ST498 Capstone Project 2024-25

Building a Foundation Model for Modelling the World's Oceans

Candidate Numbers:
50270
50450
51348
45278

Submitted for the Master of Science in Data Science,
London School of Economics, University of London

Contents

List of Figures

Executive Summary

This capstone builds a practical forecasting pipeline for ocean-colour variables to provide short-range predictions when satellite coverage is patchy. We work with the Copernicus Marine Environment Monitoring Service (CMEMS) Sentinel-3/OLCI *Level-3* (L3) products because they are daily, well-documented, and include the optical and biogeochemical variables needed for coastal applications. Alternatives such as MODIS or VIIRS were reviewed, but they either rely on older sensors or lack several optical variables at the required spatial and temporal granularity. The study targets the south-west Coast of England from January 2022 to July 2025 and predicts eight variables: KD490, ZSD, RRS490, RRS443, CHL, MICRO, BBP, and CDM.

The task is non-trivial for two reasons. Firstly, Sentinel-3 satellite evidence systematic temporal discontinuity in the form of atmospheric obstructions (cloud cover), surface optical effects (sunglint), and regional seasonal limitations (less winter sunlight within the South-West England study region), contributing to large areas of missing data. Secondly, the data are large, multivariate, and spatially interdependent, so pointwise models ignore important cross-location and cross-variable signals. To make the problem tractable without discarding information, we reduce resolution from 1 km to 5 km, apply a conservative imputation scheme (forward fill capped at 15 days and nearest-neighbour spatial fill), explicitly keep prolonged winter gaps as missing rather than fabricate values. We ensured that the imputation process enhanced data completeness without compromising scientific validity or introducing unrealistic values into the dataset. On this prepared dataset we deploy a comprehensive modeling hierarchy that progresses from baseline methods to advanced architectures: simple baselines (7-day moving average; exponential smoothing), classical multivariate approaches (per-point VAR(1); k-means with cluster-level VAR; a low-rank factor model via Tucker decomposition), and three neural architectures (ConvLSTM; a temporal-attention CNN; and an edge-aware GNN + LSTM that uses geodesic distance and bearing as edge inputs). Model performance is evaluated using a held-out test dataset comprising the most recent 12 months, with three complementary metrics: Symmetric Mean Absolute Percentage Error (SMAPE), Root Mean Square Error (RMSE), and Mean Absolute Error (MAE).

The main findings from our comprehensive model comparison on the test set reveal a clear hierarchy in predictive accuracy. The k-means + VAR model emerged as the top-performing model across all eight oceanographic variables, consistently achieving the lowest SMAPE, RMSE, and MAE. This indicates that clustering grid points into homogeneous regions before applying a VAR model effectively captures both spatial and temporal dynamics, leading to more accurate forecasts than other approaches. The Exponential Smoothing baseline, while simple, provided a reasonable benchmark but was significantly outperformed. The deep learning models, ConvLSTM and Edge-Aware GNN, showed promise but did not generalize as well as the k-means + VAR model on the final test data, suggesting that their complexity may have led to overfitting or that they require more extensive tuning and larger datasets to unlock their full potential.

For end users, we advise deploying the k-means + VAR model at a 5km resolution as the main nowcaster for your region. It's important to update predictions daily as soon as new L3 satellite tiles become available. Ensuring that long winter outages remain masked during training and inference is essential to prevent adding artificial data. We also recommend keep-

ing the Exponential Smoothing model as a transparent, fast baseline and a reliable rollback option. The current model is trained for a single coastal region and excludes unobserved winters by design. Future work needs to expand training to areas and decades for scalability and explore additional evaluation metrics suitable for high-dimensional oceanographic data so that performance assessment can be aligned with the multivariate and capture the full spatiotemporal complexity of marine systems.

1 Introduction

Ocean forecasting is the scientific discipline focused on predicting the state of the marine environment and is fundamental for understanding dynamic ocean processes. This in turn plays a critical role in supporting sustainable ocean use, ensuring the safety of livelihoods and marine ecosystems. For example, ocean forecasting is vital for disaster preparedness and response [Link et al., 2023, Visbeck, 2018]. By enabling a predictive system to notify the authorities in advance, the ocean forecasts help them to anticipate and mitigate the impact of extreme events including tsunamis [Tsushima and Ohta, 2014, Sugawara, 2021], storm surges [Pérez Gómez et al., 2022, Morim et al., 2023, Chaigneau et al., 2023], marine heatwaves [Hartog et al., 2023, Bonino et al., 2024], and oil spill accidents [Cucco et al., 2024, Keramea et al., 2023]. Furthermore, ocean forecasting also helps optimise operational planning across various industries, such as shipping, fishery, marine resource management and coastal engineering [Holmberg et al., 2025]. Actionable decisions in these industries rely on high-resolution models that can provide accurate forecasts tailored to the local regions [Sakamoto et al., 2019, Ciliberti et al., 2021, Kärnä et al., 2021].

The sheer volume of open-source data available from satellites and sea and ground-based weather monitoring stations presents both an immense opportunity and a significant challenge for advancing ocean forecasting. For instance, the European Space Agency (ESA) stands as the third-largest data provider globally, generating an astonishing 20 terabytes of data daily [European Space Agency, 2023]. To contextualise this, contemporary artificial intelligence (AI) systems like ChatGPT are trained on 45 terabytes of all open-source text data on the internet – just over two times the data collected in a single day by ESA [Laizure, 2024]. Deriving meaningful insights from the increasing volume and complexity of this data requires a shift from traditional physical modelling to more data-driven approaches [Reichstein et al., 2019]. The authors further state that this indicates a notable gap in the earth observation community, which has been dominated by domain experts rather than data scientists equipped with the expertise to handle such large-scale computational challenges, and extract spatio-temporal features automatically.

