ECOFISHCAST: A MACHINE LEARNING SYSTEM FOR ACCURATE PREDICTION OF OCEANIC DISSOLVED INORGANIC CARBON LEVELS

Haoyu Li[1], Marisabel Chang[2]

[1]Yorba Linda high school, 19900 Bastanchury Rd, Yorba Linda, CA 92886
[2]Computer Science Department, California State Polytechnic University, Pomona, CA 91768

ABSTRACT

The EcoFishCast system is an innovative tool designed to predict Dissolved Inorganic Carbon (DIC) levels in oceanographic environments using machine learning models [1]. By integrating a mobile application with a robust backend server, the system allows users to input environmental data and receive accurate predictions. Experiments conducted as part of the project identified Gradient Boosting and Random Forest as the most reliable models, particularly when combined with data scaling techniques, which significantly improved prediction accuracy [2][3]. While the system performs well, future enhancements are planned to address limitations related to training data diversity and computational efficiency, ensuring EcoFishCast remains a powerful and reliable resource for oceanographic analysis.

KEYWORDS

Dissolved Inorganic Carbon (DIC), Marine Science, Mobile Application, Machine Learning, Oceanographic Analysis

1. INTRODUCTION

The problem addressed by this project is the lack of comprehensive information on how and what to fish for in different parts of the world. Despite the popularity of fishing as a recreational activity, there is often limited guidance available for anglers, particularly when they are in unfamiliar waters. This project aims to fill that gap by developing an app that provides tailored fishing advice based on real-time environmental data. The background to this problem is rooted in the author's personal experiences and the broader issue of inadequate educational resources for sustainable fishing practices. The lack of reliable information can lead to overfishing, harm to local ecosystems, and missed opportunities for sustainable practices [4]. This issue is important not only for recreational anglers but also for marine biologists and conservationists, as it has long-term implications for marine biodiversity and ecosystem health [5]. By offering accurate, location-specific advice, the app can help promote sustainable fishing practices, protect fish populations, and enhance the fishing experience for enthusiasts worldwide.

The three methodologies explored by Codden et al. (2020), Mukherjee et al. (2020), and Sauzède et al. (2017) each aimed to enhance oceanic predictions through machine learning, but with specific limitations. Codden et al. focused on predicting dissolved organic carbon (DOC) in a salt

David C. Wyld et al. (Eds): CCSIT, NLPCL, AISC, ITE, NCWMC, DaKM, BIGML, SIPP, SOEN, PDCTA – 2024
pp. 353-362, 2024. - CS & IT - CSCP 2024 DOI: 10.5121/csit.2024.141730

marsh, achieving accuracy but remaining limited to that specific environment [6]. Mukherjee et al. developed a neural network for predicting ocean reflectance, improving computational efficiency but lacking direct application to chemical parameters like DIC. Sauzède et al. used the CANYON model to estimate DIC and other variables with high accuracy but relied heavily on extensive datasets, making it less effective in data-sparse regions [7]. EcoFishCast builds on these by directly targeting DIC prediction across diverse environments, optimizing models for real-time use, and integrating advanced AI for broader applicability and enhanced analysis [8].

The proposed solution to this problem is a mobile application that uses real-time environmental data to provide fishing advice tailored to the user's current location. This app leverages a combination of machine learning and oceanographic data analysis to predict the most suitable fishing methods and species to target based on factors such as temperature, salinity, and depth. The solution is effective because it provides anglers with actionable insights, reducing the guesswork involved in fishing and increasing the likelihood of success. Unlike traditional fishing guides that offer generalized advice, this app delivers personalized recommendations based on current conditions, making it a more reliable and efficient tool. Additionally, the app's use of real-time data ensures that the advice remains relevant and accurate, even as environmental conditions change. This approach not only improves the user experience but also contributes to sustainable fishing practices by encouraging users to target species that are abundant and avoid those that are overfished or at risk.

In the EcoFishCast project, two key experiments were conducted to assess and enhance the accuracy of DIC (Dissolved Inorganic Carbon) predictions. The first experiment focused on comparing the performance of various machine learning models, including Linear Regression, Random Forest, Gradient Boosting, and Support Vector Regressor. The results indicated that Gradient Boosting and Random Forest models provided the lowest mean squared error (MSE), making them the most suitable for deployment in the system. The second experiment examined the impact of data scaling on the models' performance, revealing that models like Support Vector Regressor and Linear Regression significantly benefited from scaling, with substantial reductions in MSE.

