**Final Year Project Description**

**Title:** Predictive Trash Debris Tracking and Marine Conservation Using AI

**Problem Description**

The increasing accumulation of trash debris in the marine environment poses a severe threat to both marine life and human activities. Traditional methods for cleaning up these areas are resource-intensive and are not proactive in preventing the accumulation of waste. The lack of real-time data and predictive analytics also hampers conservation efforts, affecting areas like coral preservation and diving experiences.

**Project Overview**

This project aims to develop an Artificial Intelligence (AI)-based model to transform trash debris cleanup, coral preservation, and diving experiences. The central focus is to build a predictive model that leverages ocean current and weather data to forecast the presence and movement of trash debris in bays and potential dive sites around the Maltese Islands. The system will offer real-time updates and actionable insights to facilitate collaborative marine conservation efforts.

**Objectives**

**Data Integration:** To collect, preprocess, and integrate multiple datasets, including weather and ocean current data, into a unified model.

**Model Development:** To build and validate an AI-based predictive model capable of accurately forecasting and predicting the presence and movement of trash debris.

**Real-Time Updates:** To implement a web-based platform that delivers real-time updates from the predictive model, enhancing underwater exploration and marine conservation efforts.

**User Engagement:** To allow for user-generated data collection, where divers and other people can report debris locations, thereby refining the model's predictions over time.

**Sustainability:** To foster a proactive approach towards marine conservation, providing the necessary insights to both governmental and non-governmental organizations involved in these efforts.

**Data Accounting Report for Trash Debris Prediction FYP**

This report provides an overview of the available data sets, including their variables and characteristics.

**Data Sources**

**1. Ocean Current Data (NetCDF File)**

**File Name:** *med-cmcc-cur-rean-h\_1694161199370.nc*

**Variables:**

time: Time stamps

lat: Latitude

lon: Longitude

uo: Eastern Ocean Current Velocity

vo: Northern Ocean Current Velocity

**Format:** NetCDF (Network Common Data Form)

Purpose: To provide ocean current information

**2. Weather Data (Excel File)**

**File Name:** (8) August 2023.xlsx

**Variables:**

Date: Day of the month

Maximum Temperature (°C)

Minimum Temperature (°C)

Mean Temperature (°C)

Mean Wind (km/h)

Mean Gust (km/h)

Mean Wind and Gust (km/h)

Highest Gust (km/h)

Dominant Direction (°)

Rainfall (mm)

**Format:** Excel

Purpose: To provide weather information

**Data Characteristics**

**1. Ocean Current Data**

Dimensionality: 3-dimensional (time x latitude x longitude)

Temporal Coverage: Records over 3 months (1st Jan 2021 – 31st Mar 2021)

Time Stamp: Every 1 hour

Spatial Coverage:

*Longitude Domain: 13.916667 to 14.791667*

*Latitude Domain: 35.604168 to 36.3125*

**2. Weather Data**

Dimensionality: 2D (time x various weather metrics)

Temporal Coverage: August 2023

Time Stamp: Daily Average

Spatial Coverage: Single Point

**Model**

Given the characteristics of the data, **a Lagrangian dispersion model** might be a more suitable choice for several reasons:

Temporal Granularity: The ocean current data comes in hourly intervals, which would allow for more detailed tracking of individual "particles" (i.e., pieces of trash) as they move with the current.

Spatial Coverage: The ocean current data has specific latitude and longitude information, making it easier to track particles in a Lagrangian framework.

Variable Metrics: With a Lagrangian model, it would be easier to incorporate multiple types of data, like the weather and ocean current data, into the movement equations for each particle.

**Location of Particles**

Random Locations: For a more general study of how trash moves within a larger area, I can randomly distribute particles within that area.

Data-driven Locations: Observe data showing where trash is commonly found, I could use this to set the initial conditions.

Existing Studies: I can also initialize particles based on the results of previous studies or models that have investigated similar phenomena.

Uniform Grid: For a comprehensive study, particles could be initialized at all points in a uniform grid covering the area of interest. This would require significantly more computational resources but would provide a detailed picture of particle dispersion.

