model to reconstruct trajectories, detect abnormal ship behaviors and identify ship type from AIS data. Gallego et al. [23] proposed a convolutional neural network-based method, which identified the type of ships in optical aerial images. We are yet unaware of any deep neural network-based maritime traffic density prediction methods.

This paper introduces a new deep neural model called STENet (Ship Traffic Extraction Network), which is trained with AIS sensor data and predicts the traffic of a caution area. The STENet model receives as the input both ship movement data and ship attribute data, which are extracted from AIS

sensor data. The network produces as the output the predicted number of ships in the caution area at the future time points in 20, 30, 40 and 50 min.

The remainder of the paper is organized as follows: Section 2 presents some related works, and Section 3 explains the characteristics of AIS sensor data and describes how to prepare the AIS sensor data for a prediction model construction. Section 4 proposes the new deep neural network model STENet for the future traffic predictions in the caution area. Section 5 shows the experimental results of the proposed model for a real dataset for a Korean harbor. Finally, we draw the conclusions in Section 6.

## 2. Related Works

There are various works for maritime traffic prediction [12–21]. Perera et al. [12] proposed both a neural network-based method that detects and tracks multiple ships by using the radar data collected on the shore-side station and a Kalman filter-based method that predicts ship trajectories from current ship data. Xu et al. [13] proposed a short-term position prediction method that uses a multi-layered perceptron trained with ships' position, course and speed data. Their trained model showed better accuracy in the ship movement prediction than the ship motion law-based method. Perera et al. (2010) [14] proposed a curvilinear motion model-based method to predict ocean-going ship trajectories and also proposed an extended Kalman filter-based algorithm to predict ship position, speed and acceleration.

There are some works to predict medium- and long-term maritime traffic using historical trajectories. Ristic et al. [15] proposed a method to predict individual ship's trajectory at the future time points in 10, 32 and 70 min, which extracts representative trajectories by applying an adaptive kernel density estimation method to historical trajectory data. Mazarella et al. [16] proposed a method to predict the ship trajectories and voyage times, which uses a particle filter-based simulation method and a velocity model. Their method uses the ship movement vector data extracted from AIS sensor data, but it does not take into account other neighboring ships' traffic data.

Machine learning techniques have been applied for predicting maritime traffic [5,17–19], which train some machine learning models for traffic prediction with maritime data instead of developing man-made mathematical models. Xiao et al. [17] proposed a method to extract ship traffic patterns from ocean-going ship trajectories data with a DBSCAN-based clustering algorithm and to estimate short-term and long-term traffic by applying a kernel density estimation technique for the extracted ship traffic patterns. Kim et al. [5,18] proposed a method to predict ship position, speed and course in harbor areas, which trains a support vector regression model with ship trajectory data including the information about position and speed and then uses the trained regression model along with a dead reckoning estimation model. Zhang et al. [19] proposed a traffic prediction method for narrow water passage, which uses a support vector machine-based technique combined with a genetic algorithm for metaheuristic search. The method takes into account the ship trajectory data like position, speed and course, but does not consider other important factors such as ship destination, ship type and size and pilotage.

There are several neural network-based traffic prediction models that use both ship trajectory data and other traffic-related factors. Gan et al. [20] proposed a ship traffic estimation method for narrow water passage. The method first trains a neural network model with a hidden layer, which determines clusters of ship trajectory data along with ship's speed, loading capacity, weight, maximum power and water level. Then, it uses the trained neural network model to choose the corresponding trajectory cluster to individual ship trajectories and then predicts the future traffic with the chosen trajectory clusters. Daranda [21] proposed a turning point-based path prediction method, which first identifies turning points with the DBSCAN-based clustering algorithm and then trains a multi-layered perceptron, which takes as input the ship information such as ship type, speed, course, length and position, and afterward outputs the next turning points. However, they proposed the prediction models, but did not present in detail the performance evaluation results on their methods. Although they tried to use both ship trajectory data and other traffic-related data, their neural

network models were shallow networks, which are generally recognized to be inferior to recent deep neural networks.

We propose a deep neural network-based traffic prediction method for the caution area, which is trained with data consisting of ship movements and ship attributes. The proposed method is unique in that it uses a deep neural network model to predict the future traffic in the caution area with reference to all the available data about ship movements and attributes over the entire harbor area. To evaluate the performance of the proposed method, we conducted some experiments to apply it to a large real traffic dataset.

### 3. Automatic Identification System Sensor Data

#### 3.1. Characteristics of Automatic Identification System Sensor Data

Ships with more than 300 gross tonnage with passengers are equipped with an AIS transponder device that broadcasts the ship's dynamic, static and voyage information. Table 1 shows some data items in AIS messages and their broadcasting rates. An AIS message contains dynamic information about the ship's movement such as position, speed, course and voyage status. The broadcasting rates of dynamic information messages depend on a ship's movement status. For example, dynamic information is broadcasted every 3.3 s when a ship changes its course at a voyage speed of less than or equal to 14 knots and every 2 s when the voyage speed is greater than 23 knots. An AIS message conveys the ship's static information such as ship name, call sign, ship type and ship specification. The static information does not change once the AIS device is installed on a ship. Voyage information can be conveyed in an AIS message, which is the semi-static data that do not change over one voyage from its departure port to its destination port. Voyage information includes freight information, ship draught, Estimated Time of Arrival (ETA), and so on. Because both ship's static information and voyage information do not change over one voyage, where their AIS messages are broadcasted at a low frequency, e.g., every 6 min, AIS messages are sometimes delayed for a considerable amount of time due to the radio communication environment. The delay is caused by AIS device malfunctions, radio interferences with neighboring ships' radio signals or geographic obstacles like islands [24]. AIS messages for dynamic information may sometimes contain invalid data due to the occasional malfunction of onboard AIS sensors, which are installed in a severe ship environment.

For ship traffic monitoring and controlling, VTS stations receive all AIS messages in their coverage areas with shore-based AIS devices. AIS messages contain the item of the message broadcast time, and hence, VTS stations can locate and keep track of the ships in their monitoring area. AIS messages occupy a time slot in their radio channel and are received in a stream data manner. Each message is encoded in the National Marine Electronics Association (NMEA) format. Figure 1 shows some examples of raw AIS messages in NMEA format. By parsing such raw AIS messages, we can extract the items shown in Table 1. Meanwhile, VTS stations save the received AIS messages in their storage for later use like safety and security analysis, forensic and maritime statistical analysis.

```
!AIVDM,1,1,,A,16SgdhPP00a@8dfDBT92GJnd2H>8
!AIVDM,1,1,,B,16SepN0P009@=qVDBDTEbwvf2D4>
!AIVDM,1,1,,A,16Sg@ePP019>m8HD5ipiUOvgrD7p
!AIVDM,1,1,,B,16Sa<KP0h49=QbpD4UrQ92Jb042t
!AIVDM,1,1,,A,16SWdM?0009=?prCu`Aowp:gp@=d
!AIVDM,1,1,,A,15?FeB000t9>sSpD5=odl:6f0@=e
```

Figure 1. Examples of Automatic Identification Service (AIS) messages.