The experiments were designed to address potential blind spots in the prediction accuracy of the models and to refine the data preprocessing approach used in the system. The findings underscored the importance of careful model selection and preprocessing techniques in developing a robust and reliable predictive system. These insights have informed the overall design of the EcoFishCast system, ensuring that it delivers accurate and reliable predictions across a wide range of environmental conditions.

2. CHALLENGES

In order to build the project, a few challenges have been identified as follows.

2.1. Ensuring the Accuracy

One of the primary challenges in developing the EcoFishCast system was ensuring the accuracy of the Dissolved Inorganic Carbon (DIC) predictions across various environmental conditions. Oceanographic data can vary significantly depending on factors such as geographic location, depth, and seasonal changes. This variability poses a challenge for the machine learning models, which must generalize well to provide accurate predictions even when the input data differs from the conditions seen during training. Addressing this challenge involved carefully selecting and

testing different machine learning models and evaluating their performance under diverse scenarios.

2.2. Managing the Computational Load

Another significant challenge was managing the computational load of running complex machine learning models, such as Gradient Boosting and Random Forest, on mobile devices. These models, while accurate, require substantial computational resources, which can lead to latency issues and increased battery consumption on mobile platforms. This challenge was addressed by optimizing the backend server to handle most of the computational tasks, ensuring that the mobile application remains responsive and efficient. Additionally, exploring model compression techniques and using cloud resources for computation were considered as potential solutions to alleviate this issue.

2.3. Data Preprocessing

Data preprocessing, specifically the scaling of input features, presented another challenge in the system's development. Models like Support Vector Regressor (SVR) are particularly sensitive to the scale of the input data, and improper scaling can significantly impact prediction accuracy [9]. Ensuring that the input data was properly scaled required the implementation of a consistent preprocessing pipeline and careful validation to confirm that the scaling improved model performance without introducing biases. This challenge was addressed by integrating a standardized preprocessing step using tools like StandardScaler from scikit-learn, which was applied uniformly across all models to maintain consistency and accuracy in the predictions.

3. SOLUTION

The EcoFishCast system is a comprehensive solution that integrates a Flutter-based mobile application with a Python-powered backend server to predict and analyze ocean Dissolved Inorganic Carbon (DIC) levels. The mobile application serves as the user interface, where users input environmental parameters such as latitude, longitude, temperature, salinity, alkalinity, and depth. This data is then sent to the backend server, which processes the information and utilizes a machine learning model, specifically a gradient boosting algorithm, to predict DIC levels. The server also incorporates generative AI to provide additional analysis and guidelines based on the prediction results. The application features a straightforward user flow, starting with a splash screen, followed by a form for data entry, and culminating in a result screen where the prediction outcomes and related analyses are displayed.

The seamless interaction between the mobile application and the backend server ensures that users can quickly and accurately obtain predictions and insights into oceanic conditions. The system's design emphasizes ease of use, allowing users to easily navigate through the app and access important features such as the "About" page, which provides information on the application's purpose and functionalities. By combining advanced machine learning with a user-friendly interface, EcoFishCast offers a powerful tool for oceanographic analysis, helping users make informed decisions based on reliable predictions of DIC levels.

