INTERNATIONAL JOURNAL OF
COASTAL & OFFSHORE ENGINEERING
IJCOE Vol.5/No. 3/Fall 2020 (35-41)

Available online at: www. Ijcoe.org

# Positioning Using Classification and Regression: Case study of Oman Sea

**Ali Ghorbani[1], Mohammad Reza Khalilabadi [2*]**

*[1] Department of Computer Science, Shiraz University, Shiraz, Iran;alighorbani29@gmail.com*
*[2*] Assistant Professor, Faculty of Naval Aviation, Malek Ashtar University of Technology, Iran; khalilabadi@mut.ac.ir*

**ARTICLE INFO**

*Article History:*
Received: 15 Nov. 2020
Accepted: 26 Jan. 2021

*Keywords:*
Classification
Regression
Deep Learning
Ensemble Models

**ABSTRACT**

In the past few years, the location prediction played a critical role in many applications like intelligent self-learning vehicle, ocean location prediction because of the security and speed issues of GPSs. In this study, we proposed a model for location prediction on Oman's gulf using a NetCDF Data set. The proposed model is based on classification and regression which means it first mapped the data in a region on Oman's Gulf using classification and then using regression models to predict a specific location. This progress effect both response time and error of the system. And to the best of our knowledge, no researches are using the same idea. We used multiple classification models for classification tasks (both ensemble models and simple models) and two regression models (linear and XGboost regressor). The result shows reduce of man square error after using classification for regression task. Also, the result and explanation of the data capturing model are provided in the paper.

## 1. Introduction

The purpose of this study is to design and implement a positioning system at sea using machine learning algorithms based on environmental parameters. The occurrence of an event, tracking of a moving object, or monitoring the physical condition of an area are some of the applications in which the position and coordinates of the agent are very important. Also, in the case of underwater sensor networks, not knowing the location of a sensor causes the data collected by that sensor to be useless. This is especially important in underwater sensor networks, wherein many applications it is necessary to know the location of the nodes. Therefore, in underwater sensor networks as well as ground networks, determining the position of a sensor node is important.

Also, due to the risk of position information leakage of agents using satellite-based positioning systems for physical and strategic factors, the need to design and implement positioning systems without the risk of interception is essential. For example, the following can be mentioned as an example of these risks.

In data collection systems in insecure environments, an attacker may want to prevent the nodes from being positioned accurately and thus prevent the network from working properly. The enemy may compromise with some nodes and thus gain secret keys and other data stored in the nodes. This information can be used to provide misleading information to the base station as well as to other nodes in the network. Without an effective approach to refine or eliminate the effect of incorrect information, the positioning algorithm leads to an incorrect estimate of the sensor position. Therefore, it is necessary to design secure positioning algorithms that are resistant to attacks and obtain the correct position of the node in the presence of intrusion. One of the secure systems is systems based on environmental parameters that find the operating position according to characteristics such as seafloor pattern, water salinity, temperature, etc. In these systems, finding a pattern based on features is necessary for positioning, and considering the results of recent years, deep learning, and artificial networks, as well as cumulative models in finding patterns, classification, and regression, it makes sense to use these methods for this design.

The data used in this design is NetCDF format data, which after preprocessing is converted to CSV format, which is the format used and standard for data engineering and data analysis. The powerful Tensorflow library in the python language was also used to implement this design, as well as the numpy, sickitlearn, pandas, LightGBM, and XGboost libraries.

## 2. Back Ground

In this section, we provide a summary of the base concepts of our work:

### 2.1. Classification

Classification models are machine learning models that aim to assign one of the values of the discrete class k to a property called the label with respect to input x. [1] there are many classification models some of them are based on binary classification like SVM and Logistic Regression and some are not like Decision trees and Random forest. Many of these models predict the label by finding a decision boundary in x space which causes to divide the x space to different regions named decision regions. [2] some other algorithms are aimed to first map the data into another space named latent space and then tries to separate the different class labels like deep learning models. [3] Another way to categories of different algorithm models is ensembling. A summary of ensemble models and some of the ensemble models we used in this research have been provided in the next subsection and also a summary of Deep learning in the subsection after that.

### 2.2. Ensemble models

These models are models that select a set of models and hypotheses and combine their predictions for the final decision. These models are easy to understand and solve for complex problems because they break it down into simpler sub-problems. These models are also more reliable than single models. Figure 1 teaches a basic set learning model, using different training data or different learning algorithms, learn several alternative definitions of a concept. [4]

#### 2.2.1. Random forest

Is an ensemble model for classification and regression and other task using multiple decision trees.[5] The random forest is very similar to the bagging of trees [6] only difference is they use a modified tree learning algorithm that selects, at each candidate split in the learning process, a random subset of the features. This process is sometimes called feature bagging.
Fig 2 shows the difference between decision tree and random forest.