The primary objective of this project is to leverage statistical time series as well as recent developments in deep learning models to model and predict ocean dynamics, based on ESA's Sentinel-3 datasets. We want to answer the research question: *"Which modelling approaches can accurately forecast ocean variables from satellite observations, while addressing cloud cover gaps and high data dimensionality?"*

Section 2 dives into a literature review of existing modelling approaches and our novel contributions. We then provide a dataset overview in Section 3, exploring our dataset and variable selection. Section 4 follows, mentioning the challenges of working with granular satellite data and introduces our novel technique of handling missing data using a two-step interpolation approach with forward filling and k-d Tree for temporal and spatial filling respectively. Section 5 highlights the overall methodological framework, including the creation of train-validation-test splits and model evaluation. We then proceed with navigating the 3 modelling approaches: baseline, time series and deep learning models in Sections 6, 7 and 8. Lastly, we compare the model performance for all the models on the validation set in Section 9.1, and choose the best four competing models to run on the unseen test set. Section 9.2 details the best generalising model, Edge-Aware Graphical Neural Networks (GNN). Finally,

Section 10 reports our final conclusion and recommendations for future work.

When tested on the hold-out test data, k-means + VAR delivered the strongest performance across most variables, often achieving the lowest SMAPE, RMSE, and MAE, particularly for stable or spatially homogeneous patterns. The Edge-Aware GNN also performed well, especially in cases where spatial structure played a larger role, while ConvLSTM focused rather on approaching the accuracy of all variables within similar range at once due to its nature of generalised architecture. Exponential Smoothing provided reasonable results for a simple baseline, but it was consistently outperformed by the more advanced approaches.

2 Literature Review

In order to decide upon our modelling approaches, we refer to the historical as well as recent academic discoveries in the fields relevant to our research: handling missing data with interpolation, combatting high dimensionality, time series and neural network-based modelling approaches for oceanographic data.

Firstly, the extensive presence of missing data due to cloud coverage and sensor limitations is a fundamental challenge when employing ocean colour observations, such as Level 3 (L3) data from ESA's Copernicus Sentinel-3 satellite [Campbell, 1995, Franz et al., 2015]. Various methods have been developed for interpolating missing satellite data. They range from basic spatial interpolation techniques like Inverse Distance Weighting (IDW) [Oliver and Webster, 1990, Isaaks and Srivastava, 1989] to more advanced spatio-temporal approaches such as empirical orthogonal function (EOF) based methods, which reconstruct missing ocean data based on the dimensions with highest variability [Beckers and Rixen, 2003, Alvera-Azcárate et al., 2005]. Recently, machine learning techniques including neural networks and Gaussian processes can account for complex non-linear relationships when providing interpolation [Liu et al., 2017, Sun et al., 2020]. We propose a novel approach for temporal and spatial imputation tailored to our oceanographic dataset. First, we perform time-wise forward filling and second, we employ a k-d tree (k-dimensional tree) [Skrodzki, 2019] to query the k-nearest neighbours and efficiently impute spatial gaps in the dataset. To avoid data leakage, which is critical for forecasting purposes, temporal gaps are forward-filled within each (lat, lon) cell using only information available up to and including that day; we fill missing values for a location by looking at the closest nearby location on the same day. This way, coastal areas are only filled using other coastal points, and open ocean areas are filled using other open ocean points, maintaining realistic coastal–open-ocean contrasts.

The foundation for applying time series analysis to oceanographic data is the presence of temporal autocorrelation in ocean data. Studies have established that oceanographic variables exhibit strong short-term correlations with their past values due to physical processes such as diffusion [Emery and Thomson, 2001]. Hence, this justifies the use of vector autoregressive models (VAR) for understanding the relationship between multiple ocean variables is a well-studied topic [de Bodas Terassi et al., 2023],[Yan et al., 2021]. However, these models only consider the temporal trends and not spatial information. To overcome this, a more general framework for spatio-temporal analysis is the *Panel VAR* model, which extends VAR to panel data, allowing for the joint modelling of variables across multiple locations over time [Canova and Ciccarelli, 2013]. This enables the study of both cross-variable correlations and spatial spillover effects, for example, whether changes in the chlorophyll level in one region influence the chlorophyll level in neighbouring regions [Holtz-Eakin et al., 1988, Baltagi, 2013]. However, given 10899 grid points in our dataset, fitting a Panel VAR would be computational infeasible due to the large coefficient matrix. Therefore, we explore methods to work with time series in a high-dimensional setting.

Firstly, clustering-based approaches to ocean modelling is very common in the literature. For example, studies include clustering-based approach to ocean data around Antarctica [Sun et al., 2021]. Dynamic Time Warping (DTW) is another algorithm using to measure the similarity between two time series by finding the optimal alignment in their sequences and thus cluster them [Paparrizos and Gravano, 2015], [Chu et al., 2002]. Further extending the spatio-

temporal clustering framework, a paper introduced Second-Order Data-Coupled Clustering (SODCC) which embeds temporal dynamics within the clustering itself [Chidean et al., 2018]. However, these models do not perform forecasting — they only identify cluster assignments for each spatial location. Our study extends the approach proposed by Chidean et al. [2018], by bridging the spatial clustering with predictive temporal modelling. We first perform k-means to identify homogeneous ocean regimes, and then apply a VAR model on each regime to learn something about both the spatial and temporal trends in a forecasting framework. Another approach is factor models for tensor time series; tensor decompositions, like Tucker decomposition and CP decomposition, are increasingly used in environmental sciences to model data with multiple modes of variation, such as space, time, and different variables [Kolda and Bader, 2009, Rabus et al., 2019]. These methods are particularly advantageous for inherently sparse tensors, where explicit handling of missing values is critical, often by integrating a mask into the decomposition algorithm [Acar et al., 2011].

