Configuration Manual

MSc Research Project
Programme Name

Shamaie Maria Alleyne
Student ID: 24114570

School of Computing
National College of Ireland

Supervisor: Dr. Adbul Razzaq

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Shamaie Maria Alleyne
Student ID:	24114570
Programme:	Programme Name
Year:	2025
Module:	MSc Research Project
Supervisor:	Dr. Adbul Razzaq
Submission Due Date:	11/12/2025
Project Title:	Configuration Manual
Word Count:	933
Page Count:	14

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	Shamaie M. Alleyne
Date:	10th December 2025

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Configuration Manual

Shamaie Maria Alleyne
24114570

1 Google Earth Engine Data Acquisition

This section describes the configuration required to replicate the satellite imagery acquisition from Google Earth Engine (GEE) for the Dublin Bay study area.

1.1 GEE Platform Access

- **Platform:** Google Earth Engine Code Editor (`https://code.earthengine.google.com/`)
- **Account Requirement:** Valid Google Earth Engine account with approved access
- **API Version:** Earth Engine JavaScript API

1.2 Study Area Configuration

The Dublin Bay Area of Interest is defined using the following geographic boundaries with Coordinate Reference System set to WGS84 (EPSG:4326).

Listing 1: Dublin Bay Definition

```
var dublinBayGeoJson = {
    "type": "Polygon",
    "coordinates": [
        [
            [-6.281064699981442, 53.406396313634474],
            [-6.281064699981442, 53.205126344230905],
            [-6.031566544488584, 53.205126344230905],
            [-6.031566544488584, 53.406396313634474],
            [-6.281064699981442, 53.406396313634474],...
        ]
    ]
};
```

1.3 Sentinel-2 Data Collection

Table 1: Sentinel-2 Collection Configuration

Parameter	**Configuration Value**	**Rationale**
Satellite Collection	`COPERNICUS/S2_SR_HARMONIZED`	Surface Reflectance with harmonized processing
Temporal Range	2017-01-01 to 2025-12-31	Aligned with Sentinel-2 availability
Temporal Resolution	Quarterly composites	Balances detail with data quality
Cloud Filter	`CLOUDY_PIXEL_PERCENTAGE < 10%`	Ensures high-quality imagery
Compositing Method	Median composite per quarter	Reduces cloud contamination
Spatial Resolution	10 meters	Native resolution for selected bands

The following 8 Sentinel-2 bands were exported for analysis:

Table 2: Sentinel-2 Band Configuration

Band Name	**Wavelength (nm)**	**Resolution (m)**	**Purpose**
B1 (Coastal Aerosol)	443	60 (resampled to 10)	Atmospheric correction
B2 (Blue)	490	10	Water body detection
B3 (Green)	560	10	NDWI calculation
B4 (Red)	665	10	Vegetation/water discrimination
B8 (NIR)	842	10	NDWI calculation, land/water boundary
B8A (Narrow NIR)	865	20 (resampled to 10)	Enhanced vegetation analysis
B11 (SWIR 1)	1610	20 (resampled to 10)	Moisture content detection
B12 (SWIR 2)	2190	20 (resampled to 10)	Soil/vegetation discrimination

1.4 Transect Configuration

- Number of Transects:159 shore-perpendicular transects
- Transect Length: 2000 meters
- Transect Buffer: 150 meters radius per transect point
- Transect Orientation: Perpendicular to coastline, derived from adjacent point geometry
- Coordinate System for Distance Calculation: UTM Zone 29N (EPSG:32629)

1.5 Export Settings

- **Output Format:** GeoTIFF (8-band multi-band raster)
- **Destination:** Google Drive folder `Dublin_Bay_Quarterly`
- **Naming Convention:** `Dublin_S2_[YEAR]_Q[1-4]_[Season].tif`
- **Maximum Pixels:** 1×10^{13} (prevents truncation of large exports)

