Spatio-Temporal Deep Learning for Ocean Current Prediction Based on HF Radar Data

Nathachai Thongniran and Peerapon Vateekul
Department of Computer Engineering
Faculty of Engineering
Chulalongkorn University
6071009021@student.chula.ac.th
peerapon.v@chula.ac.th

Kulsawasd Jitkajornwanich
Department of Computer Science
Faculty of Science
King Mongkut's Institute of Technology Ladkrabang
kulsawasd.ji@kmitl.ac.th

Siam Lawawirojwong and Panu Srestasathiern
GISTDA Academy
Geo-Informatics & Space Technology Development Agency
siam@gistda.or.th
panu@gistda.or.th

Abstract—**Ocean surface current prediction is necessary to carry a variety of marine activities, such as disaster monitoring, search and rescue operations, etc. There are three traditional forecasting approaches: (i) numerical based approach, (ii) time series based approach and (iii) machine learning based approach. Unfortunately, their prediction accuracy was limited since they did not cooperate with spatial and temporal effects together. In this paper, we present a novel current prediction model, which is a combination between Convolutional Neural Network (CNN) to extract spatial characteristic and Gated Recurrent Unit (GRU) to find a relationship of temporal characteristic. The dataset is collected by high frequency (HF) radar station's located along coastal Thailand's gulf by GISTDA from 2014 to 2016. It was an intensive experiment comparing our method and eight existing methods, e.g., ARIMA, kNN, Perceptron, Multilayer Perceptron (MLP), etc. The results show that our network outperforms almost all baselines in terms of RMSE for 11.21% and 27.01% averaging improvement on U and V components, consecutively.**

Surface current forecasting; HF radar; deep learning; spatio-temporal; convolutional neural network; gated recurrent unit;

I. Introduction

Ocean current prediction is crucial for various tasks in marine activities, such as disaster monitoring, monitoring coastal water quality, search and rescue operations, pollution trajectories, sea navigation, etc. For each grid area on the ocean, the current direction is usually affected by its neighbors' direction. Also, the current direction during the same season is usually not much different. To accurate forecast the current direction, it is important to capture both spatial (neighbors' direction) and temporal (past direction) effects.

There were many attempts to forecast ocean currents direction, which can be categorized into three approaches. First, the numerical based approach is based on a set of predefined rules [1]. However, this approach requires extremely high computing resources [2]. Second, it is a time series approach, e.g., Moving Average (MA) and Autoregressive Integrated Moving Average (ARIMA). Unfortunately, this approach only relies on temporal effect regardless of spatial effect. Finally, many machine learning techniques were employed to predict current direction, such as kNN [3], Perceptron [4], Multilayer Perceptron (MLP), Self Organizing Map (SOM) [5], etc. Although some attempts can show interesting performance, all of them relied on either neighbors' current direction (spatial effect) or temporal effect. Furthermore, deep learning network has shown its success in many domains of application; however, it has never been successfully applied to current prediction task.

In this paper, we aim to improve the accuracy of forecasting system by using deep learning approach. We introduce Gated Recurrent Unit (GRU) to capture temporal characteristic and Convolutional Neural Network (CNN) that can extract spatial characteristic from the nearby location also have the ability to visualize [6] ocean current patterns for more understanding the behavior of it. The current data, which was collected from 18 high frequency (HF) radar stations which are located along coastal Thai gulf by GISTDA (Geo-Informatics and Space Technology Development Agency), was being used in this research, consisting of three years of data since 2014 to 2016.

This paper is organized as follows. Background knowledge such as Thai gulf current circulation and neural network related is described in Chapter II. Related works on current prediction task are summarized in Chapter III. The proposed method is presented in Chapter IV. An experimental setup which consists of data preparation and forecasting model is illustrated in Chapter V. Experimental results are presented in Chapter VI and the conclusion is summarized Chapter VII.

978-1-7281-0719-6/19/$31.00 ©2019 IEEE

II. Background

A. Ocean Current Circulation in Thai Gulf

Generally, tides, surface wind, streamflow, density and seasons are the main factors of ocean surface current circulation. Thai gulf water circulation is mainly affected by two monsoons: (i) Southwest monsoon generally starts from May to September and (ii) Northeast monsoon generally starts from October to February [7].

