

Ocean Current Prediction in the Gulf of Thailand Using Spatio-Temporal Deep
Learning on High-Frequency Radar



A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Science in Computer Science
Department of Computer Engineering
Faculty of Engineering
Chulalongkorn University
Academic Year 2019
Copyright of Chulalongkorn University

การทำนายความเร็วกระแสน้ำในอ่าวไทยด้วยแบบจำลองการเรียนรู้เชิงพื้นที่และเวลาบนข้อมูลเรดาร์
ความถี่สูง



วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต
สาขาวิชาวิทยาศาสตร์คอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2562
ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

Thesis Title Ocean Current Prediction in the Gulf of Thailand Using Spatio-
Temporal Deep Learning on High-Frequency Radar
By Mr. Nathachai Thongniran
Field of Study Computer Science
Thesis Advisor Assistant Professor PEERAPON VATEEKUL, Ph.D.
Thesis Co Advisor Siam Lawawirojwong, Ph.D.

Accepted by the Faculty of Engineering, Chulalongkorn University in Partial Fulfillment
of the Requirement for the Master of Science

THESIS COMMITTEE

..... Dean of the Faculty of Engineering
(Professor SUPOT TEACHAVORASINSKUN, D.Eng.)

..... Chairman
(Professor BOONSERM KIJSIRIKUL, D.Eng.)

..... Thesis Advisor
(Assistant Professor PEERAPON VATEEKUL, Ph.D.)

..... Thesis Co-Advisor
(Siam Lawawirojwong, Ph.D.)

..... Examiner
(Assistant Professor VEERA MUANGSIN, Ph.D.)

..... External Examiner
(Kulsawasd Jitkajornwanich, Ph.D.)

ณัฐชัย ทองนิรันดร์ : การทำนายความเร็วกระแสน้ำในอ่าวไทยด้วยแบบจำลองการเรียนรู้เชิงพื้นที่และเวลาบนข้อมูลเรดาร์ความถี่สูง. (Ocean Current Prediction in the Gulf of Thailand Using Spatio-Temporal Deep Learning on High-Frequency Radar) อ.ที่ปรึกษาหลัก : ผศ. ดร.พีรพล เวทีกุล, อ.ที่ปรึกษาร่วม : ดร.สยาม ลววิโรจน์วงศ์

การทำนายความเร็วกระแสน้ำเป็นงานที่สำคัญอย่างยิ่งในการปฏิบัติทางน้ำ ยกตัวอย่างเช่น การค้นหาและช่วยเหลือ การสังเกตการณ์ภัยพิบัติ การทำนายพลังงานไฟฟ้าที่ถูกผลิตมาจากความเร็วกระแสน้ำ และอื่นๆ ในปัจจุบันมีทั้งหมด 3 เทคนิคหลักในการทำนาย (i) การทำนายเชิงตัวเลข (ii) การทำนายเชิงเวลา และ (iii) การทำนายด้วยศาสตร์การเรียนรู้ของเครื่อง (machine learning) อย่างไรก็ตามความแม่นยำของเทคนิคเหล่านั้นยังถูกจำกัด เนื่องจากไม่ได้พิจารณาผลของเชิงพื้นที่และเวลาพร้อมกันและไม่ได้พิจารณาข้อมูลเชิงมหาสมุทร (oceanic input) งานวิจัยนี้นำเสนอแบบจำลองการทำนายความเร็วกระแสน้ำที่พิจารณาผลของเชิงพื้นที่และเวลาพร้อมกัน โดยใช้นิวรอลเน็ตเวิร์กแบบคอนโวลูชัน (Convolutional Neural Network) ร่วมกับโครงข่ายประตูป้อนกลับ (Gated Recurrent Unit) และร่วมกับข้อมูลเชิงมหาสมุทร ข้อมูลเรดาร์ความถี่สูงที่ใช้ในการทดลองได้รับมาจากสถานีซึ่งถูกติดตั้งตามชายฝั่งอ่าวไทย (Gulf of Thailand) โดยสำนักงานพัฒนาเทคโนโลยีอวกาศและภูมิสารสนเทศ (Geo-Informatics and Space Technology Development Agency) ตั้งแต่ปี 2014 ถึง 2016 การทดลองจะเป็นการเปรียบเทียบแบบจำลองที่นำเสนอกับวิธีการอื่นๆ เช่น ARIMA เพอร์เซ็ปตรอน (Perceptron) และอื่นๆ โดยใช้ค่ารากที่สองของค่าความคลาดเคลื่อนกำลังสองเฉลี่ย (RMSE) เป็นตัวชี้วัด บน ส่วนประกอบยูและวี (U and V components) ของกระแสน้ำ



สาขาวิชา วิทยาศาสตร์คอมพิวเตอร์
ปีการศึกษา 2562

ลายมือชื่อนิสิต

ลายมือชื่อ อ.ที่ปรึกษาหลัก

ลายมือชื่อ อ.ที่ปรึกษาร่วม

6071009021 : MAJOR COMPUTER SCIENCE

KEYWORD: Surface current forecasting, HF radar, deep learning, spatio-temporal, convolutional neural network (CNN), gated recurrent unit, attention mechanism, transfer learning

Nathachai Thongniran : Ocean Current Prediction in the Gulf of Thailand Using Spatio-Temporal Deep Learning on High-Frequency Radar. Advisor: Asst. Prof. PEERAPON VATEEKUL, Ph.D. Co-advisor: Siam Lawawirojwong, Ph.D.

Ocean surface current prediction is a crucial task for a variety of marine activities, such as disaster monitoring, search and rescue operations, power forecasting, and etc. There are three traditional forecasting approaches: (i) numerical based approach, (ii) time series based approach and (iii) machine learning based approach. However, their prediction accuracy was limited as they did not cooperate with spatial and temporal effects together, including oceanic knowledge is also not considered.

This paper introduces the ocean surface prediction model that accounts for spatial and temporal characteristics by a combination between CNN and GRU and also the incorporation of oceanic inputs which are month number, lunar effect, and hour number. The experiment compared the proposed model with an existing method, e.g., ARIMA, Perceptron, Temporal kNN and etc. by using RMSE as a metrics on both U and V components of dataset that was collected by high frequency (HF) radar stations located along coastal Gulf of Thailand by GISTDA from 2014 to 2016.

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

Field of Study: Computer Science

Academic Year: 2019

Student's Signature

Advisor's Signature

Co-advisor's Signature

ACKNOWLEDGEMENTS

This thesis is one of the most challenging works in my life. I am interested in a computer since I was young and always want to fall into it. It may be because I did not graduate with a bachelor's degree in computer science, so I decided to do a master's degree in this field. Throughout 2 years of studying in master degree while proceeding in the research, I have faced the obstacle, problem, hope, and happiness like all we did, my classmates. Working and studying for a master's degree taught me many lessons particularly time management, task prioritization, and mind control. I am sure that after getting through this, I can do anything. Anyway, there was plenty of support from kind people that make the thesis completed.

I have learned a lot from doing this research, especially working with my advisor. Asst. Prof. Peerapon Vateekul, Ph.D., helps me in a lot of things not just about studying. In terms of studying, he always has time for encouraging, supporting and suggestion. A variety of knowledge and techniques that he coaches me because of his solid background, make me understand it clearly which is invaluable.

Valuable data on the thesis was provided by GISTDA (Geo-Informatics and Space Technology Development Agency). I got a warm welcome from all GISTDA members, especially, Siam Lawawirojwong, Ph.D. and Panu Srestasathiern, Ph.D. every time I got stuck and tented to have a suggestion and support. A lot of excellent suggestions came from those who have a high level of combination between computer science and ocean knowledge.

Thank you all other committees which consist of Prof. Boonserm Kijirikul, D.Eng., Assistant Professor Veera Muangsin, Ph.D. and Kulsawasd Jitkajornwanich, Ph.D. that supports me by being the committees and always giving a great suggestion that is honored and valuable for me.

Many thanks for everyone in "Data Mining Group, MIND Lab" group. We have shared many things together. I would not complete the thesis if I did not have the group. In the last, I want to thank all the people around me for always supporting me for doing the project from the first day to graduation day.

Nathachai Thongniran

TABLE OF CONTENTS

	Page
ABSTRACT (THAI).....	iii
ABSTRACT (ENGLISH).....	iv
ACKNOWLEDGEMENTS.....	v
TABLE OF CONTENTS.....	vi
LIST OF TABLES.....	xi
LIST OF FIGURES.....	xiii
Chapter 1 Introduction.....	1
1.1 Motivation.....	1
1.2 Objective.....	2
1.3 Scope.....	2
1.4 Expected Result.....	2
1.5 Research Plan.....	3
1.6 Publications.....	3
Chapter 2 Background Knowledge.....	4
2.1 Ocean Current Circulation in Thai Gulf.....	4
2.1.1 Monsoon.....	4
2.1.2 Lunar Illumination.....	4
2.1.3 Sea Breeze and Land Breeze.....	5
2.2 High Frequency Radar and Data Collection.....	6
2.2.1 HF Radar Station in Thailand.....	6
2.2.2 Measurement.....	7

2.2.3 Data Loss	7
2.2.4 Current Measurement	7
2.3 Tide	9
2.3.1 Tidal Force	9
2.3.2 Spring Tide and Neap Tide.....	10
2.3.3 Tide with Different Viewpoint.....	12
2.4 Neural Network (NN).....	13
2.4.1 Perceptron.....	14
2.4.2 Activation Function.....	14
2.4.3 Loss Function.....	16
2.5 Convolutional Neural Network (CNN)	16
2.5.1 Convolution Layer	16
2.5.2 Pooling Layer	19
2.6 Recurrent Neural Network (RNN).....	19
2.7 Gated Recurrent Unit (GRU)	20
2.8 Attention Mechanism	21
2.8.1 Additive Attention.....	22
2.8.2 Multiplicative Attention	22
2.8.3 Self-Attention.....	22
2.9 Transfer Learning	23
2.10 One-Hot Encoding.....	23
Chapter 3 Related Works	24
3.1 Model Approach.....	24
3.1.1 Numerical Based Approach.....	24

3.1.2 Time Series Based Approach	24
3.1.3 Machine Learning Based Approach	24
3.2 Ocean Current Prediction in Gulf of Thailand	25
3.3 Room for improvement	26
3.3.1 Addressing Spatial and Temporal Effect Together	26
3.3.2 Domain Knowledge Input.....	26
3.3.3 Deep Learning Technique	26
Chapter 4 Proposed Method.....	27
4.1 High Frequency Dataset.....	27
4.1.1 Dataset	27
4.1.2 Data Removal	29
4.1.3 Missing Grid Imputation	30
4.1.4 Data Normalization	30
4.2 Oceanic Input Dataset.....	31
4.2.1 Month Number	31
4.2.2 Lunar Effect.....	31
4.2.3 Hour Number	32
4.3 Proposed Model.....	33
4.3.2 Oceanic Input	34
4.3.3 Attention Mechanism.....	34
4.3.4 Transfer Learning.....	35
Chapter 5 Experimental Setup.....	36
5.1 Input Data	36
5.1.1 Data.....	36

5.1.2 Data Partitioning	36
5.2 Performance Evaluation.....	36
5.3 Forecasting Model.....	37
5.3.1 Lookback Timestep	37
5.3.2 Moving Average (MA).....	37
5.3.3 Autoregressive Integrated Moving Average (ARIMA)	38
5.3.4 Temporal kNN.....	38
5.3.5 Perceptron.....	38
5.3.6 Multilayer Perceptron (MLP).....	38
5.3.7 Convolutional Neural Network (CNN).....	39
5.3.8 Gated Recurrent Unit (GRU).....	39
5.3.9 Spatio-Temporal (CNN-GRU).....	39
5.3.10 CNN-GRU-Input.....	40
5.3.11 CNN-GRU-Input (All)-Att.....	40
5.3.12 CNN-GRU-Input (All)-Att-TL	41
Chapter 6 Experiment and Result	42
6.1 Forecasting Next One Timestep.....	42
6.1.1 Dataset and Partitioning	42
6.1.2 Evaluation.....	43
6.1.3 Result.....	43
6.1.3.1 Overview.....	44
6.1.3.2 Effect of Combination Between Spatial and Temporal Characteristics	45
6.1.3.3 Effect of Oceanic Input	46

6.1.3.4 Effect of Attention Mechanism.....	47
6.1.4.5 Effect of Transfer Learning.....	48
6.2 Rolling Forecasting Up to Next 48 Timesteps	49
6.2.1 Dataset and Partitioning	50
6.2.2 Evaluation.....	51
6.2.3 Result.....	51
6.2.3.1 Overview.....	52
6.2.3.2 Temporal Effect.....	52
6.3 Forecasting Next One Timestep with Different Number of Input Timesteps	53
6.3.1 Dataset and Partitioning	54
6.3.2 Evaluation.....	54
6.3.3 Result.....	54
6.3.3.1 Overview.....	55
6.3.3.2 Tuning on Model with Number of Input Timesteps Equals 13.....	55
Chapter 7 Summary and Future Work	56
7.1 Summary.....	56
7.2 Future Work.....	57
7.2.1 Revise Model Architecture	57
7.2.2 Address Temporal Effect.....	57
7.2.3 Address Sun Position.....	57
REFERENCES	59
VITA.....	63

LIST OF TABLES

	Page
Table 1 Maximum tidal forces of the sun, moon, and planets on the earth	10
Table 2 Month number in one-hot encoding	23
Table 3 Example of records in dataset	27
Table 4 Statistical data of each year on U and V components from HF dataset	28
Table 5 Dataset size on each processing step.....	30
Table 6 Example of month number input.....	31
Table 7 Example of lunar effect input	32
Table 8 Example of hour number input	33
Table 9 Model explanation.....	33
Table 10 Year and dataset in the experiment.....	36
Table 11 Example of predicted value of Lookback model when $k = 1$	37
Table 12 Example of ARIMA parameters that have been used while tuning.....	38
Table 13 Optimal hyperparameter of CNN-GRU.....	39
Table 14 Example of oceanic inputs.....	40
Table 15 Dataset period of forecasting next one timestep experiment	42
Table 16 Model's RMSE of next one timestep forecasting on U and V component, compared to the proposed model.....	44
Table 17 Effect of combination between spatial and temporal characteristics in term of RMSE comparing to CNN-GRU	45
Table 18 Effect of oceanic inputs in term of RMSE comparing to CNN-GRU.....	47
Table 19 Effect of attention mechanism in term of RMSE comparing to CNN-GRU-Input (All).....	47

Table 20 Effect of transfer learning in term of RMSE comparing to CNN-GRU-Input (All)-Att.....	48
Table 21 Dataset period of rolling forecasting up to next 48 timesteps experiment	50
Table 22 RMSE of proposed model with different number of input timesteps on U and V components.....	54

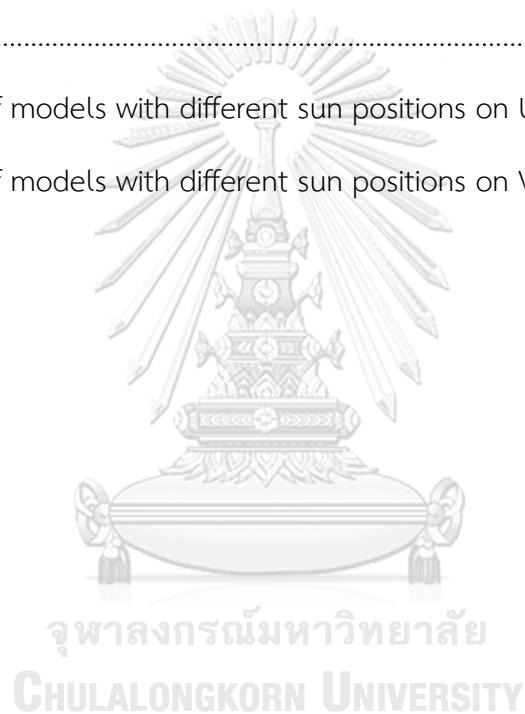


LIST OF FIGURES

	Page
Figure 1 Ocean circulation during the southwest and northeast monsoon	4
Figure 2 The circulations of land and sea breeze	5
Figure 3 Two implementation phases of HF radar stations	6
Figure 4 Transmitting and receiving of HF radar	7
Figure 5 Example of magnitude and direction of reported grid data.....	8
Figure 6 U and V component of ocean current.....	8
Figure 7 Tidal force between the earth and object in space	9
Figure 8 Tidal force of moon to the earth.....	10
Figure 9 Tidal effects from the sun and moon on the earth.....	11
Figure 10 Diagram of the moon's phases	11
Figure 11 The types of tides	12
Figure 12 Tide with different viewpoints	12
Figure 13 Tide types map.....	13
Figure 14 Perceptron architecture	14
Figure 15 Convolutional neural network architecture.....	16
Figure 16 Convolution operation	17
Figure 17 Wide convolution with padding	18
Figure 18 Convolution layer with input size equals 5x5, filter size equals 3x3, and stride size equals 2.....	18
Figure 19 Convolution with number of filters equals 3	18
Figure 20 Example of max pooling operation	19

Figure 21	A recurrent neural network.....	20
Figure 22	Gated recurrent unit	21
Figure 23	The attention mechanism: The soft attention assign weights on different locations of features using softmax.....	21
Figure 24	Example of data splitting that is used in Temporal kNN model research ..	25
Figure 25	3097 sites of data was obtained from GISTDA	28
Figure 26	Histogram of grid data availability over three years.....	29
Figure 27	1,065 of 3,097 grid points are shown in red color.....	29
Figure 28	Two steps of data imputation	30
Figure 29	Example of lunar illumination on each timestep.....	32
Figure 30	Example of lunar effect on each timestep.....	32
Figure 31	Example of CNN-GRU architecture.....	34
Figure 32	Implementation of oceanic inputs in the model	34
Figure 33	Attention mechanism part in the model	35
Figure 34	Model architecture of CNN-GRU-Input (All)-Att-TL.....	35
Figure 35	Example of forecasting value on test dataset of U component.....	43
Figure 36	Example of forecasting value on test dataset of V component	43
Figure 37	Average RMSE (hour) of models on U component	46
Figure 38	Average RMSE (hour) of models on V component.....	46
Figure 39	Average RMSE (hour) of each module on U component	48
Figure 40	Average RMSE (hour) of each module on V component.....	48
Figure 41	Example of rolling forecasting step	49
Figure 42	Input and output grid of CNN model for rolling forecasting	50
Figure 43	RMSE of rolling forecasting up to 48 hours on U component.....	51

Figure 44 RMSE of rolling forecasting up to 48 hours on V component	51
Figure 45 RMSE of rolling forecasting up to 48 hours in box plot, on U and V components respectively	52
Figure 46 Example of adding more timesteps in a model input	53
Figure 47 RMSE of proposed model with different number of input timesteps on U component	55
Figure 48 RMSE of proposed model with different number of input timesteps on V component	55
Figure 49 RMSE of models with different sun positions on U component.....	57
Figure 50 RMSE of models with different sun positions on V component	58



จุฬาลงกรณ์มหาวิทยาลัย

CHULALONGKORN UNIVERSITY

Chapter 1

Introduction

Introduction section consists of 6 subtopics which are motivation, objective, scope, expected result, research plan and publications.

