*ICIC Express Letters*
Volume **15**, Number **6**, June **2021**
 ISSN 1881-803X
pp. **627–637**

# A RECURRENT NEURAL NETWORK MODEL FOR DETECTING FISHING GEAR PATTERNS

Worawut Srisukkham[1,*], Luepol Pipanmaekaporn[2]
and Suwatchai Kamonsantiroj[2]

[1]Department of Computer Science
Faculty of Science
Chiang Mai University
239, KouyKaew Road, Suthep, Mueng, Chiang Mai 50200, Thailand
*Corresponding author: worawut.s@cmu.ac.th

[2]Department of Computer and Information Science
Faculty of Applied Science
King Mongkut's University of Technology North Bangkok
1518, Pracharaj1, Wongsawang, Bangsue, Bangkok 10800, Thailand
{ luepol.p; suwatchai }@sci.kmutnb.ac.th

Received October 2020; accepted January 2021

Abstract. *Estimation of fishing efforts is the key metric for sustainable ocean management. Previous studies have been proposed to detect fishing activities based on analysis of vessel trajectory from Vessel Monitoring System (VMS). However, identification of fishing activity without prior knowledge related to fishing gears may cause detection failure because individual gears of fishing possess specific movement patterns. It is desirable to identify vessel movements made by different fishing gears for accurately detecting fishing events. In this work, we propose a novel method that recognizes a VMS trajectory corresponding to fishing gear types by encoding sequences of GPS points with Recurrent Neural Networks (RNNs). Firstly, we segment a route trajectory using an unsupervised segmentation scheme. After that, each extracted segment is encoded into a semantic space to train a neural network model for identifying a fishing ship of a specific gear. We also demonstrate that RNNs with feature embedding can leverage the discriminative power of classifier. We conduct experiments on real trajectory data of three fishing gear types, including trawl, purse-seine and falling net, collected from a VMS database of the Thailand Command Center for Combating Illegal Fishing (CCCIF). Our experimental results demonstrate embedded bidirectional gate recurrent units achieves over 90% classification accuracy compared with state-of-the-art methods and other RNN models.*
**Keywords:** Fishing gear detection, VMS, Trajectory classification, RNN

1. **Introduction.** Fishiers managers are typical to assess fish populations by areas for sustainable management decisions. In such cases, fishing efforts are often estimated by using logbook data recorded by fishers. However, this method tends to underestimate the population sizes [1]. It is desirable to identify an effective method to detect fishing activities related to individual vessels. Among remote-sensing technologies, Vessel Monitoring System (VMS) that uses satellite communication becomes a standard tool of fisheries monitoring and surveillance [2]. VMS data allow to track events related to a fishing trip by recording information, including GPS positions (latitude/longitude), speed over ground and course over ground. In VMS, a GPS tracker is deployed on a fishing ship to transmit information to vessel operators in time intervals. For many years, previous studies utilize the VMS data for fishing studies. For example, some studies use the ship tracking data to compute fishing metrics such as fishing diversity [2] and fishing intensity [3]. Many

DOI: 10.24507/icicel.15.06.627

of them have also proposed to detect fishing activities made by a vessel [4-6]. However, these studies tend to be impractical for two reasons. Firstly, these studies are solely designed to detect fishing events related to specific fishing gears. Secondly, identification of fishing activity without knowing fishing gear types can result in inaccurate detection because different fishing gear types have different movement patterns. Because VMS data provide information on the movement of fishing vessels, previous studies were proposed to utilize the GPS tracking data to characterize fishing gears. [6] focused on fishing ship speeds from the VMS data as the main indicator for fishing gear type identification. More sophisticated approaches that use machine learning schemes have also been proposed to characterize movement patterns of fishing gears. [7] proposed a Gaussian mixture model to encode data points from different types of gear. After that, support vector machine is applied to differentiating four fishing gear types: trawl net, longline net, pole net and purse-seine net. The recent study [8] proposed to produce specific profiles of fishing gear types based on velocity. Feature selction and XGBoost are then employed to classify fishing gear types based on their profile. A probabilistic model that uses spatio-temporal information (i.e., days, months, weather, fishing hours and X-Y regions) [9] was proposed to identify fishing gear types. Recently, [10] proposed a Recurrent Neural Network (RNN) model [11,12] which focuses on extracting sequential patterns of ship trajectory to discriminate some fishing gear types. According to experimental results, their method achieved 89.7% classification accuracy. Nevertheless, we believe that the accuracy may be limited by intrinsically meaningless points tracked in VMS and complicated dynamic patterns in trajectories of fishing gear types.

Motivated by the above challenges, we propose an RNN-based method to enhance the accuracy of fishing gear type identification. Our proposed method, at the first step, extracts a single route trajectory from VMS data using an unsupervised segmentation strategy. Furthermore, Density-Based Spatial Clustering of Applications with Noise (DBSCAN) [15] with adaptive parameters is adopted to identify meaningful data points capturing specific movements of fishing ships from VMS data. We then extract various features of each point segment and encode these features into a low-dimensional embedding space to improve RNN training for the detection of specific fishing gears. Experimental results conducted on real-world VMS data of trawl, purse-seine and falling net gears have shown that the proposed RNN-based models achieve encouraging performance.