B. High-Frequency Radar and Data Collection

Ocean surface currents velocity and direction in the coastal area are measured by high-frequency radio waves. Coastal stations initiate a signal and monitor it returning back to the station then using backscattered radio wave to calculate surface currents. There are 3,097 grid points from GISTDA from 2014 to 2016 that are being used as a dataset in this study.

C. Neural Network (NN)

The model is inspired by biological nervous systems. It is trained by train data to predict unseen data which consists of many components, such as (i) activation function that being used to decide which neuron should be activated or not. Popular activation functions are sigmoid function, Rectified Linear Unit (ReLU) [8], (ii) loss function which is a method of evaluating how well your model performs on the dataset. If the model performs well on the dataset, the output from the loss function of the model should be low. It's a very important part while tuning hyperparameter.

Perceptron, a single hidden layer neural network which contains only one hidden layer between the input and output layer.

III. Related Works

Many researches utilize HF Radar data for ocean surface current forecasting with different methods that can be categorized into three categories.

A. Numerical Based Approach

Frolov et al. [9] developed short-term prediction model by using linear autoregressive technique on past 48 hours of HF-radar data as an input to predict 48 future hours based on Monterey coast since 1st Jan 2006 to 30th Oct 2010. Barrick et al. [10] presented a short-term predictive system (STPS) which only required a few hours of previous data and open-modal analysis (OMA) as a preprocessing method. However, this approach requires high computational costs and predefined parameters manually assigned by experts.

B. Time Series Based Approach

Ocean current predictions can be considered as a time series prediction. Some researchers employed ARIMA to benchmark their proposed models [3]. However, it considers only temporal effects regardless of spatial effect.

C. Machine Learning Based Approach

Kalinić et al. [5] proposed ocean current prediction by using Self-Organizing Map (SOM) that uses wind data generated from numerical weather prediction (NWP) and HP radar. Saha et al. [11] attempted to make daily predictions of ocean currents by combining an artificial neural network (ANN) with a numerical method. The system was implemented at two locations in the Indian ocean. Jirakittayakorn et al. [3] proposed Temporal kNN to predict short-term ocean current prediction based on HF radar of Thailand's gulf 3 years long from 2014 to 2016. The model is designed to capture seasonal and temporal characteristics. However, only traditional machine learning techniques have been employed and they did not corporate both spatial and temporal effects.

IV. Proposed method

Spatio-Temporal deep learning model that takes advantage of noticeable domain properties by using Convolutional Neural Network (CNN) to capture spatio property and Gated Recurrent Unit (GRU) to capture temporal property.

A. Convolutional Neural Network (CNN)

CNN originally created for digits recognition research by using filter to create a feature map as an input for the next layer, mainly consists of two main layers. Example of architecture is shown in Fig. 1.

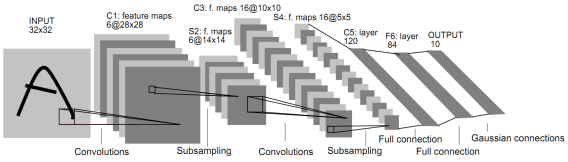

Figure 1. Convolutional neural network architecture (reference from Fig. 2. [12])

1) Convolution Layer

The layer is used to find a feature from a group of nearby input by using a dot product of matrix. In this layer, there have many components that need to be considered, such that filter size, stride size, number of filters and number of channels.

2) Pooling Layer

The layer's objective is to reduce computational cost. Popular pooling layers are max and average pooling layer.

B. Recurrent Neural Network (RNN)

Neural Network that's designed for sequential data by using internal memory. Internal memory makes RNN able to remember things which have been passed.

C. Gated Recurrent Unit (GRU)

GRU is created for solving the vanishing gradient problem of RNN by implementing a gating mechanism which consists of an update gate and a reset gate [13]. The update gate is used to decide what information to add or not. The reset gate is used to decide how much information to forget.

D. Spatio-Temporal (CNN-GRU)

The proposed model that takes advantage of noticeable domain properties by combining 2 models.

CNN is used as the feature extraction for capturing spatio property, then the result of CNN layer is used as an input for a GRU layer for finding a temporal property, as shown in Fig. 2. Other layers or customizations will be described more in Chapter V.