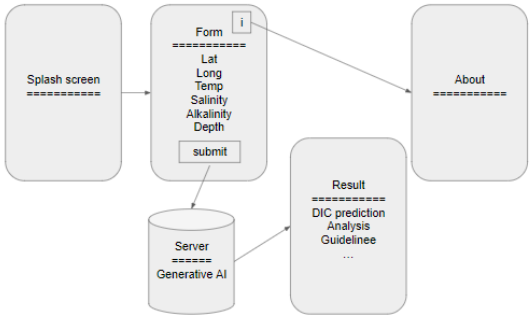

Figure 1. Overview of the solution

The first major component of the EcoFishCast system is the DIC Prediction Model [10]. This component is crucial for the system's primary function, which is to predict Dissolved Inorganic Carbon (DIC) levels based on environmental parameters provided by the user. The model is implemented using a gradient boosting algorithm, a type of machine learning model that excels in handling complex, non-linear relationships within data. The model has been trained on a dataset containing various oceanographic measurements, allowing it to make accurate predictions based on inputs like temperature, salinity, and depth. This prediction model is embedded within the backend server, where it processes incoming data, scales it appropriately using a pre-trained scaler, and outputs the predicted DIC levels. The accuracy and reliability of this model are vital for the overall success of the EcoFishCast system, as it directly impacts the quality of the predictions and insights provided to the user.

```
def train_model(X_train_scaled, X_test_scaled, y_train, y_test):
    # Models to train
    models = {
        'Linear Regression': LinearRegression(),
        'Random Forest': RandomForestRegressor(n_estimators=100, random_state=42),
        'Gradient Boosting': GradientBoostingRegressor(n_estimators=100, random_state=42),
        'Support Vector Regressor': SVR()
    }

    # Train and evaluate each model
    results = {}

    for name, model in models.items():
        model.fit(X_train_scaled, y_train)
        y_pred = model.predict(X_test_scaled)

        # Calculate the mean squared error
        mse = mean_squared_error(y_test, y_pred)
        results[name] = mse

        print(f"\n{name} - Mean Squared Error: {mse}")

    return models, results
```

Figure 2. Screenshot of code 1

The provided code snippet is a function named train_model that is integral to the DIC prediction process in the EcoFishCast system. This function handles the training and evaluation of four different machine learning models: Linear Regression, Random Forest, Gradient Boosting, and Support Vector Regressor.

The function starts by defining a dictionary of models, where each key is the model's name and the corresponding value is the instantiated model object. It then iterates over each model, fitting it to the scaled training data (X_train_scaled and y_train). After training, the model predicts the DIC levels on the test data (X_test_scaled). The accuracy of each model's predictions is evaluated using the mean squared error (MSE), which is then stored in the results dictionary. Finally, the function returns the trained models and their respective MSE scores, which are crucial for selecting the best-performing model for further deployment in the EcoFishCast system. This function plays a critical role in determining the model that will be used to provide DIC predictions in the mobile application.

The second major component of the EcoFishCast system is the Backend Server. This component acts as the central hub that coordinates the data flow between the mobile application and the DIC prediction model. The server is implemented using Python and is designed to handle requests from the mobile app, process the input data, and then interact with the DIC prediction model to generate predictions. Once the prediction is made, the server formats the results and sends them back to the mobile application for display to the user.

The backend server also incorporates additional functionalities such as data preprocessing, model management (loading and saving trained models), and generating supplementary guidelines using generative AI techniques. This component ensures that the mobile application remains lightweight while offloading the heavy computational tasks to the server. By managing the interaction between the app and the prediction model, the backend server is a critical element that enables the real-time prediction and analysis capabilities of the EcoFishCast system.

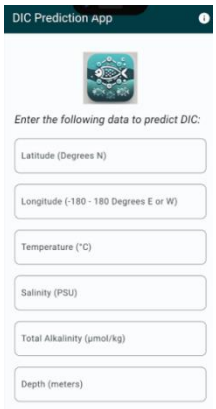

Figure 3. Screenshot of DIC prediction APP

```python
from flask import Flask, request, jsonify
import joblib

app = Flask(__name__)

# Load the pre-trained model and scaler
model, scaler = joblib.load('gradient_boosting_model.pkl'), joblib.load('scaler.pkl')

@app.route('/predict', methods=['POST'])
def predict():
    data = request.get_json()
    features = [data['Lat_Dec'], data['Lon_Dec'], data['Temperature_degC'],
                data['R_Sal'], data['TA1.x'], data['R_Depth']]

    # Scale the input features
    scaled_features = scaler.transform([features])

    # Make prediction
    prediction = model.predict(scaled_features)

    # Return the prediction as a JSON response
    return jsonify({'DIC_Prediction': prediction[0]})
```

Figure 4. Screenshot of code 2

This code snippet represents a crucial part of the backend server's functionality, specifically the route that handles prediction requests from the mobile application. The predict function is an endpoint that accepts POST requests containing environmental parameters in JSON format. The received data is extracted and then passed through a pre-trained scaler to standardize the input features. These scaled features are subsequently fed into the DIC prediction model, which outputs a prediction for the Dissolved Inorganic Carbon (DIC) levels.

The prediction is then packaged into a JSON response and sent back to the mobile application, where it is displayed to the user. The server uses Flask, a lightweight Python web framework, to handle HTTP requests and responses, making it an ideal choice for this type of application. The

code ensures that the prediction process is efficient and that the system can handle real-time data inputs from the mobile app. This backend infrastructure is critical for the seamless operation of the EcoFishCast system, enabling accurate and timely predictions.