**Table 1.** AIS sensor message information and update rates.

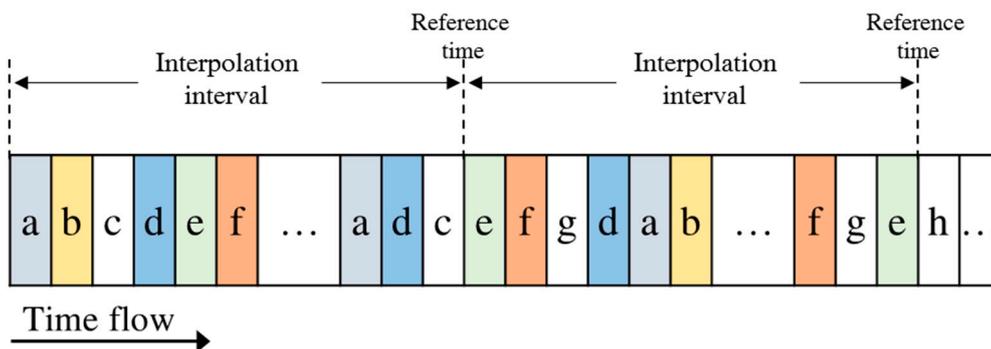
Type of Information	Information	Broadcasting Rate
Dynamic Information	Maritime Mobile Service Identity number (MMSI)	At anchor or moored (<3 kts): 3 min At anchor or moored (>3 kts): 10 s
	Ship position	Ship 0–14 kts: 10 s
	Speed over ground	Ship 0–14 kts and changing course: 3.3 s
	Course over ground	Ship 14–23 kts: 6 s
	Navigational status	Ship 14–23 kts and changing course: 2 s
	Time Etc.	Ship >23 kts: 2 s
	Static and Voyage-related Information	MMSI number
Ship type		
Length and beam		
Estimated time of arrival		
Destination Etc.		

3.2. Ship Movement Data Preparation

We are interested in developing a machine learning-based method to predict the traffic in the entire harbor area rather than a method to predict the movement of individual ships. We need to have both ship movement data and other ship movement data obtained at synchronized time points to predict the traffic at specific future time points [25]. Ship movement data are crucial because they represent the movement vectors of ships, which are essential in predicting the future locations of ships.

As mentioned in Section 3.1, the AIS messages are sometimes missing and their broadcasting intervals are different. Hence, the AIS data received at VTS stations are arranged in increasing order of their broadcasting time, as shown in Figure 2. To predict the traffic at a specific future time point, we are supposed to have movement data of the present time point. This means that all movement data should be synchronized at a specific time interval.

Figure 2 exemplifies some AIS messages sorted in increasing order of their broadcasting time where the letters in the boxes indicate the ship identifiers. We can see that the broadcasting times of received AIS messages are different from each other and the message arrival rates of ships are different from each other. To get movement data at specific intervals, we set an interpolation interval as shown in Figure 3 and interpolate movement data at the reference time points, which are the starting time of each interpolation interval.



**Figure 2.** A sequence of a received AIS messages.

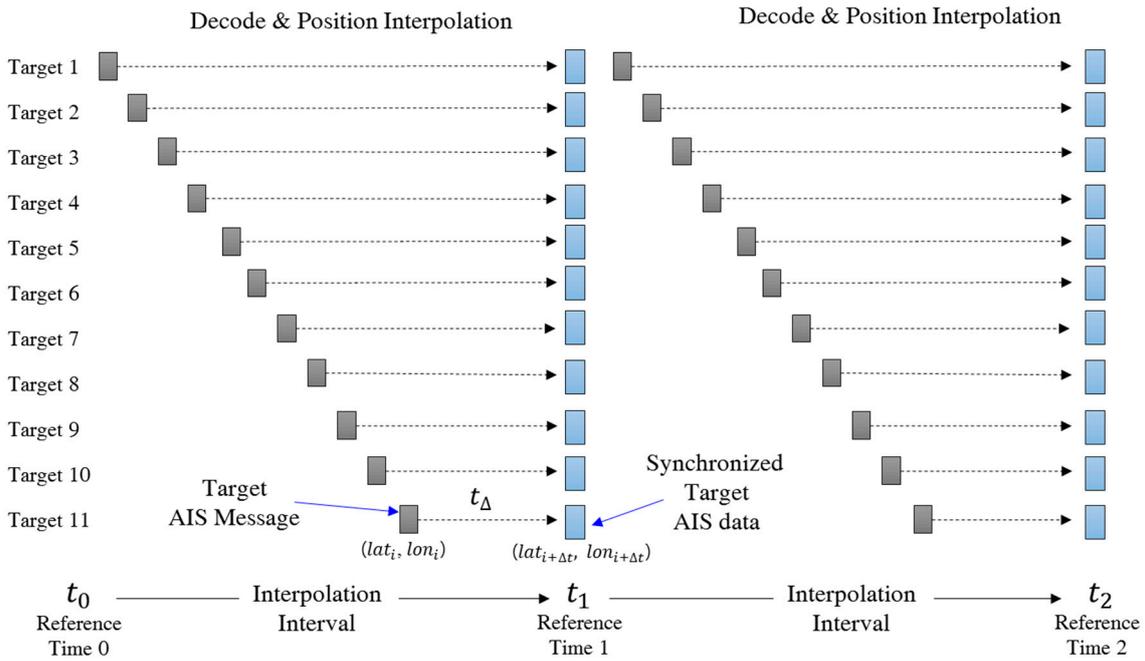


Figure 3. Synchronized AIS sensor data interpolation.

The ship movement data at reference time points are generated by the interpolation method as follows [18]: First, we remove duplicated AIS messages for the same ship except the most recent message in each interpolation interval. Then, we apply an interpolation method to obtain the ship position at the reference time point. Let  $t_k$  denote the  $k$ -th reference time point and  $time(M_i)$  denote the broadcasting time of the message  $M_i$ . When an AIS message  $M_i$  occurred in the interval between the  $k$ -th reference time and the  $k + 1$ -th reference time, i.e.,  $t_k < time(M_i) \leq t_{k+1}$ , the position  $[lat_i, lon_i]$  of the ship for  $M_i$  is replaced with the position at time  $t_{k+1}$ , where the position is determined by the interpolation with the motion vector, i.e., course  $\theta$  and speed  $v$ . Let  $\Delta t = t_{k+1} - time(M_i)$ . The following shows the interpolation equations for the new position at the reference time point [26].

$$\begin{aligned}
 lat_{i+\Delta t} &= \arcsin \left[ \sin(lat_i) \Delta \cos \frac{v \cdot \Delta t}{R} + \frac{\cos(m_k)}{\Delta t^2} \Delta \cos(lat_i) \Delta \sin \frac{V \cdot \Delta t}{R} \right], \\
 lon_{i+\Delta t} &= lon_i + \arctan \left[ \frac{\frac{\sin(m_k)}{\Delta t^2} \Delta \sin \frac{V \cdot \Delta t}{R} \Delta \cos(lat_i)}{\cos \frac{V \cdot \Delta t}{R} \Delta \sin(lat_i) \Delta \sin(lat_{i+\Delta t})} \right]
 \end{aligned} \tag{1}$$