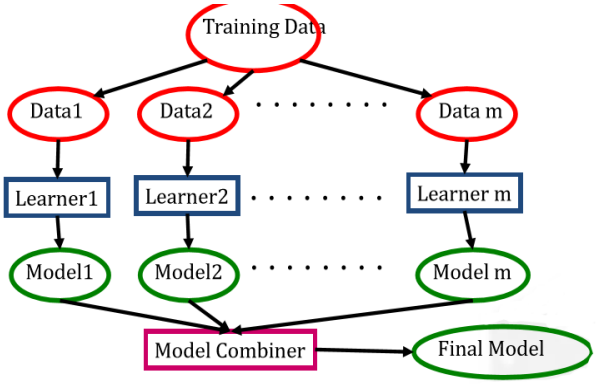


**FIGURE 1. A simple ensemble learning algorithm**

#### 2.2.2. XGboost

XGBoost is a group algorithm that recently designed distributed slope amplifier libraries that are highly efficient, flexible, and portable in machine learning and Kaggel competitions for optimized structural or tabular data. It implements machine learning algorithms under the Gradient Boosting framework. XGBoost provides parallel tree amplification (also known as GBDT, GBM) that solves many data science problems quickly and accurately. [7]

#### 2.2.3. lightGBM

LightGBM is a gradient boosting framework that uses tree-based learning algorithms. It is designed to be

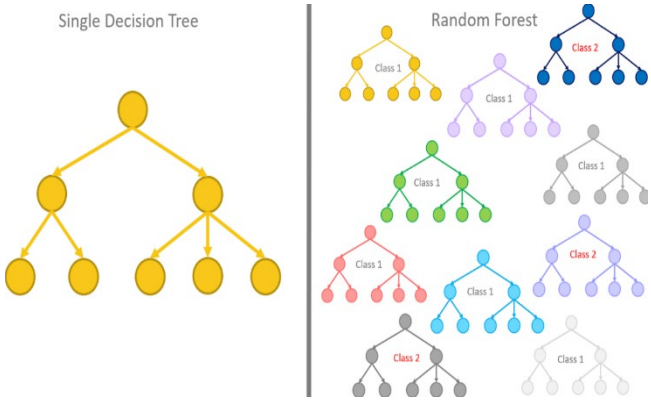


**FIGURE 2. Difference between single decision tree and random forest**

distributed and efficient with the following advantages: [8]

Faster training speed and higher efficiency.

- Lower memory usage.
- Better accuracy.
- Support for parallel and GPU learning.
- Capable of handling large-scale data.

### 2.3. Deep Learning

Deep learning is a subcategory of machine learning that tries to find high-level features from input data. [9]

Deep learning is based on the neural network (stochastic models that are directly inspired by the human and animal nervous systems and intend to find a relationship between input and output). Each neural network has two paths, forward and back-ward. In the forward, the input is mapped to the output by passing through a diagram with linear or nonlinear deformation layers, and in the backward, it calculates the cost slope according to the weights and tries to update the weights. In other words, it is based on feature and learning representation in different layers of the model. Deep learning models break down any complex

concept into simpler concepts. This process leads the model to find the basic concepts. In recent years, deep learning has achieved significant success in many areas such as machine vision, machine translation, speech recognition due to access to complex data patterns.

### 2.4. Regression

The goal of regression is to predict the value of one or more continuous target variables t given the value of a D-dimensional vector x of input variables.

## 3. Existing Works

Many studies used machine learning models for the natural area.[10] used random forest for Predictions of Seafloor Biomass which is similar to our research in the With a perspective of using ensemble models for prediction a biological metric. As the same perspective li liqi et al..[11] also used the ensemble of SVM with KNN for Eukaryotic Protein Subcellular Location Prediction.

Also, in the perspective location prediction, these models are the hot topic in machine learning and pattern recognition areas like tracking and mobility. Cadger et al. [12] used the machine learning models for location and movement prediction In mobile ad-hoc networks for optimizing the routing protocols. In this work instead of using classification models, they used regression-based machine learning algorithms that can predict coordinates as continuous variables. Stojmenovic et al.[13] proposed a depth-first search (DFS) method for routing decisions in the localized routing area. Chen et al. [14]proposed a model for geographic routing As nodes need to maintain up-to-date positions of their immediate neighbours for making effective forwarding decisions. As these works show, machine learning and artificial intelligence-based models achieved promising results in mobility and network location prediction so it is reasonable to use these models for location prediction on oceans.