Combination: I could also use a combination of the above methods to set initial particle locations.

**Libraries:**

**OceanParcels**: A Python library designed for simulating the movement of particles in ocean currents.

**OpenDrift**: A generic framework for modelling the drift of objects or substances (like oil spills, plastic, etc.) in the ocean.

**Sections for the model:**

1. Data Preparation

Importing necessary libraries and modules

Loading data files, such as ocean current data and weather data

2. Preprocessing

Data cleaning, filtering, and possibly transformation

Interpolation of data points to create a smoother field for ocean currents

3. Initialization

Setting up initial conditions for the particles, like their starting coordinates

Initializing variables and constants used in the simulation

4. Core Advection Scheme

The central differential equation used for advection

Phenomena affecting particle speed (e.g., currents, winds)

5. Interpolation

Method to find and use the closest current and wind speeds to a particle's position

6. Eddy Diffusivity and Random Walk

Accounting for small-scale phenomena that are too small to model

Implementing a random walk to simulate eddy diffusivity

7. Time Evolution and Trajectory Calculation

Repeating the process to compute a series of positions for each particle

8. Visualization and Analysis

Producing concentration maps and other visualizations

Analysing the paths of particles and identifying garbage patches

**Where can AI be incorporated?**

Parameter Optimization: I can use machine learning algorithms to optimize various parameters in the model, such as the coefficients in the advection equation or the parameters governing the random walk for eddy diffusivity.

Data-Driven Advective Scheme: Use machine learning models to predict the advection velocities (*uo* and *vo*) based on historical data. This way, instead of interpolating the velocities from the datasets, we will be predicting them using a trained model.

Anomaly Detection: Use machine learning to identify anomalies or unexpected patterns in the dispersion of particles. For example, if particles are gathering in an area where the model predicts they shouldn't.

Predictive Modelling: Train machine learning models on the outputs of the Lagrangian model to make quicker, perhaps less accurate, predictions. This could be useful for real-time decision-making.

Data Preprocessing: Machine learning algorithms can be used to clean and preprocess the data more effectively, especially if there’s missing or unreliable data points.

Weather Prediction Component:

Incorporate a machine learning model that predicts future weather conditions based on past data. This predicted weather data could then be used in the advection calculations. This would make the model more dynamic and capable of "forecasting" particle movements based on predicted future weather conditions.

**Steps to Integrate Machine Learning into the Model:**

1. Data Preparation and Preprocessing

1.1 Load ocean current data

- Read ocean current data from NetCDF or similar file formats.

1.2 Load historical weather data

- Read historical weather data from Excel, CSV, or other sources.

- Extract relevant weather variables like wind speed, wind direction, etc.

1.3 Preprocess and clean the data

- Handle missing values, outliers, and other anomalies.

- Normalize or standardize the data as required.

1.4 Feature engineering for weather data

- Create additional features that might be useful for machine learning, like moving averages of wind speed, etc.

2. Machine Learning Model for Weather Prediction

2.1 Select an appropriate ML model

- Choose a model suitable for time-series data, such as LSTM, ARIMA, or Prophet.

2.2 Train the ML model

- Split the historical weather data into training and test sets.

- Train the model on the training set.

2.3 Evaluate the model

- Use the test set to evaluate the model's performance metrics like RMSE, MAE, etc.

2.4 Model tuning

- If the model's performance is not satisfactory, tune its hyperparameters or consider using a different model.

3. Initialization of Physics-based Model

3.1 Initialize particles

- Generate initial coordinates for particles based on specified or random locations.

3.2 Initialize variables and constants

- Set up any other initial conditions, parameters, or constants needed for the simulation.

4. Core Advection Scheme with AI Component

4.1 For each time step:

- 4.1.1 Use ML model to predict future weather

- Input current and past weather conditions into the ML model to forecast future conditions.

- 4.1.2 Calculate advection

- Use both ocean currents and predicted weather conditions to calculate particle advection.

- 4.1.3 Incorporate uncertainty

- If the ML model provides confidence intervals, incorporate this as an uncertainty measure in the advection calculations.