Here,  $R$  indicates the radius of the earth, and  $m_k$  denotes the angle between the x-axis and the course direction  $\theta$ .  $m_k$  is computed using the course direction as follows:

$$m_k = \begin{cases} 90 - \theta & (0 \leq \theta < 90) \\ 90 + \theta & (90 \leq \theta < 180) \\ 270 - \theta & (180 \leq \theta < 270) \\ 270 + \theta & (270 \leq \theta < 360) \end{cases} \tag{2}$$

The speed of a ship is measured by the unit knot (abbreviated as kt); one knot is the speed at which the ship travels one nautical mile for one hour. The ship speed is ranged over the interval from 0 knot–30 knots. The range of the course value is 0–359°. The course angles of 359° and 0° look very different, but they are very close. When we use the angle values in deep neural network models, we have to convert them into another representation, which allows the difference to be used

for similarity computation. The new horizontal and vertical movement vectors ( $V_x, V_y$ ) are computed as follows:

$$V_x = \cos(m_k) \cdot v, V_y = \sin(m_k) \cdot v \quad (3)$$

### 3.3. Association of Ship Attribute Data with AIS Data

In maritime traffic prediction, it is necessary to have ship movement data with the attributes such as position, velocity and course. In addition, there are other traffic-related factors such as ship length, ship type, ship destination, Pilot Onboard (POB) and Caution Area Estimated Time of Arrival (CAETA). Ship movement data, ship length and type information are directly obtained from AIS data. However, ship destination, POB and CAETA data are obtained by processing AIS sensor data. Ship length is extracted from a static information message of each ship, which is important in the traffic prediction because the longer a ship, the heavier it is, and hence, heavy ships show late responses for speed-up, speed-down or turning operations. The ship length affects the ship traffic prediction especially when a ship changes its speed or changes its course. For the convenience of handling in the prediction model, the length is normalized to have a value in the interval [0,1]. The normalized ship length is computed by dividing the ship length by the maximum ship length.

Ship type code in AIS messages ranges over the integers 0–99, i.e., there are 100 categories of ships. In the traffic prediction, the detailed categories of ships are not needed. Hence, we group 100 categories into three macro-categories, i.e., cargo ships, tanker ships and other ships. Cargo ships load their freight above the deck and have a low block coefficient, so that their navigation has a low influence from the under-water fluid. Therefore, they can travel fast, but take more time to change a course than other types of ships [27]. Tanker ships have the tankers under the deck, which are loaded with crude oil and chemical products. They have a high block coefficient, and hence, they are usually slower, but take a shorter time to change course than cargo ships. Other ships indicate small-sized ships like a pilot boat, operation boat, fishing boat, tug boat, and so on. They may travel on an arbitrary route and even cross the regular routes, if needed. They can easily change their course and can reduce their speed in a short time. Due to these characteristics, we regroup the 100 ship types into cargo ship, tanker ship and other ship.

The proposed method is concerned with predicting the future traffic in the caution area. Hence, the destination of ships is also an important factor to influence the caution area traffic. If the destination of a ship is located at the berth across the caution area, the ship should pass through the caution area. If a destination is near the caution area, the ship slows down its speed to come alongside the berth. The destination information in AIS sensor data is neither the berth's name, nor the coordinate in the map, but a port name such as Port of South Louisiana and Busan Port. It is different to infer the destination berth from AIS sensor data. To estimate the destination berth for the training data, we examine AIS sensor data of each ship to see where the ship stops within a berth location range. Then, a ship destination is assigned with the ship location ( $s.lat, s.lon$ ). Once the traffic prediction model is designed, then the destination of a ship is obtained by the VTS operator from the port management information system.

POB information indicates whether a pilot embarks or disembarks. On pilot embarkation or disembarkation, the ship slows down, and hence, POB information affects the traffic. When a ship comes into a port or departs from a port, then it slows down to 5–6 knots, which is the boarding speed if a pilot is scheduled to be on board. When constructing the training data, we examine the AIS sensor data of each ship to determine whether a pilot has gotten on board. We decided that a pilot has gotten on board if the ship slowed down to 4–6 knots in the pilot location range. Figure 4 shows the procedure used to extract POB and destination information of AIS sensor data for constructing the training data.

CAETA indicates the time taken for a ship  $i$  to travel from the current location ( $s_i^t.lat, s_i^t.lon$ ) to the center of the caution area ( $c.lat, c.lon$ ), at the present speed  $s_i^t.spd$  at time  $t$ . Here, ( $s_i^t.lat, s_i^t.lon$ ) and

( $c.lat$ ,  $c.lon$ ) indicate the pairs of the latitudinal and longitudinal positions for the ship's location and the center of the caution area. CAETA can be computed by the following equation (Equation (4)):

$$CAETA = \frac{\sqrt{(s_i^t.lat - c.lat)^2 + (s_i^t.lon - c.lon)^2}}{s_i^t.spd} \quad (4)$$

In the training data, CAETA is normalized to be in the range [0,1] by dividing CAETA by the expected maximum time. If CAETA is greater than the expected maximum time, its normalized value is set to one.

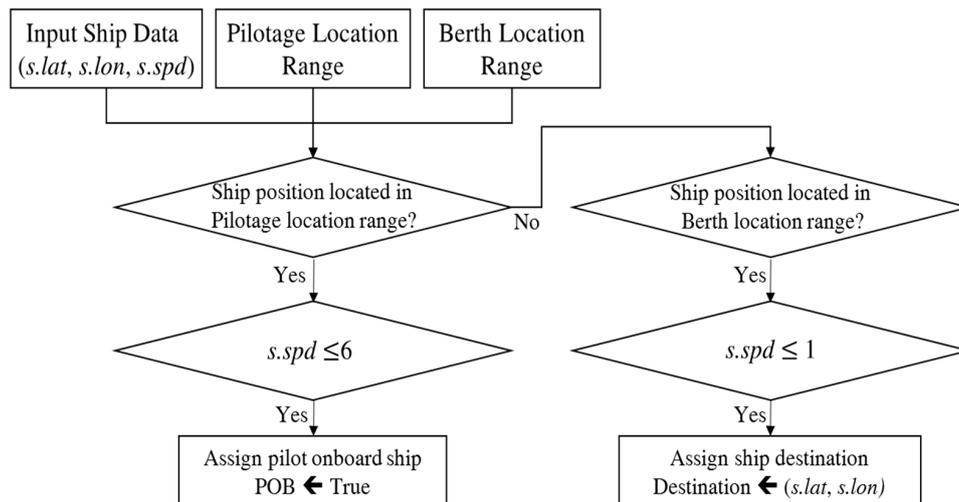


Figure 4. Pilot Onboard (POB) and destination information extraction procedure from AIS sensor data.

CAETA is useful information with which we can compute the estimated arrival time to the caution area under the condition that a ship maintains the present speed. Therefore, CAETA is taken as an input attribute for traffic density prediction in the caution area.