1.1 Motivation

Ocean current prediction is a crucial task for various marine activities, such as disaster monitoring, monitoring coastal water quality, search and rescue operations, pollution trajectories, sea navigation, and etc. For each grid area on the ocean, the surface ocean current is usually affected by its neighbors. Also, the nature of ocean current circulation is mainly affected by its seasonality such as monsoons, lunar illumination, sea breeze, and land breeze.

Lately, an increasing of computing power and ability to handle big data of machine make a deep learning based model has successfully been implemented to a variety of fields for instance image classification, medical disease detection and self-driving car due to fast growing development and capacity to handle a number of data. Using recent promising deep learning techniques will improve the model accuracy by making it more accurate. The addition of oceanic input data also helps the model more understanding about its nature pattern. There is a great chance to improve prediction task which can support all marine activities by implementing these listed techniques above.

There were three approaches of many attempts for surface ocean current prediction tasks. First, the numerical based approach which 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., Lookback, Moving Average (MA) and Autoregressive Integrated Moving Average (ARIMA). Unfortunately, this approach only relies on temporal effect. Lastly, machine learning techniques were deployed to predict surface current velocity, such as Temporal kNN [3], Perceptron [4], Multilayer Perceptron (MLP), Self Organizing Map (SOM) [5], and etc. Some research shows interesting result in term of model accuracy, but still has opportunity to be improved such as consideration temporal and spatial together, establish recent deep learning techniques, and involved with oceanic domain inputs.

In this thesis, we shoot to improve the accuracy of forecasting system by using deep learning approach. We introduce various of modules which consist of, (i) Gated Recurrent Unit (GRU) to capture temporal characteristic, (ii) Convolutional Neural Network (CNN) that can extract spatial characteristic from the nearby location, and also has an ability for visualization [6] for more understanding its behavior, (iii) a combination of CNN and GRU, (iii) utilizing oceanic inputs as follows, month number, lunar effect, and hour number to help the model to understand oceanic seasonal effects, (iv) attention mechanism [7] was employed to focus on a significant grid, (v) and transfer learning was used to utilize pre-trained knowledge.

The ocean surface current dataset, since 2014 to 2016, in this research was collected from 18 high frequency (HF) radar stations located along coastal Gulf of Thailand which are implemented by GISTDA (Geo-Informatics and Space Technology Development Agency).

1.2 Objective

Ocean current prediction model uses deep learning techniques to improve performance, which focuses on noticeable characteristics of its nature, by combining CNN and GRU in order to take advantage of spatio-temporal characteristics.

1.3 Scope

1. Data was obtained from Geo-Informatics and Space Technology Development Agency (GISTDA)
2. Ocean current data from year 2014 to 2016
3. Measurement by Root-mean-square error (RMSE) of predicted U and V components by using year 2014 for training, 2015 for validation, and year 2016 for test datasets

1.4 Expected Result

1. Better model performance, evaluated by RMSE of a predicted value
2. Able to use deep learning technique to take advantage of spatio-temporal characteristics of HF radar data

1.5 Research Plan

1. Study related theories and literature review
2. Study neural network and deep learning
3. Design experiment, develop and collect result
4. Summarize result
5. Thesis proposal topic examination
6. More experiment according to the proposal
7. Writing and publishing a research paper
8. Collect result and conclude thesis
9. Thesis examination

1.6 Publications

“Spatio-Temporal Deep Learning for Ocean Current Prediction Based on HF Radar Data” by Nathachai Thongniran, Peerapon Vateekul, Kulsawasd Jitkajornwanich, Siam Lawawirojwong, and Panu Srestasathiern in “2019 - 16th International Joint Conference on Computer Science and Software Engineering (JCSSE 2019)” conference which is established at Amari Pattaya hotels, Chonburi, Thailand on 10-12 July 2019.

“Combining Attentional CNN and GRU Networks for Ocean Current Prediction Based on HF Radar Observations” by Nathachai Thongniran, Kulsawasd Jitkajornwanich, Siam Lawawirojwong, Panu Srestasathiern, and Peerapon Vateekul in “2019 - 8th International Conference on Computing and Pattern Recognition (ICCP 2019)” conference which is established at Grand Gongda Jianguo Hotel, Beijing, China on 23-25 October 2019.

Chapter 2

Background Knowledge

This section describes the background knowledge in this thesis which mainly composed by two topics which are oceanic knowledge and deep learning techniques.

2.1 Ocean Current Circulation in Thai Gulf

Normally, tides, surface wind, streamflow, density, monsoon, sea breeze, and land breeze are the main factors of ocean surface current circulation.

2.1.1 Monsoon

Gulf of Thailand current circulation is mainly affected by two monsoons: (i) Southwest monsoon usually starts from May to September and (ii) Northeast monsoon generally starts from October to February [8], as shown in Figure 1. There are two periods of transition between two monsoons, the first inter-monsoon starts from March to April and the second inter-monsoon occurs in October.

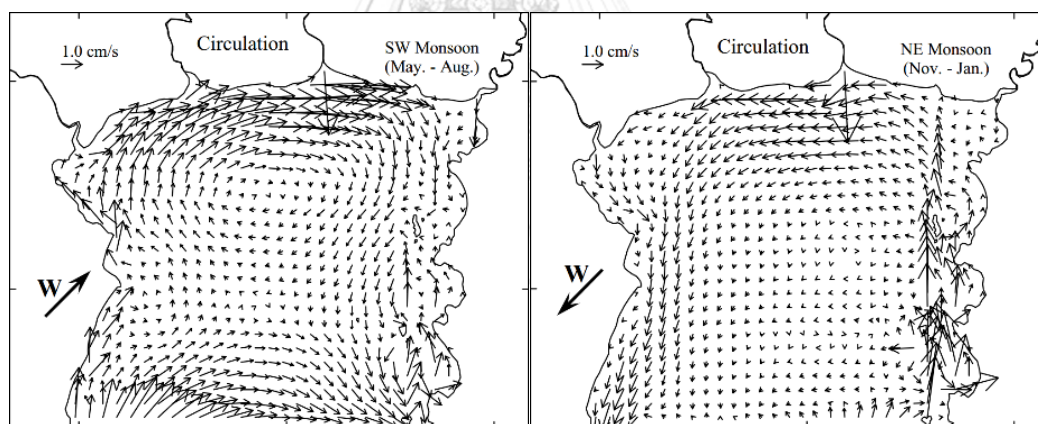


Figure 1 Ocean circulation during the southwest and northeast monsoon

(From: Figure 4 [8])

2.1.2 Lunar Illumination

Tide is mainly affected by moon orbit around the earth. A highest and lowest lunar illumination which are full moon and new moon causes the spring tide to make increasing of sea level. In contrast to the neap tide causes decrement of sea level.

2.1.3 Sea Breeze and Land Breeze

Differentiation of temperature between ocean and coastal land area causes sea breeze and land breeze. In the day, the temperature of land area is higher than ocean area causes air over the land area to float up while the colder wind from ocean area flows to replace the floating one on land area which called “sea breeze” which drives the ocean current toward a coastal area. In contrast to the “land breeze”, wind that drive from land area toward to ocean area, as shown in Figure 2.

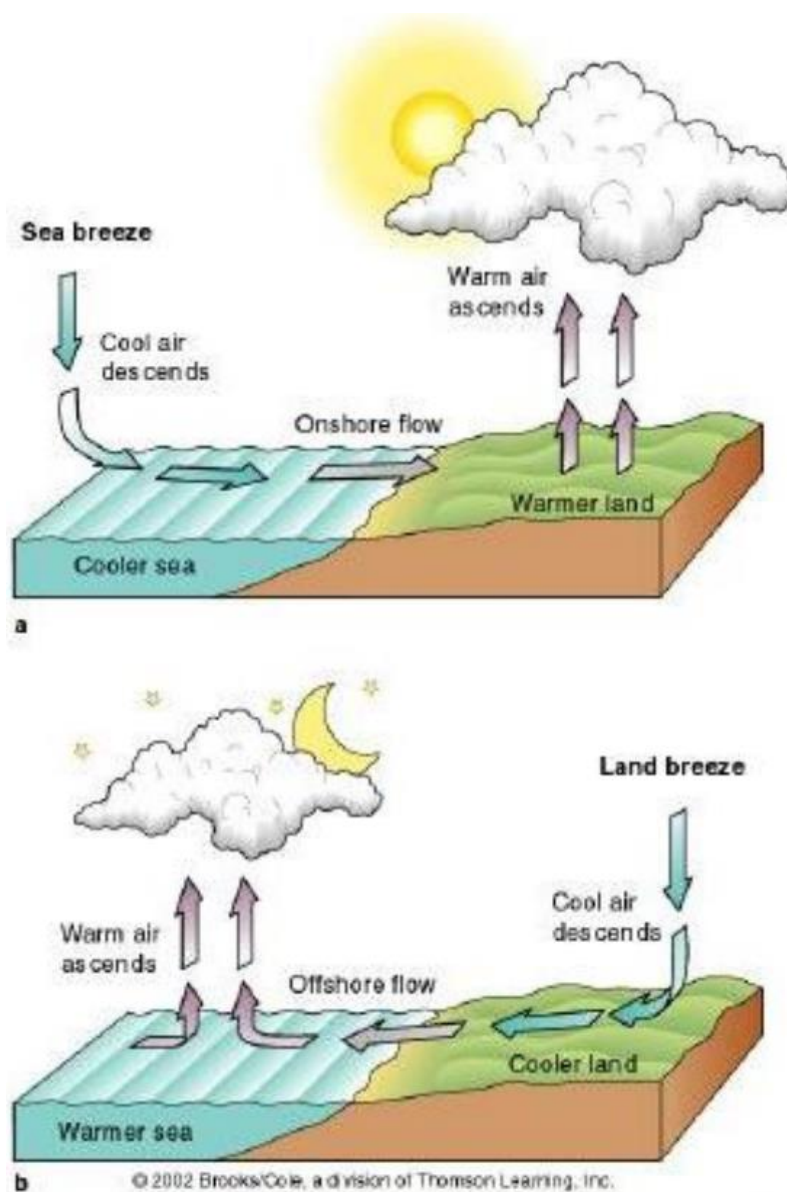


Figure 2 The circulations of land and sea breeze

(From: Figure 14 [9])

2.2 High Frequency Radar and Data Collection

This section specifically describes to our main dataset in this research which is collected by High Frequency (HF) Radar stations.

2.2.1 HF Radar Station in Thailand

Thailand has faced many ocean disasters from an inefficient monitoring and management which is caused from a lack of data for monitoring and understanding of the ocean's behavior. In the first phase, GISTDA deployed 18 HF coastal radar stations located along the gulf of Thailand since 2012 and more 6 stations which are currently in phase two. Figure 3 shows an area HF coastal radar which consists of two project phases, the first phase is appeared in purple color and second phase is displayed in yellow color.

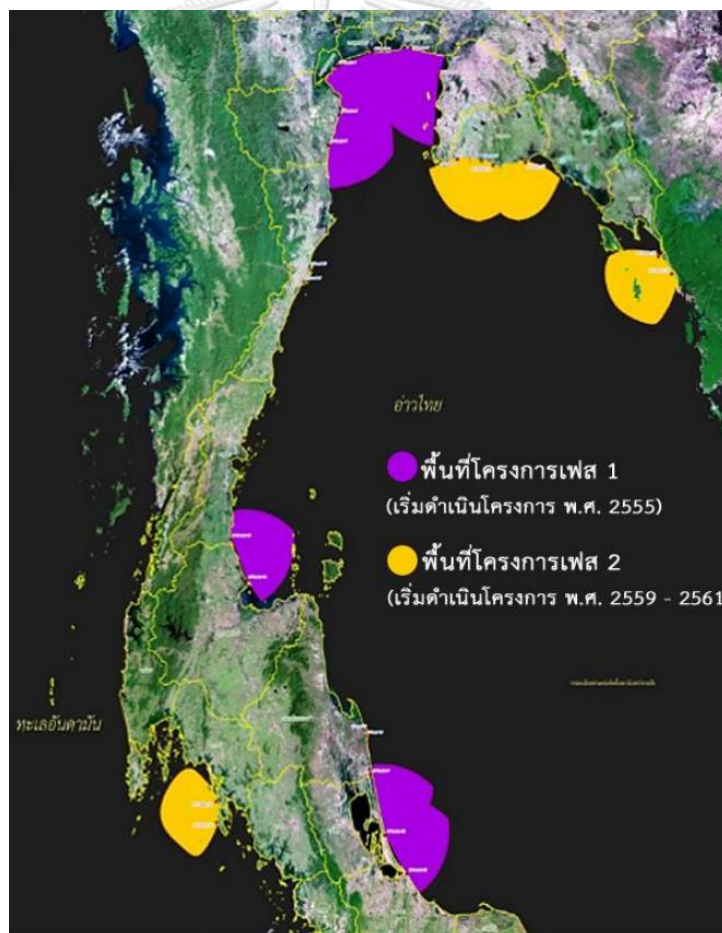


Figure 3 Two implementation phases of HF radar stations

[From:

https://www.gistda.or.th/main/sites/default/files/news/1458/file/session5_kaarekhaathuengrabba_ihbrikaarkhmuulerdaarchaayfang-1458-7574.pdf Accessed: December 1, 2019]

2.2.2 Measurement

Ocean surface current velocity and direction in the coastal area are measured by HF radio waves. Coastal stations initiate a signal and monitor it returning to the station then using backscattered radio waves to calculate surface current as shown in Figure 4. There are 3,097 grid points from GISTDA from 2014 to 2016 that are being used as a dataset in this study.

HF radar station which is operated by GISTDA collects varied of values, such as latitude longitude, magnitude, direction, U component and V component.

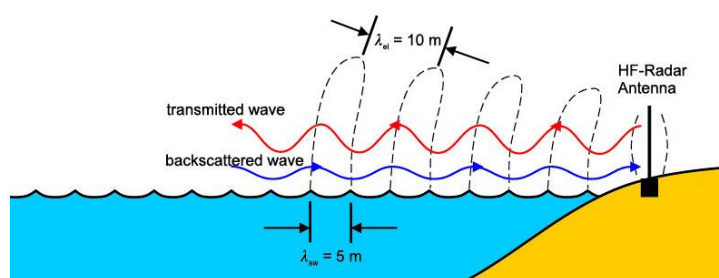


Figure 4 Transmitting and receiving of HF radar

[From: <http://coastalradar.gistda.or.th/wp/> Accessed: December 1, 2019]

2.2.3 Data Loss

The reported data from HF radar stations generally lost by many causes, for example (i) the station equipment and sensor does not work functionally, (ii) a high ocean wave which can block the signal that's sent by the station, (iii) a long distance between target and HF station that signal cannot reach to, (iv) noise or obstacle in coastal area disturb the signal, and (v) the angle of signal base before firing if angle is too high so the station will not able to measure the short distance area near the station.

2.2.4 Current Measurement

Ocean surface current normally represents in two approaches, (i) magnitude and direction which describe the velocity and direction of the current toward to, as shown in Figure 5, (ii) U and V components, two separate flow speeds measured along two orthogonal axes which are u and v components. U component represents the horizontal flow in the east-west direction while v component represents the vertical flow in the north-south direction as shown in Figure 6.

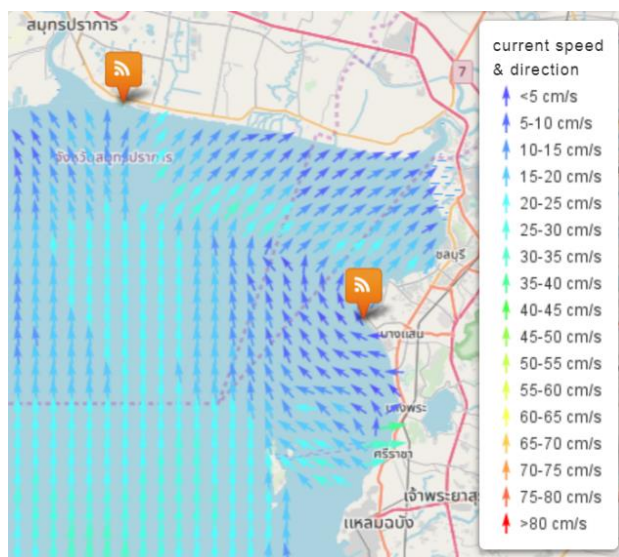


Figure 5 Example of magnitude and direction of reported grid data which is shown on top of the Gulf of Thailand map

[From: <http://coastalradar.gistda.or.th> Accessed: December 1, 2019]

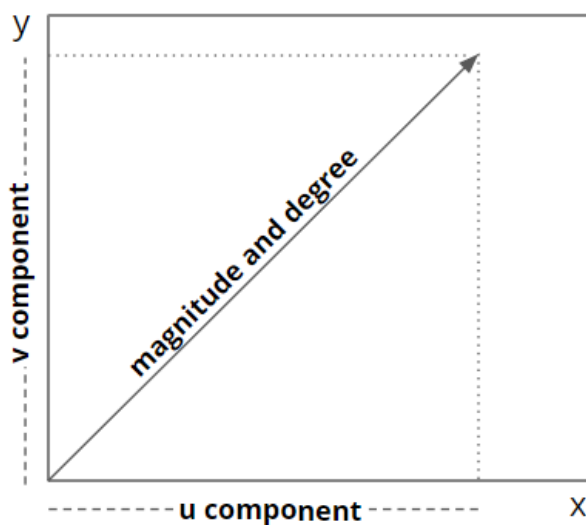


Figure 6 U and V component of ocean current

Two current measurements are used differently depending on the purpose. Magnitude and direction will be more suitable for boat sailor for knowing how fast they go and direction they heading to. On the other hands, finding a new object position when a period passes would be easier if we calculate with U and V component which we can performs an averaging a pair of current measurements obtained at two different times.

2.3 Tide

Tide phenomenon can be considered as seasonal effect of ocean which effects to ocean current. The topic is explained the tide and how it effects to ocean current.

2.3.1 Tidal Force

Tidal forces are based on the gravitational force which depends on distance between two objects than their mass, as shown in equation (1) and Figure 7.

$$F = \frac{2GMm}{r^2} \left(\frac{R}{r} \right) \propto \frac{m}{r^3} \quad (1)$$

Where G is universal gravitational constant, M is mass of the earth, m is mass of space object, r is distance between earth and object and R is a radius of the earth.