5. Interpolation, Eddy Diffusivity and Random Walk

5.1 For each particle:

- Apply a random walk algorithm to simulate the effect of eddy diffusivity on particle movement.

6. Time Evolution and Trajectory Calculation

6.1 Update particle positions

- Use the calculated advection and random walk to update each particle's position.

6.2 Record positions

- Store the new positions for later analysis and visualization.

7. Visualization and Analysis

7.1 Produce visualizations

- Generate concentration maps or other relevant visualizations.

7.2 Analysis

- Identify areas where there is a high probability of debris accumulation.

8. Evaluation and Iteration

8.1 Assess performance

- Evaluate how well the combined model performs in terms of accuracy and computational efficiency

8.2 Re-train and fine-tune

- As more data becomes available, re-train the ML model and make any necessary adjustments to the physics-based model.

**Questions about model**

How should the datasets be combined?

Where should the initial particles be placed?

How should the model predict per timestamp (example every day)?

What is the best AI feature to incorporate into this model?

How will the AI feature work with the model?

How should the weather model be?

**Weather Forecasting Model and Integration with Lagrangian Model**

1. Function Weather\_Forecast\_Model(training\_data):

1.1 Preprocess training\_data

1.2 Perform feature engineering

1.3 Split data into train and test sets

1.4 Select appropriate ML model // Could be LSTM, ARIMA, etc.

1.5 Train the model on train set

1.6 Evaluate the model on test set

1.7 Return trained model

2. Function Get\_Weather\_Prediction(trained\_model, current\_weather\_data):

2.1 Use trained\_model to predict future weather conditions from current\_weather\_data

2.2 Return predicted\_weather\_data

1. Function Lagrangian\_Model(ocean\_current\_data, initial\_particle\_positions, trained\_weather\_model):

1.1 Initialize particles with initial\_particle\_positions

1.2 Initialize variables and constants for simulation

// Loop over each time step

1.3 For each time\_step in simulation\_duration:

// Get Weather Prediction

1.3.1 predicted\_weather\_data = Get\_Weather\_Prediction(trained\_weather\_model, current\_weather\_data)

// Update Advection Scheme

1.3.2 For each particle in particles:

1.3.2.1 ocean\_current\_velocity = Get\_Ocean\_Current\_Velocity(ocean\_current\_data, particle.position)

1.3.2.2 predicted\_wind\_velocity = Get\_Wind\_Velocity(predicted\_weather\_data, particle.position)

1.3.2.3 advection\_velocity = α \* ocean\_current\_velocity + β \* predicted\_wind\_velocity

// Optionally, incorporate uncertainty

1.3.2.4 If model\_provides\_confidence\_intervals:

1.3.2.4.1 Integrate confidence\_intervals into advection\_velocity

// Update Particle Position

1.3.2.5 particle.position += advection\_velocity \* time\_step\_duration

// Eddy Diffusivity and Random Walk

1.3.3 For each particle in particles:

1.3.3.1 Apply random\_walk to particle.position

// Update particle positions for analysis

1.3.4 Record particle positions

1.4 Return final\_particle\_positions

The equation Vplastic = Vcurrent + αVwind + Veddy diffusivity describes how the velocity of a plastic particle in the water is influenced by multiple factors. The equation is derived from fluid dynamics and is commonly used in models that simulate the dispersion of particles in fluid flows.

1. Vcurrent is the velocity due to the ocean current. This is the base velocity of the particle as it is carried along by the water.

2. Vwind is the velocity imparted by the wind. Wind can push the plastic particles along the water surface and influence their speed.

3. Veddy diffusivity accounts for small-scale phenomena, like eddies and turbulent flows, that also contribute to the particle's velocity but are too detailed for the model to simulate directly.

The variable α is a coefficient that weighs the influence of the wind velocity on the plastic particle. It's essentially a scaling factor that allows you to adjust how much the wind contributes to the overall velocity of the plastic. The value of α could be determined empirically or through other modelling approaches.

So, the equation integrates these three major factors to give an overall velocity for the plastic particle at a given time. This velocity is then used to update the particle's position as it moves through the water.