The main contributions of our study are summarized as follows.

- We propose an RNN-based method for fishing gear type identification which uses VMS data.
- We demonstrate that RNN models with an embedding space achieve improved accuracy of recognition for trawl, purse-seine and falling net vessel trajectories compared with state-of-the-art approaches.
- We carry out experiments to evaluate the effectiveness of the proposed models on real-world vessel movement data in Thailand.

## 2. Preliminaries.

### 2.1. VMS trajectory.

VMS originally records information about a vessel's identity, location and activity in time intervals. Table 1 shows VMS samples (points) of vessel's events [2]. Each VMS record consists of vessel's ID number, status of ship movement, date/time, Speed over Ground (SoG), distance from last move status, latitude/longitude and Course over Ground (CoG). In this work, we focus on identifying VMS data of three fishing gear types: trawl, purse-seine and falling net. Table 2 and Figure 1 show different characteristics of each of the gear types. Trawlers involve fishing with pulling the net through the water behind a fishing vessel. Fishing with purse-seine gear catches the prey by quickly encircling the net with high speeds. Falling net gear is primarily designed to

Table 1. Samples of VMS data

| Fishing ID | Date/Time | SoG | Distances | Latitudes | Longitudes | CoG |
|---|---|---|---|---|---|---|
| A0004 | 4/1/2016, 0:34:27 | 0.54 | 0 | 11.3169 | 102.4657 | 300 |
| A0004 | 4/1/2016, 1:32:27 | 2.16 | 0.07 | 11.3179 | 102.6451 | 270 |
| A0004 | 4/1/2016, 2:32:27 | 0.00 | 1.74 | 11.2964 | 102.4807 | 315 |

Table 2. Summary of fishing vessel behavior

| Gear types | Descriptions | Speed (knots) | Time (h) |
|---|---|---|---|
| Trawl | Catch the prey by dragging a net through the water behind a boat | 3-6 | 2-5 |
| Purse-seine | Catch the prey by quickly encircling the net with high speeds | 0-2.5 | 2 |
| Falling net | Catch prey by clapping down the prey | 1 | 8 |

(a) (b) (c)

Figure 1. VMS gear tracks of (a) trawl, (b) purse-seine and (c) falling net

catch prey in shallow water. While operating, the vessel falls the net to capture the prey from above.

2.2. **Density Based Spatial Clustering of Applications with Noise (DBSCAN).** DBSCAN [15] is a density-based clustering method. DBSCAN is an important and widely used algorithm for group identification in spatial databases [16,17] because it can detect arbitrary shaped clusters. Given a set of objects (points), DBSCAN marks points as core if they are close to each other within a given radius and more than a minimum number of points required for a cluster. If points within the radius are less than the minimum number of points required, then they are marked as border; otherwise, they are assigned as outlier (noise). Finally, clusters are formed by differentiating between points marked as border and noise. Despite this, the main drawback when applying classical DBSCAN to vessel trajectory segmentation is meaningless results generated by grouping spatial-temporal GPS points [18]. In this paper, a modified DBSCAN based segmentation algorithm of VMS trajectory data is presented.

2.3. **Recurrent Neural Networks (RNNs).** RNN [11,12] is a category of neural networks designed for modelling sequential data and has success in recent applications such as stock price prediction [13] and network protocol prediction [14]. Unlike standard feed-forward neural networks, RNNs employ recurrent layers to capture temporal dependencies among time steps defined by

$$h_t = \sigma(W_{xh}x_t + W_{hh}h_{t-1} + b_h) \quad (1)$$

$$y_t = \sigma(W_{hy}h_{t-1} + b_y) \quad (2)$$

where $y_t$ and $h_t$ are the output state and the hidden state at time step $t$. $W_{xh}$, $W_{hh}$, $W_{hy}$ denote weight metrics of each unit. $b_h$, $b_y$ are bias terms and $\sigma$ is the logistic sigmoid function with an output in $[0, 1]$. Long Short-Term Memory (LSTM) networks [11] were proposed to address the gradient vanishing problem in RNNs. It has several gates that enables to learn which information can keep or throw away. For many years, there is a variant of LSTM architecture while many experiments have reported that Gated Recurrent Unit (GRU) [12] well performs for trajectory classification. GRU is formally defined by

$$r_t = \sigma(W_{xr}x_t + W_{hr}h_{t-1} + b) \tag{3}$$

$$z_t = \sigma(W_{xz}x_t + W_{hz}h_{t-1} + b) \tag{4}$$

$$\tilde{h}_t = \sigma(W_x x_t + W_h(r_t \odot h_{t-1})) \tag{5}$$

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t \tag{6}$$

where $r_t$, $z_t$ denote the reset and update gates for time step $t$. $\tilde{h}_t$, $h_t$ denote the candidate and the final states at time step $t$ respectively. $W_x$, $W_h$ denote weight metrics of input and recurrent layers. $\odot$ is the operation of element-wise multiplication. When compared to LSTM, GRU uses only two gates to replace the forget gate and the input gate, and does not preserve a memory cell state. This results in the simplified architecture with fewer parameters that are easy to be optimized.