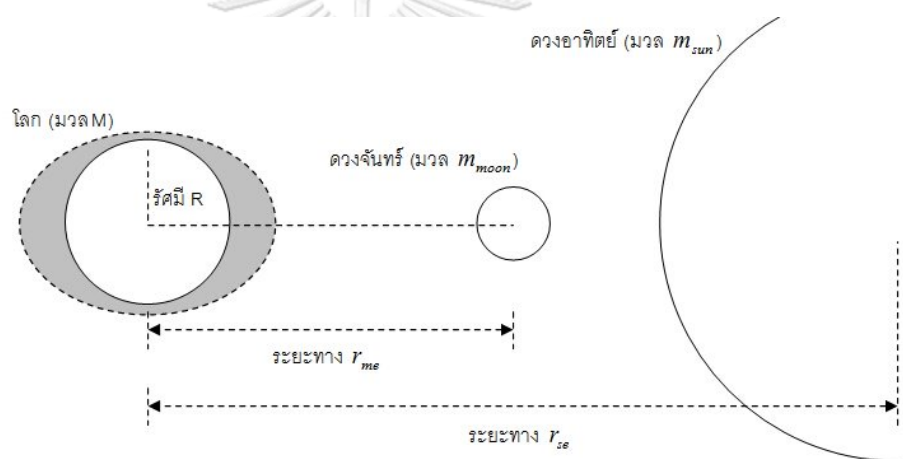


Figure 7 Tidal force between the earth and object in space

[From: <http://www.narit.or.th/index.php/astronomy-article/96-oceantides>

Accessed: December 1, 2019]

Regarding tidal forces to the earth between sun and moon, the sun only has 46% of moon tidal force to the earth due to its distances. Moon, which has a higher of tidal force will affect to tide more than the sun. In addition, effect of tidal force makes circular object shape-changing to ellipse with a same volume. The tidal force of other planets has been shown in Table 1 below.

Table 1 Maximum tidal forces of the sun, moon, and planets on the earth

[From: https://science.nasa.gov/science-news/science-at-nasa/2000/ast04may_1m

Accessed: December 1, 2019]

Solar system object	Tidal force
Moon	2.1
Sun	1
Venus	0.000113
Jupiter	0.0000131
Mars	0.0000023
Mercury	0.0000007
Saturn	0.0000005
Uranus	0.00000001
Neptune	0.000000002
Pluto	0.0000000000001

2.3.2 Spring Tide and Neap Tide

The effect of tidal force from space object to the earth especially sun and moon cause tide phenomenon as shown in Figure 8. In the solar system, any time sun, earth, and moon are exactly or very closely aligned, it generates highest tidal force to the earth which causes spring tide and neap tide phenomenon as shown in Figure 9.

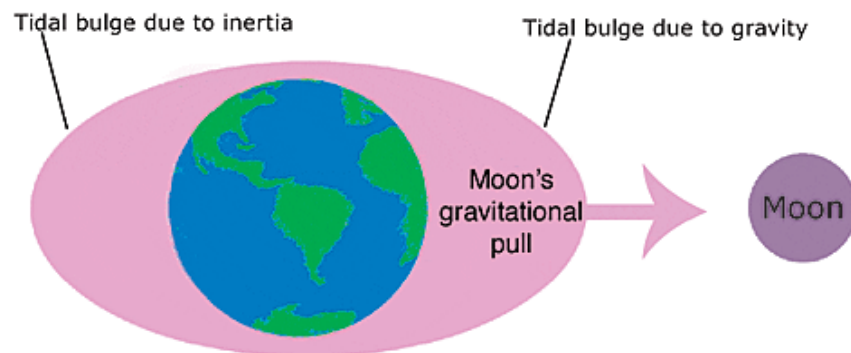


Figure 8 Tidal force of moon to the earth

[From: <http://oceanservice.noaa.gov> Accessed: December 1, 2019]

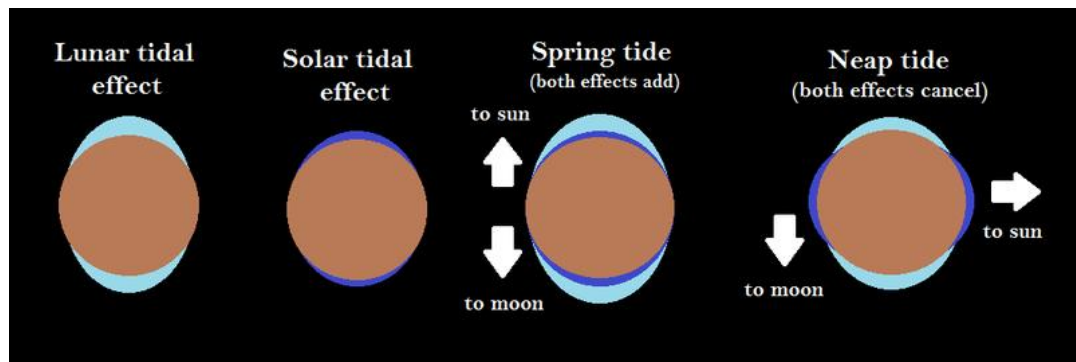


Figure 9 Tidal effects from the sun and moon on the earth

[From: https://energyeducation.ca/encyclopedia/Tidal_force Accessed: December 1, 2019]

Spring tide happens when New Moon and Full Moon occurred which have highest lunar illumination and lowest lunar illumination, respectively as shown in Figure 10 and relation between lunar illumination and tide is shown in Figure 11.

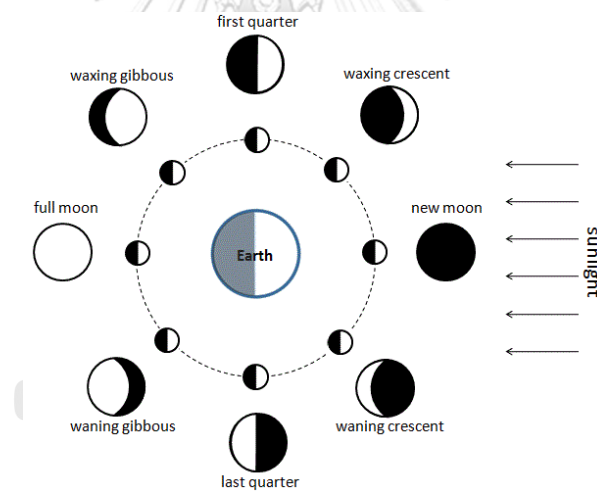


Figure 10 Diagram of the moon's phases

[From: https://simple.wikipedia.org/wiki/Phases_of_the_Moon Accessed: December 1, 2019]

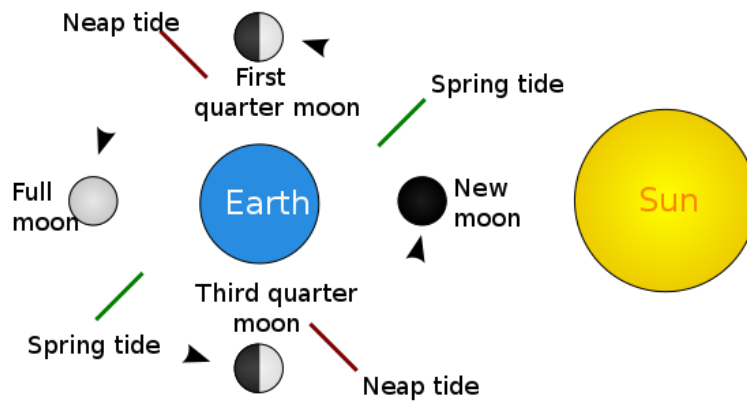


Figure 11 The types of tides

[From: <https://en.wikipedia.org/wiki/Tide> Accessed: December 1, 2019]

2.3.3 Tide with Different Viewpoint

Within one day, number of tide phenomenon depends on latitude and longitude of the locations, type of tide can be categorized in three groups.

1) Semidiurnal Tide (Equatorial Tide)

If we are at the equator line, our observation point is taken passed bulging point (A and A' in Figure 12) two times in a day that the earth is rotating itself, so it lets we see rising tide two times and falling tide two times in a day with the same water level.

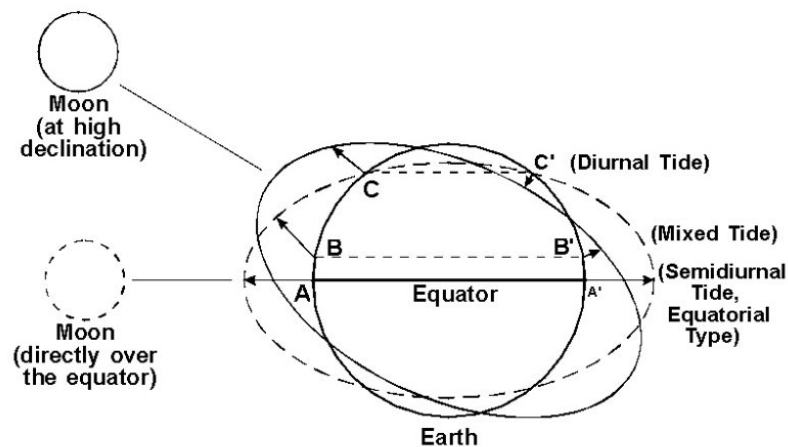


Figure 12 Tide with different viewpoints

[From: <http://www.narit.or.th/index.php/astronomy-article/96-oceantides>

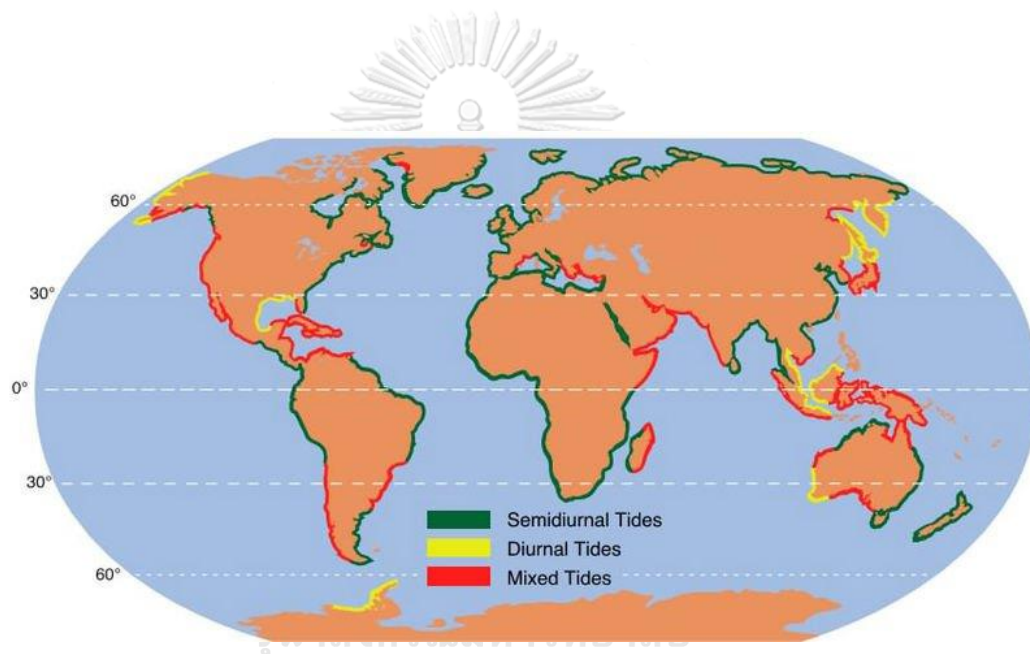
Accessed: December 1, 2019]

2) Mixed Tide

If we are above or lower the equator line less than 28.5 degree, we will saw two times (B and B' in Figure 12) of tide like Semidiurnal Tide but with difference water level.

3) Diurnal Tide

In the position and above or lower the equator line more than or equals 28.5 degree, we will see only one time (C and C' in Figure 12) of tide (one tidal cycle per day). In left side Gulf of Thailand area (dataset we used for this research), we will see diurnal tide as shown in Figure 13.



CHULALONGKORN UNIVERSITY
Figure 13 Tide types map

[From: <https://en.wikipedia.org/wiki/Tide> Accessed: December 1, 2019]

2.4 Neural Network (NN)

The model is inspired by biological nervous systems as shown in Figure 14. 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) [10], (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 is a very important part while tuning hyperparameter.

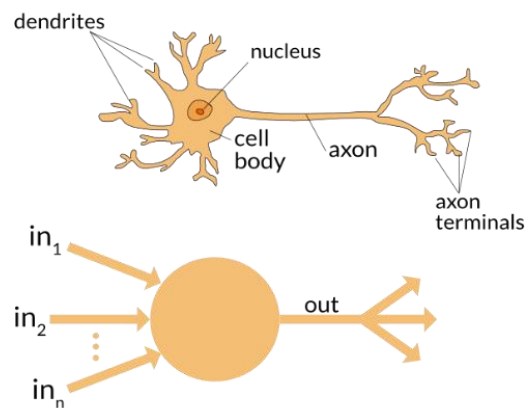


Figure 14 Perceptron architecture

[From: <https://appliedgo.net/perceptron/> Accessed: December 1, 2019]

2.4.1 Perceptron

Perceptron, a single hidden layer neural network which contains only one hidden layer between the input and output layer. A perceptron receives \mathbf{n} number of inputs since $\mathbf{x}_1, \dots, \mathbf{x}_n$ then multiply with its weight which are $\mathbf{w}_1, \dots, \mathbf{w}_n$ after that the result will be added by \mathbf{w}_0 and will be passed through step function as shown in (2).

$$f(x) = 1 \text{ if } (w_0 + \sum_{i=1}^n w_i x_i) > 0, -1 \text{ otherwise} \quad (2)$$

Where \mathbf{w} is weights, \mathbf{w}_0 is bias and \mathbf{x} is an input and \mathbf{n} is number of inputs. In the learning process, can be described by two equations below.

$$w_i \leftarrow w_i + \Delta w \quad (3)$$

$$\Delta w = \alpha (\hat{y} - y) x_i \quad (4)$$

Where α is learning rate, which is used to define how much weight will be updated on each training round.

2.4.2 Activation Function

The function is used to decide whether a neuron should be activated or not by nonlinear transformation an output before sending to a next layer. Popular activation functions are as follows.

1) Sigmoid Function

The function refers to the special case of the logistic function. Output of all real numbers that is passed sigmoid function will be bounded from 0 to 1, as shown in (5).

$$\sigma(x) = \frac{1}{1 + e^{-x}} \quad (5)$$

2) Rectified Linear Unit (ReLU)

In the last few years, ReLU has become very popular. It simply thresholded at zero, as shown in (6) that make it less computational cost [10].

$$f(x) = \max(0, x) \quad (6)$$

3) Softmax Function

The function returns value between 0 to 1 and sum of all nodes will be equals 1 which can be compared to possibility of each nodes. Defines x as number of nodes in dense layer, node result in layer is replaced with z , so the result of softmax function of j or f_j can be shown in the equation (7).

$$f(z)_j = \frac{e^{z_j}}{\sum_{i=1}^k e^{z_i}} \quad (7)$$

4) Hyperbolic Tangent Function

The function returns a value in range -1 to 1. The function can be called in **tanh** which is can be calculated from the equations (8).

$$\tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}} \quad (8)$$

5) Threshold Function

The function determines whether a value equality of its inputs exceeded a t threshold. The function can be written with the equation below (9).

$$f(z, t) = \begin{cases} 0 & \text{if } z < t \\ z & \text{if } z \geq t \end{cases} \quad (9)$$

2.4.3 Loss Function

The loss function 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 is a very important part while tuning hyperparameter.

2.5 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 Figure 15.

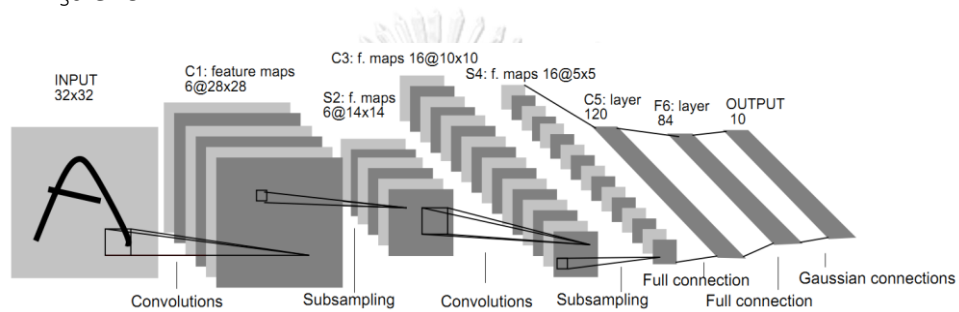


Figure 15 Convolutional neural network architecture

(From: Fig. 2. [11])

2.5.1 Convolution Layer

The layer is used to find a feature from a group of nearby input by using a dot product of matrix with the filter as shown in Figure 16. In this layer, there are many components that need to be considered, such that filter size, stride size, number of filters and number of channels. Where \mathbf{I} is input matrix, \mathbf{K} is filter matrix with $\mathbf{h} \times \mathbf{w}$ size, result of convolution is shown in equation (10). There are components that must be considered as list below.

$$(\mathbf{I} * \mathbf{K})_{xy} = \sum_{i=1}^h \sum_{j=1}^w \mathbf{K}_{ij} \cdot \mathbf{I}_{x+i-1, y+j-1} \quad (10)$$

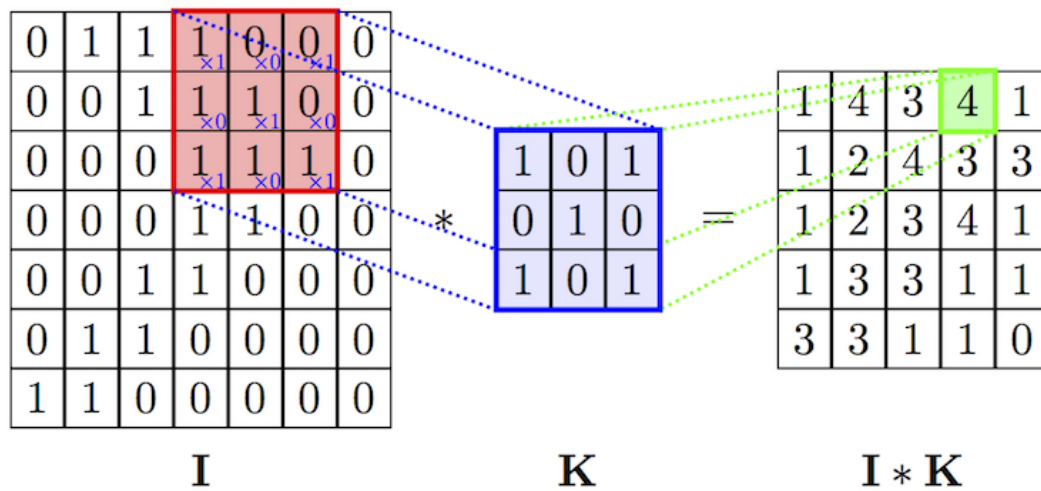


Figure 16 Convolution operation

[From: <http://www.deeplearningessentials.science/convolutionalNetwork/>

Accessed: December 1, 2019]

1) Filter Size

Filter size is weight and height of filter that used in the convolution layer (\mathbf{h} and \mathbf{w} of equation (10) are equal 3 and 3 is used in Figure 16).

2) Convolution Type

Narrow convolution, convolution type is being used in general. It does not operate out of input matrix size. Result of convolution of input matrix $\mathbf{N} \times \mathbf{N}$ with filter size $\mathbf{M} \times \mathbf{M}$ will result matrix size $(\mathbf{N} - \mathbf{M} + 1) \times (\mathbf{N} - \mathbf{M} + 1)$, an example of it is shown in Figure 16.