Listing 2: GEE Export Configuration

```
Export.image.toDrive({
    image: bands.clip(dublinBay),
    description: 'Dublin_S2_YYYY_QX',
    folder: 'Dublin_Bay_Quarterly',
    region: dublinBay,
    scale: 10,
    crs: 'EPSG:4326',
    maxPixels: 1e13
});
```

1.6 Shoreline Extraction Methodology

Shorelines are extracted from quarterly imagery using the Normalized Difference Water Index (NDWI) Formula:

$$\mathrm{NDWI} = \frac{\mathrm{Green} - \mathrm{NIR}}{\mathrm{Green} + \mathrm{NIR}} \tag{1}$$

Threshold: NDWI > 0 classified as water post-processing:

- Binary opening (2 iterations) to remove noise
- Binary closing (2 iterations) to fill gaps
- Minimum segment length filter: 100 meters
- Output format: GeoJSON LineString geometries

2 Python Environment and Software Dependencies

Python Version: 3.8+ (tested on 3.10)
Package Manager: Conda or pip

2.1 Required Python Libraries

2.1.1 Geospatial and GIS Libraries

Table 3: Required Geospatial Libraries

Library	Version	Purpose
`rasterio`	Latest	Raster data I/O (GeoTIFF processing)
`geopandas`	Latest	Vector data manipulation
`shapely`	Latest	Geometric operations
`pyproj`	Latest	Coordinate transformations
`fiona`	Latest	Vector file I/O backend

2.1.2 Machine Learning Libraries

Table 4: Required Machine Learning Libraries

Library	Version	Purpose	Configuration Notes
`lightgbm`	Latest	Gradient boosting (Model A & C)	Requires OpenMP for parallel training
`scikit-learn`	1.0+	Preprocessing, metrics	Used for scalers, encoders
`tensorflow`	2.10+	Deep learning (Model B)	GPU support recommended
`keras`	Bundled	High-level neural network API	—
`torch`	2.0+	PyTorch for MAE pretraining	CUDA 11.8+ required
`torchvision`	Compatible	Image transformations	—
`timm`	Latest	Pre-trained vision models	Used for ConvNeXt

2.1.3 Data Processing Libraries

Table 5: Required Data Processing Libraries

Library	Version	Purpose
`numpy`	1.23+	Numerical computation
`pandas`	1.5+	Tabular data manipulation
`scipy`	Latest	Statistical functions

2.1.4 Visualization Libraries

Table 6: Required Visualization Libraries

Library	Version	Purpose
`matplotlib`	Latest	Static plotting
`seaborn`	Latest	Statistical visualizations
`plotly`	Latest	Interactive visualizations

2.1.5 Utility Libraries

Table 7: Required Utility Libraries

Library	Version	Purpose
`joblib`	Latest	Model serialization
`tqdm`	Latest	Progress bars
`warnings`	Standard library	Suppress warnings

2.2 PyTorch Device Configuration

```
DEVICE = "cuda" if torch.cuda.is_available() else "cpu"
```

2.3 Random Seed Configuration

For reproducibility, the following seeds are set:

```
SEED = 42
np.random.seed(SEED)
torch.manual_seed(SEED)
tf.random.set_seed(SEED)
```

3 Model Architecture and Configuration

3.1 Phase 1: Masked Autoencoder (MAE) Pretraining

3.1.1 LICS Dataset for Self-Supervised Pretraining

The MAE pretraining phase utilizes the Landsat Irish Coastal Segmentation (LICS) Dataset v2 (O'Sullivan and Dev; 2024), a specialized coastal imagery dataset designed for deep learning-based water body segmentation.