3. **The Proposed Method.** As shown in Figure 2, our proposed method is firstly to acquire data points from each trip of fishing vessel recorded by VMS. After that, the data points are divided into a sequence of segments by using a trip segmentation algorithm to capture specific movements of vessel made by fishing gear. We then perform extraction of various features in each segment and form a feature vector that describes each point segment. Finally, the embedded RNN, which is trained from training sequences of point segments, is used for identifying fishing gear patterns.

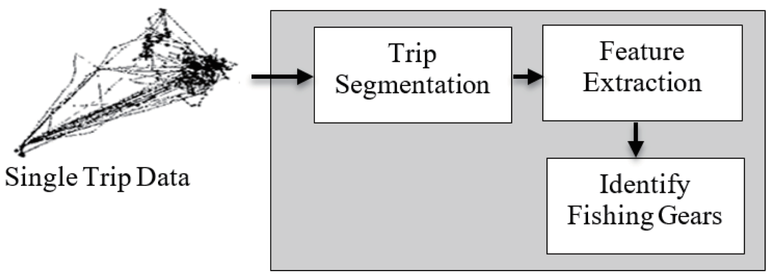


Figure 2. The proposed method

3.1. **Single trip extraction.** Many trajectory classification tasks typically divide a trajectory into segments of individual trips. Several methods are used to segment GPS-based trajectories. Some of them are transition-based method [19], clustering-based method [20], and window-based method [21]. Because of differentiation of movements in fishing vessels, we apply DBSCAN, a density-based clustering technique, for trip segmentation. The DBSCAN algorithm enables to automatically discover significant clusters based on a notion of density level. For each data point in a cluster, the neighborhood within a radius $\varepsilon$, namely $N_\varepsilon$, has to contain at least a minimum number of points, i.e., a threshold $\tau$.

However, traditional DBSCAN cannot deal with the complicated issue related to repeated visits to a fishing activity location within the same trajectory. Figure 3 depicts a sample trajectory with a starting point and the end point. With $\tau = 4$, DBSCAN will assign both $P$ and $Q$ as a core point since there are four points within the radius $\varepsilon$. However, this leads misinterpretation of the core points. DBSCAN typically lacks a

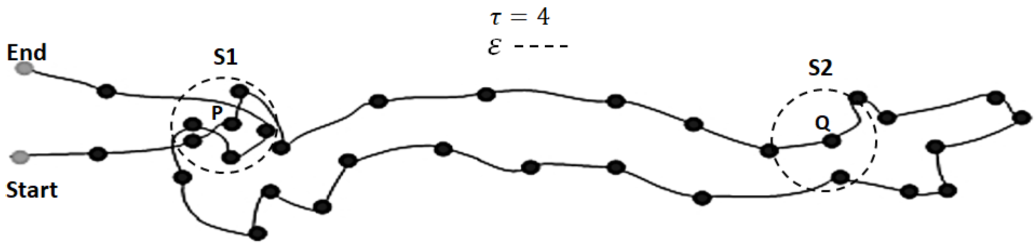


FIGURE 3. The meaningless results in DBSCAN

mechanism to process the temporal sequence of GPS points. To overcome the aforementioned drawback, we extend DBSCAN by adding a time constraint $T$. Given $p_k$ is a point taken at time $k$, during which the distance between $p_k$ and any other point taken at time step $t$, $p_t$, is less than radius $\varepsilon$, where $|t-k| < T$. We then define the neighborhood of $p_k$ as

$$Neighbors(p_k) = \{p_t \in P | dist(p_k, p_t) \leq \varepsilon, |t-k| < T\} \tag{7}$$

As shown in Figure 3, the above rule, Equation (7), is used to ensure that both $P$ and $Q$ are ruled out owing to the fact that some neighbors do not satisfy the condition. We also propose a heuristic strategy to determine the appropriate values of $\varepsilon$ and $\tau$. Given a data point, a distance from the point to its nearest neighbor is calculated. After that, we count each point neighbor. For the radius $\varepsilon$, the distribution of distances from each point to it nearest neighbor is computed. The value $\varepsilon$ can be automatically estimated by keeping the majority of data points that lies within the ordered distances. The threshold value $\tau$ is determined by using the distribution of counts for each point's neighbors with the chosen value $\varepsilon$.