Wide convolution, this convolution type operates out of input matrix size (out of input matrix size was considered as 0), which is called padding for preventing information losing at the matrix border. Result of convolution of input matrix $\mathbf{N} \times \mathbf{N}$ with filter size $\mathbf{M} \times \mathbf{M}$ will result matrix size $(\mathbf{N} + \mathbf{M} + 1) \times (\mathbf{N} + \mathbf{M} + 1)$, an example of it is shown in Figure 17.

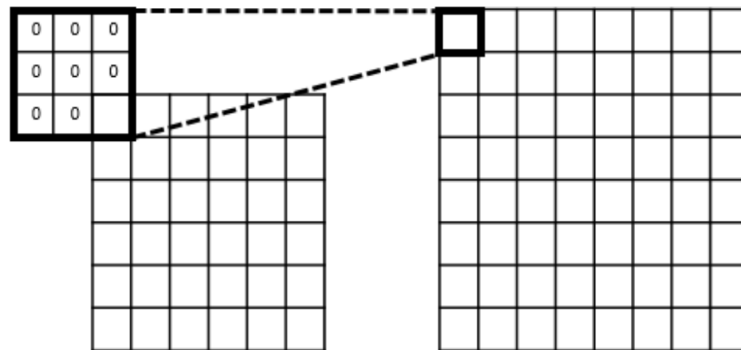


Figure 17 Wide convolution with padding

(From: Fig. 2.6 [12])

3) Stride Size

It is a number of input slots that will be used for calculating for each result slot, which have default value equals 1. An example of convolution layer which has stride size equals 2 is shown in Figure 18.

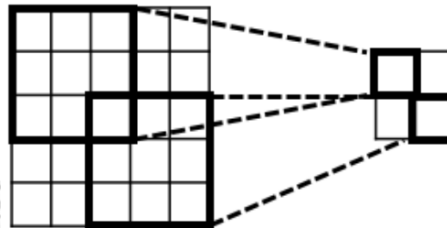


Figure 18 Convolution layer with input size equals 5x5,
filter size equals 3x3, and stride size equals 2

(From: Fig. 2.7 [12])

4) Number of Filters

It is able to have many filters, each filter can have different weight. The number of filters will result to number of inputs in the next layer, as shown in Figure 19.

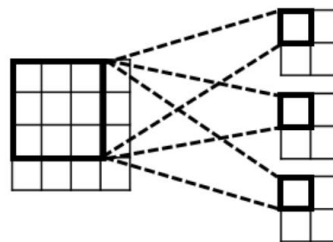


Figure 19 Convolution with number of filters equals 3

(From: Fig. 2.8 [12])

4) Number of Channels

Number of channels (input's depth) is normally equals 3 in image research / task, which represent primary colors or number of input's depth in previous layers. The result of convolution layer with number of channels is shown in equation (11), where k is number of channels.

$$z_{ij}^l = \sum_{c=0}^{k-1} \sum_{a=0}^{m-1} \sum_{b=0}^{m-1} w_{a,b}^l a_{c,i+a,j+b}^{l-1} + b^l \quad (11)$$

2.5.2 Pooling Layer

The layer's objective is to reduce computational cost by only keeping important information. Popular pooling layers are max and average pooling layer. An example of max pooling is shown in Figure 20.

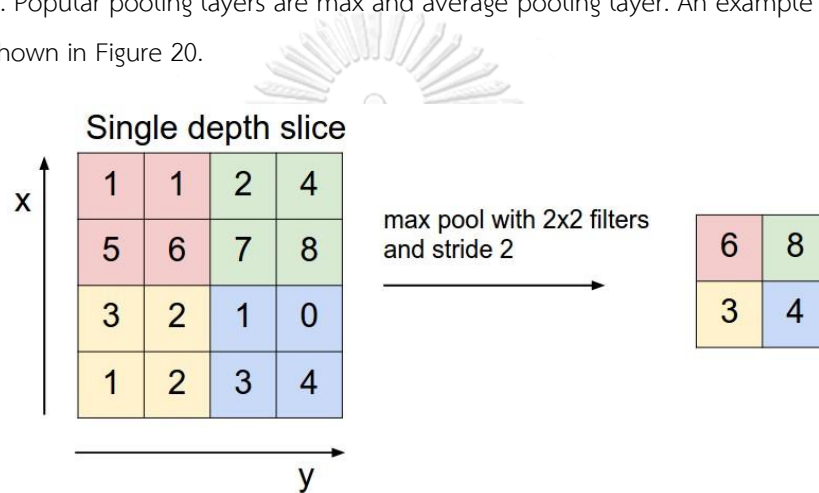


Figure 20 Example of max pooling operation

[From: <http://cs231n.github.io/convolutional-networks> Accessed: December 1, 2019]

2.6 Recurrent Neural Network (RNN)

Neural Network that is designed for sequential data by using internal memory. Internal memory makes RNN able to remember things which have been passed as shown in Figure 21 and equation (12), where each variable is listed below.

- x_t is input at timestep t
- s_t is hidden state at timestep t , also called network memory
- U is weight of input at timestep t
- W is shared weight
- b is a bias
- σ is activation function

$$s_t = \sigma(Ws_{t-1} + Ux_t + b) \quad (12)$$

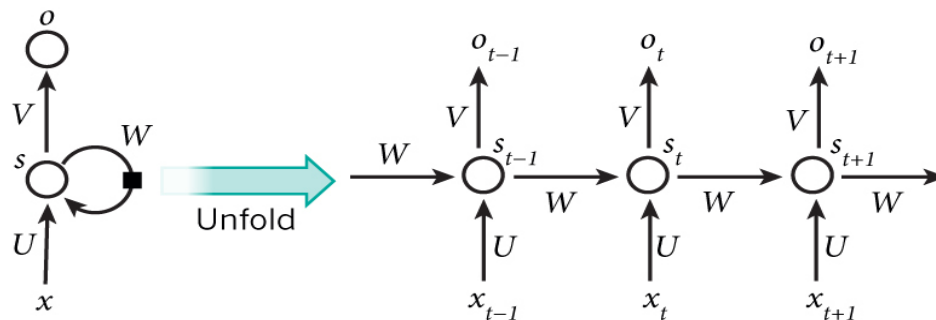


Figure 21 A recurrent neural network

[From: <http://www.wildml.com/2015/09/recurrent-neural-networks-tutorial-part-1-introduction-to-rnns/> Accessed: December 1, 2019]

In the training phase of model training, we will use “Back Propagation Through Time” (BPTT) for update neural weight. This weight-updating approach (BPTT) can cause issues called gradient explosion and vanishing when input length is too long. In BPTT, we use chain rule for weight updating, weight which is calculated from gradient can be exploded (when weight value more than 1) or vanishing (when weight value range between 0 and 1) when we have to multiply it many times according to the input length.

2.7 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 if information to be considered or not. The reset gate is used to decide how much information to forget. The GRU is shown in Figure 22 and the calculation of each component is shown in equation (13-16) where each variable is listed below while **tanh** and **sigmoid** are used for bound the result for preventing exploding and vanishing gradient.

- i is neuron index
- j is time index
- \hat{h} is result of previous input
- h is result of current input
- x_t is input value at timestep t
- z_t is value of update gate at timestep t
- r_t is value of reset gate at timestep t
- U_z is weight of update gate
- U_r is weight of update gate
- W is shared weight

$$h_j^i = (1 - z_t^j)h_{t-1}^j + z_t^j\hat{h}_t^j \quad (13)$$

$$\hat{h}_t^j = \tanh^j(Wx_t + U(r_t \odot h_{t-1})) \quad (14)$$

$$z_t^j = \text{sigmoid}^j(W_z x_t + U_z h_{t-1}) \quad (15)$$

$$r_t^j = \text{sigmoid}^j(W_r x_t + U_r h_{t-1}) \quad (16)$$

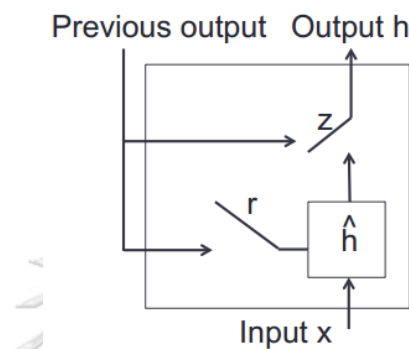


Figure 22 Gated recurrent unit

[From: https://github.com/ekapolc/nlp_course Accessed: December 1, 2019]

2.8 Attention Mechanism

A mechanism that used for focusing on a subset of its inputs, it tells which input to looking at (or pay attention to) [14]. The system that applies the attention mechanism will need to determine where to focus on. There are several ways of this mechanism implementation for example the implementation with CNN as shown in Figure 23.

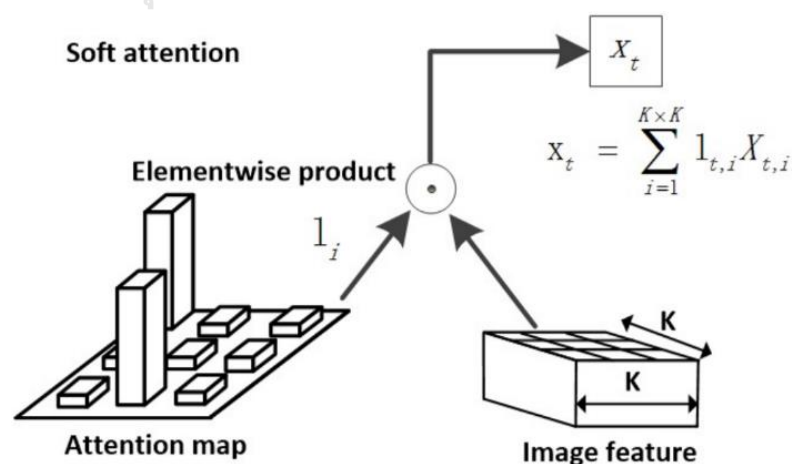


Figure 23 The attention mechanism: The soft attention assign weights on different locations of features using softmax

(From: Figure 2 [15])

The mechanism generally used in sequential inputs. An example of attention mechanism in machine translation is shown in equation (17-18) and the variable are list below.

$$c_i = \sum_j a_{ij} h_j \quad (17)$$

$$a_i = \text{softmax}(f_{\text{att}}(s_{i-1}, h_j)) \quad (18)$$

- c_i is context vector
- a_{ij} is attention score
- h_j is encoder state at index j
- a_i is an attention scores (weight vector) of all decoder index j
- s_{i-1} is previous hidden state / decoder state
- h_j is encoder state at index j
- f_{att} is function for calculating attention score which have many types

2.8.1 Additive Attention

The attention function which calculate the score with concattion of 2 sequences together as shown in equation (19).

$$f_{\text{att}}(s_{i-1}, h_j) = \tanh(W_1 s_{i-1} + W_2 h_j) \quad (19)$$

2.8.2 Multiplicative Attention

This attention calculates attention score by multiplication with weight, which is faster and more efficient than additive attention. The equation of this attention's type is shown in (20).

$$f_{\text{att}}(s_{i-1}, h_j) = s_{i-1}^T W_a h_j \quad (20)$$

2.8.3 Self-Attention

Aforementioned attention types are computed by using 2 sequential input, but this type only require itself (only one input required), as shown in equation (21-22) where W_{s1} is a weight matrix, W_{s2} is a vector of parameters.

$$H = (h_1, h_2, h_3, \dots, h_n) \quad (21)$$

$$a_i = \text{softmax}(W_{s2} \tanh(W_{s1}, H^T)) \quad (22)$$

2.9 Transfer Learning

Transfer learning concept has been widely used in the machine learning field. This concept can be implemented in several ways for example, using existing model and weight as an initial model's state then continuously fining tune for the new task, and using existing model as a front layer of our model. The idea utilize knowledge from other pre-trained models or tasks to another task which mostly related to each other (leverage knowledge from what we have learned in the past).

For example, in a cat classification task instead of training from scratch, we using trained model that was trained on ImageNet dataset (which already have good understanding in general shape or pattern of things in many categories), as a feature extraction layer of the new model (such as car classification) will usually help the our task to have a better initial state and reduce training time.

2.10 One-Hot Encoding

One-hot encoding [16] is a technique that used for representing a value. Length of result vector will be equals number of items, value inside the vector will have only one position that have 1 as a value, otherwise 0. Example of month number encoding which have range from 0 (January) to 11 (December), is shown in Table 2.

Table 2 Month number in one-hot encoding

Month number	Month number (one-hot encoding)
0	[1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
1	[0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
2	[0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0]
3	[0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0]
4	[0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0]
5	[0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0]
6	[0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0]
7	[0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0]
8	[0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0]
9	[0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0]
10	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0]
11	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1]

Chapter 3

Related Works

This section describes existing researches in this field which is composed by three topics: (i) model approach, (ii) research in the same dataset, and (iii) room for improvement.

3.1 Model Approach

Existing model approaches will be summarized into three groups which are numerical based approach, time series based approach and machine learning based approaches.

3.1.1 Numerical Based Approach

Frolov et al. [17] 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. [18] 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. This approach is not included in the experiment.

3.1.2 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. The approach will be considered as one of the baseline models.

3.1.3 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. [19] 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 incorporate both spatial and temporal effects.

3.2 Ocean Current Prediction in Gulf of Thailand

Latest research on the same dataset [3], adapt machine learning technique k-nearest neighbors (kNN) with seasonal attributes. Dataset which was provided by GISTDA was filtered out by selecting only grid areas that have missing data less than 5% over three years of data since 2014 to 2016 and normalizing with min-max normalization before using it as model inputs.

The research proposed 2 feature extractions as shown below.

- 1) Point-Hour (PH) look back sequence: Each grid point data was transformed to equation (23), where $y_{hr,d}$ is target value at hour hr of day d

$$(month, x_{hr,d-1}, x_{hr,d-2}, \dots, x_{hr,d-n}, y_{hr,d}) \tag{23}$$

- 2) Time Series (TS) look back sequence: Each grid-point data was transformed to equation (24)

$$\begin{aligned}
 & month [x_{hr-m,d-n}, x_{hr-2,d-n}, \dots, x_{hr-1,d-n}, x_{hr,d-n}] \\
 & \dots \\
 & [x_{hr-m,d-2}, x_{hr-2,d-2}, \dots, x_{hr-1,d-2}, x_{hr,d-2}] \\
 & [x_{hr-m,d-1}, x_{hr-2,d-1}, \dots, x_{hr-1,d-1}, x_{hr,d-1}] y_{hr,d}
 \end{aligned} \tag{24}$$

They separate each model for each month due to seasonality effect concerned and the dataset was splitted into two parts. Example of data splitting is shown in Figure 24.

- Year 2014: training dataset
- Year 2015 and 2016: in each month, first 10 days will be training dataset, then randomly selected 10 days for testing dataset and the rest will be additional training dataset (additional 8 to 11 days for training dataset)

At the end, the research result shows performance of their proposed model better than all baseline models.

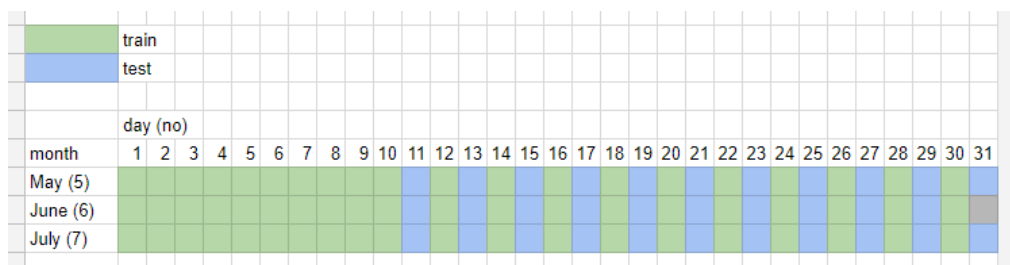


Figure 24 Example of data splitting that is used in Temporal kNN model research

3.3 Room for improvement

From the aforementioned researches in ocean surface current forecasting, there are still many of rooms for the model to be improved. In this research, we will address three components below.

3.3.1 Addressing Spatial and Temporal Effect Together

Previous research either address spatial or temporal effect, they did not cooperate with spatial and temporal effects together.

3.3.2 Domain Knowledge Input

Many researchers already applied some domain knowledge such as, seasonality effect by separately creating model for each month but there are still a lot of domain knowledge that not applied yet such as, land breeze and sea breeze.

3.3.3 Deep Learning Technique

Recently deep learning techniques have been showed successfully applied in many fields. So, it is a great chance to bring the techniques into the field.

Chapter 4

Proposed Method

In this chapter, we will propose model that is used to forecast ocean surfer current. The proposed model can be separated into 5 modules which consist of CNN, GRU, oceanic inputs, attention mechanism, and transfer learning. The proposed model takes advantage of noticeable domain properties by using CNN to capture spatial effect and combining GRU to extract temporal effect of its nature. It has two CNN blocks as feature extraction layers in front of the model, then the passed through GRU layer which is connected to it. There are three oceanic input that we add as additional inputs into the model which are month number, lunar effect, and hour number. At last, we extended the model with two modern deep learning techniques, attention mechanism to tell us which part that the model should focus and transfer learning to give the model more knowledge from another task. The chapter will be spitted into two parts: data preprocessing and the proposed model.

4.1 High Frequency Dataset

The high frequency dataset is used for our main dataset. The section covered the dataset and data pre-processing process as follows, data removal, missing grid imputation, and data normalization.

4.1.1 Dataset

U and V components of ocean surface currents in this research are measured from high-frequency (HF) radar stations which are located alongside Thai gulf. Total of 3,097 grid points are reported since 1 Jan 2014 to 23 Dec 2016, as shown in Figure 25 with UTC±00 offset. The dataset resolution is 2x2 kilometer in spatial and one hour in temporal resolutions. An example of records in the dataset is shown in Table 3.

Table 3 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
2015/6/22 6:00	12.6876	99.9726	-9.9550	-17.7540
2015/6/22 6:00	12.6695	99.9726	-5.8360	-18.2980
2015/6/22 6:00	12.6514	99.9727	6.2560	-18.2790

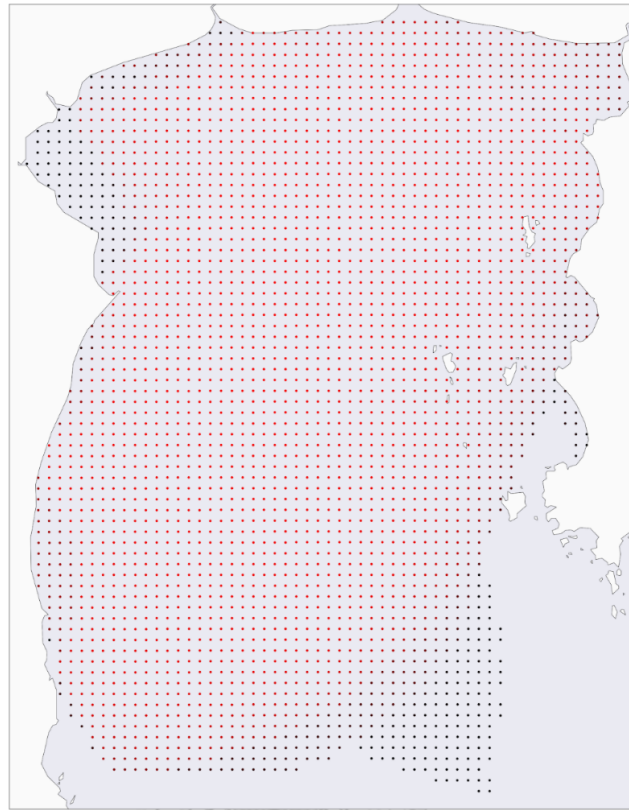


Figure 25 3097 sites of data was obtained from GISTDA

Statistical data of U and V component data was used for data normalization process. Max and min value of each year has almost the same value which is about 100 for max and -100 for min values, as shown in Table 4.