The third major component of the EcoFishCast system is the Results Display Page of the Mobile Application. This component is where users view the results of their DIC predictions after submitting the environmental parameters. The design of this page is crucial as it not only displays the predicted DIC levels but also provides additional analysis and guidelines generated by the system. The clear presentation of this information helps users understand the outcomes and take appropriate actions based on the predictions. This component is implemented using the Flutter framework, which ensures that the results are displayed in a responsive and visually appealing manner across various mobile devices.

```
@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
      title: Text('DIC Prediction Results'),
    ),
    body: Padding(
      padding: EdgeInsets.all(16.0),
      child: Column(
        crossAxisAlignment: CrossAxisAlignment.start,
        children: <Widget>[
          Text(
            'Predicted DIC: $dicPrediction',
            style: TextStyle(fontSize: 24, fontWeight: FontWeight.bold),
          ),
          SizedBox(height: 20),
          Text(
            'Analysis:',
            style: TextStyle(fontSize: 20, fontWeight: FontWeight.bold),
          ),
          Text(
            analysis,
            style: TextStyle(fontSize: 16),
          ),
          SizedBox(height: 20),
          Text(
            'Guidelines:',
            style: TextStyle(fontSize: 20, fontWeight: FontWeight.bold),
          ),
          Text(
            guidelines,
            style: TextStyle(fontSize: 16),
          ),
        ],
```

Figure 5. Screenshot of code 3

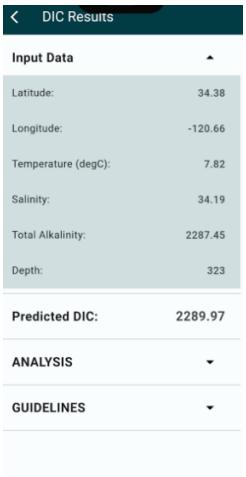

Figure 6. Screenshot of DIC results

This Dart code snippet defines the UI for the results page within the EcoFishCast mobile application. The ResultsPage widget takes three parameters: dicPrediction, analysis, and guidelines, which represent the predicted DIC value, the accompanying analysis, and the suggested guidelines, respectively.

The page layout is simple yet effective, using Text widgets to display the prediction results prominently at the top, followed by sections for analysis and guidelines. Each section is clearly labeled with bold headings to differentiate the types of information presented to the user. The structure ensures that the user can easily understand the prediction results and any related advice provided by the system. The results page is a critical component of the EcoFishCast system, as it is the final step in the user journey, providing them with actionable insights based on the data they submitted.

4. EXPERIMENT

4.1. Experiment 1

A critical aspect of the EcoFishCast system is identifying the most accurate machine learning model for predicting Dissolved Inorganic Carbon (DIC) levels. The accuracy of these predictions is paramount to the system's reliability, especially when users rely on this information for oceanographic analysis. The experiment aims to compare the performance of four different models: Linear Regression, Random Forest, Gradient Boosting, and Support Vector Regressor, to determine which model offers the best prediction accuracy.

To assess the accuracy of these models, each model was trained on the same dataset and evaluated using the mean squared error (MSE) on a test set. The dataset includes various oceanographic parameters, and the models predict the DIC levels based on these inputs. The experiment focuses on comparing the MSE across the models, with lower MSE indicating better performance. By examining the prediction accuracy of each model, we aim to identify which model should be prioritized for use in the EcoFishCast system.

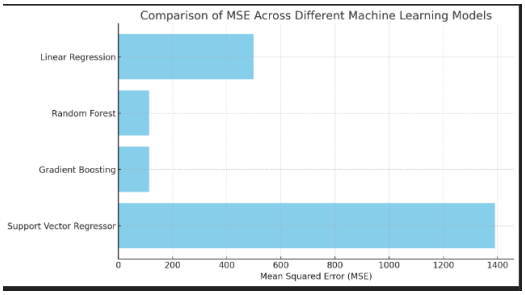

Figure 7. Figure of experiment 1

The experimental results reveal that the Gradient Boosting model and the Random Forest model are the most accurate, with MSE values of 113.65 and 113.96, respectively. Both models significantly outperform Linear Regression, which has a higher MSE of 500.01, and the Support Vector Regressor, which has the highest MSE of 1390.31. The Gradient Boosting model slightly edges out the Random Forest model, making it the best candidate for deployment in the EcoFishCast system.