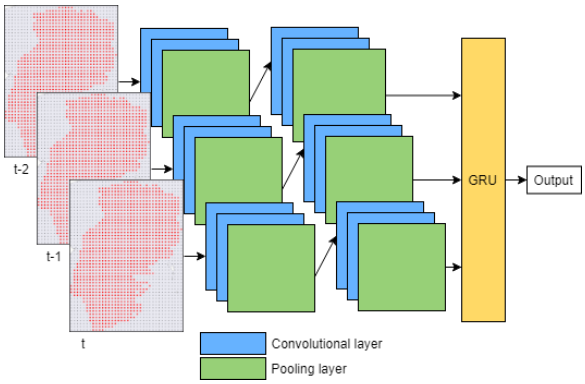

Figure 2. Example of CNN-GRU architecture

V. Experimental Setup

In this section, we are going to discuss how to prepare the data, details of experimental models, and evaluation.

A. Data Preparation

1) Dataset

U and V components of ocean surface currents were collected from HF radar stations which are located alongside Thai gulf. Total of 3,097 grid points were reported since 01/01/2014 to 23/12/2016. The dataset resolution is 2x2 kilometer in spatial resolution and one hour in temporal resolution. Example of records in the dataset is shown in Table I.

TABLE I. EXAMPLE OF RECORDS IN DATASET

Timestamp	Latitude	Longitude	U comp	V comp
2015/6/22 6:00	12.7418	99.9725	0.1560	-14.0680
2015/6/22 6:00	12.7238	99.9725	-7.8070	-16.3430
2015/6/22 6:00	12.7057	99.9725	-13.7860	-17.8560

2) Data Removal

Data availability of HF radar stations is a significant issue that would affect forecasting value [14]. To avoid the problem, we proceeded with two more steps which are removal and imputation. Histogram of grid's data availability over three years is shown in Fig. 3.

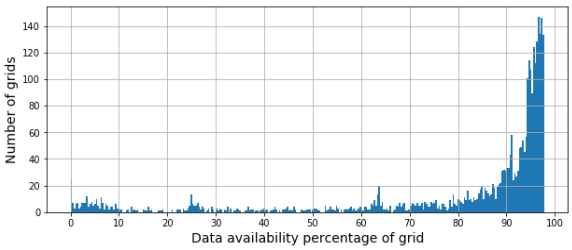

Figure 3. Histogram of grid's data availability over three years

Any grid points that had missing data more than five percent over three long years were removed. As a result, 1,065 remaining grid points are still remaining for our experiment, as shown in Fig. 4.

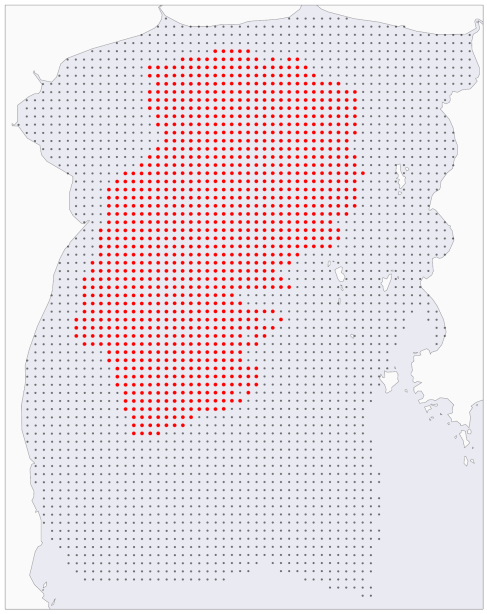

Figure 4. 1,065 of 3,097 grid points are shown in red color

3) Missing Grid Imputation

We basically performed simple two imputation steps, as shown in Fig. 5 below.

Figure 5. Two steps of data imputation

a) Imputation at Boundary Grids

We filled left and right boundaries with the value of the nearest data point, as shown in Step 1 of Fig. 5.

b) Imputation at Non-boundary Grids

We filled the rest with an average of closest available data points, as shown in Step 2 of Fig. 5.

4) Data Normalization

We used min-max normalization as shown in (1) to normalize data. The minimum value was mapped to 0 and the maximum value was mapped to 1.

$$y_i = \frac{x_i - min(x)}{max(x) - min(x)} \quad (1)$$

Where x_i is an original value, y_i is a normalized value, $min(x)$ is the minimum and $max(x)$ is the maximum value of the entire dataset.