Dataset Specifications

Table 8: LICS Dataset Specifications

Attribute	Details
Dataset Name	Landsat Irish Coastal Segmentation (LICS) Dataset
Source	Zenodo (DOI: 10.5281/zenodo.8414665)
Creators	Conor O'Sullivan (UCD), Soumyabrata Dev (ADAPT SFI)
Publication Date	October 6, 2023 (Version 0.1)
License	Creative Commons Attribution 4.0 International
Purpose	Aid development of DL methods for coastal segmentation

Training Set Configuration:

- **File:** `training.zip` (16.8 GB)
- **Number of Patches:** 30,000 image chips
- **Channels:** 9 channels per image
  - Channels 1-7: Landsat spectral bands
  - Channel 8: Landsat Quality Assessment (QA) band
  - Channel 9: Rough training annotation (segmentation mask)
- **Image Format:** NumPy arrays (`.npy` files)
- **Data Type:** 16-bit unsigned integer (normalized to float32)

Fine-tuning Set Configuration

- **File:** `finetune.zip` (subset from `test.zip`)
- **Number of Patches:** 100 image chips
- **Channels:** 11 channels per image
  - Channels 1-7: Landsat spectral bands
  - Channel 8: Landsat QA band
  - Channel 9: Rough training annotation
  - Channel 10: Test mask annotation (high-quality segmentation)
  - Channel 11: Test edge annotation (shoreline boundaries)

Geographic Coverage: Irish coastline which aligns with Dublin Bay study area preprocessing techniques (O'Sullivan et al.; 2024).

```
# Normalization to [0, 1] range
img = img.astype(np.float32) / 65535.0

# Channel selection for MAE training
img = arr[:, :, 0:7]  # Spectral bands only
qa = arr[:, :, 7:8]   # QA band (optional)
img_8ch = np.concatenate([img, qa], axis=2)  # 8-channel input
```

3.1.2 Architecture Configuration

Table 9: MAE Architecture Parameters

Parameter	Value	Description
Input Channels	8	Sentinel-2 spectral bands
Image Size	256×256 pixels	Resized during preprocessing
Patch Size	16×16 pixels	Tokenization grid
Embedding Dimension	256	Transformer hidden dimension
Encoder Layers	6	Transformer encoder depth
Attention Heads	8	Multi-head attention
Feed-Forward Dimension	1024	4× embedding dimension
Mask Ratio	0.75	75% of patches masked

3.1.3 Training Configuration

```
EPOCHS = 20
BATCH_SIZE = 32
LEARNING_RATE = 1e-4
OPTIMIZER = "Adam"
LOSS_FUNCTION = "MSE (on masked patches only)"
```

Data Preprocessing

- Normalization: Divide by 65535 (16-bit to float32)
- Data Augmentation: None (self-supervised reconstruction)

3.2 Phase 2: Supervised Fine-Tuning

3.2.1 Deep Feature Extraction from LICS-Pretrained Encoder

After MAE pretraining on the LICS dataset, the encoder is used to extract deep temporal features for the hybrid modeling pipeline

Feature Extraction Configuration:

Table 10: Feature Extraction Specifications

Component	**Specification**
Encoder Source	MAE pretrained on 30,000 LICS patches
Encoder Architecture	6-layer Transformer with 256-dim embeddings
Input Sequence Length	4 timesteps (quarterly observations)
Feature Layer	`Dense_64` (second-to-last dense layer)
Output Dimensionality	32-dimensional deep feature vector
Aggregation Method	Global Average Pooling (GAP) over patches
Projection Head	Linear layer: 256 $\rightarrow$ 64 dimensions

Extraction Process

1. **Input:** Temporal sequence of 4 quarterly Sentinel-2 images ($256\times256\times8$ each)
2. **Patch Embedding:** 16×16 patch tokenization
3. **Encoder Forward Pass:** Extract 256-dimensional embeddings per patch
4. **GAP Pooling:** Average across all patches $\rightarrow$ 256-dim vector
5. **Projection:** 256-dim $\rightarrow$ 64-dim $\rightarrow$ Final 32-dim feature vector

File Mapping

- Fine-tuning images: `data/lics/finetune/*.npy`
- Extracted features: `data/lics/features/X_deep_features_sequence_MAE_UNet.npy`
- Feature-label alignment: Based on Landsat scene date and transect ID

Key Insight The LICS-pretrained encoder learns coastal-specific visual representations (water/land boundaries, sediment patterns, tidal effects) that are then transferred to the Dublin Bay shoreline change prediction task.