#### 4. The Proposed Ship Traffic Prediction Method

This section presents a new deep neural network model, called the STENet (Ship Traffic Extraction Network) model, to predict the future ship traffic at the caution area, which is trained with AIS sensor data. The proposed deep neural network uses a Convolutional Neural Network (CNN) [28] as its subnetwork. CNN is representative of a deep neural network model, which takes multiple channels of two-dimensional data as input and repeatedly transforms them in convolution operations and optionally in the pooling operations. CNN extracts valuable features for problem solving from the input data by repetitive and consecutive convolution and pooling operations. The output of CNN is typically served as an input to a Fully-Connected Neural Network (FCNN), i.e., a multilayered perceptron. Section 4.1 presents how to represent the training data for the STENet model. Section 4.2 describes the STENet architecture in detail.

##### 4.1. Encoding of AIS Data for the STENet Model

The STENet model predicts the number of ships in the caution area at a medium-term and long-term future time from the harbor traffic status at the moment. The prediction model is trained with training data constructed from AIS sensor data. In Sections 3.2 and 3.3, how the training data are constructed from AIS sensor data is presented. Because the information of ships is associated with geographical locations, the proposed method encodes the input part of the training data in 10 channels of a two-dimensional array, as shown in Figure 5. The training data for the STENet model consist of the movement vector, ship length, CAETA, destination, ship type and POB for input and the number of ships in the caution area at the designated future time point for output. The water area in a harbor

is partitioned into an equi-sized  $m \times m$  grid structure, each grid cell of which corresponds to a small water area and is associated with longitudinal and latitudinal coordinates. The grid size is designed enough to hold only one large ship with surrounding safety space, and it is assumed that at most one ship is located in a grid cell. In a two-dimensional array representation, each element corresponds to a grid cell in the water area. Hence, a grid cell is located by the index of the two-dimensional array. Each element in the two-dimensional array represents the information associated with the ship, if any, located at the corresponding grid cell.

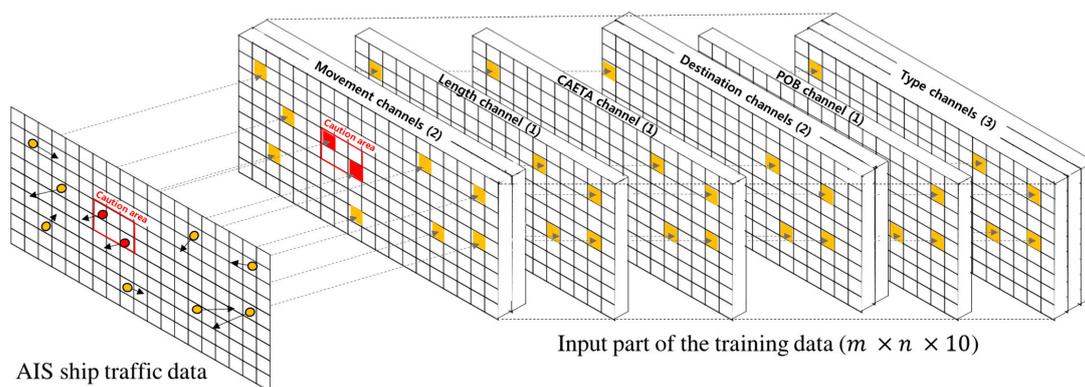


Figure 5. Input data encoding for the Ship Traffic Extraction Network (STENet) model.

Suppose that there are  $n$  ships in a harbor area at a time point  $t$ . The information  $S^t$  for the ships is expressed as  $S^t = \{s_1^t, s_2^t, \dots, s_n^t\}$ , where  $s_i^t$  indicates all information for the  $i$ -th ship at time point  $t$ . For  $s_i^t$ , its position index  $(p_i^t, q_i^t)$  at the two-dimensional arrays is found by comparing its position  $(s_i^t.lat, s_i^t.lon)$  with the grid cell locations. The information of a ship is stored at the array element with the index.

The movement vector channels express the movement vectors of ships that are made of two components. The first channel contains the vector component  $V_x$  in the  $x$ -axis, and the second channel contains the vector component  $V_y$  in the  $y$ -axis. Hence, the movement vector channels are represented by an  $m \times n \times 2$  array. The movement vector channels at index  $(p_i^t, q_i^t)$  indicate the movement vector of the ship at the corresponding grid cell. The information of other ship attributes is also stored at the elements of the corresponding index  $(p_i^t, q_i^t)$  for their channels. Length, CAETA and POB are expressed in a single numerical value. Hence, each of them is stored in a single channel, i.e.,  $m \times n \times 1$ , respectively.

The destination information is represented by a vector  $D$  from the position of a ship to the destination coordinate. Suppose that  $(s_i^t.d_{lat}, s_i^t.d_{lon})$  denotes the coordinate of the destination of the  $i$ -th ship at time point  $t$ . Then,  $D = (s_i^t.d_{lat}, s_i^t.d_{lon}) - (s_i^t.lat, s_i^t.lon)$ . The destination channels are made of two channels, one of which represents the vector component in the  $x$ -axis, and the other of which represents the vector component in the  $y$ -axis.

The proposed prediction model classifies ship type into one of cargo ship, tanker ship and other ship. Hence, the ship type information of a ship is encoded in an  $m \times n \times 3$  array, i.e., in three channels. Each channel corresponds to one type of ship, and the ship type information is expressed in one-hot encoding as shown in Table 2. In the table, we see that Ship A is a cargo ship located at the grid corresponding to the index (2,3).

Table 2. Examples of each channel and the layer index of the layer type.

Ship ID (Position Index)	Ship A (2,3)	Ship B (4,8)	Ship C (5,3)	Ship D (7,2)	Ship E (5,9)
Cargo ship channel	1	0	1	0	0
Tanker ship channel	0	0	0	1	0
Other ship channel	0	1	0	0	1

On constructing the training dataset for the prediction model, the output is the predicted value. The STENet model is trained to predict the number of ships in the caution area at the future time points. For an input that consists of the values of six attributes at a specific time point  $t$ , its output is determined by computing the ships in the caution area for the interpolation interval at a future time  $t + \Delta$ .

#### 4.2. STENet Architecture

Figure 6 shows the proposed STENet architecture that is organized into a hierarchical model, of which the front part consists of a CNN module and five FCNN for feature extraction, of which the rear part is a fully-connected network to predict the future traffic using the extracted features. One CNN module is called the ship movement feature extraction module, which transforms the movement vector channels into a smaller feature map. The fully-connected network modules are called the ship attribute extraction modules, which take CAETA, ship length, destination, POB and channel type as the input and extract effective features from them. The outputs of both CNN module and the fully-connected modules of the front part are concatenated and flattened into one-dimensional data. The flattened data are fed into the rear-part of the fully-connected network, which produces the predicted number of ships in the caution area.