5) *Data Partitioning*

According to the nature of ocean currents which is affected by seasonality and the dataset contains only three years of data. We kept the year 2016 as test dataset and year 2014 to 2015 as training and validation datasets.

Splitting year 2014 to 2015 dataset into training and validation datasets, we further splitted it into 2 types: (a) non-time related and (b) time related which depends on the model.

a) Non-Time Related

For non-time related models such as Perceptron, and CNN, the data was divided into 33% and 67% as validation and training datasets, respectively.

b) Time Related

For time related model such as Lookback, GRU, and CNN-GRU, the data during 2014 was used as training data, while the remaining data (2015) was used as validation data.

B. Forecasting Model

In the experiment, there were nine algorithms including our proposed method. Each algorithm has been optimized its performance using various hyperparameters on the validation dataset.

1) Lookback Timestep

A naive model, forecasted value equals to last k timesteps of observed value as shown in (2). We performed varied of k parameters, as follow, 1, 2, 3, 4, 5, 24, 168, 720, 2,160 and 8,640. The result indicates that 1 is an optimal parameter for k on both U and V components.

$$y_t = x_{t-k} \quad (2)$$

Where x_{t-k} is k previous timesteps of observed value, y_t is a predicted value.

2) Moving Average (MA)

Moving Average is an unweighted mean of the previous k timesteps as shown in (3). k parameter since 2 to 4 has been deployed for tuning. The result showed that 2 was an optimal k on both U and V components.

$$y_t = \frac{1}{k} \sum_{i=1}^{k} x_{t-i} \quad (3)$$

Where x_{t-i} is i previous timesteps of observed value, y_t is a predicted value.

3) Autoregressive Integrated Moving Average (ARIMA)

Number of parameters have been deployed in the tuning process while p is the order of the autoregressive model, d is the degree of difference, and q is the order of the moving-average model also including seasonal order parameters which contain (P, D, Q and s) order of the seasonal component for AR parameters, differences, MA parameters, and periodicity, respectively. Optimal parameters are No. 1 and No. 3 in Table II for U and V components, respectively.

TABLE II. EXAMPLE OF ARIMA PARAMETERS THAT HAVE BEEN USED WHILE TUNING

No.	p	d	q	P	D	Q	s
1	0	1	1	-	-	-	-
2	1	0	1	-	-	-	-
3	2	0	0	-	-	-	-
4	2	1	0	-	-	-	-
5	0	0	1	0	0	1	1

4) Temporal kNN

There are three parameters which are k (number of neighbors), *weight* and n (number of lookback days) of Temporal kNN model. We used k from 1 to 6, distance and uniform for *weight* and n from 1 to 6 for search space of the model. The result showed optimal k, *weight* and n parameters are 6, uniform and 6 for both U and V components, respectively.

5) Perceptron

We tuned only the number of hidden units on a hidden layer, fixed ReLU for an activation function of the hidden layer and fixed sigmoid function for an activation function of output layer. In tuning process, 1,024, 2,048 and 4,096 were used as number of hidden units of hidden layer. The result shows 2,048 is an optimal value for both U and V components.

6) Multilayer Perceptron (MLP)

By stacking hidden layers, this allowed the model to be able to extract more hierarchical structure information. In this paper, there are two hidden layers in MLP, where the number of hidden units of both layers are the same. To obtain the most suitable number of hidden units, we varied and compared different the number of hidden units: 512, 1,024, 2,048, 4,096 and 8,192. From our experiment, the best number of hidden units is 4,096.

7) Convolutional Neural Network (CNN)

Since CNN architecture is varied, the combination of architectures and hyperparameters is infinite. We used a mature architecture such as VGG [15] as a starting point then fine tune from it. The optimal architecture is closed to CNN-GRU without GRU layer that will be illustrated in CNN-GRU section.

8) Gated Recurrent Unit (GRU)

We fixed the number of GRU layers to be one. Only number of units was varied. As the result, an optimal number of units value on both U and V components was 2,048.

9) Spatio-Temporal (CNN-GRU)

This is a forecasting model that is used in our framework. We started with our optimal CNN and GRU then fine tune from that, new activation function, such as ELU [16] and PELU [17] were also being used in a hyperparameter tuning process. The optimal hyperparameters were summarized in Table III for both U and V components.