## 4. Data Collection

In this research, we used the Massachusetts Institute of Technology general circulation model (MITgcm). This model solves the completely nonlinear and non-hydrostatic Navier-Stokes equations under the Boussinesq approximation for the inflexible fluid by discretizing the finite space in an orthogonal computational network. The model formulation includes implicit free surface topography and partial step. The MITgcm formulation described in detail by [15]and its source code and documentation are available at the MITgcm group Website [16].

The proportional design selected for this study is a finite third-order direct space-time flux design , [17], which is unconditionally stable.

Turbulent closure parameters for the viscosity and vertical penetration provided by [18] was used in this configuration:

$$\vartheta = \frac{\vartheta_0}{(1+\alpha Ri)^n} + \vartheta_b \quad , \qquad \mathrm{k} = \frac{\vartheta}{(1+\alpha Ri)} + k_b \qquad (1)$$

Where $Ri = \frac{N^2(z)}{(u_z^2+v_z^2)}$ is the Richardson number, $\vartheta_b = 1.5 \times 10^{-4} m^2 s^{-1}$, $k_b = 1 \times 10^{-7} m^2 s^{-1}$, and $\vartheta_0 = 1.5 \times 10^{-2} m^2 s^{-1}$, $\alpha = 5$ and $n = 1$ are adjustable parameters. Horizontal diffusivity coefficient is $k_h = 1 \times 10^{-2} m^2 s^{-1}$, While variable horizontal viscosity uses parameterization of [19]As shown by [20], Using these parameters, the numerical model gives quite solid results even if the wave breaks. This configuration is part of the numerical modeling of internal wave modeling in the Gulf of Oman. The main domain is between $56 - 59^{\circ}E$ and $23.4 - 27.4^{\circ}N$ and was discretized by a non-uniform orthogonal grid of 480*342 points. Spatial resolution along the longitudinal axis, $\Delta x$, ranges between 500m (near the sill region) to 1000m, and along the latitudinal axis, $\Delta y$, is 1000m. This model has 32 z-levels where the thickness of the layers increases from the surface down. Topographic data were obtained from the National Surveying Center (NCC) of Iran using high-resolution batimetric diagrams. No slipping was imposed on the bottom and side borders.

Average monthly sea surface temperature (SST), sea surface salinity (SSS) of the WOA database and climatic data (wind and heat budget components) database from NOAA [21]. These data prescribed in the model for 12 months of the year. The model range has two open borders on both the west and east sides. Western Open Boundary Conditions Imposed by Hourly Observation Data on Salinity, Temperature, and Flow Profiles from Surface to Bottom Layer at a Distance of 10 m, and Eastern Open Boundary Conditions Imposed by Hourly Observation Data of Sea Surface Height (SSH) predicted data of salinity, temperature, and current profiles. This data is prescribed in the Open Conditions section of the model.To validate the MITgcm model, the monthly average temperature and January salinity profile obtained from the WOD program are compared with the MITgcm simulation results.

Figure 3 shows theses comparison at a point (which situated at $24.7^{\circ}$N and $57.4^{\circ}$E). These plots show a reasonable comparison between WOD and the numerical results.

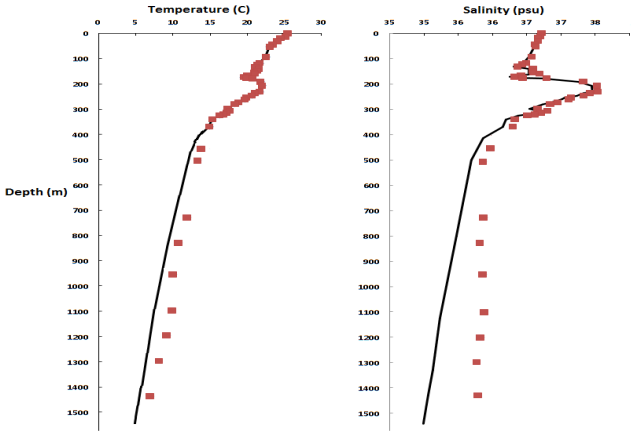


**FIGURE 3. A comparison between the monthly averages of temperature and salinity profiles obtained from the WOD program and the numerical results.**