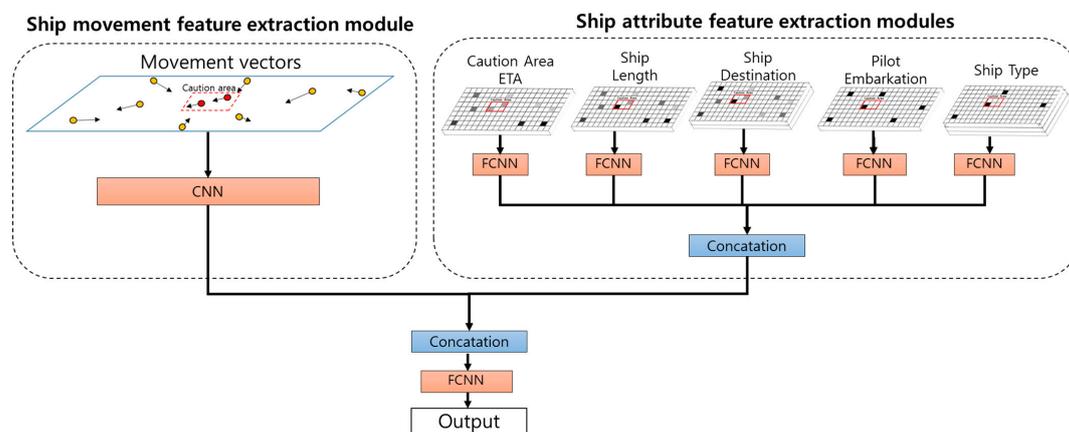


Figure 6. STENet architecture.

##### 4.2.1. Ship Movement Feature Extraction Module

The ship movement feature extraction module is implemented by a CNN model as shown in Figure 7. The CNN model is supposed to extract useful features for the spatial dependencies between overall ship positions and caution area. The input to the CNN model is the two channels of the ship movement vectors. The model is organized into the following architecture with seven layers:

$$\text{Input}(m, n, 2) [-\text{Conv} - \text{Conv} - \text{Conv} - \text{Maxpool}] \times 2 - \text{Output}(m/4, n/4, 2)$$

Here, *Conv* indicates a convolution layer, which uses a kernel  $W$  to extract some features; *Maxpool* indicates the max pooling operation; and  $[\ ] \times 2$  indicates the repetition of the subnetwork in the bracket  $[\ ]$ . In a CNN model, the kernels are learned from training data, while in conventional signal processing, the developers have to set the kernels appropriate for a given task manually. The convolution is a transformation operator for an input with the kernel as follows: Suppose that input data  $C^{n-1}$  is two-dimensional data of size, i.e.,  $C_x^{n-1} \times C_y^{n-1}$ , and the kernel  $W^n$  is a two-dimensional array of size, i.e.,  $K_x \times K_y$ . Let  $*$  denote the convolution operation, and  $f$  is an activation function such as the

sigmoid [29] or ReLU function [30]. When the convolution kernel  $W^n$  is applied to the input  $C^{n-1}$ , then the output  $C^n$  is computed as follows (here,  $b^n$  is a bias term):

$$C^n = f \left( \sum_{i=1}^{C_x^{n-1}} \sum_{j=1}^{C_y^{n-1}} C_{(i,j)}^{n-1} * W_{(i,j)}^n + b^n \right) \quad (5)$$

Max pooling indicates an operation to choose the maximum value for a specified region, which is usually a square region in its input. It plays the role of selecting the maximum feature values and of reducing the input into a smaller one.

In the CNN model for ship movement feature extraction, the convolution layers use a  $3 \times 3$  convolution kernel with equi-padding, which makes the convolution result have the same dimension as its input, and max pooling operations are carried out with a  $2 \times 2$  window. The input is given as an  $m \times n \times 2$  array, which represents the ship movement vectors expressed in two channels. On the other hand, the output is produced in an  $(\frac{m}{4}) \times (\frac{n}{4}) \times 2$  array.

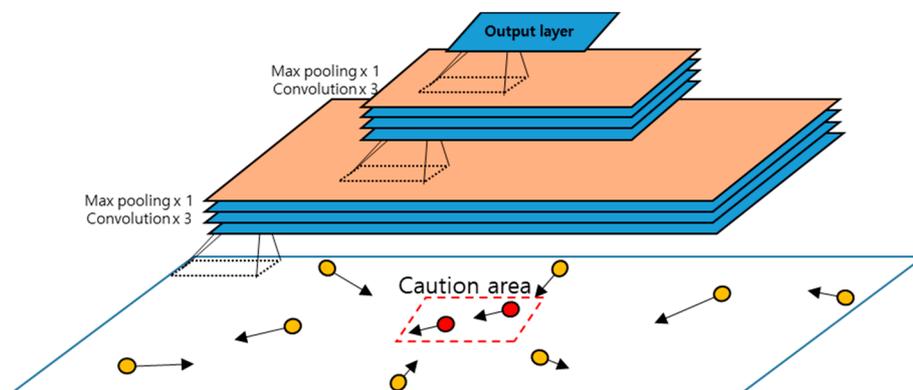


Figure 7. The CNN model for the ship movement feature extraction module.

#### 4.2.2. Ship Attribute Feature Extraction Modules

The ship attribute extraction modules are made of five fully-connected networks, which take as the input the two-dimensional channels of CAETA, ship length, destination, POB and ship type, respectively. Each fully-connected network module has the following architecture with three layers, each of which has  $p$  nodes:

$$Input(m,n,c)-FC(p)-FC(p)-FC(p)-Output(p)$$

Here,  $FC(p)$  indicates a fully-connected layer with  $p$  nodes. For each fully-connected module, its input dimension is  $m \times n \times c$ , where  $c$  is the number of channels for the corresponding attribute, and the output dimension is  $p \times 1$ . The modules transform the ship attributes into compact feature vectors of dimension  $p$  that are effective for predicting future traffic in the caution area.

#### 4.2.3. Prediction Module

The prediction module consists of a fully-connected network with three hidden layers, and each of layers has 50, 30 and 10 nodes, respectively, and the output layer with a single node. The architecture has the outputs of the feature extraction modules as an input to the prediction module. It produces the predicted number of ships in the caution area in the medium-term and long-term futures. Moreover, it contains some additional operational layers for batch normalization and dropout to improve the performance.

In the fully-connected layers, each node is connected to all nodes of the very preceding layer. The output  $N_i^{l+1}$  of the  $i$ -th node at layer  $l$  is computed as follows:

$$N_i^{l+1} = f \left( \sum_{j=1}^n w_{ij}^{l+1} N_j^l + b_i^{l+1} \right) \quad (6)$$

Here,  $w_{ij}^{l+1}$  is the connection weight between the  $i$ -th node at layer  $l$  and the  $j$ -th node at layer  $l + 1$ ;  $b_i^{l+1}$  is the bias term for the  $i$ -th node at layer  $l + 1$ ; and  $f$  denotes the activation function. In FCNN, the exponential linear unit (ELU) function is used as the activation function, which is defined as follows [31]:

$$f(x) = \begin{cases} x & , x > 0 \\ \alpha(e^x - 1), & x \leq 0, \end{cases} \quad (7)$$

Batch normalization is an operation that makes the input data to the next layer preserve a distribution of zero mean and unit variance. It is experimentally shown that the batch normalization is helpful for improving the performance and stability of a network.