TABLE III. OPTIMAL HYPERPARAMETER OF CNN-GRU

Convolutional layer 1 and 2	#filters	32
	Kernel size	3x3
	Activation function	PELU
Pooling layer	Type	Max
	Pool size	2x2
Dropout layer	Rate	0.2
Convolutional layer 3 and 4	#filters	64
	Kernel size	3x3
	Activation function	PELU
Pooling layer	Type	Max
	Pool size	2x2
Dropout layer	Rate	0.2
GRU layer	#units	256

C. Evaluation

RMSE of both U and V components on next one timestep, which is equal to the next one hour on the year 2016 of the dataset. The RMSE equation is shown in (4).

$$RMSE = \sqrt{\frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n}(y_{predict} - y_{actual})^2} \quad (4)$$

Where $y_{predict}$ is predicted value, y_{actual} is observed value, n is a number of grid points and m is a number of timesteps.

VI. Experimental Results

In this section, the result of nine models is illustrated and discussed.

Table IV and V show the overall performance and compare to the CNN-GRU model. The winner is CNN-GRU by 11.21% and 27.01%, in term of RMSE on U and V components which are calculated from averaging over "Differences (%)" column of all other models.

TABLE IV. AVERAGE RMSE ON U COMPONENT COMPARING TO CNN-GRU

Model	RMSE (cm/s)	Differences (%)
Lookback	5.068	-11.03
MA	5.422	-16.84
ARIMA	5.252	-14.14
Temporal kNN	7.840	-42.49
Perceptron	4.529	-0.43
MLP	4.603	-2.04
CNN	4.518	-0.20
GRU	4.625	-2.50
CNN-GRU	**4.509**	

TABLE V. AVERAGE RMSE ON V COMPONENT COMPARING TO CNN-GRU

Model	RMSE (cm/s)	Differences (%)
Lookback	11.167	-33.69
MA	14.324	-48.30
ARIMA	11.128	-33.46
Temporal kNN	11.862	-37.58
Perceptron	9.263	-20.06
MLP	9.201	-19.52
CNN	8.799	-15.84
GRU	8.018	-7.64
CNN-GRU	**7.405**	

In addition, CNN-GRU performed better than CNN and GRU individually on both U and V components. This proves that a combination of spatial and temporal effects is important as we expected from noticeable domain properties.

Average RMSE of each hour on both U and V components are shown in Fig. 6 and Fig. 7. In V component, obviously shows that temporal property might be the dominant property over a spatial property from the result of GRU model and CNN-GRU which clearly sees that they are winning overall models at any point of an hour with a large gap.