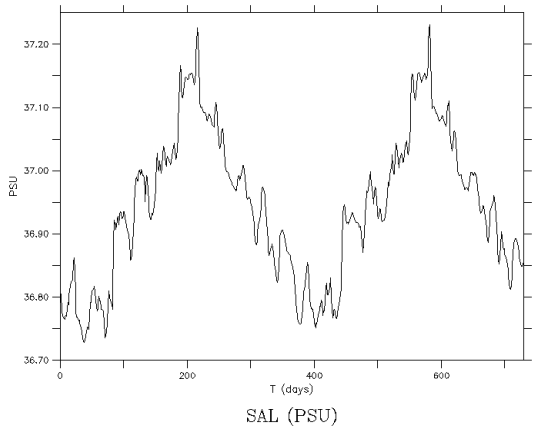


**FIGURE 4. Time series of this surface layer salinity site**

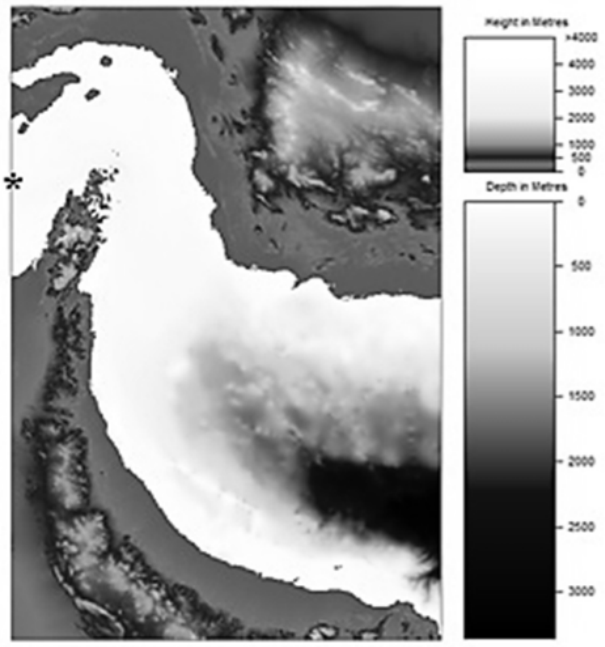


**FIGURE 5. Topology of the study area**

## 5. Proposed methodology

In this section, we will describe our model and its benefits in two parts which in part A, the description of our novel model based on classification and regression are provided, and benefits of it compared to the regression model will discuss in part B.

### 5.1. Region detection using classification and latitude-longitude prediction using Regression

In this research because of the high extent of the area in the Oman's Gulf first, we split the whole area into smaller regions (split latitude into 15 regions and longitude into 10 regions). After that, we assign a new label to data which is the number of the region each data is on it. The classification model job is to train using this new label and in test time predict the region of the data. After that, we train regression models for each region (150 regression models), and in the test time after finding the region of data we used the corresponding regression model to predict the latitude and longitude. We used the ensemble models for classification also Deep learning model result has been provided for comparison and regression we used linear regression, XGboost Regressor, and also Deep learning model for regression. See algorithm 1.

| Algorithm 1: coordinate prediction using classification and regression |
|---|
| Input: Data X_train, X_test, latitude X_train, longitude X_train, latitude X_test, longitude X_test<br>Output: latitude and longitude for X_test<br>X_train ←Preprocessing(X_train)<br>X_test ← Preprocessing(X_test)<br>Region_train ← split latitude into 15 regions and longitude into 10 regions<br>Clf ← train classification model (input = X_train, label= Region_train)<br><br>For i←1 to number of Region_train do<br>Regressor [i] ← Train Regressor model ( input = X_train in Region_train [i] )<br>Region_test ← Clf.predict(X_test)<br>For i←1 to size X_test do<br>Latitude[i], longitude[i] ← Regressor[Region_test].predict(X_test[i]) |

**5.2 benefits of the proposed model**

the first benefit of our model is the low error because the maximum error of the regression model is the distance of two boundaries of each region. Another benefit of our model compared to simple regression is training and testing speed even with the high number of regressors because these models are much simpler than one regressor they train and test very much faster.

## 6. Evaluation and result

Our proposed model implemented using GPU-enabled TensorFlow for deep learning models and sklearn library, python XGboost framework, python GPU enabled lightgbm framework. The models are performed on 64-bit Ubuntu 14.04 on PC with Intel(R) Core (TM) i5-6400 CPU at 2.70GHz, 16 GB ram, and NVIDIA GTX 1070 GPU.