Table 4 Statistical data of each year on U and V components from HF dataset

Year	Statistical Data	U component	V component
2014	Max	99.8110	99.9460
	Mean	1.6161	-2.1626
	Min	-99.9920	-99.8980
	Standard Deviation	9.8523	20.1035
2015	Max	99.9750	99.7160
	Mean	0.9495	-1.8671
	Min	-99.9560	-99.9110
	Standard Deviation	11.7686	21.4724
2016	Max	99.7820	99.9710
	Mean	1.1678	-1.7078
	Min	-99.9650	-99.9210
	Standard Deviation	11.8899	22.8146

4.1.2 Data Removal

We only keep grid points that have data appeared more than or equal ninety-five (same as previous research in the same dataset [3]) percent over three year reported due to the significant issue of data availability from HF radar stations that would affect model prediction result [20]. After the process, 1,065 of 3,097 grid points will be used in the experiment. The histogram of data availability of all grid points over three year is shown in Figure 26 and the total 3,097 grid points are shown in black dot and remaining 1,065 grid points are shown red dot in Figure 27.

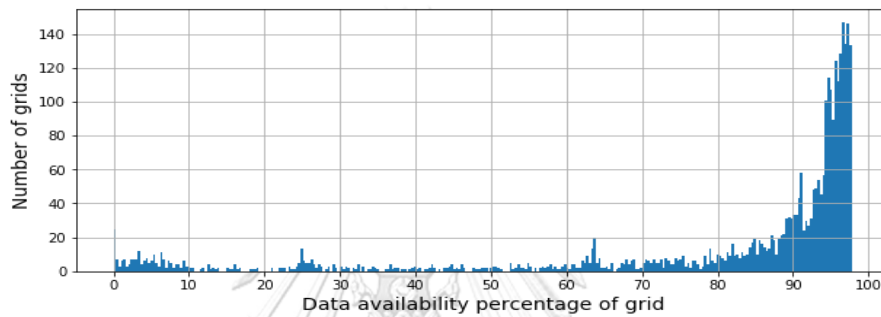


Figure 26 Histogram of grid data availability over three years

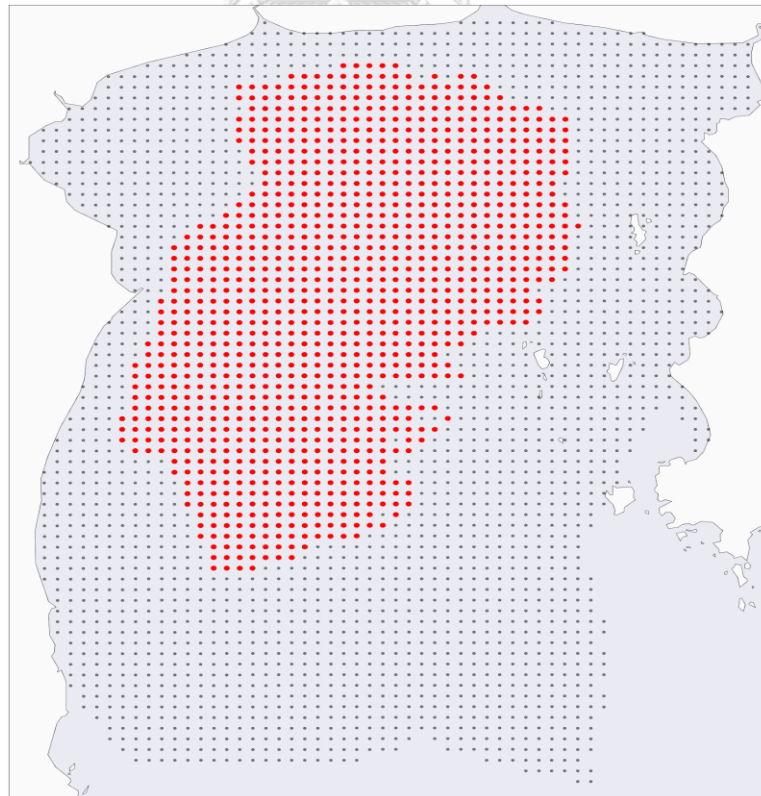


Figure 27 1,065 of 3,097 grid points are shown in red color

4.1.3 Missing Grid Imputation

After the data removal step, we still have a few missing data on each grid point which need to be filled. We proceed two basic steps of imputation to achieve it, as shown in Figure 28 below.

Timestep	1	2	3	4	...	17	18	19	20	21	22	23	24				
Original		21	18				20	26	23		19	20					
Step 1	21	21	18				20	26	23		19	20	20				
Step 2	21	21	18	19	19	19	20	26	23		19	20	20				$(20+18) / 2 = 19$
		21	21	18			20	26	23	21	19	20	20				$(23+19) / 2 = 21$

Figure 28 Two steps of data imputation

1) Imputation at Boundary Grids

We filled missing left and right boundaries with the value of the nearest data point, as shown in Step 1 of Figure 28.

2) Imputation at Non-boundary Grids

We filled the rest with an average of closest available data points, as shown in Step 2 of Figure 28.

4.1.4 Data Normalization

Min-max normalization as shown in (25) is used to normalize the data. The minimum value was mapped to 0 and the maximum value was mapped to 1 and dataset size for each step is shown Table 5.

$$y_i = \frac{x_i - \min(x)}{\max(x) - \min(x)} \quad (25)$$

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

Table 5 Dataset size on each processing step

Step	Processing	Size (GB)
1	Raw (unzip)	11.69
2	Step 1 + Removing metadata	7.76
3	Step 2 + Keep only considered data	4.54
4	Step 3 + Min-max normalization	2

4.2 Oceanic Input Dataset

We use additional oceanic input to help a model to learn more about ocean current background. New inputs that have been added consist of month number, lunar effect, and hour number.

4.2.1 Month Number

The month number is used to represent seasonal effect of two monsoons (southwest monsoon and northeast monsoon) which generally start on certain months consistently. The data is picked directly from the month of each reported timestep then using one hot encoding to encode the data before being used by the model as additional input, as shown in Table 6.

Table 6 Example of month number input

Timestamp	Month number	Month number (one-hot encoding)
2015/12/31 11:00	12	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1]
2015/12/31 12:00	12	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1]
2015/12/31 13:00	12	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1]
2015/12/31 14:00	12	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1]
2015/12/31 15:00	12	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1]
2015/12/31 16:00	12	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1]
2015/12/31 17:00	12	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1]

4.2.2 Lunar Effect

The lunar effect is used to represent tide effect which helps the model about how much to the current will be changed in the day according to the spring tide and neap tide phenomenon. The lunar illumination is calculated from equation in chapter 48 [21] then passed to equation (26) to reflect the effect of spring tide and neap tide. Example of lunar illumination and effect are shown in Table 7 Figure 29 and 30.

$$y_i = |x_i - 0.5| \times 2 \quad (26)$$

Where x_i is a lunar illumination which ranges from zero to one, y_i is lunar effect value with the range of value.

Table 7 Example of lunar effect input

Timestamp	Lunar effect
2015/12/31 11:00	0.32591
2015/12/31 12:00	0.31843
2015/12/31 13:00	0.31094
2015/12/31 14:00	0.30344
2015/12/31 15:00	0.29591
2015/12/31 16:00	0.28837
2015/12/31 17:00	0.28082

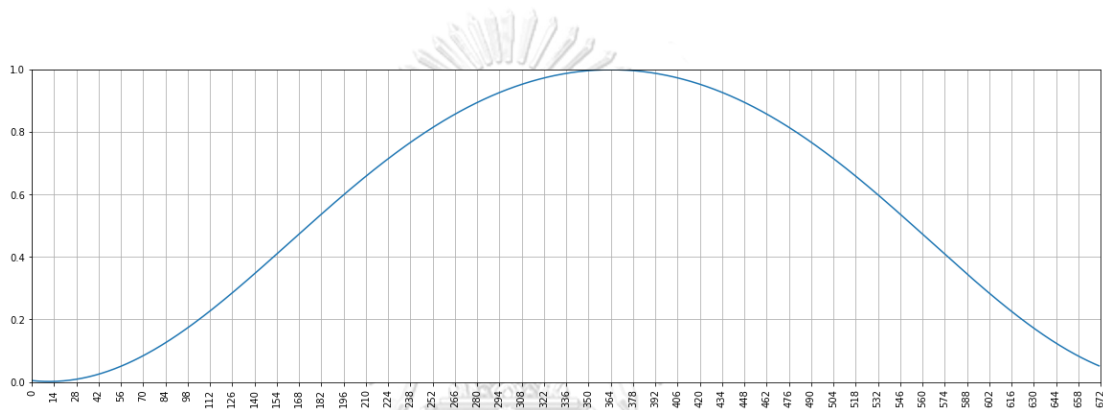


Figure 29 Example of lunar illumination on each timestep

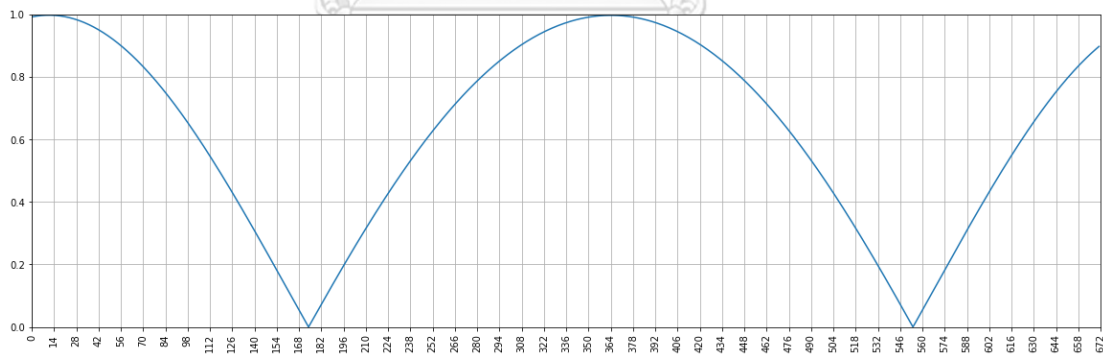


Figure 30 Example of lunar effect on each timestep

4.2.3 Hour Number

The hour number is a representative of land breeze and sea breeze effects which is directly affected by the time in the day. The data is picked directly from the hour of each reported timestep then using one hot encoding to encode the data before being used by the model as additional input, as shown in Table 8.

Table 8 Example of hour number input

Timestamp	Hour number	Hour number (one-hot encoding)
2015/12/31 11:00	11	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
2015/12/31 12:00	12	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
2015/12/31 13:00	13	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
2015/12/31 14:00	14	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0]
2015/12/31 15:00	15	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0]
2015/12/31 16:00	16	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0]
2015/12/31 17:00	17	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0]

4.3 Proposed Model

The model is combined with a variety of sub modules that will be explained in this topic. Each step of adding the module will be created as a new model to shows the effect of added module, as shown in Table 9.

Table 9 Model explanation

Model	Description
CNN-GRU	Spatio-Temporal model that aims to capture two main characteristics which are spatial and temporal characteristics by using CNN and GRU respectively.
CNN-GRU-Input (Month Number)	Enhanced CNN-GRU model with month number as a new input.
CNN-GRU-Input (Lunar Effect)	Enhanced CNN-GRU model with the lunar effect as a new input.
CNN-GRU-Input (Hour Number)	Enhanced CNN-GRU model with an hour number as a new input.
CNN-GRU-Input (All)	Enhanced CNN-GRU model with month number, the lunar effect and an hour number as new inputs.
CNN-GRU-Input (All)-Att	CNN-GRU-Input (All) with the utilization of soft attention mechanism.
CNN-GRU-Input (All)-Att-TL	The proposed model combines all techniques which are adding all oceanic inputs, soft attention mechanism, and transfer learning.

4.3.1 Combination of CNN and GRU

A Spatio-Temporal model takes advantage of obvious domain characteristics by using Convolutional Neural Network (CNN) to capture spatial characteristic and Gated Recurrent Unit (GRU) to capture temporal characteristic. The mode architecture is shown in Figure 31.

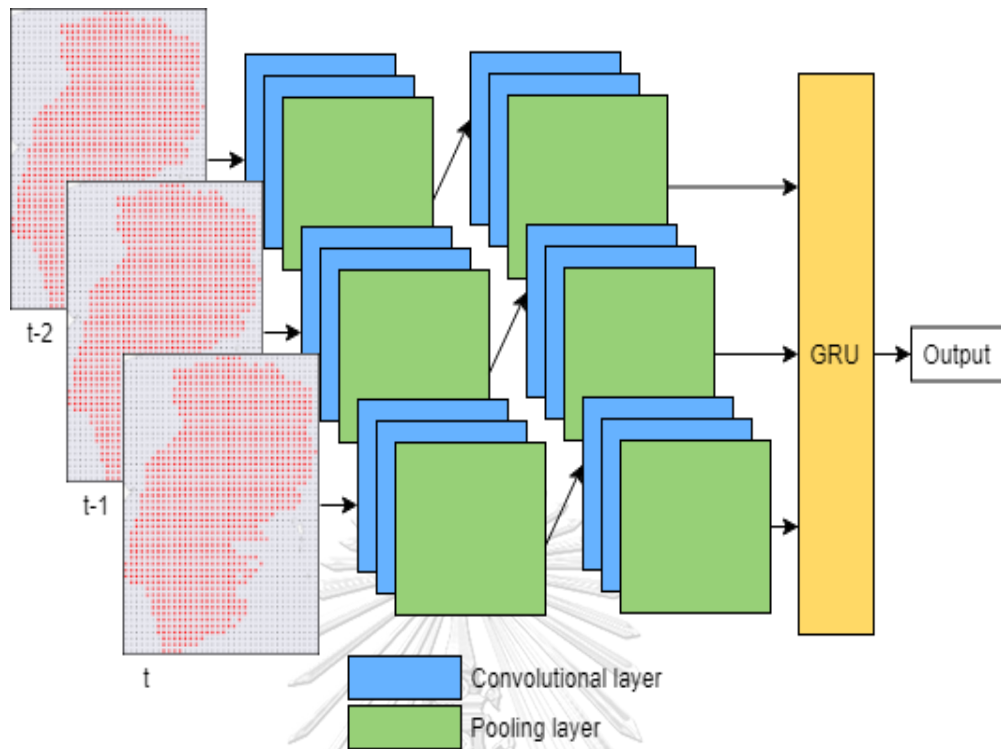


Figure 31 Example of CNN-GRU architecture

4.3.2 Oceanic Input

Oceanic inputs are added and connected to two fully connected layers except lunar effect input, the implementation as shown in Figure 32.

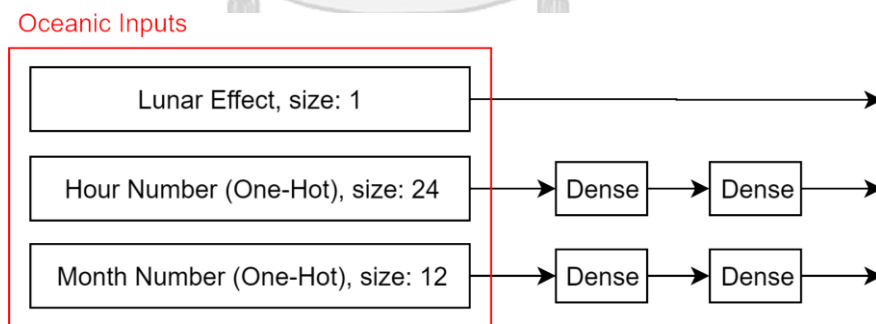


Figure 32 Implementation of oceanic inputs in the model

4.3.3 Attention Mechanism

The implementation idea is even we have two convolutional blocks in the front of the model for feature extraction layer, but there are still many parts that the model must take care of, so we use attention mechanism for telling which part that a model should pay attention to. The attention mechanism was added after the first convolution block.

The feature map values from the first CNN block was sent to tanh for normalization, dense was used a weight function, and softmax was used as an output layer to give the possibility of each feature map then element wise multiplication it together, as shown in Figure 33.

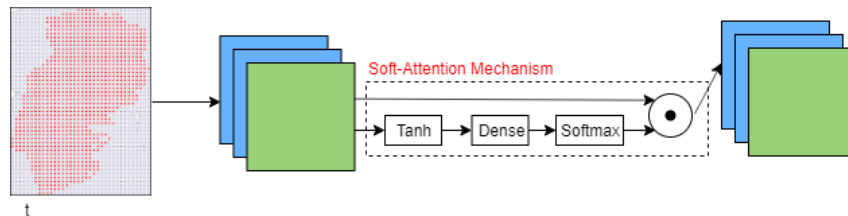


Figure 33 Attention mechanism part in the model

4.3.4 Transfer Learning

We have deployed concept into the model by using another model’s component as an initial weight of the model. For instance, pre-trained weight of U component model is used as an initial weight of V component model, in the same way pre-trained weight of V component model is used as an initial weight of V component model in training process.

The proposed model combines all described topics above which are combination of CNN-GRU, oceanic inputs, attention mechanism, and transfer learning, as shown in Figure 34.

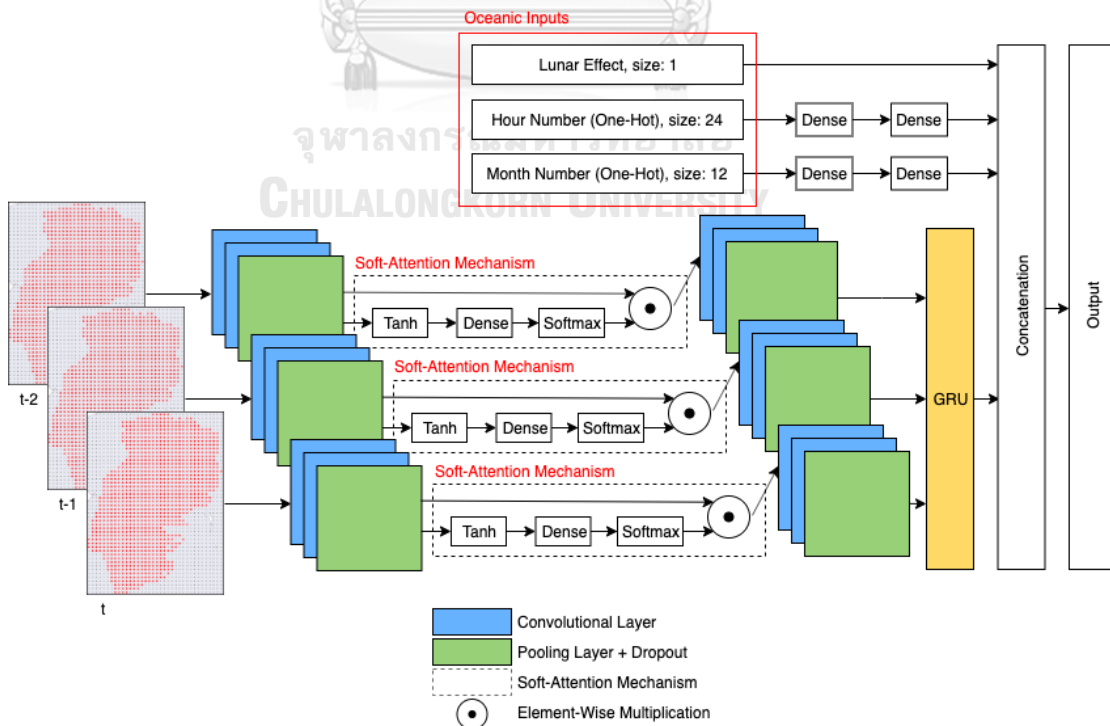


Figure 34 Model architecture of CNN-GRU-Input (All)-Att-TL

Chapter 5

Experimental Setup

In the section, we discuss how to prepare the experiment, and details of all experiment models which can be categorized in three parts such that data preparation, performance evaluation, and forecasting model.