Moreover, deep learning models have emerged as highly effective tools for spatio-temporal forecasting due to their capacity to learn complex patterns. Convolutional Neural Networks (CNNs) excel at extracting spatial features, while Recurrent Neural Networks (RNNs), particularly Long Short-Term Memory (LSTM) networks, are adept at capturing temporal dependencies [Hochreiter and Schmidhuber, 1997]. The integration of these two architectures, leading to Convolutional LSTMs (ConvLSTMs), have shown good prediction capacity for this type of data [Shi et al., 2015]. Applications of ConvLSTMs in oceanography include sea surface temperature prediction, ocean current forecasting, and chlorophyll-a anomaly detection [Kim et al., 2019, Li et al., 2021]. Additionally, the concept of attention, another way for capturing temporal dependencies, has also been presented in geospatial prediction to form a so called temporal attention CNN (TACNN) [Lin et al., 2021]. Next to the ConvLSTM and TACNN, transformers are increasingly competitive for spatio-temporal geoscience because they tokenise each frame into patches and use self-attention to capture long-range temporal and spatial dependencies that CNNs/LSTMs may only model indirectly, such a interactions between distant ocean regions [Dosovitskiy et al., 2021].

Another approach is graph-based neural networks, which have gained traction in oceanographic forecasting due to their ability to capture complex spatial dependencies in irregularly gridded data. For example, a paper explores Graph Memory Neural Network (GMNN) for Sea Surface Temperature (SST) prediction [Liang et al., 2023]. While these studies mark significant progress, they have largely been limited to univariate forecasting of SST or similarly single-parameter fields. In this report, we address this gap by extending the framework to multivariate prediction of multiple optical and biogeochemical variables using an edge-aware graph neural network (GNN) coupled with a temporal module to jointly model spatial connectivity and temporal dynamics.

Overall, our research aims to build upon these existing methodologies by developing models for forecasting ocean variables based on spatio-temporal satellite data. We employ a two-stage interpolation procedure to deal with vast amounts of missing data. Also, we develop time series models spanning vector autoregressions, clustering and factor models, and deep learning architectures focused on attention and edge-aware mechanisms. We then evaluate all these models against a baseline model and conclude which model leads to the most accurate ocean characteristic predictions.

3 Dataset overview

3.1 Choosing our data source

To study and forecast ocean characteristics accurately, we require a reliable, consistent, and scientifically validated dataset. After comparing various global providers online, we selected the Copernicus Marine Environment Monitoring Service (CMEMS), supported by the European Space Agency (ESA). we provided raw merged satellite products here: Google Drive (Raw Dataset) and our preprocessed dataset here: Google Drive (Processed Dataset). While other global providers such as NASA's MODIS-Aqua and Terra sensors have been widely used for ocean colour data, they present much older missions which provide lower spatial resolution data and are known to have more gaps in cloudy or polar conditions [Campbell, 1995, Franz et al., 2015]. NOAA's VIIRS mission offers improvements in data continuity, but is primarily focused on physical variables like sea surface temperature. Not all required optical and biogeochemical parameters were available at the desired spatial and temporal resolution.

In contrast, Sentinel-3's OLCI (Ocean and Land Colour Instrument) delivers 300–1000m resolution satellite images, provides daily observations, and includes a wider range of optical variables such as RRS443, RRS490, CDM, or MICRO [Copernicus Marine Environment Monitoring Service, 2023b, International Ocean Colour Coordinating Group, 2018]. Additionally, CMEMS provides pre-processed Level-3 products with regular gridding and cloud masking (see Section 3.3 for more details about levels of data in oceanographic contexts). Thus, we chose Sentinel-3's OLCI datasets which are openly accessible via APIs and well-documented, making it easier to automate large-scale spatio-temporal data workflows.

3.2 Variable selection

Ocean datasets spans across a wide range of variables, typically grouped into categories such as physical, optical, and biogeochemical properties. We focused on four ecologically important categories measurable by satellites: transparency, reflectance, plankton, and optical scattering. They capture key processes and metrics that regulate the light availability in oceans, primary production of aquatic organisms and animals, as well as water quality and health of marine ecosystems. Changes in the levels of these variables can signal pollution or climate-driven changes, which would need urgent countermeasures to preserve the ocean ecosystem. From these four categories, we selected the following eight important variables for our study:

- **KD490 (Diffuse Attenuation Coefficient at 490nm), per meter(m)**: Indicates the rate at which sunlight is absorbed or scattered as it travels deeper into the ocean at the wavelength of 490 nanometers (nm). When KD490 is high, light does not travel far below the surface, which usually means the water is more turbid or cloudy. This is important for estimating the euphotic depth — the depth at which sufficient sunlight is available for photosynthesis to occur. Understanding light penetration helps us estimate where phytoplankton can grow and how much primary production of aquatic organisms is possible in different parts of the ocean [Lee et al., 2005].

- **ZSD (Secchi Disk Depth), in m**: Traditional, easy-to-understand measure of water

clarity, based on how deep a white disk (called a Secchi disk) can be seen from the ocean surface. Although we derive this value from satellite algorithms instead of a physical disk, it still acts as a simple and reliable reference. ZSD can help verify or validate other optical measurements such as KD490, making it useful for cross-checking our data for quality issues or unusual optical conditions [Lee et al., 2015].

- **RRS443 and RRS490 (Remote Sensing Reflectance at 443nm and 490nm), per steradian (solid angle unit for reflectance)**: These are measurements of how much sunlight is reflected from the ocean surface back into space at specific wavelengths (443nm and 490nm). RRS443 is more sensitive to chlorophyll and dissolved organic matter, making it useful for identifying biological content like phytoplankton. RRS490 is widely used in ocean colour algorithms and helps us understand overall water quality. These reflectance values are often used as inputs in models that estimate chlorophyll and other water properties [Mobley, 1999, International Ocean Colour Coordinating Group, 2018].