Calculating α

* Empirical Analysis: If you have observational data that shows how plastic particles move under different wind conditions, you can use statistical methods to estimate.
* Sensitivity Analysis: You can run your model with different values of α and see which one produces results that best match observed or expected behaviours. This is useful if you have some ground truth data to compare with.
* Literature Review: Sometimes, values for parameters like α are available in scientific literature, based on empirical studies or other modelling work.
* Machine Learning: More complex but potentially more accurate, you can use machine learning techniques to train a model that can estimate α based on features of the system you are modelling. This would require a dataset where α values are known or can be accurately inferred.

A screenshot of a computer

Description automatically generated

A black background with white text

Description automatically generated

**Literature Review**

**Lagrangian Model**

1. **Modelling plastics in the ocean, principles of dispersion modelling -**<https://theoceancleanup.com/updates/forecasting-ocean-plastic-around-the-globe-a-deep-dive-into-modeling-the-garbage-patches/>
2. **OpenDrift -** <https://opendrift.github.io>
3. **OceanParcels -** <https://oceanparcels.org>
4. **Gibraltar Strait PARticle Tracking model -** <https://personal.us.es/rperianez/gispart.htm>
5. **Lagrangian dispersion models -** <https://personal.us.es/rperianez/lesson7.pdf>
6. **An Overview of the Lagrangian Dispersion Modeling of Heavy Particles in Homogeneous Isotropic Turbulence and Considerations on Related LES Simulations** **-** <https://www.mdpi.com/2311-5521/6/4/145>
7. **Efficiently simulating Lagrangian particles in large-scale ocean flows — Data structures and their impact on geophysical applications -** <https://www.sciencedirect.com/science/article/pii/S0098300423000262?via%3Dihub>
8. **(Chapter 8) Lagrangian Dispersion Models -** <https://link.springer.com/book/10.1007/978-1-4757-4465-1>
9. **A Lagrangian Particle Dispersion Model Compatible with WRF -** <https://www.researchgate.net/publication/267994680_A_Lagrangian_Particle_Dispersion_Model_Compatible_with_WRF>
10. **The Lagrangian particle dispersion model FLEXPART version 10.4 -** <https://gmd.copernicus.org/articles/12/4955/2019/>
11. **SIMULATION OF URBAN-SCALE DISPERSION USING A LAGRANGIAN STOCHASTIC DISPERSION MODEL -** <https://link.springer.com/article/10.1023/A:1018973813500>
12. **The Lagrangian particle dispersion model FLEXPART-WRF version 3.1 -** <https://gmd.copernicus.org/articles/6/1889/2013/>
13. **A New High Performance Version of the Lagrangian Particle Dispersion Model Spray, Some Case Studies -** <https://link.springer.com/chapter/10.1007/978-1-4615-4153-0_51>
14. **A lagrangian dispersion model for calculating concentration distribution within a built-up domain-** <https://www.sciencedirect.com/science/article/abs/pii/1352231096001446>
15. **The Parcels v2.0 Lagrangian framework: new field interpolation schemes -** <https://www.sciencedirect.com/science/article/abs/pii/1352231096001446>
16. **Parcels v0.9: prototyping a Lagrangian Ocean Analysis framework for the petascale age -** <https://gmd.copernicus.org/articles/10/4175/2017/>

**Weather Forecasting Model**

1. **Modelling plastics in the ocean, principles of dispersion modelling -**<https://theoceancleanup.com/updates/forecasting-ocean-plastic-around-the-globe-a-deep-dive-into-modeling-the-garbage-patches/>
2. **OpenDrift -** <https://opendrift.github.io>
3. **OceanParcels -** <https://oceanparcels.org>
4. **Gibraltar Strait PARticle Tracking model -** <https://personal.us.es/rperianez/gispart.htm>
5. **Lagrangian dispersion models -** <https://personal.us.es/rperianez/lesson7.pdf>
6. **An Overview of the Lagrangian Dispersion Modeling of Heavy Particles in Homogeneous Isotropic Turbulence and Considerations on Related LES Simulations** **-** <https://www.mdpi.com/2311-5521/6/4/145>