3.2.2 Model B: Pure Transformer (Improved)

Table 11: Transformer Model Hyperparameters

Hyperparameter	Value	Justification
Sequence Length	4 timesteps	Captures one-year temporal patterns
Transformer Layers	2	Balance capacity and overfitting
Attention Heads	4	Multi-scale temporal attention
Feed-Forward Dimension	224	$2\times$ input feature dimension
Dropout Rate	0.2	Regularization
Batch Size	16	Constrained by GPU memory
Learning Rate	0.001	Adaptive with ReduceLROnPlateau
Optimizer	Adam	Effective for transformers
Loss Function	Huber Loss	Robust to outliers
Epochs	100	Early stopping patience = 20

Ensemble Configuration

- Number of Models: 3
- Aggregation Method: Mean prediction
- Seed Variation: 42, 142, 242

Data Augmentation

- Gaussian noise: $\sigma = 0.02$ (2% of feature scale)
- Augmentation factor: $2\times$ (doubles training data)

3.2.3 Model A: Classical Machine Learning (LightGBM)

Table 12: LightGBM Hyperparameters

Hyperparameter	Value	Description
Objective	`regression`	Continuous position prediction
Metric	`rmse`	Root Mean Squared Error
Boosting Type	`gbdt`	Gradient Boosted Decision Trees
Number of Estimators	1000 (early stop)	Maximum boosting rounds
Learning Rate	0.05	Conservative rate
Max Depth	8	Tree depth
Number of Leaves	64	Maximum leaf nodes
Min Child Samples	20	Minimum samples in leaf
Subsample	0.8	Row sampling ratio
Colsample by Tree	0.8	Column sampling ratio
Regularization Alpha	0.1	L1 regularization
Regularization Lambda	0.1	L2 regularization
Early Stopping Rounds	50	Validation patience

Target Variable: $\text{position}_t = \text{lag}_1 + \text{target\_change}$ (Absolute shoreline position at time t)

3.2.4 Model C: Hybrid Fusion (Transformer → LightGBM)

Architecture

1. **Stage 1:** Transformer feature extraction (frozen Model B encoder)
   - Extracts 32-dimensional deep temporal features from 4-timestep sequences
   - Uses `Dense_64` layer output as feature representation
2. **Stage 2:** Feature fusion
   - Concatenates 32 deep features + 97 classical features = 129 total features
3. **Stage 3:** LightGBM regression
   - Same hyperparameters as Model A
   - Trained on fused feature set

Key Configuration

```
SEQUENCE_LENGTH = 4
TRANSFORMER_PATH = "models/model_b_pure_transformer.keras"
FEATURE_LAYER = "dense_33"   # Layer before final output
TARGET_COL = "position_t"    # Absolute position prediction
```

3.3 Feature Engineering Configuration

3.3.1 Temporal Features

Table 13: Temporal Feature Types

Feature Type	Details	Purpose
Lag Features	lag_1, lag_2, lag_3, lag_4, lag_8	Historical positions
Change Features	change_1q, change_4q	Short-term & annual changes
Rolling Statistics	rolling_mean_4q, rolling_std_4q	Temporal smoothing
Momentum	First derivative of change	Rate of change acceleration
Normalized Change	change / rolling_std	Scale-invariant metric

3.3.2 Seasonal Features

Table 14: Seasonal Feature Encoding

Feature	Encoding	Range
Quarter (Sin)	$\sin(2\pi \times \text{quarter}/4)$	[-1, 1]
Quarter (Cos)	$\cos(2\pi \times \text{quarter}/4)$	[-1, 1]
Year (Normalized)	$(\text{year} - \text{min\_year})/(\text{max\_year} - \text{min\_year})$	[0, 1]
Time Index	$(\text{year} - \text{min\_year}) \times 4 + (\text{quarter} - 1)$	[0, T]