- **CHL (Chlorophyll-a Concentration), in mg/m^3**: Chlorophyll-a is a pigment found in all photosynthetic organisms, especially phytoplankton. Measuring the concentration of chlorophyll-a gives us a direct way to estimate phytoplankton biomass — essentially the degree of microscopic plant life present in the water. It is one of the most important indicators of biological activity in the ocean and is used in almost all marine ecosystem and oceanographic productivity models [Behrenfeld et al., 2006].

- **MICRO (Microphytoplankton Proportion), in mg/m^3**: This variable tells us the percentage of phytoplankton in the water that are considered "micro" in size — usually the larger types of phytoplankton. When MICRO values are high, it often means the water has plenty of nutrients and supports the growth of large-celled phytoplankton like diatoms. These organisms play a key role in the marine food chain and can influence carbon cycling, making MICRO an important ecological indicator [Uitz et al., 2006].

- **BBP (Backscattering Coefficient), per m**: BBP measures how much light is scattered back toward the satellite by particles in the water. These particles can include sediment, plankton, or other suspended matter. BBP is especially useful for understanding water turbidity and particle concentration, which can change due to river runoff, storms, or algal blooms [Loisel and Morel, 1998]. It also helps in detecting events like resuspension of sediments or changes in plankton density.

- **CDM (Coloured Dissolved Organic Matter), per m**: CDM tracks the amount of dissolved organic substances in the water that absorb light — typically the brown or yellowish colour that comes from dead plants or land-based runoff. These substances do not reflect light like particles do but instead absorb it. High CDM values are usually linked to freshwater input from rivers or the breakdown of biological material in coastal waters. CDM plays an important role in regulating how deep light can travel, which in turn affects marine ecosystems [Siegel et al., 2002].

Although the CMEMS dataset includes several other variables, we decided not to include them all in our study, because they were mostly quite similar to our chosen variables, or did

not meet the required spatial resolution. This observation was made because of the following reasons. For example, physical variables like Sea Surface Temperature (SST) and Sea Surface Height (SSH) were excluded because they were not consistently available in the Level-3 optical data we were using. These types of variables are usually derived from different sensors (like altimeters or radiometers) and are more commonly processed in Level-4 physical datasets, such as those used in ocean circulation modelling [Copernicus Marine Environment Monitoring Service, 2023a, Groom et al., 2009]. Moreover, since the focus of our study is primarily on optical and biological water properties, these physical variables were outside the scope of our analysis. We also left out the variables indicating the uncertainty of an ocean variable (e.g. CHL uncertainty, KD490 uncertainty). While they are useful for validating sensors or comparing algorithms [Lee et al., 2015, International Ocean Colour Coordinating Group, 2018], our modelling goal is to predict the ocean conditions themselves — not their associated error margins. Including these would introduce redundancy and unnecessarily complicate the feature space. The group of detailed plankton types were also excluded from our study (e.g. DIATO, DIANO, GREEN, HAPTO, NANO, PICO, PROCLO, and PROKAR). These are derived taxonomic groups that serve as subsets of the broader phytoplankton community. Many of these variables are strongly correlated with CHL or MICRO [Uitz et al., 2006, Behrenfeld et al., 2006]. Using all of them would risk multicollinearity where too many similar inputs can lead to unstable estimates and reduce interpretability. Similarly, we excluded additional reflectance bands like RRS412, RRS555, and RRS670. These bands capture light reflected from the ocean at different wavelengths, but in practice they are often highly correlated with RRS443 and RRS490, which we already included [International Ocean Colour Coordinating Group, 2018, Mobley, 1999]. Finally, we excluded also SPM (Suspended Particulate Matter). Although it is related to water clarity like BBP and KD490, the SPM variable had a high number of missing values in our selected region, and showed strong correlation with BBP, which is more consistently observed in satellite data [Loisel and Morel, 1998, Palmer et al., 2015]. For model simplicity and reliability, BBP served as a better proxy for particle concentration. Overall, we narrowed our variables down to the most informative and complete features to ensure the model would remain interpretable, manageable, and scientifically robust.

3.3 Choosing the dataset level

Satellite-derived ocean datasets are available at different processing levels. The two most relevant for our work are Level-3 (L3) and Level-4 (L4): On the one hand, L3 datasets are quality-controlled, mapped to a uniform grid and they preserve true observational patterns without model-based smoothing. These datasets are available at the daily resolution, which is especially important for identifying short-term oceanic events and training time series forecasting models. We are interested in predicting the variables on a daily basis [Copernicus Marine Environment Monitoring Service, 2023a]. However, L3 datasets may contain spatial gaps due to cloud coverage, sensor dropouts, or land masking. On the other hand, L4 datasets provide spatially and temporally complete maps by interpolating over missing areas using model assimilation techniques. While this makes them easier to work with, L4 datasets are generally offered at weekly or monthly resolution and often exclude many optical and biogeochemical parameters critical to our study — such as RRS443, CDM, and MICRO

[Lee et al., 2015, International Ocean Colour Coordinating Group, 2018]. This loss of detail could negatively affect model performance, especially for time-series forecasting [International Ocean Colour Coordinating Group, 2018, Lee et al., 2015]. Furthermore, only 2 out of the 4 core variable categories used in our analysis were available in L4, and no daily data products were found for these variables. Moreover, L3 products are easy to access via the Copernicus Marine Service data portal at: `https://marine.copernicus.eu`. Users can filter by dataset level, satellite mission (e.g., Sentinel-3 OLCI), spatial region, and variable type, and download data via APIs or the interactive GUI.

Because our goal is to construct a model that could learn from detailed changes in ocean conditions over time and space, which is grounded in daily data, we found that L3 data was a better fit than L4 data. It offers daily data, covers a wider range of variables, and keeps the original patterns observed by the satellite. This leaves smoothing, interpolating, and preprocessing the data to us in order to have the data ready to be inserted into our models. In addition to that, it is important to note massive data volume of L3 data, high sparsity, and processing overhead. These aforementioned issues were carefully addressed during our data cleaning and imputation workflow (refer to Section 4.1).