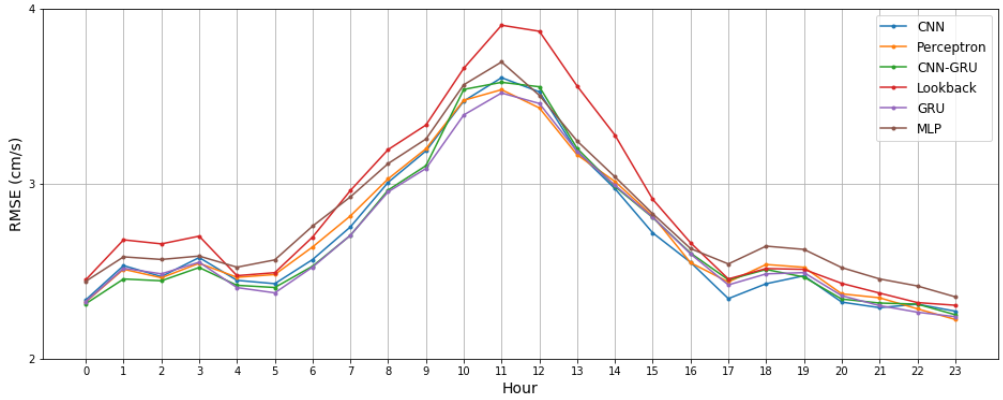

Figure 6. Average RMSE (hour) of top 6 models on U component

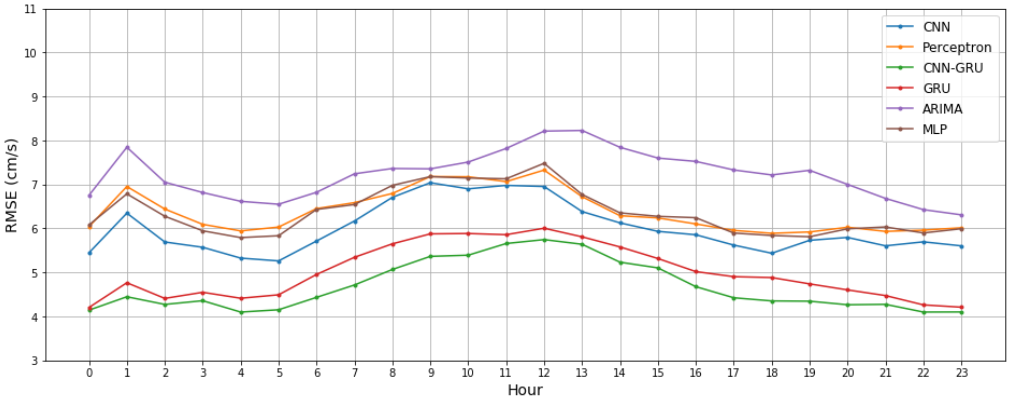

Figure 7. Average RMSE (hour) of top 6 models on V component

VII. Conclusion

This paper proposes the Spatio-Temporal model (CNN-GRU) that takes advantage from noticeable domain properties by using CNN to capture spatio effect and GRU to capture temporal effect, respectively. The model is evaluated and compared to other eight existing models on the current data in Gulf of Thailand from 2014 to 2016. The experimental result shows that our model outperforms almost all models on both U and V components for 11.21% and 27.01% on average.

References

[1] Allard, R., et al. (2014). "The US Navy coupled ocean-wave prediction system." Oceanography 27(3): 92-103.

[2] Rozier, D., et al. (2007). "A reduced-order Kalman filter for data assimilation in physical oceanography." SIAM review 49(3): 449-465.

[3] Jirakittayakorn, A., et al. (2017). Temporal kNN for short-term ocean current prediction based on HF radar observations. Computer Science and Software Engineering (JCSSE), 2017 14th International Joint Conference on, IEEE.

[4] Ren, L., et al. (2018). "Short-Term forecasting of coastal surface currents using high frequency radar data and artificial neural networks." Remote Sensing 10(6): 850.

[5] Kalinić, H., et al. (2017). "Predicting ocean surface currents using numerical weather prediction model and Kohonen neural network: a northern Adriatic study." Neural Computing and Applications 28(1): 611-620.

[6] Qin, Z., et al. (2018). "How convolutional neural network see the world-A survey of convolutional neural network visualization methods." arXiv preprint arXiv:1804.11191.

[7] Buranapratheprat, A. (2008). Circulation in the upper gulf of Thailand: A review.

[8] Nair, V. and G. E. Hinton (2010). Rectified linear units improve restricted boltzmann machines. Proceedings of the 27th international conference on machine learning (ICML-10).

[9] Frolov, S., et al. (2012). "Improved statistical prediction of surface currents based on historic HF-radar observations." Ocean Dynamics 62(7): 1111-1122.

[10] Barrick, D., et al. (2012). "A short-term predictive system for surface currents from a rapidly deployed coastal HF radar network." Ocean Dynamics 62(5): 725-740.

[11] Saha, D., et al. (2016). "A combined numerical and neural technique for short term prediction of ocean currents in the Indian Ocean." Environmental Systems Research 5(1): 4.

[12] Lecun, Y., et al. (1998). "Gradient-based learning applied to document recognition." Proceedings of the IEEE 86(11): 2278-2324.

[13] Cho, K., et al. (2014). "Learning phrase representations using RNN encoder-decoder for statistical machine translation." arXiv preprint arXiv:1406.1078.

[14] Richard E. Thomson, W. E. (2014). "Data Analysis Methods in Physical Oceanography." 3.

[15] Simonyan, K. and A. Zisserman (2014). "Very deep convolutional networks for large-scale image recognition." arXiv preprint arXiv:1409.1556.

[16] Clevert, D.-A., et al. (2015). "Fast and accurate deep network learning by exponential linear units (elus)." arXiv preprint arXiv:1511.07289.

[17] Trottier, L., et al. (2017). Parametric exponential linear unit for deep convolutional neural networks. 2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA), IEEE.