3.3.3 Environmental Features

Data Sources

1. **Tidal Data:** Dublin Port Water Level (Marine Institute; 2025b)
   - File: `tidal_data_RAW_dublin.csv`
   - Source: Marine Institute of Ireland
   - Temporal Resolution: Hourly
   - Aggregation: Annual statistics (mean, std, range, extremes)
2. **Wind Data:** Dublin Airport Meteorological Station (Government of Ireland; 2025)
   - File: `wind_data_RAW_DUBLIN_AIRPORT.csv`
   - Source: Met Éireann
   - Temporal Resolution: Hourly
   - Aggregation: Annual statistics (speed, direction, pressure)
3. **Wave Data:** Irish Weather Buoy Network M2 Buoy (Marine Institute; 2025a)
   - File: `wave_data_M2.csv`
   - Source: Marine Institute of Ireland
   - Temporal Resolution: Hourly
   - Aggregation: Quarterly statistics (height, period, energy)
4. **DEM (Elevation):** Copernicus DEM 30m (OpenTopography; 2025)
   - File: `output_hh.tif`
   - Source: OpenTopography Platform
   - Resolution: 30 meters
   - Extraction: Statistics along transect buffer (mean, std, range)

Lagging Strategy All environmental features are lagged by 1 quarter to prevent data leakage (environmental conditions at time $t-1$ cause changes at time t).

3.3.4 Spatial Features

Table 15: Spatial Feature Engineering

Feature	**Method**	**Purpose**
Spatial Cluster	K-Means ($k = 10$)	Groups transects by proximity
Cluster Statistics	Mean & std of residuals	Captures regional behavior

3.4 Outlier Detection and Detrending

3.4.1 Outlier Detection

Method: Median Absolute Deviation (MAD)

```
modified_z_score = 0.6745 * (values - median) / MAD
THRESHOLD = 3.5
```

Handling: Outliers are replaced with linear interpolation (not removed).

3.4.2 Detrending

Method: Linear trend removal per transect

```
trend = slope * time_index + intercept
residual = distance_clean - trend
```

Purpose: Tree-based models cannot extrapolate linear trends; detrending ensures the model learns deviations from the long-term trend.

4 Hardware Requirements

4.1 Minimum Hardware Specifications

Table 16: Hardware Requirements

Component	Minimum Requirement	Recommended
CPU	4 cores, 2.5 GHz	8+ cores, 3.0+ GHz
RAM	16 GB	32 GB
GPU	NVIDIA GPU 6GB VRAM	NVIDIA RTX 3080 (10GB)
Storage	50 GB SSD	100 GB SSD
OS	Windows 10, macOS 10.15+, Linux	Ubuntu 20.04+

4.2 GPU Configuration

CUDA Requirements - CUDA Version: 11.8 or 12.x

4.3 Computational Performance Expectations

4.3.1 Training Time Estimates

Table 17: Training Time Estimates (with recommended GPU)

Model	Dataset Size	Training Time	Epochs
MAE Pretraining	60,000 patches	∼7 hours	20
Model A (LightGBM)	1,590 samples	∼2 minutes	1000
Model B (Transformer)	1,908 sequences	∼15 min × 3	100
Model C (Hybrid)	954 sequences	∼5 minutes	1000

Total Pipeline Runtime: ∼8-10 hours for complete replication

4.3.2 Memory Usage Profile

Table 18: Peak Memory Usage

Process	Peak RAM	Peak GPU VRAM
Data Loading	~4 GB	N/A
Feature Engineering	~8 GB	N/A
MAE Training	~6 GB	~8 GB
Transformer Training	~4 GB	~6 GB
LightGBM Training	~2 GB	N/A

4.4 File Structure and Data Storage

The experimental setup expects the following directory structure:

Listing 3: Project Directory Structure

```
project_root/
|-- data/
|   |-- gee/
|   |   |-- Dublin_Bay_Quarterly_Images/*.tif (17 images)
|   |   |-- Dublin_Bay_Shorelines/*.geojson (17 files)
|   |   |-- transects_manual_exact.geojson
|   |   `-- analysis/
|   |       |-- transect_timeseries.csv
|   |       |-- features_engineered_improved.csv
|   |       `-- hybrid_splits/
|   |           |-- train_hybrid_assumption.csv
|   |           `-- test_hybrid_assumption.csv
|   |-- environmental/
|   |   |-- tidal_data_RAW_dublin.csv
|   |   |-- wind_data_RAW_DUBLIN_AIRPORT.csv
|   |   |-- wave_data_M2.csv
|   |   `-- dem_rasters_COP30/output_hh.tif
|   |-- lics/
|   |   |-- training/*.npy (60,000 patches)
|   |   |-- finetune/*.npy (100 patches)
|   |   `-- features/
|   |       |-- X_deep_features_sequence_MAE_UNet.npy
|   |       `-- final_hybrid_ml_data_*.csv
|   `-- model_b_dl_change_predictions.pkl
|-- models/
|   |-- best_MAE_UNet.pth
|   |-- best_ConvNeXt_UNet.pth
|   |-- best_Siamese_MAE.pth
|   `-- model_b_pure_transformer.keras
|-- encoder/
|   `-- mae_encoder_pretrained.pth
|-- results/
|   |-- model_a_position_results.pkl
|   |-- model_b_improved_results.pkl
```

```
|    `-- model_c_hybrid_results.pkl
|-- Step 1 Feature Engineering.ipynb
|-- Step 2 Masked AutoEncoder Pretraining.ipynb
|-- Step 3 Model Fusion.ipynb
|-- Step 4 Model Visualizations.ipynb
`-- Quarterly_GEE Dublin Bay_COPERNICUS.js
```

Critical Notes

- All CSV files use UTF-8 encoding
- Raster files are GeoTIFF format (`.tif`)
- Vector files are GeoJSON format (`.geojson`)
- Model checkpoints: PyTorch (`.pth`), Keras (`.keras`), Joblib (`.pkl`)

4.5 Parallel Processing Configuration

4.5.1 LightGBM Parallelization

```python
lgb_params = {
    'n_jobs': -1,   # Use all available CPU cores
    'verbose': -1
}
```

4.5.2 Data Loading (PyTorch)

```python
NUM_WORKERS = 0   # Set to 0 for Windows compatibility
```

Note: On Linux/macOS, `NUM_WORKERS=4` can accelerate data loading.

4.6 Reproducibility Configuration

To ensure deterministic results across runs:

1. **Set Random Seeds**

```python
SEED = 42
np.random.seed(SEED)
torch.manual_seed(SEED)
tf.random.set_seed(SEED)
```

2. **Disable Non-Deterministic Operations:**

```python
torch.backends.cudnn.deterministic = True
torch.backends.cudnn.benchmark = False
```

References

Government of Ireland (2025). Dublin airport hourly data. Accessed: 10 December 2025.
URL: *https://data.gov.ie/dataset/dublin-airport-hourly-data*

Marine Institute (2025a). Erddap iwb network. Accessed: 10 December 2025.
URL: *https://erddap.marine.ie/erddap/tabledap/IWBNetwork.html*

Marine Institute (2025b). Irish national tide gauge network (erddap). Accessed: 10 December 2025.
URL: *https://erddap.marine.ie/erddap/tabledap/IrishNationalTideGaugeNetwork.html*

OpenTopography (2025). Copernicus glo-90 digital surface model, UC San Diego.
URL: *https://doi.org/10.5069/g9028pqb*

O'Sullivan, C. and Dev, S. (2024). The Landsat Irish Coastal Segmentation (LICS) Dataset, *Zenodo (CERN European Organization for Nuclear Research)* .
URL: *https://doi.org/10.5281/zenodo.13742222*

O'Sullivan, C., Kashyap, A., Coveney, S., Monteys, X. and Dev, S. (2024). Enhancing coastal water body segmentation with Landsat Irish Coastal Segmentation (LICS) dataset, *Remote Sensing Applications Society and Environment* **36**: 101276.
URL: *https://doi.org/10.1016/j.rsase.2024.101276*