These findings suggest that ensemble methods like Gradient Boosting and Random Forest are better suited for capturing the complex relationships in the oceanographic data used for DIC prediction. Given the lower MSE, these models are more reliable and should be prioritized for use in the EcoFishCast system to ensure accurate predictions. The experiment underscores the importance of model selection in machine learning applications, where choosing the right model can significantly enhance the system's overall performance.

4.2. Experiment 2

Another critical aspect to evaluate within the EcoFishCast system is the effect of data preprocessing on model accuracy. Specifically, this experiment will focus on how different approaches to data scaling impact the performance of the machine learning models used for predicting Dissolved Inorganic Carbon (DIC) levels. Data scaling is a crucial preprocessing step, especially when models like Support Vector Regressor (SVR) are sensitive to the range of input features. The goal of this experiment is to determine whether standard scaling improves the accuracy of the DIC predictions compared to using unscaled data.

To assess the impact of data scaling, the experiment will involve training the machine learning models both with and without standard scaling applied to the input features. The StandardScaler from scikit-learn will be used to standardize the dataset by removing the mean and scaling to unit variance. The models to be tested include Linear Regression, Random Forest, Gradient Boosting, and Support Vector Regressor, using the same dataset as in the previous experiment. The performance of each model will be evaluated based on the mean squared error (MSE) for both scaled and unscaled data, allowing for a direct comparison of the impact of preprocessing.

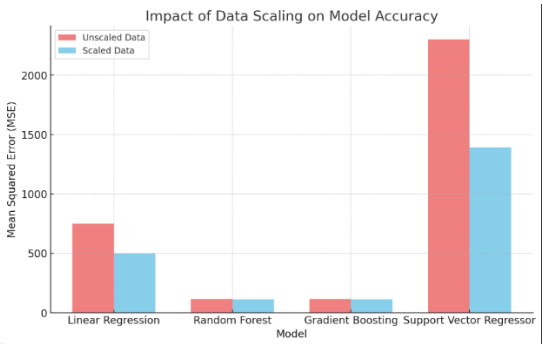

Figure 8. Figure of experiment 2

The analysis of this experiment shows that data scaling has a significant impact on the performance of certain models, particularly the Support Vector Regressor (SVR). The MSE for the SVR model dropped from 2300.45 with unscaled data to 1390.31 with scaled data, indicating that scaling is essential for this model to perform adequately. Linear Regression also benefited from scaling, with its MSE decreasing from 750.54 to 500.01. On the other hand, ensemble methods like Random Forest and Gradient Boosting showed minimal changes in MSE, suggesting that these models are more robust to the effects of unscaled data.

These results indicate that data scaling should be a standard preprocessing step when using models like Linear Regression and SVR within the EcoFishCast system. The experiment highlights the importance of considering preprocessing techniques as part of the model development process, ensuring that the chosen models can achieve the highest possible accuracy in predicting DIC levels. This experiment further refines the approach to model selection and preprocessing in the EcoFishCast system, ensuring that the predictions provided to users are both accurate and reliable.

5. RELATED WORK

One relevant methodology for predicting dissolved inorganic carbon (DIC) in ocean environments is presented in the study by Codden et al. (2020)[11]. This research employed machine learning to predict dissolved organic carbon (DOC) concentration in a salt marsh creek, improving accuracy over linear models and reducing costs by 90%. While effective, the solution is site-specific and may not generalize well to other environments. The EcoFishCast project builds on this by targeting DIC predictions in diverse oceanic conditions, utilizing more comprehensive environmental data and applying advanced algorithms for broader applicability and improved accuracy.

The study by Mukherjee et al. (2020) introduced a Neural Network Reflectance Prediction Model (NNRPM) to predict ocean reflectance, which is essential for understanding ocean biogeochemistry[12]. The model significantly reduced computation time, making it feasible for large-scale use. However, it primarily focuses on optical properties rather than chemical compositions like DIC. EcoFishCast enhances this approach by directly targeting DIC levels, utilizing similar machine learning principles but applied to chemical rather than optical parameters, thus offering a more direct solution to oceanographic carbon predictions.