To protect the fully-connected layers from overfitting the noisy or erroneous data, the dropout operations are used in the training phase [32]. When dropout is applied, randomly selected nodes are ignored by the network. The dropout operation is ignored in the inference phase.

#### 4.3. Error Function and Performance Evaluation

STENet is trained to minimize the error function using a gradient descent-based training algorithm. As the error function, the network uses the Mean Absolute Percentage Error (MAPE), which is defined as follows [33]:

$$MAPE = \left( \frac{1}{n} \sum_{j=1}^n \left| \frac{\hat{y}_j - y_j}{y_j} \right| \right) \times 100 \quad (8)$$

where  $y_j$  is the target output of the network and  $\hat{y}_j$  is the predicted value by STENet. MAPE can measure the model with relative accuracy in the range of 0–100 in the training and validation phases.

## 5. Experiments

### 5.1. Data Preparation

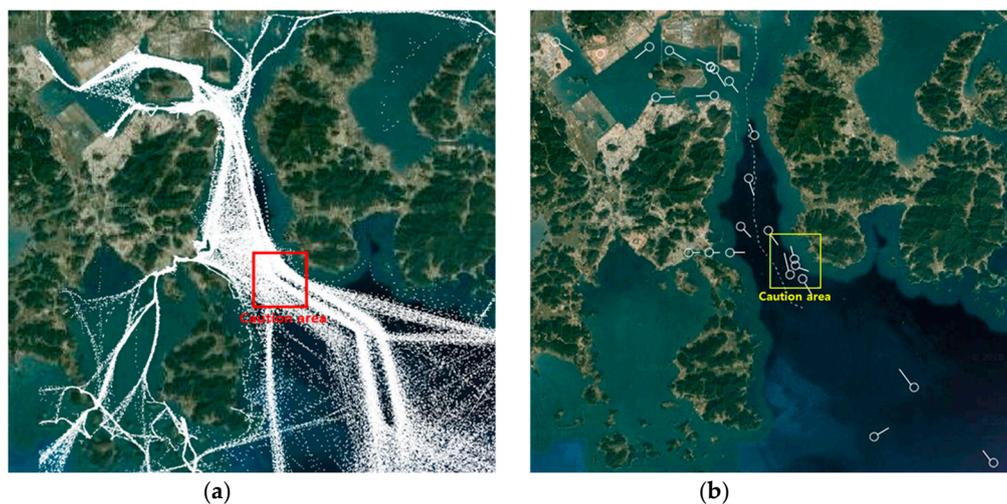
To evaluate the performance of the proposed STENet model, we have used a real AIS sensor dataset collected over the two years (2015–2016) in Yeosu, which is a harbor located at the southern part of the Korean peninsula. According to the stability of their AIS sensor data, ships are categorized into either Class A or Class B, where ships to broadcast valid AIS messages are labeled with Class A and ships broadcasting invalid messages are labeled with Class B. The training data have been constructed from the AIS sensor data for Class A ships. Figure 8 shows the harbor water area for which AIS sensor data are collected by a VTS station. Figure 8a shows the distribution of ship trajectories over a day. In Figure 8b, the region indicated by a rectangle is the caution area for which we want to predict the future traffic. The region is the caution area because many inbound and outbound voyage routes are merged, crossed or split. It is important to control the number of ships in the region to be a manageable size for safety assurance. In the experiments, we implemented four prediction models to predict the future traffic at the caution area. We trained the models with the historical ship trajectory data constructed from AIS sensor data. The trained models are supposed to predict the caution area traffic at the future time points in 20, 30, 40 and 50 min with the real-time AIS sensor data over the entire harbor water area.

The dataset was constructed by the methods described in Sections 3.2 and 3.3, where the interpolation interval was 10 s and the entire harbor water area was partitioned into a  $100 \times 100$  grid. The size of merchant ships is on average 120 m, and in the normal situation, no such ships get closer to each other within

a shorter distance than three-times the ship length to secure their safety. Hence the size of a grid cell was set  $360 \times 360 \text{ m}^2$  in the experiments.

Hence, the harbor traffic at a specific time point was expressed in a  $100 \times 100 \times 10$  tensor. The synchronized data at the reference time points become the input part of the training data. The number of ships in the caution area at the future time points in 20, 30, 40 and 50 min become the output part of the training data. The number of ships ranged over the integer values from 0–9. If the number of ships was greater than 9, we clamped it to 9. For each future time point, a separate prediction model was trained. In total, four prediction models have been constructed. We acquired 25,470 distinct AIS trajectory data for the harbor over two years. Each trajectory consists of a sequence of AIS data from a harbor area limit to a berth in the harbor. There are 3 harbor area limits and 75 berths in the harbor; hence, there are 225 distinct routes. The training data for the prediction models, STENet and VGGNet, are generated by sampling the information of ship movement and attributes every 10 s. We get 8640 training data for a day. In a day, at many sampling times, there are no ships in the caution area, and hence, we selected the data in a way that too many data with zero ships in the caution area are included. Hence, the number of training data available is too small to train deep learning models like STENet and VGGNet.

Hence, we generated the training data by using the following augmentation method for the trajectory data. All the available trajectory data were partitioned into 225 groups according to their route. For each group, the occurrence probabilities of travels were computed over the hourly intervals in a day. The probability distributions show how frequently the corresponding trajectories happen in the routes. Every 10 s over a day, some groups are randomly selected according to the distributions. For each selected group, a trajectory is randomly selected from the group. Then, according to the trajectory, a ship is assumed to start at the starting point, i.e., harbor area limit or berth, of the trajectory, and the corresponding AIS data are observed. At the same time, a training data record is constructed with the 10 two-dimensional channels to describe the traffic situation at the moment as the input and the number of ships in the caution area as the output. With the augmentation method, we generated the training dataset of a size of 10,000,000. Then, 80% of the dataset was used to train the prediction model, and the remaining 20% was used to test the trained model.



**Figure 8.** (a) The distribution of ship navigation trajectories and (b) the caution area.

## 5.2. Performance Evaluation

We compared the four prediction models: the Dead Reckoning (DR) model [16], Support Vector Regression (SVR) model [5], a new CNN-based model and the STENet model. In the DR model, it is assumed that the routes for ships are fixed in the harbor area and the ships travel over their route at their current speed. Then, after, the model determines the positions of ships at a specified time point

by making them travel on their route at their current speed up to a prediction time point. Finally, the ships in the caution area are counted as the output. In the SVR model, the routes of ships are fixed, and the speeds at the segments of routes are trained by the SVR model. The future positions of ships are computed by making ships travel on the segments of their routes at the speeds for corresponding segments estimated by the SVR model. Similar to DR, the ships in the caution area are counted as the output.