3.2. **Trajectory feature extraction.** After segmentation, we perform feature extraction for each point segment. We define that a point segment is represented by a sequence of 5 tuple-elements: $P = \langle\langle x_1, y_1, s_1, c_1, t_1\rangle, \ldots, \langle x_n, y_n, s_n, c_n, t_n\rangle\rangle$ where $P_i = \langle x_i, y_i, s_i, c_i, t_i\rangle$, $x_i$ and $y_i$ denote the position of the vessel at time $t_i$, $s_i$ and $c_i$ denote SoG and CoG at time $t_i$, respectively. The length of segment is denoted as $n = |P|$ and we, firstly, extract features based on timestamp, including starting hour $t_1$, ending hour $t_n$ and duration of hours, i.e., $t_n - t_1$. Secondly, we compute statistical features based on SoG, including maximum, minimum, average and variance of speeds for the segment. Besides using SoG, we define structural features based on distance, angles and position, respectively. As shown in Figure 4, given a segment $S = \langle P_1, P_2, P_3, P_4, P_5\rangle$, the total distance of segment $D_S$ is calculated by summing $d_{21}$, $d_{32}$, $d_{43}$, and $d_{54}$, where each $d_{ji}$ denotes the Euclidean distance between two consecutive points $P_i$ and $P_j$. We also define the angle $\theta_i$ as the change of two directions $\alpha_1$ and $\alpha_2$ from two consecutive points $P_{i-1}P_i$ and $P_iP_{i+1}$ as follows:

$$\theta_i = \begin{cases} |\alpha_2 - \alpha_1|, & |\alpha_2 - \alpha_1| < \pi \\ 2\pi - |\alpha_2 - \alpha_1|, & \text{otherwise} \end{cases} \tag{8}$$

As seen in Figure 4, $\theta_1$ is calculated by the two directions between $P_1P_2$ and $P_2P_3$, respectively. After that, we encode the angles: $\theta_1$, $\theta_2$, $\theta_3$, $\theta_4$ and $\theta_5$ into the angular distribution of vessel movement. There are twelve bins of turning angles between two directions quantized at $\frac{\pi}{2}$ radian interval.

We finally incorporate geo-information by using differential longitude and latitude. Differential longitude is the difference of absolute longitudes between the current and previous data points. As shown in Figure 4, the absolute longitudes take from $[|lon_2 - lon_1|, |lon_3 -$

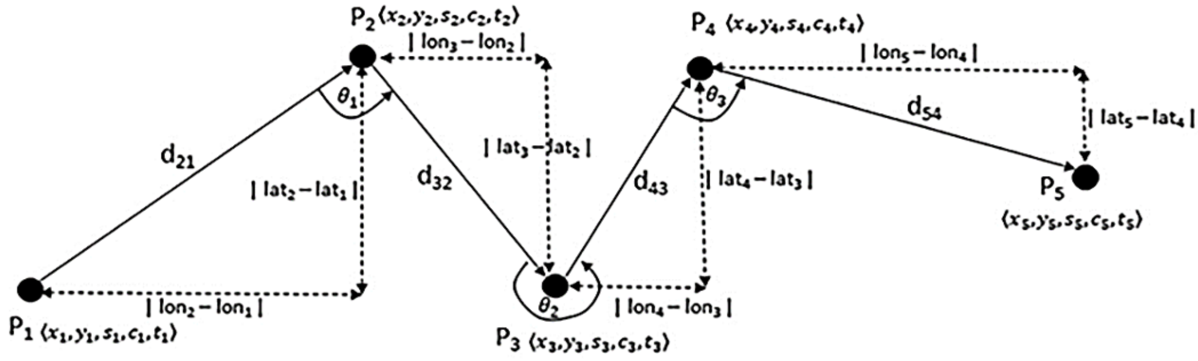


FIGURE 4. Feature representation of point segment

TABLE 3. Summary of feature descriptors extracted from each trajectory segment

| Descriptors | # of descriptors | Descriptions |
|---|---|---|
| Time | 3 | Starting hour, ending hour and duration of hour in the segment |
| SoG | 4 | Maximum, minimum, average, variance of ship speed |
| Distance | 1 | Total distance $D_S$ beginning from the start point $P_1$ to the end point $P_n$ in the segment $S$ |
| CoG | 12 | Histogram of twelve angles between two consecutive points |
| Differential longitudes | $n-1$ | List of differential longitude between two consecutive points $\lvert lon_2 - lon_1\rvert, \lvert lon_3 - lon_2\rvert, \ldots, \lvert lon_n - lon_{n-1}\rvert$ |
| Differential latitudes | $n-1$ | List of differential latitudes between two consecutive points $\lvert lat_2 - lat_1\rvert, \lvert lat_3 - lat_2\rvert, \ldots, \lvert lat_n - lat_{n-1}\rvert$ |

$lon_2|, |lon_4 - lon_3|, |lon_5 - lon_4|]$. Similarly, we transform absolute latitudes into differential form $[|lat_2 - lat_1|, |lat_3 - lat_2|, |lat_4 - lat_3|, |lat_5 - lat_4|]$. Table 3 summarizes feature descriptors extracted from each point segment. All the extracted features capture characteristic of fishing gear type.