5.1 Input Data

Input data is used in our experiment, which consists of (i) U and V components dataset and (ii) data partitioning for training, validating, and testing phases.

5.1.1 Data

U and V components were received from GISTDA since 2014 to 2016 and were pre-processed, as described in chapter 4.

5.1.2 Data Partitioning

Since the ocean surface currents in Gulp of Thailand has its seasonality effect and the dataset contains three years of data, we split the year 2014, year 2015 and year 2016 as training, validation and test datasets, respectively as shown in Table 10. Training dataset is used in model training process, validation dataset is used to pick the optimal model, and test dataset is used for performance evaluation.

Table 10 Year and dataset in the experiment

Year	Dataset
2014	training
2015	validation
2016	test

5.2 Performance Evaluation

Root-mean-square error (RMSE) used as a metric for evaluating the model on both U and V components on next one timestep which equals to the next one hour in this research. The RMSE equation is shown in (27).

$$\text{RMSE} = \sqrt{\frac{1}{mn} \sum_{i=1}^m \sum_{j=1}^n (y_{\text{predict}} - y_{\text{actual}})^2} \quad (27)$$

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

5.3 Forecasting Model

In this intensive experiment, there are thirteen models including our proposed method. Each model has been optimized its performance using various hyperparameters on the validation dataset. The proposed model has been tested sequentially to shows the effect of each module applied.

5.3.1 Lookback Timestep

A naive model's forecasting value equals to last k timesteps of observed value as shown in (28), for example if $k = 1$ then a predicted value will be equals 1 previous timestep (hour) in the same grid, as shown Table 11. 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.

$$y_t = x_{t-k} \quad (28)$$

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

Table 11 Example of predicted value of Lookback model when $k = 1$

Timestamp	Grid no	Label value	Predicted value
2015/12/31 11:00	1643	10	5
2015/12/31 12:00	1643	11	10
2015/12/31 13:00	1643	15	11
2015/12/31 14:00	1643	17	15
2015/12/31 15:00	1643	20	17
2015/12/31 16:00	1643	24	20
2015/12/31 17:00	1643	32	24

5.3.2 Moving Average (MA)

Moving Average is an unweighted mean of the previous k timesteps as shown in (29). k parameter from 2 to 4 has been deployed for tuning. The result showed that 2 was an optimal k on both U and V components.

$$y_t = \frac{1}{k} \sum_{i=1}^k x_{t-i} \quad (29)$$

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

5.3.3 Autoregressive Integrated Moving Average (ARIMA)

A suit of standard structures in time series data provides a simple and powerful method of time series forecasting. The acronym stands for each aspect of the model itself.

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, 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 12 for U and V components, respectively.

Table 12 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

5.3.4 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 12, distance and uniform for weight and n from 1 to 40 for search space of the model. The result showed 12 and 40, 12 and 28 are optimal parameters for k and n with uniform weight of U and V components, respectively.

5.3.5 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.

5.3.6 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.

5.3.7 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 [22] 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.

5.3.8 Gated Recurrent Unit (GRU)

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

5.3.9 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 [23] and PELU [24] were also being used in a hyperparameter tuning process. The optimal hyperparameters were summarized in Table 13 for both U and V components. In this model we fixed 3 as a number of input timesteps.

Table 13 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

5.3.10 CNN-GRU-Input

The model extended setup from the previous model and combined with new oceanic inputs such consists of month number, lunar effect, and hour number, as shown in Table 14. Four experimental models have been tested, one for each additional oceanic input and all inputs combined for the fourth model.

Optimized hyperparameters are the same for CNN-GRU model except for the number of hidden units on dense layers of the additional input layer. The result showed that 4 and 8 are an optimal number of hidden units for month number and hour number input for two dense layers of both U and V components. We further test the fourth model (CNN-GRU-Input (All)) which uses all new oceanic inputs to see the effect of its combination by using optimal hyperparameter for each additional oceanic input.

Table 14 Example of oceanic inputs

Timestamp	Month number (one-hot encoding)	Lunar effect	Hour number (one-hot encoding)
2015/12/31 11:00	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1]	0.32591	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
2015/12/31 12:00	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1]	0.31843	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
2015/12/31 13:00	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1]	0.31094	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
2015/12/31 14:00	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1]	0.30344	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
2015/12/31 15:00	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1]	0.29591	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0]
2015/12/31 16:00	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1]	0.28837	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0]
2015/12/31 17:00	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1]	0.28082	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0]

CHULALONGKORN UNIVERSITY

5.3.11 CNN-GRU-Input (All)-Att

Number of hidden units 16x16, 32x32 and 62x62 have been used for tuning phase inside two additional dense layers of implemented self-attention mechanism on a spatial property (feature map which is the result of CNN) to gain more accurate. The result shows that 32x32 is an optimal value for both U and V components.

5.3.12 CNN-GRU-Input (All)-Att-TL

The proposed model, after tuning previous model (CNN-GRU-Input (All)-Att), we did apply transfer learning concept by using trained model and weight as an initial state of the model. For example, in modeling process of V component, we used CNN-GRU-Input (All)-Att of U component as an initial weight and we normally tuning it again on U component dataset. The proposed model is shown in Figure 34.



Chapter 6

Experiment and Result

This section describes how all experiments we did perform and the result of it, which consists of (i) forecasting next one timestep, to demonstrate effect of each module we proposed (ii) rolling forecasting up to next 48 timesteps, to show model performance when it have to predict up to 2 days, and (iii) forecasting next one timestep with different number of input timesteps, to present the number of input timesteps' effect in the proposed model.

6.1 Forecasting Next One Timestep

This experiment compares all model forecasting values on the next one timestep (next one hour), on both U and V components by using RMSE as a performance evaluation. Models used in the experiment are Lookback, MA, ARIMA, Temporal kNN, Perceptron, MLP, CNN, GRU, CNN-GRU, CNN-GRU-Input (Hour Number), CNN-GRU-Input (Lunar Effect), CNN-GRU-Input (Month Number), CNN-GRU-Input (All), CNN-GRU-Input (All)-Att, and CNN-GRU-Input (All)-Att-TL, as described in Table 9.

6.1.1 Dataset and Partitioning

U and V components from GISTDA from 2014 to 2016 and generated oceanic inputs are used for particular models such as CNN-GRU-Input (All), CNN-GRU-Input (All)-Att, and CNN-GRU-Input (All)-Att-TL.

The U and V components dataset will be splitted into 3 parts which are list below, and the period of each dataset is shown in Table 15.

- 1) Training dataset will be used for training a model.
- 2) Validation will be used for selecting an optimal model which has highest accuracy.
- 3) Test dataset will be used for performance evaluation.

Table 15 Dataset period of forecasting next one timestep experiment

Dataset	Period
Training	1 Jan 2014 00:00 to 31 Dec 2015 23:00
Validation	1 Jan 2015 00:00 to 31 Dec 2015 23:00
Test	2 Jan 2016 00:00 to 31 Dec 2016 23:00

6.1.2 Evaluation

This experiment will evaluate all models on test dataset that described above by using RMSE on both U and V component for calculate error between label and prediction value.

6.1.3 Result

Result of the experiment will be shown and discussed into the section below. The example of forecasting value on test dataset is shown in Figure 35 and 36 and average of RMSE (group by hour) on each component is also shown in Figure 37 and 38 to demonstrate the difference between baseline models and CNN-GRU model.

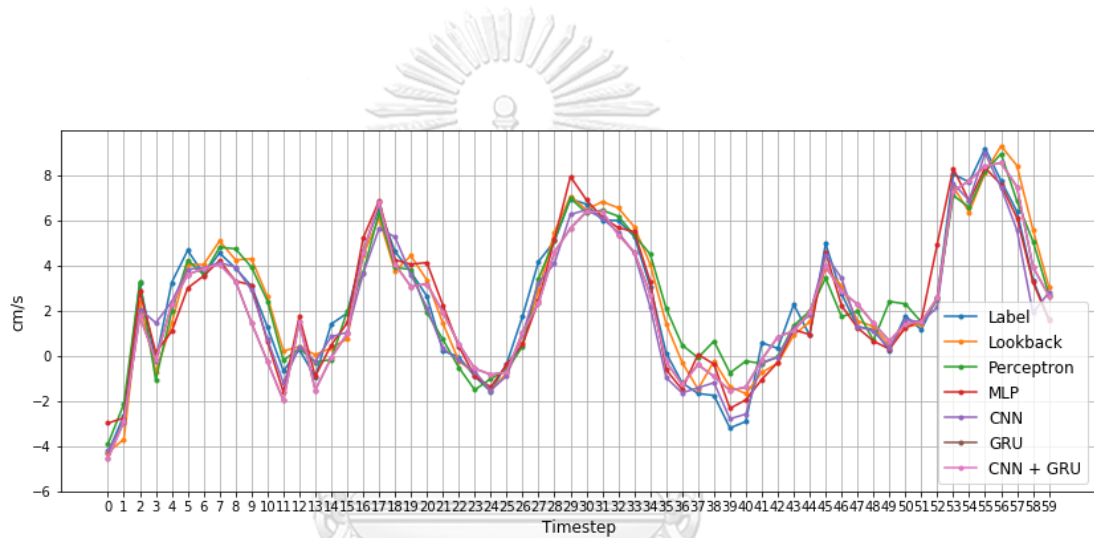


Figure 35 Example of forecasting value on test dataset of U component

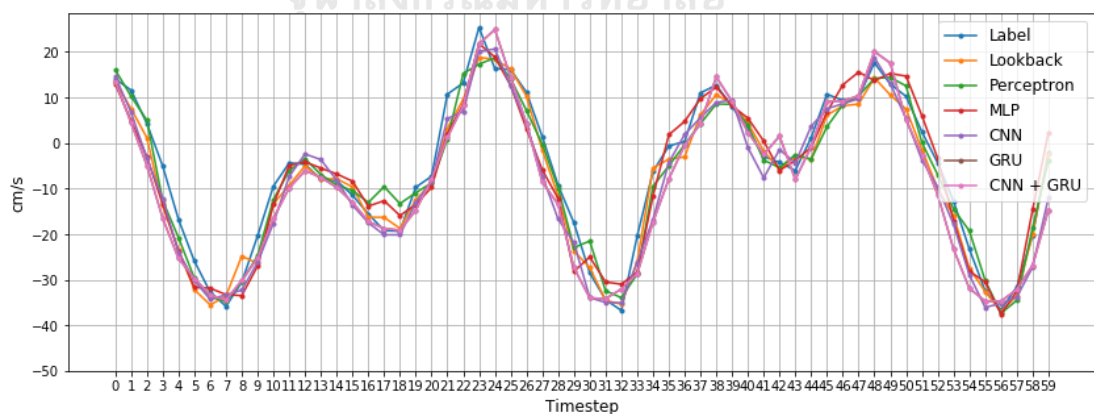


Figure 36 Example of forecasting value on test dataset of V component

6.1.3.1 Overview

This experiment evaluated all models on test dataset and compare all models with the proposed model (CNN-GRU-Input (All)-Att-TL). Table 16, RMSE on V component, shows that temporal property might be the dominant property over a spatial property from the result.

From the experiment result, RMSE is shown Table 16, deep learning based model is shown interesting performance such as CNN, GRU, and CNN-GRU etc. Each additional module which are oceanic inputs and modern deep learning techniques show its contribution. The proposed model which combines all modules has the highest accuracy (lowest RMSE) on both U and V components.

Table 16 Model's RMSE of next one timestep forecasting on U and V component, compared to the proposed model

Model	U component		V component	
	RMSE (cm/s)	Differences (%)	RMSE (cm/s)	Differences (%)
Lookback	5.0678	-15.28	11.1669	-56.00
MA	5.4222	-23.34	14.3236	-100.09
ARIMA	5.2516	-19.46	11.1284	-55.46
Temporal kNN	7.3207	-66.52	10.9683	-53.22
Perceptron	4.6360	-5.45	9.5185	-32.97
MLP	4.6951	-6.80	9.3969	-31.27
CNN	4.5354	-3.17	9.2040	-28.58
GRU	4.6246	-5.19	8.0180	-12.01
CNN-GRU	4.5090	-2.57	7.4050	-3.44
CNN-GRU-Input (Hour Number)	4.4856	-2.03	7.3618	-2.84
CNN-GRU-Input (Lunar Effect)	4.4969	-2.29	7.3793	-3.09
CNN-GRU-Input (Month Number)	4.4944	-2.23	7.3803	-3.10
CNN-GRU-Input (All)	4.4798	-1.90	7.3407	-2.55
CNN-GRU-Input (All)-Att	4.4213	-0.57	7.1988	-0.56
CNN-GRU-Input (All)-Att-TL	4.3962	-	7.1584	-

6.1.3.2 Effect of Combination Between Spatial and Temporal Characteristics

CNN-GRU has better RMSE than CNN and GRU individually on both U and V components. The combination of spatial and temporal effects (CNN-GRU) model shows an improvement of model performance as we expected, by considering both noticeable characteristics together which is compared between CNN-GRU and CNN (representative of spatial effect) or GRU (representative of temporal effect), as shown in Table 17. This supports that a combination of spatial and temporal effects is important.

Table 17 Effect of combination between spatial and temporal characteristics in term of RMSE comparing to CNN-GRU

Model	U component		V component	
	RMSE (cm/s)	Differences (%)	RMSE (cm/s)	Differences (%)
Lookback	5.0678	-12.39	11.1669	-50.80
MA	5.4222	-20.25	14.3236	-93.43
ARIMA	5.2516	-16.47	11.1284	-50.28
Temporal kNN	7.3207	-62.36	10.9683	-48.12
Perceptron	4.6360	-2.82	9.5185	-28.54
MLP	4.6951	-4.13	9.3969	-26.90
CNN	4.5354	-0.59	9.2040	-24.29
GRU	4.6246	-2.56	8.0180	-8.28
CNN-GRU	4.5090	-	7.4050	-

Regarding to the result Table 17, CNN-GRU showed a huge improvement by combination of CNN and GRU, especially in V component by 13.51% and 36.74% in average on U and V components, respectively.

The model (CNN-GRU) performance is improved on both U and V component which is shown in Figure 37 and 38, especially on V component in Figure 38 that temporal based model such as GRU and CNN-GRU perform very well.

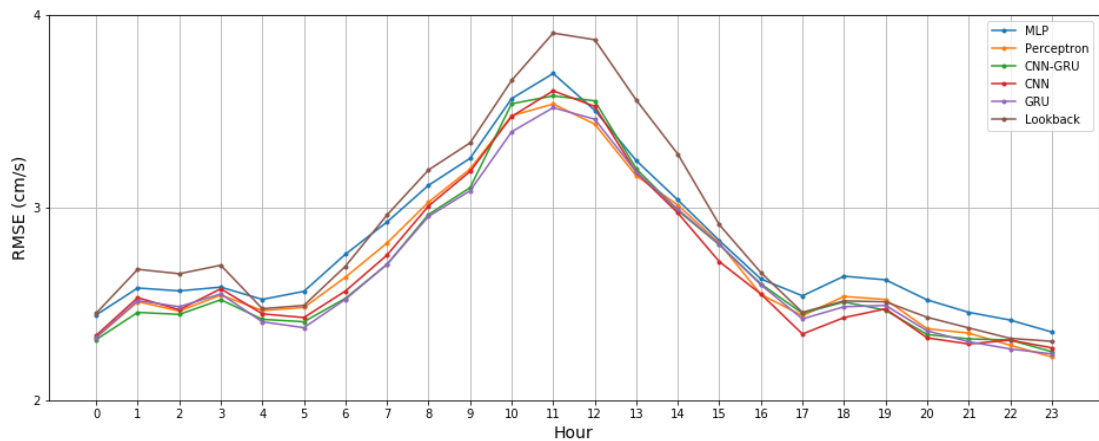


Figure 37 Average RMSE (hour) of models on U component

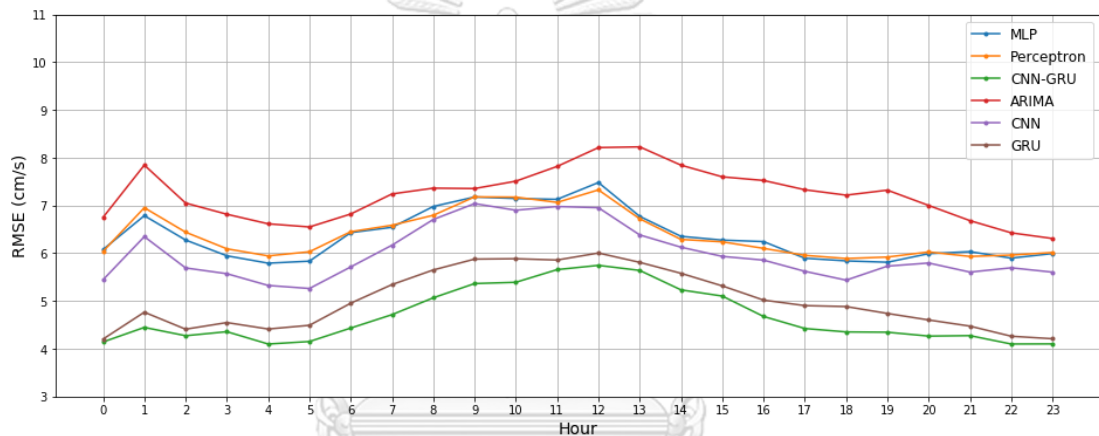


Figure 38 Average RMSE (hour) of models on V component

6.1.3.3 Effect of Oceanic Input

Adding ocean inputs as additional model inputs improve the performance, which is calculated between CNN-GRU-Input (All) and CNN-GRU. Each ocean input showed its contribution about the seasonal effect by improving the model accuracy (as shown in list below), and its combination as well. Oceanic input models are listed below.