Before developing the STENet model, we developed a new CNN-based prediction model that uses a simplified architecture of the VGGNet, which is generally known as a successful deep neural network model [34]. The VGGNet-based prediction model has the following layered architecture with 11 layers:

Input(100,100,10)-[Conv] × 2-Maxpool-[Conv] × 2-Maxpool-[Conv] × 3-Maxpool-[Conv] × 3-Maxpool-[FC]-Output(1)

Here, *Conv* indicates a 3 × 3 convolution layer; *Maxpool* indicates the max pooling operation with the 2 × 2 window; and *FC* indicates a fully-connected layer with a single output node. Later on, this model is denoted as the VGGNet model.

The same input data of size 100 × 100 × 10 prepared for the STENet model have been used for training the VGGNet prediction model. The STENet model has been trained with the prepared training dataset. The ADAM optimizer [35] was used for training the model with the learning rate  $\eta$  of 0.001 and hyperparameters  $\beta_1 = 0.9$  and  $\beta_2 = 0.999$ . It is a gradient-descent method for function optimization that adjusts weights, each of which has its own learning rate. Each weight  $w_i$  is tuned with the following update equation:

$$w_i^{(t+1)} = w_i^{(t)} - \frac{\eta}{\sqrt{\hat{v}^{(t)} + \epsilon}} \hat{m}^{(t)} \quad (9)$$

Here,  $w_i^{(t)}$  indicates the weight value of  $w_i$  at time  $t$ ,  $\eta$  is a hyperparameter and  $\epsilon$  is a small positive value to avoid dividing by zero. Meanwhile,  $\hat{m}^{(t)}$  and  $\hat{v}^{(t)}$  are the estimates of the mean and variance of the gradients for  $w_i$ , which are adjusted by the parameters  $\beta_1$  and  $\beta_2$ , respectively.

The prediction models have been developed for the future time points in 20, 30, 40 and 50 min. The 20- and 30-min future predictions are regarded as the medium-term predictions, and the 40- and 50-min future predictions are regarded as the long-term predictions. A very long future prediction is not meaningful because most ships travel across the harbor within an hour.

The Mean Absolute Error (MAE) was used to evaluate the performance of the prediction models, which is defined as follows ( $y_j$  is the true number of the ships in the caution area, and  $\hat{y}_j$  is the estimated number of ships by the proposed prediction model):

$$MAE = \sum_{j=1}^n \frac{|\hat{y}_j - y_j|}{n} \quad (10)$$

We compared the performance of the prediction models with respect to the performance of the SVR model. The relative performance improvement  $RPI(M)$  of a prediction model  $M$  is computed as follows ( $MAE(SVR)$  is the MAE of the SVR model):

$$RPI(M) = \frac{MAE(SVR) - MAE(M)}{MAE(SVR)} \times 100 \quad (11)$$

Table 3 shows the performance of the four prediction models with respect to the MAE and the Relative Performance Improvement (RPI).

**Table 3.** Experiment results of ship traffic prediction.

			DR	SVR	VGGNet	STENet
Middle-term Prediction	20-min prediction	MAE	1.004	0.904	0.821	0.415
		(PRI)	(−11.1%)	(0%)	(9.2%)	(54.1%)
	30-min prediction	SD	1.105	1.034	1.031	0.566
		MAE	1.541	1.232	1.152	0.651
Long-term Prediction	40-min prediction	(PRI)	(−25.1%)	(0%)	(6.5%)	(47.2%)
		SD	1.714	1.410	1.441	0.859
	50-min prediction	MAE	2.510	1.710	1.153	0.717
		(PRI)	(−46.8%)	(0%)	(32.6%)	(58.1%)
	50-min prediction	SD	2.822	1.922	1.591	0.779
		MAE	3.5413	1.938	1.436	0.829
		(PRI)	(−82.7%)	(0%)	(25.9%)	(57.2%)
		SD	3.673	2.057	1.754	1.077

\*\* MAE: Mean Absolute Error, RPI: Relative Performance Improvement, SD: Standard Deviation, DR: Dead Reckoning.

The experimental results showed that the performance of the STENet model is improved on average by 50.65% for the medium-term predictions and 57.65% for the long-term predictions over the baseline model, i.e., the SVR-based model.

## 6. Conclusions

“Safety comes first” is at every corner of working environments. Maritime traffic shows different characteristics from ground traffic because of the difficulty in swiftly steering the ships on the water. Ship collision is a top-priority concern in ship monitoring and controlling services, especially in the caution area where many ship routes are merged, split or cross each other. The VTS operators are required to understand both ongoing situations and expected future situations in the caution area. We proposed a new deep neural network-based model called STENet for predicting future traffic in the caution area. The STENet model consists of the front-part feature extraction modules and the rear-part prediction module. All pieces of vessel information related to traffic are spatially encoded into two-dimensional channels from which features are separately extracted by a convolutional network and five fully-connected layers in the front-part of the model. The separated feature extraction helps the model performance improve by keeping unrelated attributes from crosstalking. The fully-connected layers of the rear-part of the model predict traffic density in the caution area by using those features together. It is also presented how to construct the training data from the AIS sensor data. The model predicts the medium- and long-term future traffic in the caution area. In the experiments on a real ship traffic dataset, the STENet model has shown superior performance to the reference model by a big margin.

The STENet model has the following advantages: First, it does not resort to a mathematical modeling for traffic predictions, but learns the prediction model from a large volume of AIS sensor data, which is easily collected on the VTS stations. Second, it can find excellent prediction models for medium- and long-term traffic predictions. Third, the encoding scheme for both ship movement and ship attribute data is effective at capturing the traffic characteristics. That can be justified by the fact that the STENet model has shown excellent performance on traffic prediction in the caution area. It is hence expected that if the output attribute is changed to other indicators like vessel collision risk or congestion rate, then STENet may find some good model for the indicator.

**Author Contributions:** Conceptualization, K.I.K. Methodology, K.M.L. Software, K.I.K. Validation, K.I.K. Data curation, K.I.K. and K.M.L. Writing, original draft preparation, K.I.K. Writing, review and editing, K.M.L. Visualization, K.I.K. Supervision, K.M.L. Project administration, K.M.L. Funding acquisition, K.M.L.

**Funding:** This research was supported by the Next-generation Information Computing Development Program through the National Research Foundation of Korea (Grant No. NRF-2017M3C4A7069432).