3.3. **Learning embedding space.** As mentioned in Section 3.2, the feature vector extracted from raw VMS data is typically mixed both continuous and categorical values. Directly using these mixed features to train an RNN model can lead to poor generalization of classification performance due to overfitting effect. In this work, we develop a semantic representation of the mixed features to improve performance of RNN for gear trajectory classification. Inspired by word embedding [19] in Natural Language Processing (NLP), a word is mapped into a continuous and distributed vector representation that captures word semantics, resulting in the improved performance of text classification. To end this, we first convert the continuous features into a one-hot vector that indicates the interval in which a feature takes its values. After that, we transform the one-hot encoded vector into the semantic space by using matrix multiplication operation. Let us consider a simple example of how we calculate the semantic vector. Given $f_1 = 1.5$ and $f_2 = 2.5$ are two continuous features extracted from a trajectory segment, we convert both features into the one-hot vectors. Suppose that each of the features has 3 interval classes of value, including $[0.0, 1.0]$, $[1.1, 2.0]$ and $[2.1, 3.0]$, respectively. As seen in Figure 5, $f_1 = 1.5$, $f_2 = 2.5$ can be encoded into the one-hot vectors $i = [0, 1, 0]$ and $j = [0, 0, 1]$, respectively. Given two transform matrices $W$ and $U$, both $i$ and $j$ are merged into the semantic space by $iW + jU = [w_{21} + u_{31}, w_{22} + u_{32}, w_{23} + u_{33}]$. In this way, all the extracted features can be effectively mapped into a semantic vector space. The transformed matrices can be also learned through back-propagation while training the RNN model.

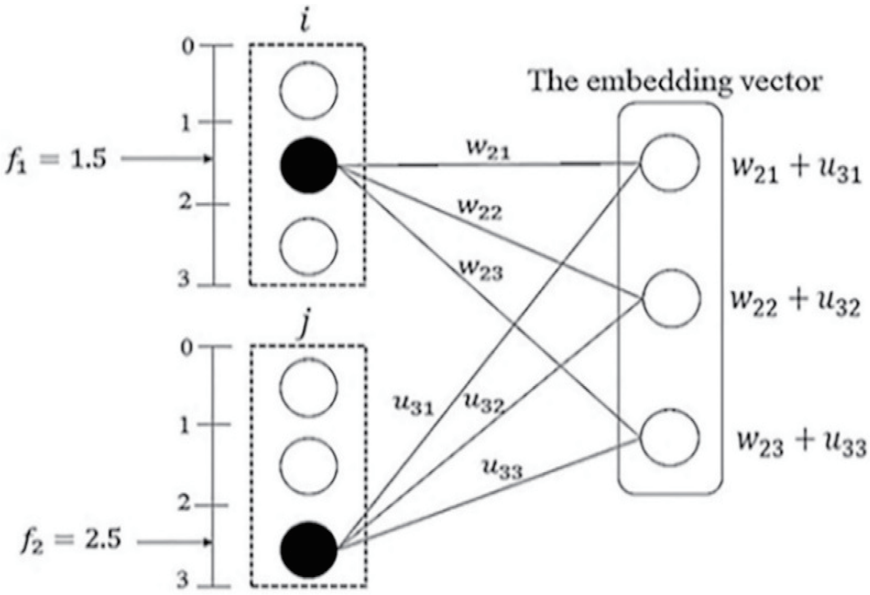


FIGURE 5. Feature embedding

3.4. **Network architecture.** Figure 6 illustrates the network architecture of this research. The first layer learns the embedding space that encodes each trajectory segment $P$ into a semantic vector and then the second layer learns contextual information from sequence of the extracted segments $P(i)$, where $i = 1, 2, \ldots, n - k$. We use GRU-based bidirectional RNN architecture to capture bidirectional flows of sequence information. After that, the state of each GRU unit is concatenated and fed into a max-pooling layer to reduce the dimensionality. A softmax classifier is learned to predict the appropriate label to the vessel gear track.