### 6.1. evaluation metrics

To performed our evaluation, we used Accuracy rate (AC) for classification and mean square error (MSE) and mean absolute error for regression.

1. Accuracy rate:
   Percentage of correctly classified records overall records.

$$AC = \frac{number\ of\ correctly\ predicted}{number\ of\ all\ samples} \times 100\% \qquad (2)$$

2. Mean square error (MSE): The average squared difference between the estimated values and the actual value.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \widehat{Y}_i)^2 \qquad (3)$$

3. Mean absolute error (MAE): The average difference between the estimated values and the actual value.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \widehat{Y}_i) \qquad (4)$$

### 6.2. Data pre-processing

#### 6.2.1. Delete Zero samples

In the used dataset there are some samples with zero values for all features with can be happened cause of noise etc. In the first step, we delete these samples to prevent the biasing of models.

#### 6.2.2. Feature selection

The next step of pre-processing is to remove less important or irrelevant features from the dataset to reduce the complexity of the model and also prevent the negative impact of these features on the model. We used the correlation of each feature with labels for feature selection. The QLAT, QSOL, QNLW, NET are the features with minimum correlation with latitude and longitude of data so these features had been removed from the data set.

#### 6.2.3 Normalization

Many features of this data sets consist of a broad range between maximum and minimum, which achieves computational complexity during the learning and testing process. Therefore, we normalized these features using min-max normalization map each feature to range between 0 and one according to equation 5.

$$x_i = \frac{x_i - Min}{Max - min} \qquad (5)$$

Where xi is a data point, min is the minimum value from all data points, and max is the maximum value for each feature.

### 6.3. Result and comparison

Because the relation between samples would be so different over the year which could hurt the model, we randomly select 10 weeks from the dataset and performed our algorithm. The process not even impact results but also on the complexity of models. For showing these impacts the result using all data also provided. This process means that for predicting the Coordinates in the current week you only need the data for last week.

#### 6.3.1. Classification

In this section, the result of classification (region detection) considering different classification models (Random Forrest, XGboost, Lightgbm, Deep learning model) has been provided. We used 5-fold cross-validation using .7 of data as the training set and .3 data as the test set. As is shown in Table 1 we compared models in terms of the mean of accuracy, training, and

testing time for the randomly chosen 10 weeks and whole data. In the Deep learning model, we used 5 fully connected layers with sigmoid activation function and cross-entropy as loss and Nesterov momentum as the optimizer. As the result shows the XGboost model got the best accuracy rate and the Lightgbm model is the fastest.

**Table 1. Training and testing time and accuracy for randomly chosen 10 weeks and full data**

| Data | Method | Training time | Testing time | Latitude Accuracy | longitude Accuracy |
|---|---|---|---|---|---|
| **10 weeks** | XGboost | 1:22:30 | 3 min | 92.97 % | 93.41 % |
| **10 weeks** | Random Forrest | 14 min | 1.3 min | 95.70 % | 96.84 % |
| **10 weeks** | Lightgbm | 7 min | 30 sec | 87.93 % | 91.48 % |
| **10 weeks** | Deep neural net | 2:33:22 | 2 min | 64.3 % | 68.2 % |
| **Full data** | XGboost | 3:02:20 | 5 min | 82.69 % | 83.48 % |
| **Full data** | Random Forrest | 1:32:20 | 4 min | 81.22 % | 84.26 % |
| **Full data** | Lightgbm | 42 min | 1 min | 69.7 % | 72.3 % |
| **Full data** | Deep neural net | 5:33:12 | 10 min | 48.3 % | 52.2 % |

### 6.3.2. Regression

In this section the result of regression models has been provided we consider two regression models linear and XGboost regressor which is similar to the XGboost classifier but just with the continuous label as objective. For showing the effect of classification the result of regression only using one regressor (no classification) has been provided in Table 2. As the result shows the classification effect significantly on MSE and MAE and also, we can see it makes the regression problem very simpler because if we use the classification the number of samples and also the area of regressor reduced.