3.4 Spatio-temporal scope: January 2022 to July 2025, south-west coast of England

From the temporal perspective, we use the L3 daily satellite data from January 2022 to July 2025, which gives us a time range of about 3.5 years. First, incorporating multiple years and therefore three full seasonal cycles was the main hypothesis for the period selection which would lead to both strong model training and reliable testing. This is important because many ocean variables such as CHL and MICRO change with the seasons [Behrenfeld et al., 2006, Uitz et al., 2006]. This helps the model learn oceanographic dynamics through the recurring patterns caused by natural biological cycles. Second, using recent data helps improve the model's ability to work well on future unseen data — a key requirement for generalization in machine learning [Franz et al., 2015]. Third, this time span gives us a good balance between having enough data to train models and keeping the dataset manageable in size.

From the spatial perspective, we focused on a specific part of the ocean — the south-west coast of England, covering areas like the Celtic Sea, Bristol Channel, and part of the English Channel. Figure 1 shows the bounding box of our area of interest.

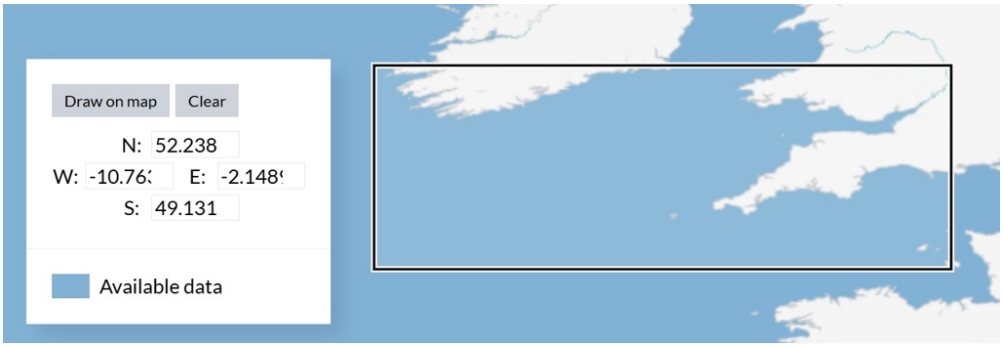


Figure 1: Study region: Bounding box drawn over the South-West Coast of England.

This region was selected for several important reasons: First, this is an ecologically rich

zone. The waters here support seasonal phytoplankton blooms, which are a major part of the ocean food web. These blooms are known to change with temperature and sunlight, making this area useful for studying biological changes over time [Groom et al., 2009, Hardman-Mountford et al., 2020]. Secondly, the region benefits from the UK mainland's river runoff which deposits nutrients and organic matter in the ocean. This leads to higher levels of coloured dissolved organic matter (CDOM) which can be easily observed using satellite sensors like Sentinel-3 [Palmer et al., 2015, Groom et al., 2019]. Third, the south-west coast is climatically important. It is a transition area where waters from the open Atlantic mix with more enclosed coastal waters. This creates natural variations in ocean colour, turbidity, and phytoplankton, which are valuable for model testing [Quante and Colijn, 2016]. Lastly, this region gives us a good mix of coastal and open-ocean environments, helpful for training models that can work in both shallow coastal areas and deeper ocean zones. This region was also among the recommended zones for CMEMS ocean colour analysis. Due to its strong ecological value and the scientific challenge, it provides a good region of interest for oceanographic forecasting.

3.5 Exploratory data analysis

Temporal Analysis (2022–2024): We visualise the 8 univariate time series of the ocean variables by computing the daily average of each variable across all spatial grid points. As seen in Figure 2, the time series for KD490 (per m) and CHL (mg/m^3) appear to be relatively stable, fluctuating around the mean values of 0.10 and 0.15 respectively, except for the steep spikes around May 2024. ZSD (in m) follows a similar stable trend with the time series reverting towards the mean value of 10. Variables RRS490 and RRS443 (both per steradian), BBP (per m) and CDM (per m) reflect a recurring annual pattern, as seen in the four segments of their time series. This could potentially indicate towards the presence of annual seasonality in the time series, far from a random walk. In terms of MICRO (mg/m^3), this variable exhibits both seasonality and change in trend with the high variability, especially during spring and summer. This aligns with known biological patterns such as phytoplankton blooms. RRS bands and BBP are more stable but still reflect short-term changes. CDM shows occasional spikes, likely tied to runoff events.

Spatial Analysis (15 July 2023): Figure 3 presents spatial snapshots for each variable on a representative date. We observe distinct coastal-to-open-sea differences. KD490 and ZSD are higher near the shore, indicating lower clarity. CHL, MICRO, and BBP show hotspots near river mouths. RRS values vary with water type, and CDM is strongly concentrated near terrestrial inputs. These patterns confirm that our selected variables capture meaningful physical and ecological signals — both over time and across space — making them strong candidates for predictive modelling.