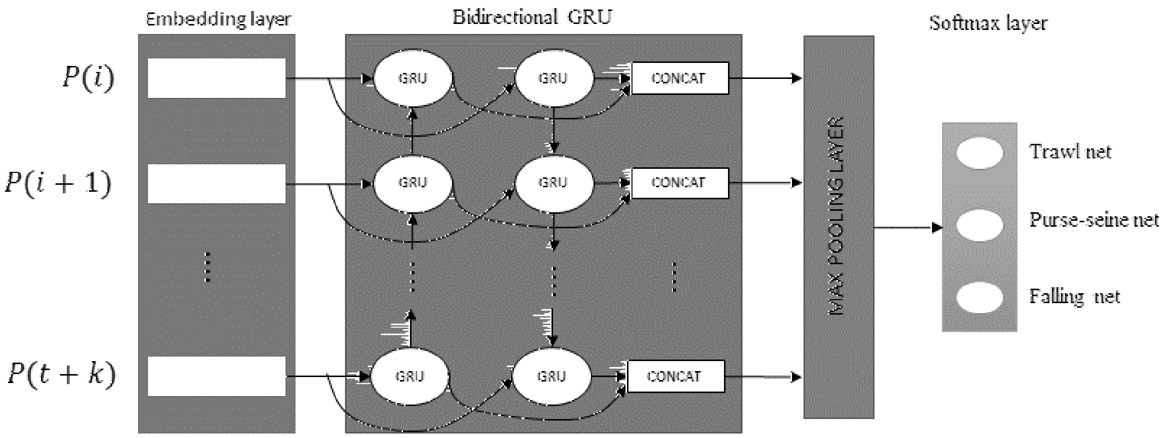


FIGURE 6. Network architecture

## 4. **Evaluation.**

4.1. **Data collection.** Our experimental dataset consists of 1,049 vessel trajectories of trawl, purse-seine and falling net originally gathered by VMS between August 2018 and February 2019. For each trajectory, we perform the following steps. We, firstly, extract a single round trip while a vessel departs and arrives a port. Secondly, the data points nearby the harbour are eliminated. Each VMS points outside the harbour are formed as a route trajectory bounded by a departure and an arrival harbour. After data preprocessing, we obtain a total number of 1,798 trip trajectories used as the training and testing data. Table 4 illustrates the dataset descriptions.

Table 4. Dataset descriptions

| Fishing gears | #points | #routes | Percentage |
|---|---|---|---|
| Trawl | 224,158 | 691 | 38.47% |
| Purse-seine | 193,412 | 612 | 34.07% |
| Falling net | 215,080 | 495 | 27.45% |

4.2. **Experimental setting.** Truncated backpropagation through time is applied to training the weights of our proposed network in order to optimize the cross entropy loss with Adam optimizer [22]. We use the learning rate to 0.002 and the momentum to 0.50. We also use a validation-based strategy of early stopping to prevent the effects of model's overfitting [23]. We have experimented with different network structures and found that a two-layer network with 50 hidden units achieves the best value in our experiments. We also discretize each feature into 25 intervals. For temporal segmentation, we employ $\varepsilon = 0.05$ and $\tau = 5$. To evaluate the performance, we compare our proposed approach to two candidate models with machine learning scheme and they also use features derived from pure VMS data.

1) *GMM+SVM* [7]. This method focuses on extraction of gear-specific mixture models from speed and tuning-angle distribution of fishing vessels. Such mixture models are used to derive feature vectors for training SVM classifier. In this experiment, we compute 32-d feature vector and use four-component mixture models to represent gear trajectory data. Finally, SVM is trained to characterize fishing gear trajectories.

2) *FVID proposed by* [8]. The authors extract 61 features related to speed, direction, location and timestamp from raw VMS data. Then, a feature selection is employed to identify the sets of candidate features for individual gear types. According to their experiment, we compute 61 features and perform feature selection to discover a common feature set for trajectory representation. After that, XGBoost [24] is applied to training a model of gear trajectory identification.

In this experiment, four classical metrics are chosen to evaluate the model's effectiveness, including classification accuracy (Accuracy), Precision, Recall and average F1 score (F1 score).

4.3. **Results.** Table 5 demonstrates the performance comparison of all the methods on the testing data. Overall, our proposed model, bidirectional GRU with feature embedding (BGRU+FE), achieves an encouraging 96.3% accuracy and outperforms all the existing baselines, including GMM+SVM and FVID. Compared to GMM+SVM, the improvement in F1 score is +9.76%, +12.73% and +12.56% for trawl, purse-seine and falling net classes respectively. Our proposed model also reveals superior performance over the measures than FVID that uses XGBoost with single feature selection. These results support superiority of sequence embedding for detecting gear-specific trajectories. Table 6 also shows the recognition rates per class. As seen in this table, recognition of trawl fishing achieves the highest precision (97.10%) and recall (98.40%) rates, compared with others. This is owing to specific movement of trawl vessels. Prediction of purse-seine category performs

Table 5. Performance comparison of the proposed model with different methods

| Methods | F1 score | | | | Accuracy |
|---|---|---|---|---|---|
| | Trawl | Purse-seine | Falling net | Average | |
| BGRU+FE | **0.978** | **0.956** | **0.941** | **0.958** | **0.963** |
| GMM+SVM [7] | 0.891 | 0.848 | 0.836 | 0.858 | 0.862 |
| FVID [8] | 0.887 | 0.832 | 0.820 | 0.846 | 0.850 |

Table 6. Recognition rate on different types of gears

| Predicted classes | Ground truth | | | Precision |
|---|---|---|---|---|
| | **Trawl** | **Purse-seine** | **Falling net** | |
| Trawl | 312 | 6 | 3 | **0.971** |
| Purse-seine | 2 | 295 | 10 | **0.960** |
| Falling net | 3 | 9 | 202 | **0.943** |
| Recall | **0.984** | **0.951** | **0.939** | |

the second best recognition rate, where 295 out of 310 class samples are obtained. These experimental results highlight accurate predictions made by RNN.

4.3.1. *The effects of RNN architectures.* Table 7 shows performance comparisons on different RNN architectures. As shown in this table, the bidirectional architectures (i.e., BGRU and BLSTM) tend to perform better results than single-directional RNN architectures (i.e., GRU and LSTM) over all the measures. These results highlight the expressive power of bidirectional RNNs to capture complex relationships among each segment pattern compared with a single direction RNN architecture. We also compare between BGRU and BLSTM for fishing gear detection. As seen in this table, BGRU achieves at least +1.2% absolute improvement in F1 score. These results also highlight the effectiveness of GRU that is easier to be trained than LSTM due to less parameters.