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

## References

1. Kim, K.I.; Jeong, J.S.; Lee, B.G. Study on the analysis of near-miss ship collisions using logistic regression. *J. Adv. Comput. Intell. Inform.* **2017**, *21*, 467–473. [[CrossRef](#)]
2. Park, D.W.; Park, S.H. Syntactic-level integration and display of multiple domains' S-100-based data for e-navigation. *Clust. Comput.* **2017**, *20*, 721–730. [[CrossRef](#)]
3. Kim, K.I.; Lee, K.M. Dynamic Programming-Based Vessel Speed Adjustment for Energy Saving and Emission Reduction. *Energies* **2018**, *11*, 1273. [[CrossRef](#)]
4. International Maritime Organization (IMO). *Guidelines for Vessel Traffic Services; Res A.857*; International Maritime Organization (IMO): London, UK, 1997.
5. Kim, J.S. Vessel Target Prediction Method and Dead Reckoning Position Based on SVR Seaway Model. *Int. J. Fuzzy Log. Intell. Syst.* **2017**, *17*, 279–288. [[CrossRef](#)]
6. Kim, K.I.; Lee, K.M. Ship Encounter Risk Evaluation for Coastal Areas with Holistic Maritime Traffic Data Analysis. *Adv. Sci. Lett.* **2017**, *23*, 9565–9569. [[CrossRef](#)]
7. Tsou, M.C. Discovering knowledge from AIS database for application in VTS. *J. Navig.* **2010**, *63*, 449–469. [[CrossRef](#)]
8. Lee, H.W.; Kim, N.R.; Lee, J.H. Deep neural network self-training based on unsupervised learning and dropout. *Int. J. Fuzzy Log. Intell. Syst.* **2017**, *17*, 1–9. [[CrossRef](#)]
9. Jeon, W.S.; Rhee, S.Y. Fingerprint Pattern Classification Using Convolution Neural Network. *Int. J. Fuzzy Log. Intell. Syst.* **2017**, *17*, 170–176. [[CrossRef](#)]
10. Yao, L.; Liu, Y.; He, Y. A Novel Ship-Tracking Method for GF-4 Satellite Sequential Images. *Sensors* **2018**, *18*, 2007. [[CrossRef](#)] [[PubMed](#)]
11. Wu, D.; Xia, L.; Geng, J. Heading estimation for pedestrian dead reckoning based on robust adaptive Kalman filtering. *Sensors* **2018**, *18*, 1970. [[CrossRef](#)] [[PubMed](#)]
12. Perera, L.P.; Oliveira, P.; Soares, C.G. Maritime traffic monitoring based on vessel detection, tracking, state estimation, and trajectory prediction. *IEEE Trans. Intell. Transp. Syst.* **2012**, *13*, 1188–1200. [[CrossRef](#)]
13. Xu, T.; Liu, X.; Yang, X. A novel approach for ship trajectory online prediction using BP neural network algorithm. *Int. J. Adv. Inf. Sci. Serv. Sci.* **2012**, *4*, 271–277.
14. Perera, L.P.; Soares, C.G. Ocean vessel trajectory estimation and prediction based on extended Kalman filter. In Proceedings of the Second International Conference on Adaptive and Self-Adaptive Systems and Applications, Lisbon, Portugal, 21–26 November 2010; pp. 14–20.
15. Ristic, B.; La Scala, B.F.; Morelande, M.R.; Gordon, N.J. Statistical analysis of motion patterns in AIS Data: Anomaly detection and motion prediction. In Proceedings of the 2008 11th International Conference on Information Fusion, Cologne, Germany, 30 June–3 July 2008; pp. 1–7.
16. Mazzarella, F.; Arguedas, V.F.; Vespe, M. Knowledge-based vessel position prediction using historical AIS data. In Proceedings of the Sensor Data Fusion: Trends, Solutions, Applications (SDF), Bonn, Germany, 6–8 October 2015; pp. 1–6.
17. 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](#)]
18. Kim, J.S.; Jeong, J.S. Extraction of Reference Seaway through Machine Learning of Ship Navigational Data and Trajectory. *Int. J. Fuzzy Log. Intell. Syst.* **2017**, *17*, 82–90. [[CrossRef](#)]
19. Zhang, H.; Xiao, Y.; Bai, X.; Chen, L. GA-support vector regression based ship traffic flow prediction. *Int. J. Control Autom.* **2016**, *9*, 219–228. [[CrossRef](#)]
20. Gan, S.; Liang, S.; Li, K.; Deng, J.; Cheng, T. Ship trajectory prediction for intelligent traffic management using clustering and ANN. In Proceedings of the 2016 UKACC 11th International Conference on Control (CONTROL), Belfast, UK, 31 August–2 September 2016; pp. 1–6.
21. Daranda, A. Neural Network Approach to Predict Marine Traffic. *Trans. Balt. J. Mod. Comput.* **2016**, *4*, 1–26.
22. Van Nguyen, D.; Vadaine, R.; Hajduch, G.; Garello, R.; Fablet, R. Multi-task Learning for Maritime Traffic Surveillance from AIS Data Streams. In Proceedings of the International Conference on Data Science and Advanced Analytics (DSAA), Turin, Italy, 1–4 October 2018; pp. 1–10.
23. Gallego, A.J.; Pertusa, A.; Gil, P. Automatic Ship Classification from Optical Aerial Images with Convolutional Neural Networks. *Remote Sens.* **2018**, *10*, 511. [[CrossRef](#)]

24. Kim, K.I.; Lee, K.M. Mining of Missing Ship Trajectory Pattern in Automatic Identification System. *Int. J. Eng. Technol.* **2018**, *7*, 167–170.
25. Jiang, Y.; Wu, J.; Zhang, S. An Improved Positioning Method for Two Base Stations in AIS. *Sensors* **2018**, *18*, 991. [[CrossRef](#)] [[PubMed](#)]
26. Sang, L.Z.; Yan, X.P.; Wall, A.; Wang, J.; Mao, Z. CPA calculation method based on AIS position prediction. *J. Navig.* **2016**, *69*, 1409–1426. [[CrossRef](#)]
27. Hayler, W.B.; Keever, J.M. *American Merchant Seaman's Manual*; Cornell Maritime Pr.: Baltimore, MD, USA, 1980; ISBN 0-87033-549-9.
28. Kim, D.S.; Arsalan, M.; Park, K.R. Convolutional Neural Network-Based Shadow Detection in Images Using Visible Light Camera Sensor. *Sensors* **2018**, *18*, 960. [[CrossRef](#)] [[PubMed](#)]
29. Han, J.; Moraga, C. The influence of the sigmoid function parameters on the speed of backpropagation learning. In Proceedings of the International Workshop on Artificial Neural Networks, Torremolinos, Spain, 7–9 June 1995; pp. 195–201.
30. Nair, V.; Hinton, G.E. Rectified linear units improve restricted boltzmann machines. In Proceedings of the 27th International Conference on Machine Learning (ICML-10), Haifa, Israel, 21–24 June 2010; pp. 807–814.
31. Clevert, D.A.; Mayr, A.; Unterthiner, T.; Hochreiter, S. Rectified factor networks. In Proceedings of the Advances in Neural Information Processing Systems, Montreal, QC, Canada, 7–12 December 2015; pp. 1855–1863.
32. Hinton, G.E.; Srivastava, N.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R.R. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv*, 2012; arXiv:1207.0580.
33. Lee, Y.H.; Ho, C.R.; Su, F.C.; Kuo, N.J.; Cheng, Y.H. The use of neural networks in identifying error sources in satellite-derived tropical SST estimates. *Sensors* **2011**, *11*, 7530–7544. [[CrossRef](#)] [[PubMed](#)]
34. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv*, 2014; arXiv:1409.1556.
35. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. In Proceedings of the 3rd International Conference for Learning Representations, San Diego, CA, USA, 7–9 May 2015.



© 2018 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/>).