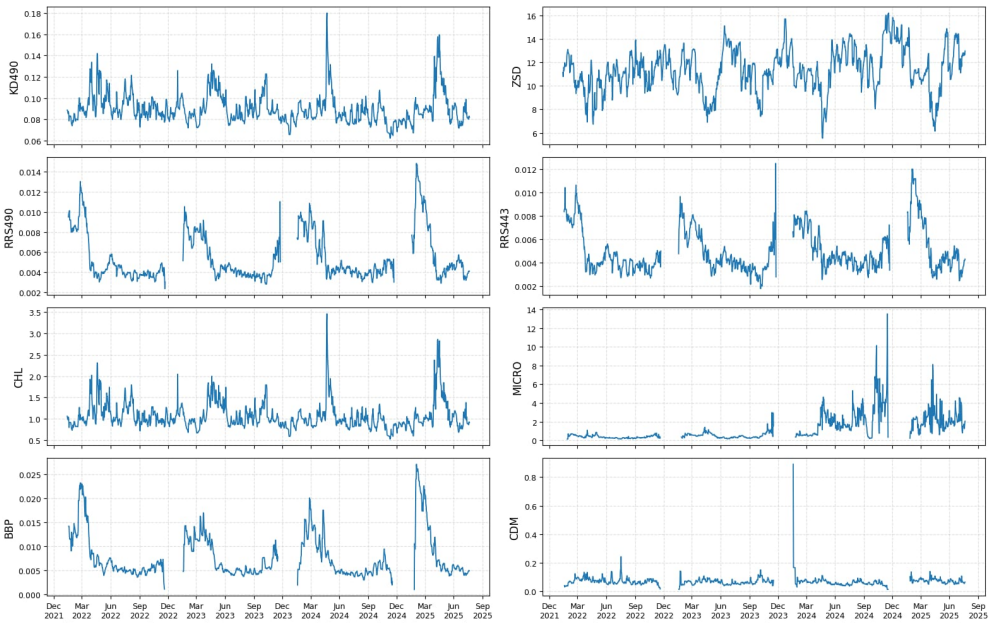


Figure 2: Time series of the eight oceanographic variables (KD490, ZSD, RRS490, RRS443, CHL, MICRO, BBP, CDM) for the full dataset (2022–2025). The observed gaps are due to seasonal and observational constraints, as described in Section 4.2.3. This dataset is subsequently split into training, validation, and test sets, as detailed in Section 5.

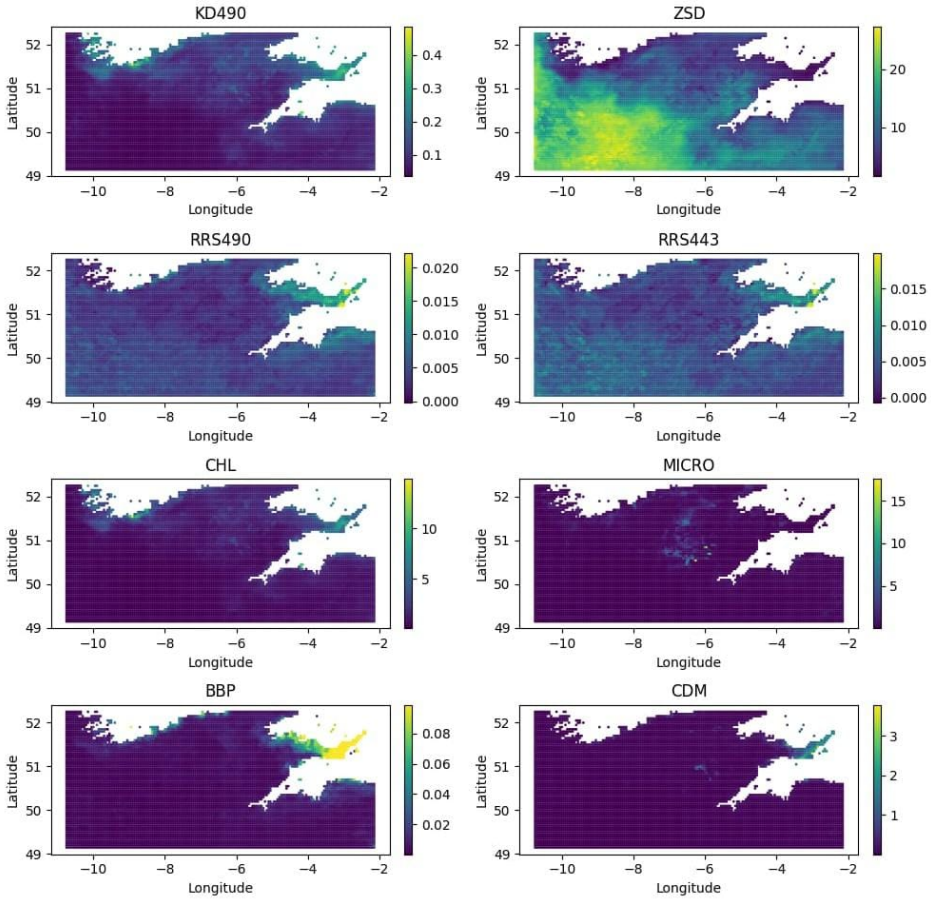


Figure 3: Spatial distribution of selected variables on 15 July 2023.

4 Dataset preparation

4.1 Computational challenges

One of the biggest hurdles when working with Copernicus Marine's ocean satellite datasets is their massive scale, both in spatial and temporal resolution. Initially, we employed 1km x 1km L3 daily data, which quickly resulted in over 180 million rows for just two years. Since our goal was to train on at least two full years, validate on half a year and test on one year of recent data (total: 3.5 years from Jan 2022 to July 2025), this volume posed a serious computational burden. Given the team's 16 GB RAM workstations, memory limitations made it impractical to train high-dimensional models on the full-resolution dataset. To manage this, we narrowed our geographic focus to a small coastal region in the North Atlantic and eventually downsampled the dataset to a 5km x 5km resolution, which brought the total row count to around 10 million. The process was addressed in Section 4.3.

4.2 Handling missing values

Apart from scale, the second major challenge was missing values, a well-documented issue in satellite-derived oceanographic remote sensing datasets [Gregg and Casey, 2007]. These gaps are primarily caused by cloud cover, low solar angles in winter, and sensor geometry, particularly for Sentinel-3, which operates in a sun-synchronous orbit with limitations in high-latitude winter coverage [CMEMS, 2023]. As discussed in Section 3.3, while L4 data offers fewer missing values due to prior interpolation, we specifically choose L3 to retain daily-level granularity, an essential requirement for training our daily-level forecasting models. To address this challenge, we adopted a systematic and hybrid imputation strategy that leverages both temporal and spatial information. The overview of the data processing in the complete pseudocode is provided in Section 4.2.2 below.