**Table 2. Regression result after using classification and without using classification**

| Using classification | Method | $\sqrt{MSE}$ X | MAE X | $\sqrt{MSE}$ y | MAE y |
|---|---|---|---|---|---|
| **Yes** | Linear | 16.70 | 14.0 | 12.87 | 10.87 |
| **Yes** | XGboost | 4.53 | 2.74 | 3.55 | 2.44 |
| **Yes** | Deep neural net | 15.41 | 11.3 | 10.3 | 8.7 |
| **No** | Linear | 162.18 | 126.02 | 180.22 | 150.02 |
| **No** | XGboost | 36.218 | 20.00 | 75.02 | 45.52 |
| **No** | Deep neural net | 150.52 | 111.22 | 140.74 | 127.29 |

## 7. CONCLUSION AND FUTURE WORKS

In this research, a novel model for location prediction based on the NetCDF data set on Oman's Gulf has been proposed. This model is based on two tasks a classification and regression. The classification task provides a region labeling for each sample which cause reduces on complexity and error between predicted location and reallocation. For evaluation, we used 5-fold cross using .7 of data for train and .3 for the test. Also, in evaluation, we performed our model on randomly chosen 10 weeks to reduce the complexity of the model but the result on the whole data, also provided. This is the first paper using both classification and regression for positioning. The result shows magnificent improvement in terms of MSE and MAE compared to the scenario of not using classification. Also, the results show that after using the classification the complexity reduced dramatically, which is a great achievement as the model most competes with GPSs. Also, the MSE on regression shows improvement compare to models with no classification. For the feature, we plane to use time-series models for both classification and regression to observe the effect of time on the accuracy, MSE, and MAE.

## 8. References


1. Bishop, C.M., *Pattern recognition and machine learning*. 2006: springer.
2. Mitchell, T.M., J.G. Carbonell, and R.S. Michalski, *Machine learning: a guide to current research*. Vol. 12. 1986: Springer Science & Business Media.
3. Bishop, C.M., *Neural networks for pattern recognition*. 1995: Oxford university press.
4. Dietterich, T.G. *Ensemble methods in machine learning*. in *International workshop on multiple classifier systems*. 2000. Springer.
5. Ho, T.K. *Random decision forests*. in *Proceedings of 3rd international conference on document analysis and recognition*. 1995. IEEE.
6. Breiman, L., *Bagging predictors.* Machine learning, 1996. 24(2): p. 123-140.
7. *XGBoost Documentation — xgboost 1.3.0-SNAPSHOT documentation.*
8. Ke, G., Q. Meng, and T. Finley, *Welcome to LightGBM's documentation*. LightGBM.
9. Deng, L. and D. Yu, *Deep learning: methods and applications.* Foundations and trends in signal processing, 2014. 7(3–4): p. 197-387.
10. Wei, C.-L., et al., *Global patterns and predictions of seafloor biomass using random forests.* PloS one, 2010. 5(12): p. e15323.
11. Li, L., et al., *An ensemble classifier for eukaryotic protein subcellular location prediction using gene ontology categories and amino acid hydrophobicity.* PLoS One, 2012. 7(1): p. e31057.

12. Cadger, F., et al. *MANET location prediction using machine learning algorithms*. in *International Conference on Wired/Wireless Internet Communications*. 2012. Springer.
13. Stojmenovic, I., M. Russell, and B. Vukojevic. *Depth first search and location based localized routing and QoS routing in wireless networks*. in *Proceedings 2000 International Conference on Parallel Processing*. 2000. IEEE.
14. Chen, Q., S.S. Kanhere, and M. Hassan, *Adaptive position update for geographic routing in mobile ad hoc networks.* IEEE Transactions on Mobile Computing, 2012. 12(3): p. 489-501.
15. Marshall, J., et al., *Hydrostatic, quasi-hydrostatic, and nonhydrostatic ocean modeling.* Journal of Geophysical Research: Oceans, 1997. 102(C3): p. 5733-5752.
16. Adcroft, A., et al. *Overview of the formulation and numerics of the MIT GCM*. in *Proceedings of the ECMWF seminar series on Numerical Methods, Recent developments in numerical methods for atmosphere and ocean modelling*. 2004.
17. Hundsdorfer, W., B. Koren, and J. Verwer, *A positive finite-difference advection scheme.* Journal of computational physics, 1995. 117(1): p. 35-46.
18. Pacanowski, R. and S. Philander, *Parameterization of vertical mixing in numerical models of tropical oceans.* Journal of Physical Oceanography, 1981. 11(11): p. 1443-1451.
19. Leith, C.E., *Diffusion approximation for two-dimensional turbulence.* The Physics of Fluids, 1968. 11(3): p. 671-672.
20. Vlasenko, V., N. Stashchuk, and K. Hutter, *Baroclinic tides: theoretical modeling and observational evidence*. 2005: Cambridge University Press.
21. Boyer, T.P., et al., *World ocean database 2013.* 2013.