Table 7. Accuracy on different RNN architectures

| Architectures | Precision | Recall | F1 score |
|---|---|---|---|
| GRU | 0.926 | 0.921 | 0.923 |
| LSTM | 0.925 | 0.907 | 0.916 |
| Bidirectional GRU (BGRU) | 0.958 | 0.958 | 0.958 |
| Bidirectional LSTM (BLSTM) | 0.950 | 0.943 | 0.946 |

4.3.2. *The effect of embedding vector.* Figure 7 shows the effect of the bidirectional RNN architectures on different embedding dimensions. As seen in this figure, the embedded BGRU achieves the best performance with 30 dimensions and is more stable compared with embedded BLSTM. This improvement is owing to the fact that the embedded space

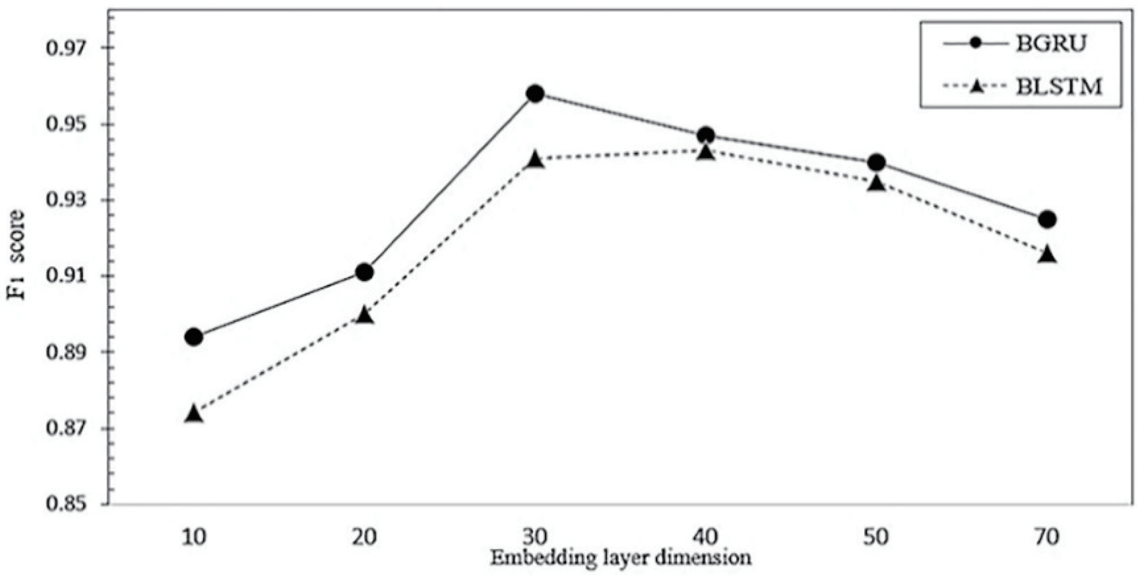


Figure 7. Performance results on F1 score varying different embedding sizes

can capture a highly nonlinear function in the original space while being easier to train the network model. We believe that the representation learned from the embedding vector space leverages the predictive power of RNNs for characterizing movement patterns of fishing gear.

5. **Conclusion.** A deep learning approach to detecting types of fishing gear from ship movement data has been proposed in this study. Our approach first deals with the data quality issue in VMS data records by identifying segments of data points that capture movement patterns of different gear with the extended DBSCAN. Once the segmentation was performed, we extract feature vectors of each segment to describe patterns of fishing gear. Finally, we designed an embedded GRU architecture that encodes a sequence of feature vectors to characterize fishing gear types. Compared with traditional models, we demonstrated that the ship tracking data predicted by our proposed approach has the advantages of high accuracy and stability. This approach helps to reduce the limitations of human supervision in detection of fishing activity. In the future work, the GRU network needs to be further optimized to improve precision results on different types of fishing gear. Moreover, we will work out the method of trajectory segmentation to improve the model's accuracy.

**Acknowledgement.** We would like to be thankful to be the Thailand Command Center for Combating Illegal Fishing for providing VMS data and all technical supports for this study. The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

## REFERENCES

[1] S.-K. Chang, Application of a vessel monitoring system to advance sustainable fisheries management – Benefits received in Taiwan, *Marine Policy*, vol.35, no.2, pp.116-121, 2011.

[2] M. J. Witt and B. J. Godley, A step towards seascape scale conservation: Using Vessel Monitoring Systems (VMS) to map fishing activity, *PLoS One*, vol.2, no.10, DOI: 10.1371/journal.pone.0001111, 2007.

[3] Y. Wang, Y. Wang and J. Zheng, Analyses of trawling track and fishing activity based on the data of Vessel Monitoring System (VMS): A case study of the single otter trawl vessels in the Zhoushan fishing ground, *J. Ocean Univ. China*, vol.14, no.1, pp.89-96, DOI: 10.1007/s11802-015-2467-6, 2015.