4.2.1 Temporal imputation

Specifically, we first identified all `(lat, lon)` cells with runs of missing data longer than 15 days and temporarily masked these so they were excluded from this temporal imputation step. We then forward-filled the remaining cells. Finally, the masked values were restored to `NaN` so they could be handled later by the day-by-day spatial imputation. [Kamalov and Sulieman, 2021].

4.2.2 Spatial imputation

After temporal imputation, the remaining gaps, which were often spatially isolated, were addressed using spatial imputation based on the k-d tree (k-dimensional tree) algorithm, as visualised in Figure 4. For each day, we first identified all valid spatial locations with non-missing values. k-d tree, an efficient nearest-neighbour search algorithm, was then used to find the closest (k=1) valid (latitude, longitude) point for each missing observation. The missing value was replaced with the value from its nearest spatial neighbour. This way, coastal pixels borrow only from coastal neighbours, and open-ocean pixels from open-ocean neighbours, preserves their natural contrast. [Virtanen et al., 2020]

Algorithm 1 Ocean Data Cleaning, spatio-temporal Imputation, and Resolution Reduction

Require: Raw dataset `df_raw` (flags, timestamps, coordinates, variables)
Ensure: Cleaned, imputed, and downsampled dataset `df_5km`

Initialize: `df` ← `df_raw.copy()`
Remove rows where `flags` = 1 (Land data)
`THRESHOLD_STREAK` ← 15
Sort `df` by (`latitude`, `longitude`, `time`)
for all unique locations $(\ell_{\text{lat}}, \ell_{\text{lon}})$ in `df` **do**
    for all variables v in ocean-variable set **do**
        Identify consecutive NaN streaks in v at $(\ell_{\text{lat}}, \ell_{\text{lon}})$
        if streak length $\geq$ `THRESHOLD_STREAK` **then**
            Set those entries to (temporary placeholder)
        end if
    end for
end for
Group `df` by (`latitude`, `longitude`) and apply forward fill along time
Replace all (temporary placeholder) values with NaN
`filled_df` ← empty list
for all unique dates d in `df` **do**
    g ← subset of `df` where `date` = d
    for all variables v in g with NaNs **do**
        Build k-d tree from valid (`lat`, `lon`) pairs in g
        For each NaN in v, assign nearest k=1 neighbor's value (same-day)
    end for
    Append g to `filled_df`
end for
`df_filled` ← concatenate all entries in `filled_df`
Mark winter dates with fully missing data as -2.0 (Arbitrary but numerically distant from natural ranges)
Reattach land pixels (previously removed) with -1.0 flag
Round coordinates to 0.05° to create `lat_5km`, `lon_5km`
Group by (`lat_5km`, `lon_5km`, `date`) and compute daily means (decreased resolution)
return `df_5km`

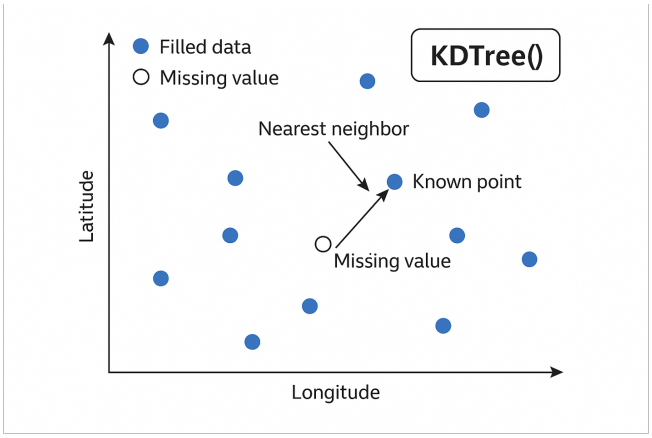


Figure 4: Illustration of k-d tree-based spatial imputation: blue dots represent valid observations; white circles denote missing values filled using the nearest spatial neighbor.

4.2.3 Winter data gaps

Some dates, especially from mid-November to late January, had complete data loss across the spatial grid in our study region over the North Atlantic. These windows could not be imputed either temporally (via forward fill) or spatially (via k-d tree), due to the absence of any valid observations. This is a known limitation in ocean color datasets from Sentinel-3, particularly during the winter months in mid- to high-latitude regions like the North Atlantic, where data collection is impacted by reduced solar illumination, persistent cloud cover, and the satellite's orbit characteristics.

Sentinel-3 operates in a sun-synchronous orbit with a local equator crossing time of approximately 10:00 AM ([Copernicus Marine Service, nd]). This means it observes each location at a consistent local solar time daily. However, during winter months in northern latitudes, this orbit leads to low solar zenith angles, limiting the availability of usable sunlight for passive optical sensors and increasing atmospheric path lengths, making ocean color retrieval unreliable or impossible .

To address this, we masked the affected days using a placeholder value of `-2.0` a unique low number not naturally present in the dataset. This allowed our models to bypass these days during training without introducing artifacts through over-imputation. chosen over extreme values (e.g., `-1000`) to avoid distorting scaling and keep files more compressible. We also set land pixels to `-1.0`, allowing a clear separation between permanent no-data (`-1.0`) and temporary gaps (`-2.0`) for imputation and modelling. Such flag-value masking is standard practice in climate and Earth-observation datasets [Garnesson et al., 2019].

4.3 Resolution reduction

Once imputation was complete, To reduce data size and enable efficient model training, we downsampled the dataset from 1km × 1km to 5km × 5km (0.05°). Latitude and longitude values were binned to the nearest 0.05 degrees, and for each day, we computed the mean of valid values within each bin. Placeholder flags (-1 for land, -2 for fully missing) were preserved where appropriate. This reduced the number of unique grid points from 246,000 to 10,900, making the dataset more manageable for deep learning [Zheng et al., 2021].