- 1) CNN-GRU-Input (Month Number): add month number as additional input
- 2) CNN-GRU-Input (Lunar Effect): add lunar effect as additional input
- 3) CNN-GRU-Input (Hour Number): add hour number as additional input
- 4) CNN-GRU-Input (All): add all oceanic inputs as additional inputs

Table 18 Effect of oceanic inputs in term of RMSE comparing to CNN-GRU

Model	U component		V component	
	RMSE (cm/s)	Improvement (%)	RMSE (cm/s)	Improvement (%)
CNN-GRU	4.5090	-	7.4050	-
CNN-GRU-Input (Hour Number)	4.4856	0.52	7.3618	0.58
CNN-GRU-Input (Lunar Effect)	4.4969	0.27	7.3793	0.35
CNN-GRU-Input (Month Number)	4.4944	0.32	7.3803	0.33
CNN-GRU-Input (All)	4.4798	0.65	7.3407	0.87

Table 18 showed the contribution of each oceanic inputs. One oceanic input models which consist of CNN-GRU-Input (Hour Number), CNN-GRU-Input (Lunar Effect), and CNN-GRU-Input (Month Number) show an improvement comparing to CNN-GRU. The combination of all oceanic inputs, CNN-GRU-Input (All) also show the effect of all inputs combined by 0.65% and 0.87% comparing to CNN-GRU.

6.1.3.4 Effect of Attention Mechanism

Attention mechanism contributed by 1.31% and 1.93% on U and V components comparing to CNN-GRU-Input (All) model, as shown in Table 18.

Table 19 Effect of attention mechanism in term of RMSE comparing to CNN-GRU-Input (All)

Model	U component		V component	
	RMSE (cm/s)	Improvement (%)	RMSE (cm/s)	Improvement (%)
CNN-GRU-Input (All)	4.4798	-	7.3407	-
CNN-GRU-Input (All)-Att	4.4213	1.31	7.1988	1.93

According to the result table, the attention mechanism makes the model perform better with big contribution, as shown in Table 19 and visibly seen large gap in Figure 39 and 40 between CNN-GRU-Input (All) and CNN-GRU-Input (All)-Att model.

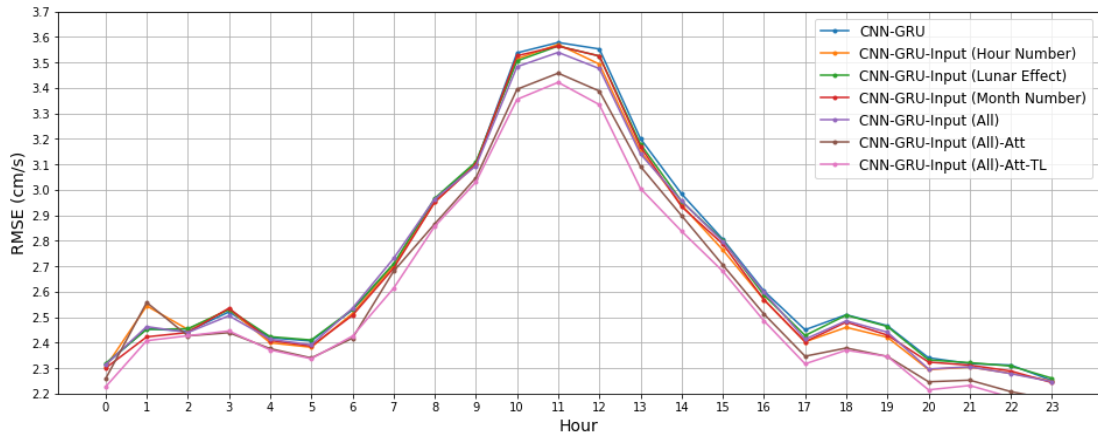


Figure 39 Average RMSE (hour) of each module on U component

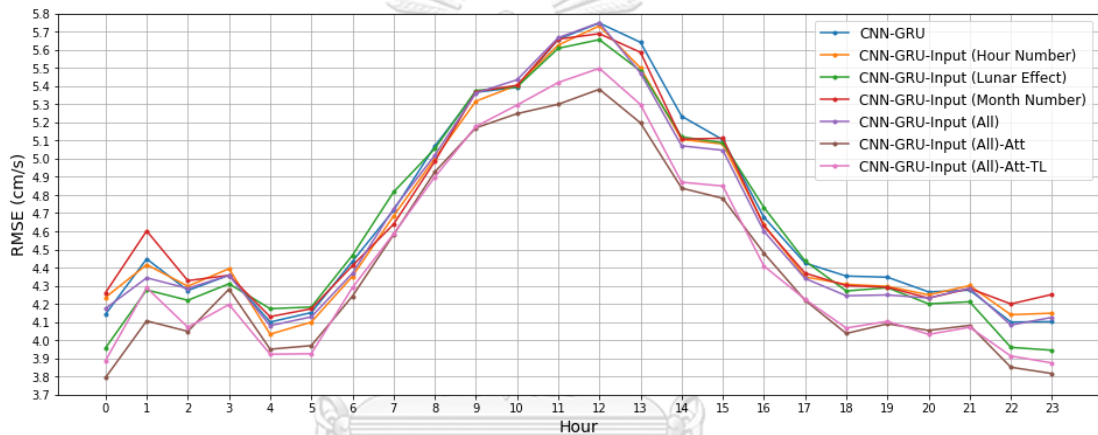


Figure 40 Average RMSE (hour) of each module on V component

6.1.4.5 Effect of Transfer Learning

Transfer learning concept was implemented into the model, can be implied that model has been trained with more data which we already described in chapter 4. The result showed that transfer learning technique improved the model accuracy by 0.57% and 0.56% on U and V components, as shown in Table 20.

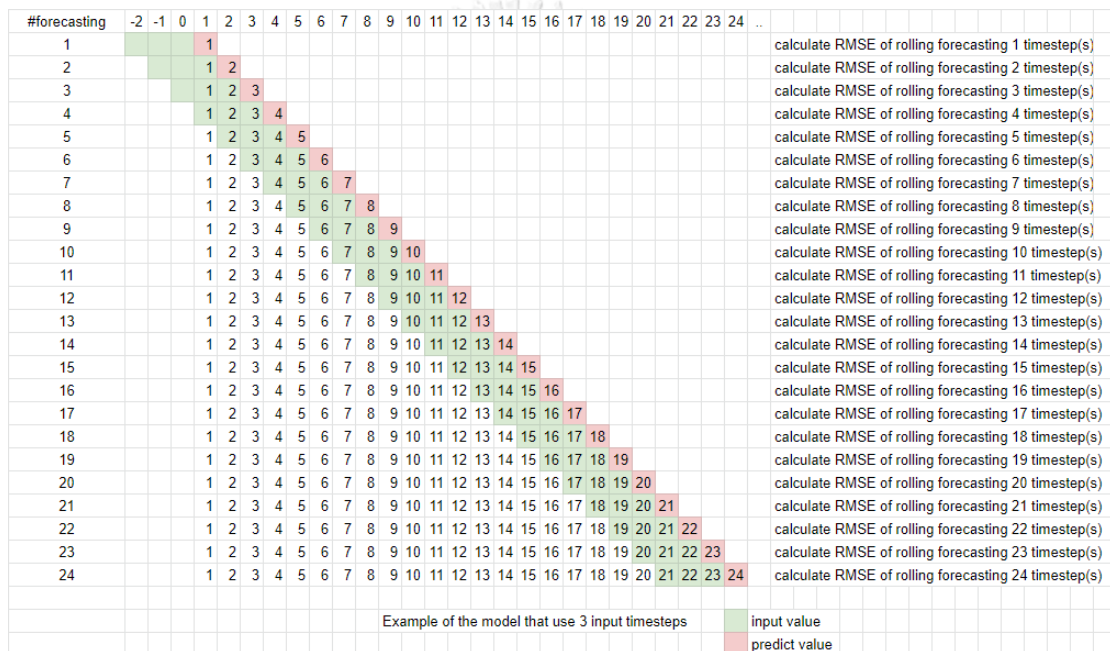
Table 20 Effect of transfer learning in term of RMSE comparing to CNN-GRU-Input (All)-Att

Model	U component		V component	
	RMSE (cm/s)	Improvement (%)	RMSE (cm/s)	Improvement (%)
CNN-GRU-Input (All)-Att	4.4213	-	7.1988	-
CNN-GRU-Input (All)-Att-TL	4.3962	0.57	7.1584	0.56

6.2 Rolling Forecasting Up to Next 48 Timesteps

This experiment evaluated the model on further next timesteps up to next 48 timesteps (hours). In some use cases, we need to forecast more than one timestep, for example in search and secure operation. We extend a number of forecasting timesteps from one to 48 for seeing the model performance in real cases.

In the testing period, we first forecast next 1 timestep and use forecasting value as an input for forecasting next 2 timesteps and we continually do it until reaches next 48 hours, as shown in Figure 41.



CHULALONGKORN UNIVERSITY
Figure 41 Example of rolling forecasting step

In model training phase, three models (CNN, CNN-GRU and proposed model) must be re-trained due to the input and output size. For example, to forecast next 1 timestep with CNN model, we use grid no in red square in Figure 42 (1728 grids) for an input because CNN required squared input and output was in red dot in Figure 42 (1065 grids) but in rolling forecasting we need output to be squared as well. In training process of 3 models, we have to using 1728 grids as a input and output and only validate on 1065 grids as well as MLP model, as shown in Figure 42.

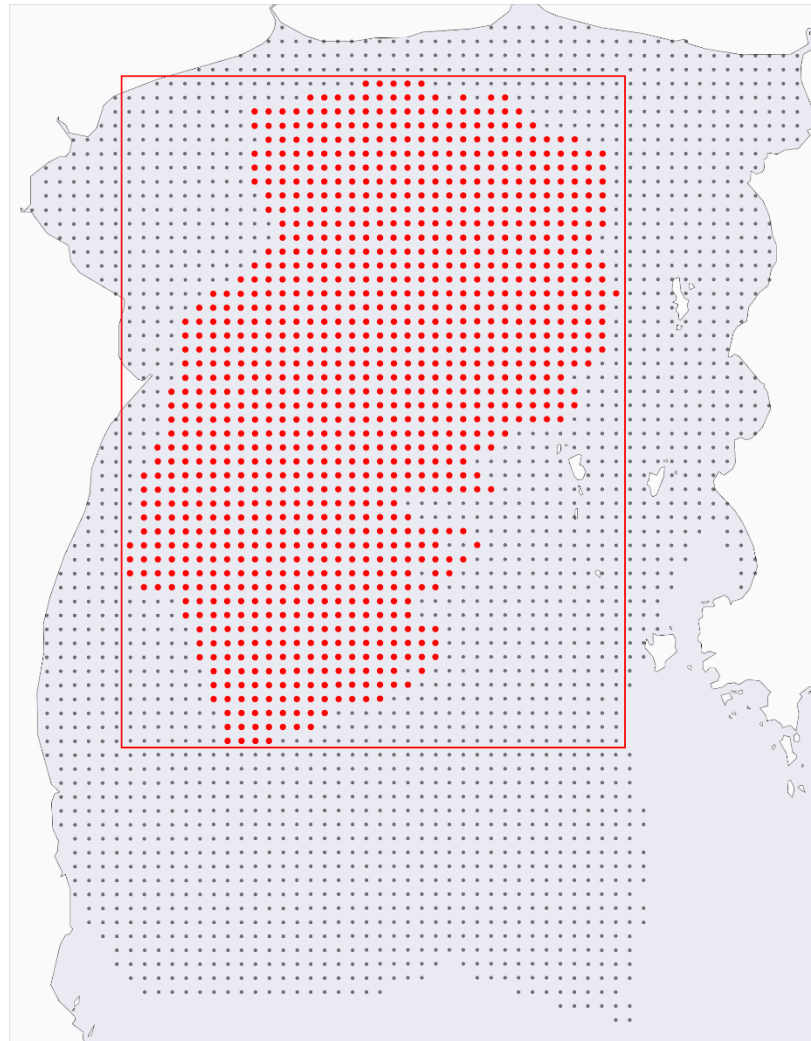


Figure 42 Input and output grid of CNN model for rolling forecasting

6.2.1 Dataset and Partitioning

The dataset is the same as the previous experiment, but the period is different due to number of next forecasting timesteps. The experiment model was tested on the next 48 hours of forecasting values. So, the dataset period is splitted into 3 parts as shown in Table 21.

Table 21 Dataset period of rolling forecasting up to next 48 timesteps experiment

Dataset	Period
Training	1 Jan 2014 00:00 to 31 Dec 2015 23:00
Validation	1 Jan 2015 00:00 to 31 Dec 2015 23:00
Test	3 Jan 2016 00:00 to 31 Dec 2016 23:00

6.2.2 Evaluation

This experiment will evaluate models on test dataset by using RMSE on both U and V components on each next forecasting timesteps until 48 timesteps (48 hours).

6.2.3 Result

The result of each forecasting timestep with 4 models on U and V components is plotted with line and box, as shown in Figure 43, 44 and 45. The proposed model (CNN-GRU-Input (All)-Att-TL) still shows its performance.

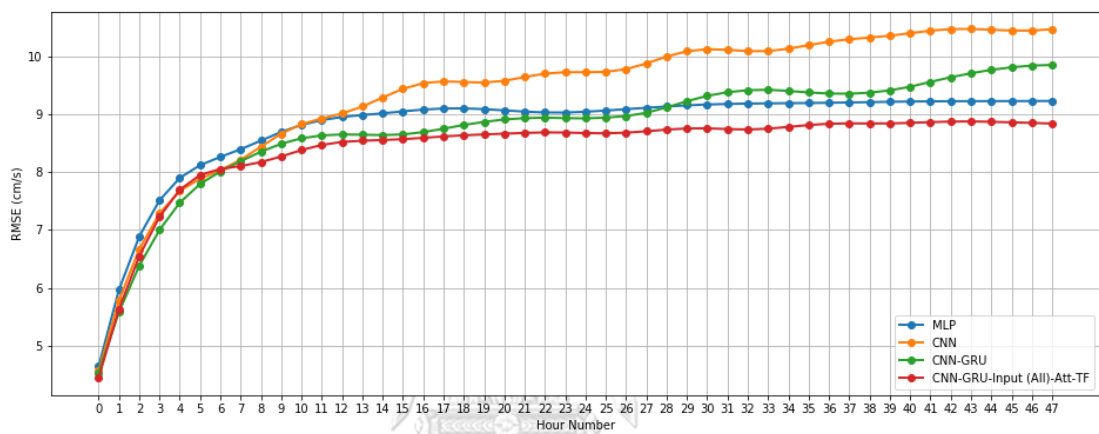


Figure 43 RMSE of rolling forecasting up to 48 hours on U component

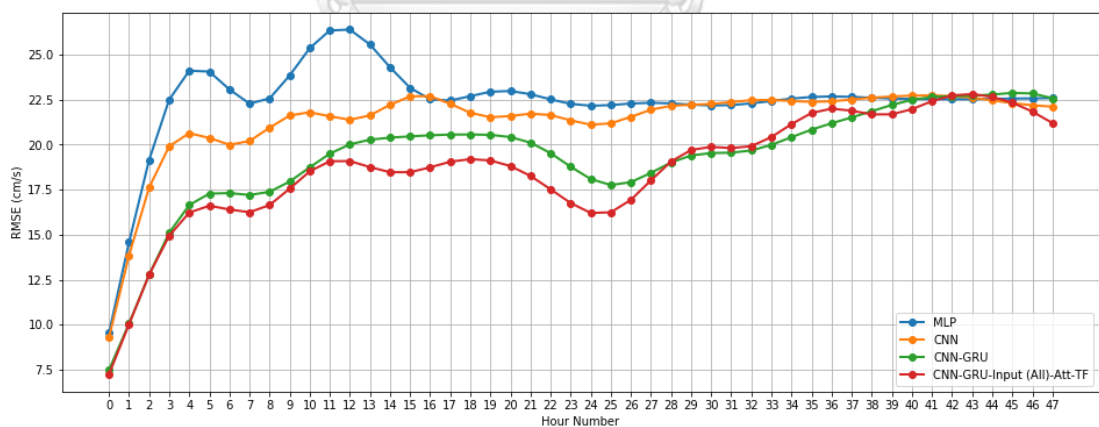


Figure 44 RMSE of rolling forecasting up to 48 hours on V component

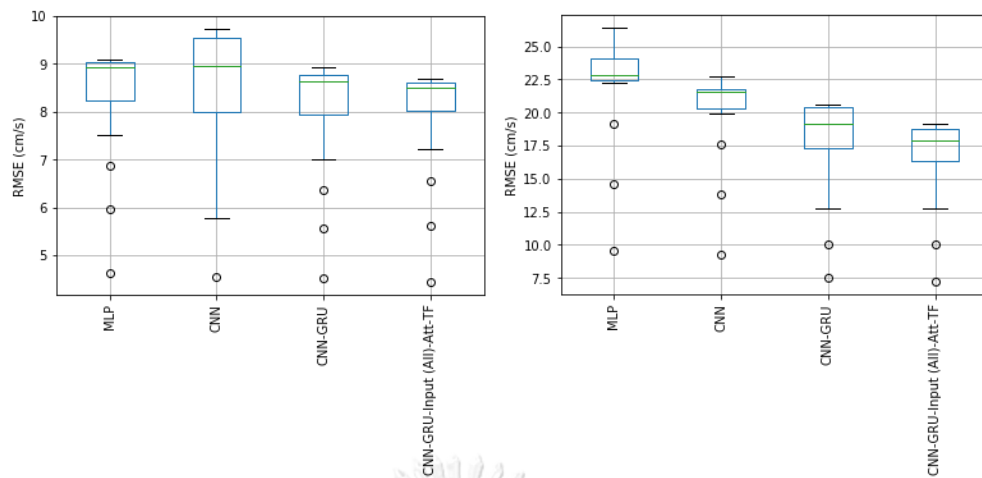


Figure 45 RMSE of rolling forecasting up to 48 hours in box plot, on U and V components respectively

6.2.3.1 Overview

In the results of Figure 43, 44 and 45, show the error continually increasing when a number of next forecasting timesteps is increased, as we expected. The proposed model (CNN-GRU-Input (All)-Att-TL) showed that model has the highest accuracy in average and better in almost all forecasting timesteps.

6.2.3.2 Temporal Effect

The result on U component is almost stable, unlike in V component. Figure 44, forecasting's RMSE on V component has the interesting point which is a large gap of an accuracy differentiation between non-temporal model and temporal model (CNN-GRU and proposed model) that address on temporal effect, especially on Hour Number 0 – 23 (forecasting up to 1 day). We obviously see better performance with a big gap on the graph in CNN-GRU and proposed model. There are 2 waves on result of CNN and proposed model that happened due to the nature of the data (daily seasonal effect).

6.3 Forecasting Next One Timestep with Different Number of Input Timesteps

The experiment is intended to reflect a number of input timesteps which we thought if we increase the number of input timesteps, the model should be able to remember the pattern. For example, if we use 13 as a number of input timesteps, the model should be able to see the effect of land breeze or sea breeze (which have about 12 timesteps as time period). In the section, we denote number of an input timesteps as #timesteps.

The winner model of first experiment (forecasting only 1 next timestep) was taken for this experiment. The winner model (CNN-GRU-Input (All)-Att-TL) only use 3 timesteps as a #timesteps. In this experiment, we will try with different #timesteps from 1 to 14, as shown in Figure 46.