[4] X. Jiang, D. L. Silver, B. Hu, E. N. de Souza and S. Matwin, Fishing activity detection from AIS data using autoencoders, in *Advances in Artificial Intelligence. Canadian AI 2016. Lecture Notes in Computer Science*, R. Khoury and C. Drummond (eds.), Cham, Springer International Publishing, 2016.

[5] Y. Zong, H. Huang, F. Hong, Y. Zhen and Z. Guo, Recognizing fishing activities via VMS trace analysis based on mathematical morphology, *2016 Techno-Ocean*, Kobe, Japan, pp.465-470, DOI: 10.1109/Techno-Ocean.2016.7890699, 2016.

[6] E. N. de Souza, K. Boerder, S. Matwin and B. Worm, Correction: Improving fishing pattern detection from satellite AIS using data mining and machine learning, *PLoS One*, vol.11, no.9, 2016.

[7] M. I. Marzuki, P. Gaspar, R. Garello, V. Kerbaol and R. Fablet, Fishing gear identification from vessel-monitoring-system-based fishing vessel trajectories, *IEEE J. Ocean. Eng.*, vol.43, no.3, 2018.

[8] H. Huang, F. Hong, J. Liu, C. Liu, Y. Feng and Z. Guo, FVID: Fishing vessel type identification based on VMS trajectories, *J. Ocean Univ. China*, vol.18, no.2, pp.403-412, 2019.

[9] P. Adibi et al., Predicting fishing effort and catch using semantic trajectories and machine learning, *MASTER@PKDD/ECML*, pp.83-99, 2019.

[10] L. Pipanmekaporn and S. Kamonsantiroj, A deep learning approach for fishing vessel classification from VMS trajectories using recurrent neural networks, *Advances in Intelligent Systems and Computing*, vol.1152, pp.135-141, 2020.

[11] S. Hochreiter and J. Schmidhuber, Long short-term memory, *Neural Comput.*, vol.9, no.8, pp.1735-1780, DOI: 10.1162/neco.1997.9.8.1735, 1997.

[12] J. Chung, C. Gulcehre, K. Cho and Y. Bengio, Empirical evaluation of gated recurrent neural networks on sequence modeling, *arXiv.org*, arXiv: 1412.3555, 2014.

[13] S. Rikukawa, H. Mori and T. Harada, Recurrent neural network based stock price prediction using multiple stock brands, *International Journal of Innovative Computing, Information and Control*, vol.16, no.3, pp.1093-1099, 2020.

[14] Y. Wang, X. Chen and X. Ye, Prediction of network protocol data flow based on a recurrent neural network, *International Journal of Innovative Computing, Information and Control*, vol.15, no.4, pp.1381-1395, 2019.

[15] J. Sander, M. Ester, H. P. Kriegel and X. Xu, Density-based clustering in spatial databases: The algorithm GDBSCAN and its applications, *Data Min. Knowl. Discov.*, vol.2, no.2, pp.169-194, 1998.

[16] A. F. Mohammed and W. R. Baiee, The GIS based criminal hotspot analysis using DBSCAN technique, *IOP Conf. Ser. Mater. Sci. Eng.*, vol.928, DOI: 10.1088/1757-899X/928/3/032081, 2020.

[17] M. Perumal and B. Velumani, Design and development of a spatial DBSCAN clustering framework for location prediction – An optimization approach, *2018 the 3rd ICCES*, 2018.

[18] D. Birant and A. Kut, ST-DBSCAN: An algorithm for clustering spatial-temporal data, *Data Knowl. Eng.*, vol.60, no.1, pp.208-221, DOI: 10.1016/j.datak.2006.01.013, 2007.

[19] Y. Zheng, Q. Li, Y. Chen, X. Xie and W.-Y. Ma, Understanding mobility based on GPS data, *Proc. of the 10th International Conference on Ubiquitous Computing (UbiComp'08)*, 2008.

[20] A. S. Júnior, B. N. Moreno, V. C. Times, S. Matwin and L. dos A. F. Cabral, GRASP-UTS: An algorithm for unsupervised trajectory segmentation, *Int. J. Geogr. Inf. Sci.*, vol.29, no.1, 2015.

[21] E. Miluzzo et al., Sensing meets mobile social networks: The design, implementation and evaluation of the cenceme application, *Proc. of the 6th ACM Conference on Embedded Network Sensor Systems (SenSys'08)*, DOI: 10.1145/1460412.1460445, 2008.

[22] D. P. Kingma and L. J. Ba, Adam: A method for stochastic optimization, *arXiv.org*, arXiv: 1412.6980, 2014.

[23] Y. Bengio, Practical recommendations for gradient-based training of deep architectures, in *Neural Networks: Tricks of the Trade. Lecture Notes in Computer Science*, G. Montavon, G. B. Orr and K. R. Müller (eds.), Berlin, Heidelberg, Springer, 2012.

[24] T. Chen and C. Guestrin, XGBoost: A scalable tree boosting system, *Proc. of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, vol.42, no.8, pp.785-794, DOI: 10.1145/2939672.2939785, 2016.