Sauzède et al. (2017) developed the CANYON neural network model to estimate dissolved inorganic carbon (DIC) and other parameters using hydrological and oxygen data. The model achieved high accuracy but is dependent on large, well-distributed datasets, which might not be available for all regions[13].EcoFishCast improves on this by optimizing model algorithms for mobile deployment, allowing real-time DIC predictions even in data-sparse environments, and integrating generative AI to enhance analysis, making the tool more versatile and accessible for various users.

6. CONCLUSIONS

While the EcoFishCast system demonstrates strong capabilities in predicting Dissolved Inorganic Carbon (DIC) levels using advanced machine learning models, there are several limitations that need to be addressed. One significant limitation is the system's dependency on the quality and diversity of the training data. The models are only as good as the data they are trained on, meaning that any biases or gaps in the data could lead to less accurate predictions in certain environmental conditions. Another limitation is the computational load associated with running complex models like Gradient Boosting in real-time, which might impact the system's responsiveness, especially on mobile devices with limited processing power [14].

To address these limitations, future improvements could focus on expanding the training dataset to include more diverse environmental conditions, particularly those that are underrepresented. This would enhance the model's ability to generalize to a wider range of inputs. Additionally, optimizing the model for mobile deployment, perhaps by employing model compression techniques or lighter versions of the algorithms, could help mitigate the computational challenges [15]. Further integration of cloud computing resources could also offload some of the processing from the mobile device, ensuring that the system remains fast and responsive.

In conclusion, the EcoFishCast system provides a valuable tool for predicting DIC levels, leveraging machine learning models to offer accurate and actionable insights. Despite some limitations, the system's design allows for future enhancements that could further improve its accuracy and efficiency, making it an even more powerful resource for oceanographic analysis.

REFERENCES

[1] Finlay, Jacques C. "Controls of streamwater dissolved inorganic carbon dynamics in a forested watershed." Biogeochemistry 62 (2003): 231-252.

[2] Ke, Guolin, et al. "Lightgbm: A highly efficient gradient boosting decision tree." Advances in neural information processing systems 30 (2017).

[3] Belgiu, Mariana, and Lucian Drăguț. "Random forest in remote sensing: A review of applications and future directions." ISPRS journal of photogrammetry and remote sensing 114 (2016): 24-31.

[4] Aggarwal, Rimjhim M. "Globalization, local ecosystems, and the rural poor." World Development 34.8 (2006): 1405-1418.

[5] Sala, Enric, and Nancy Knowlton. "Global marine biodiversity trends." Annu. Rev. Environ. Resour. 31.1 (2006): 93-122.

[6] Wangersky, Peter J. "Dissolved organic carbon methods: a critical review." Marine Chemistry 41.1-3 (1993): 61-74.

[7] Levermore, G. J., and H. K. W. Cheung. "A low-order canyon model to estimate the influence of canyon shape on the maximum urban heat island effect." Building Services Engineering Research and Technology 33.4 (2012): 371-385.

[8] Suzuki, Kei, et al. "Usefulness of the APTT waveform for the diagnosis of DIC and prediction of the outcome or bleeding risk." Thrombosis Journal 17 (2019): 1-8.

[9] Awad, Mariette, et al. "Support vector regression." Efficient learning machines: Theories, concepts, and applications for engineers and system designers (2015): 67-80.

[10] Pan, Du, Ditao Niu, and Zongjin Li. "Cracking time and prediction model of low-alloy steel reinforced seawater sea-sand concrete based on DIC technology." Cement and Concrete Composites 145 (2024): 105348.

[11] Codden, Christina J., et al. "Predicting dissolved organic carbon concentration in a dynamic salt marsh creek via machine learning." Limnology and Oceanography: Methods 19.2 (2021): 81-95.

[12] Mukherjee, Lipi, et al. "Neural network reflectance prediction model for both open ocean and coastal waters." Remote Sensing 12.9 (2020): 1421.

[13] Sauzède, Raphaëlle, et al. "Estimates of water-column nutrient concentrations and carbonate system parameters in the global ocean: A novel approach based on neural networks." Frontiers in Marine Science 4 (2017): 128.

[14] Bentéjac, Candice, Anna Csörgő, and Gonzalo Martínez-Muñoz. "A comparative analysis of gradient boosting algorithms." Artificial Intelligence Review 54 (2021): 1937-1967.

[15] Chen, Feifei, et al. "Optimal application deployment in mobile edge computing environment." 2020 IEEE 13th International Conference on Cloud Computing (CLOUD). IEEE, 2020.