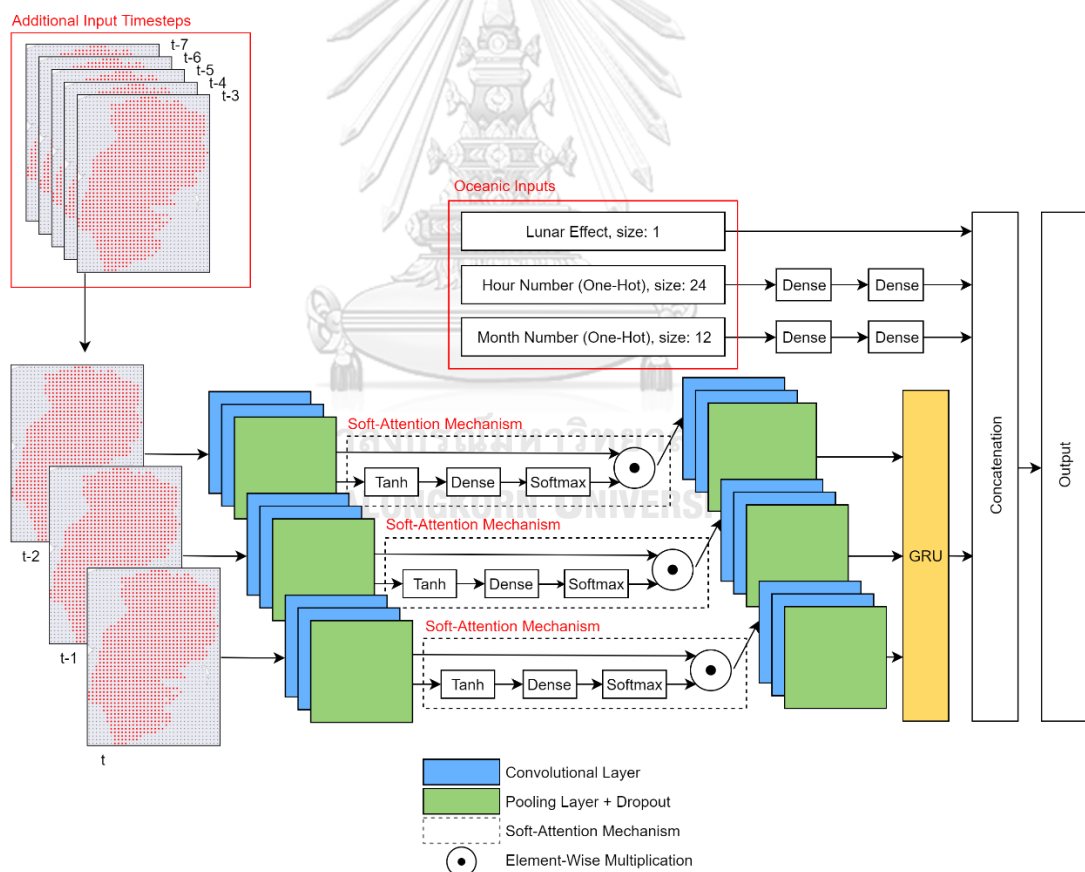


Figure 46 Example of adding more timesteps in a model input

In training or tuning process of the model, we used existing optimal hyperparameter for #timesteps equals 3. Number of units in GRU layer is only one hyperparameter that will be tuned in the experiment which are 64, 128, 256, 512, 1,024, 2,048 and 4096. The result shows 1,024 is an optimal value for both U and V components.

6.3.1 Dataset and Partitioning

Dataset in the experiment is the same as first experiment which are U and V components from GISTDA since 2014 to 2016 and generated oceanic inputs. Data partitioning also the same as first experiment.

6.3.2 Evaluation

This experiment was evaluated as same as the first experiment which is RMSE on both U and V components by calculating the difference between label and forecasting value.

6.3.3 Result

The experiment result showed the bigger number of input timesteps did not make model performance better but worse instead, as shown Table 24 and Figure 47 - 48.

Table 22 RMSE of proposed model with different number of input timesteps on U and V components

#timesteps	U component	V component
	RMSE (cm/s)	RMSE (cm/s)
1	4.54467	8.78182
2	4.45720	7.33044
3	4.42225	7.17287
4	4.44608	7.16332
5	4.45484	7.25531
6	4.45063	7.36691
7	4.47095	7.39322
8	4.49096	7.48469
9	4.48413	7.52422
10	4.51280	7.47238
11	4.50787	7.42888
12	4.51145	7.44740
13	4.54283	7.47230
14	4.55442	7.47417

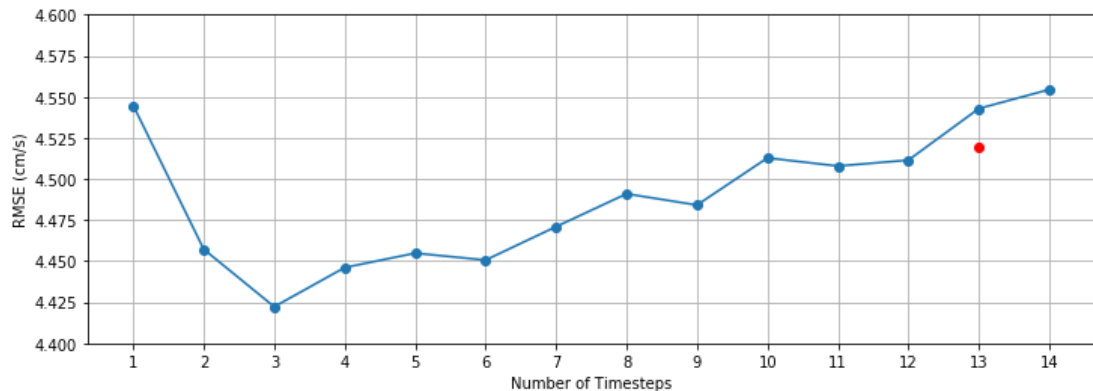


Figure 47 RMSE of proposed model with different number of input timesteps on U component

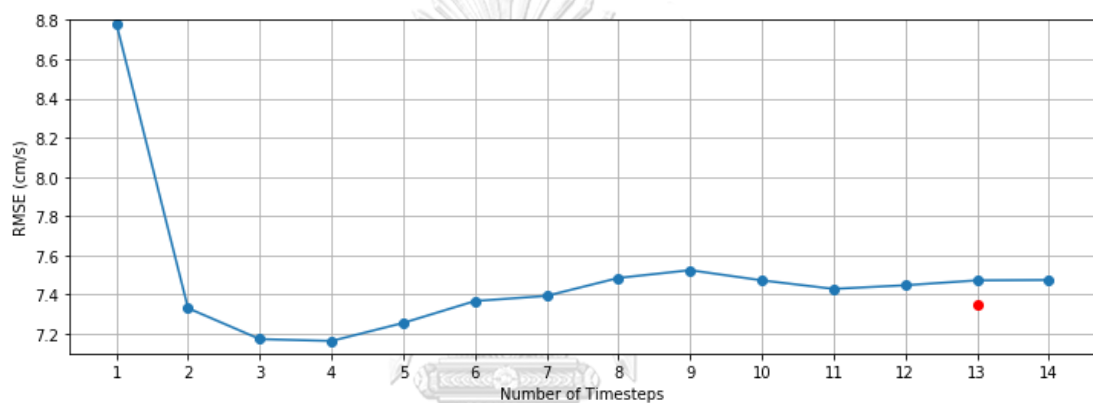


Figure 48 RMSE of proposed model with different number of input timesteps on V component

6.3.3.1 Overview

The RMSE result of the experiment is not as we expected as shown in Table 24. Figure 47 and 48 clearly showed the effect of different #timesteps. The model RMSE started decreasing when we added more #timesteps until 3 and 4 on U and V components then it began to increase.

6.3.3.2 Tuning on Model with Number of Input Timesteps Equals 13

The unexpected result form #timesteps 5 to 14 in blue line in Figure 47 and 48 occurred because we used optimal hyperparameter from the mode that using 3 as #timesteps, so it might not work with other models that use different #timesteps. We further tuned model on #timesteps equals 13 and plot it with the red dot in the same Figure (47 and 48). The result of tuned model that using 13 #timesteps which is shown in red dot still not perform better. This may because of the oceanic inputs we added that might cover the effect of big #timesteps, so that is why the result of big #timesteps is not as good as we expected.

Chapter 7

Summary and Future Work

This conclusion will summarize our work in this thesis and possibility of an improvement in ocean current prediction task.

7.1 Summary

This thesis proposed short-term ocean current prediction model which can be used to support all marine activities by deploying recent deep learning technique and implemented domain knowledge as a model input. HF radar dataset from 2014 to 2016 was provided by GISTDA. The model was expected to have highest accuracy among the other models which is evaluated on U and V components by using RMSE on the test dataset.

We applied three deep learning techniques as follow, (i) combination of CNN and GRU to consider noticeable spatial and temporal characteristics together, (ii) attention mechanism to express the grid area that model should focus on, and (iii) transfer learning to take advantage of knowledge from other components by using it as initial model weight.

Hour number, lunar effect, and month number are used as ocean inputs to represent the effect of ocean current seasonality. Hour number represented land breeze and sea breeze phenomenon. Lunar effect represented tide phenomenon and month number represented monsoon effect.

Three experiments performed to reflect aspect of its nature. First experiment, forecasting next one timestep was performed to address the effect of each added modules and the result showed that our proposed model is the winner among other models. Second experiment, rolling forecasting up to next 48 timesteps tested the model with longer timesteps of prediction values to see how the model perform when it has to forecast longer. Our proposed model still shows its impressive performance in the 2nd experiment. The last experiment was tested only on the proposed model to see the effect of different number of input timesteps. The result shows that the increasing of the number of input timesteps makes the model more accurate.

In term of applying to the application, the accuracy improvement of proposed model can help the real use case especially when the big period of time passes by. When the accuracy is in error at 1 cm/s, it is 0.86 km/day which is important for many cases such as oil spill trajectory, and search and rescue operation.

7.2 Future Work

Future works in ocean current forecasting model can be described in 3 topics.

7.2.1 Revise Model Architecture

Our proposed model architecture is one way of combining all modules into one model. There are many possible architectures that can be deployed and may have better performance without adding new contribution.

7.2.2 Address Temporal Effect

The model used only GRU to address temporal effect. GRU did excellent on performance improvement, but the model still cannot address longer information (big number of input timesteps) which already shown in 3rd experiment result (forecasting next one timestep with different number of input timesteps).

7.2.3 Address Sun Position

In our result of first experiment (forecasting next one timestep), it shows the RMSE grouped by hour number. In Figure 49 and 50, the RMSE started increasing from hour number 5 which is noon in local time and start decreasing when sunset. This behavior happened in both U and V components, so if we can address this topic, the model might have a big improvement in term of RMSE. Provided dataset timestamp offset is UTC±00, but in the local time is UTC+07.

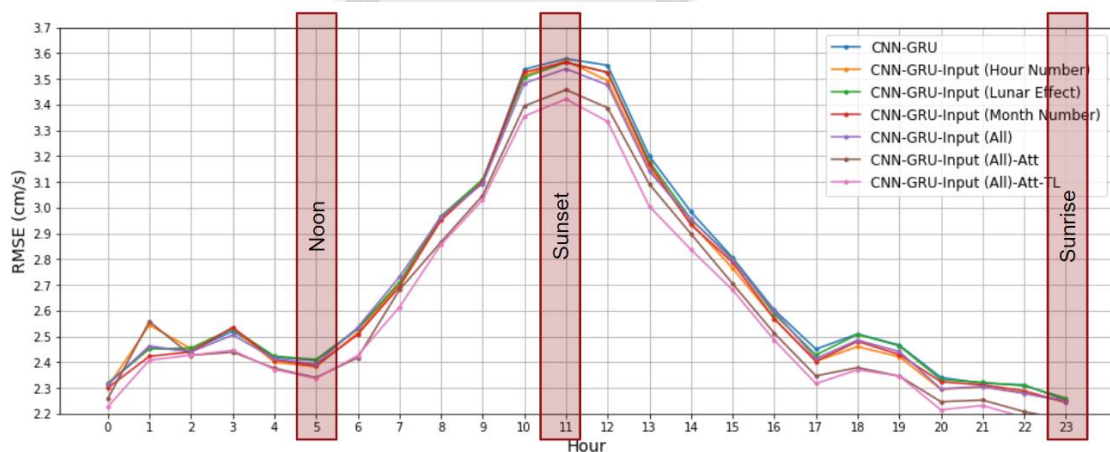


Figure 49 RMSE of models with different sun positions on U component

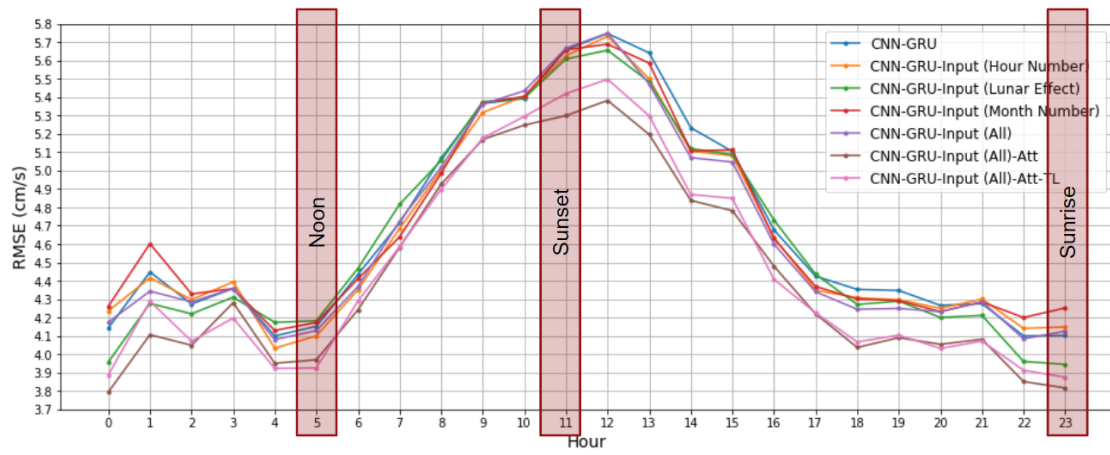


Figure 50 RMSE of models with different sun positions on V component



REFERENCES

1. Allard, R., et al., *The US Navy coupled ocean-wave prediction system*. Oceanography, 2014. **27**(3): p. 92-103.
2. Rozier, D., et al., *A reduced-order Kalman filter for data assimilation in physical oceanography*. SIAM review, 2007. **49**(3): p. 449-465.
3. Jirakittayakorn, A., et al. *Temporal kNN for short-term ocean current prediction based on HF radar observations*. in *Computer Science and Software Engineering (JCSSE), 2017 14th International Joint Conference on*. 2017. IEEE.
4. Ren, L., Z. Hu, and M. Hartnett, *Short-Term forecasting of coastal surface currents using high frequency radar data and artificial neural networks*. Remote Sensing, 2018. **10**(6): p. 850.
5. Kalinić, H., et al., *Predicting ocean surface currents using numerical weather prediction model and Kohonen neural network: a northern Adriatic study*. Neural Computing and Applications, 2017. **28**(1): p. 611-620.
6. Qin, Z., et al., *How convolutional neural network see the world-A survey of convolutional neural network visualization methods*. arXiv preprint arXiv:1804.11191, 2018.
7. Xu, K., et al. *Show, attend and tell: Neural image caption generation with visual attention*. in *International conference on machine learning*. 2015.
8. Buranapratheprat, A., *Circulation in the upper gulf of Thailand: A review*. Vol. 13. 2008. 75-83.
9. Ahmad, F., et al., *An Analysis of Anomalous Propagation in Peninsular Malaysia*.
10. Nair, V. and G.E. Hinton. *Rectified linear units improve restricted boltzmann machines*. in *Proceedings of the 27th international conference on machine learning (ICML-10)*. 2010.
11. Lecun, Y., et al., *Gradient-based learning applied to document recognition*. Proceedings of the IEEE, 1998. **86**(11): p. 2278-2324.

12. Koomsubha, T., *Text Categorization for Thai Corpus Using Character-Level Convolutional Neural Network*. 2016.
13. Cho, K., et al., *Learning phrase representations using RNN encoder-decoder for statistical machine translation*. arXiv preprint arXiv:1406.1078, 2014.
14. Bahdanau, D., K. Cho, and Y.J.a.p.a. Bengio, *Neural machine translation by jointly learning to align and translate*. 2014.
15. Yan, S., et al., *Hierarchical Multi-scale Attention Networks for action recognition*. 2018. **61**: p. 73-84.
16. Potdar, K., T.S. Pardawala, and C.D.J.I.J.o.C.A. Pai, *A comparative study of categorical variable encoding techniques for neural network classifiers*. 2017. **175**(4): p. 7-9.
17. Frolov, S., et al., *Improved statistical prediction of surface currents based on historic HF-radar observations*. *Ocean Dynamics*, 2012. **62**(7): p. 1111-1122.
18. Barrick, D., et al., *A short-term predictive system for surface currents from a rapidly deployed coastal HF radar network*. *Ocean Dynamics*, 2012. **62**(5): p. 725-740.
19. Saha, D., et al., *A combined numerical and neural technique for short term prediction of ocean currents in the Indian Ocean*. *Environmental Systems Research*, 2016. **5**(1): p. 4.
20. Richard E. Thomson, W.E., *Data Analysis Methods in Physical Oceanography*. 2014. **3**.
21. Meeus, J., *Astronomical Algorithms*. 1998: p. 477.
22. Simonyan, K. and A. Zisserman, *Very deep convolutional networks for large-scale image recognition*. arXiv preprint arXiv:1409.1556, 2014.
23. Clevert, D.-A., T. Unterthiner, and S. Hochreiter, *Fast and accurate deep network learning by exponential linear units (elus)*. arXiv preprint arXiv:1511.07289, 2015.

24. Trottier, L., P. Gigu, and B. Chaib-draa. *Parametric exponential linear unit for deep convolutional neural networks*. in *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*. 2017. IEEE.





จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

VITA

NAME Nathachai Thongniran

DATE OF BIRTH 25 March 1991

PLACE OF BIRTH Bangkok

INSTITUTIONS ATTENDED B.Eng. (First Class Honours), Electronic and Telecommunication Engineering, King Mongkut's University of Technology Thonburi (2009-2013)

HOME ADDRESS 133, Charansanitwong 3 Road, Wat Tha Phra, Bangkok Yai, Bangkok 10600

PUBLICATION N. Thongniran, P. Vateekul, K. Jitkajornwanich, S. Lawawirojwong, and P. Srestasathiern, "Spatio-temporal deep learning for ocean current prediction based on HF radar data", 2019 - 16th International Joint Conference on Computer Science and Software Engineering (JCSSE 2019)

N. Thongniran, K. Jitkajornwanich, S. Lawawirojwong, P. Srestasathiern, and P. Vateekul, "Combining Attentional CNN and GRU Networks for Ocean Current Prediction Based on HF Radar Observations", 2019 - 8th International Conference on Computing and Pattern Recognition (ICCP 2019)