4.4 Case Study: Effectiveness of MICRO variable imputation

To evaluate the performance of our hybrid spatio-temporal imputation strategy, we conducted a case study on the MICRO variable for the North Atlantic region. This example illustrates how missing data, often caused by wintertime satellite limitations, can be reliably reconstructed using a two-step process: forward-filling over time, followed by spatial interpolation using KDTree.

Figure 5 presents the imputation workflow and the results for October 22–23, 2024. The left panels show satellite-derived MICRO values for October 22 (top) and October 23 (bottom), both exhibiting large missing patches. The right panel displays the reconstructed map for October 23 after applying our hybrid imputation approach. The reconstruction clearly restores large spatial gaps while maintaining coherent spatial gradients across oceanographic features.

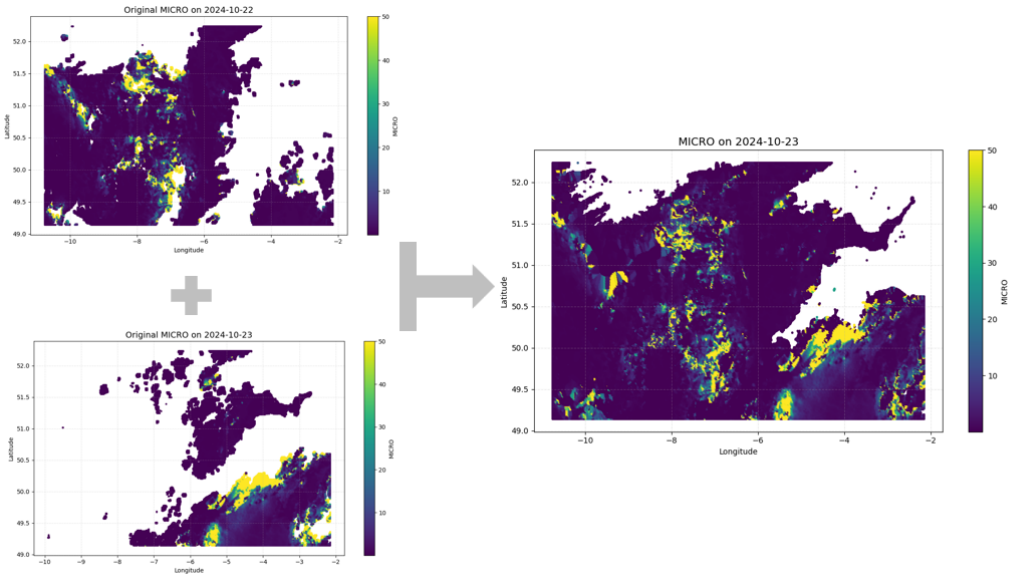


Figure 5: Illustration of the hybrid imputation strategy applied to the MICRO variable.Left: Original L3 MICRO data for October 22 (top) and October 23, 2024 (bottom), showing extensive missing regions.Right: Reconstructed MICRO field for October 23 after temporal forward-filling (up to 15 days) and spatial KDTree-based nearest-neighbor imputation.

To further demonstrate the transformation from the original resolution to a model-ready format, Figure 6 compares the same imputed data at 1km and 5km resolution. The top panel shows the temporally and spatially imputed MICRO variable at 1km × 1km resolution, while the bottom panel shows the downsampled 5km × 5km version used in subsequent modelling.

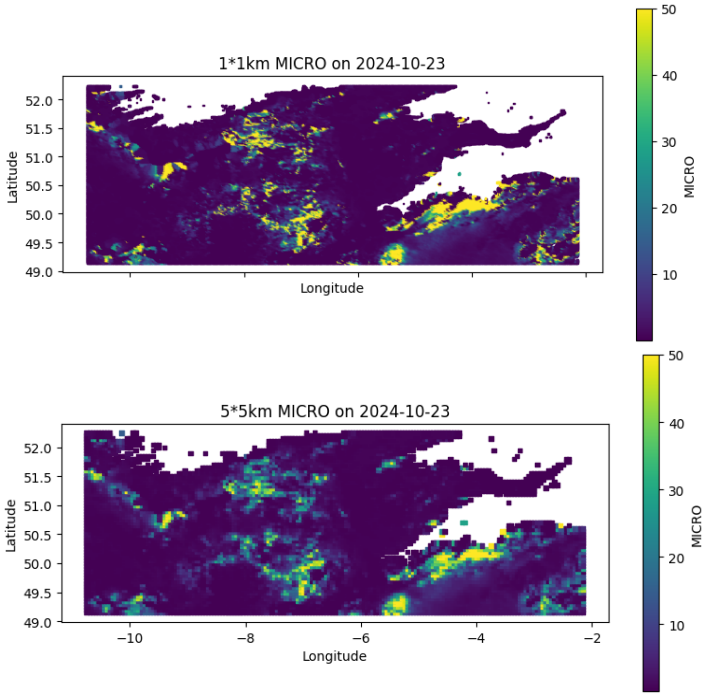


Figure 6: Left: Reconstructed MICRO values on October 23, 2024 at native 1km × 1km resolution. Right: Same data after 5km × 5km downsampling using spatial binning and aggregation.

5 Methodology framework

Algorithm 2 Oceanographic Data Modelling & Forecasting Pipeline

Require: Sentinel-3 CMEMS Level-3 oceanographic data (South-West coast of England)
Require: Modelling approaches →
    Baseline models: 7-day Moving Average, Exponential Smoothing
    Time Series models: VAR, k-means + VAR per cluster, Factor model
    Convolutional neural networks (CNN): ConvLSTM, TACNN
    Graph Neural Networks (GNN): Edge-Aware GNN
Ensure: 1-day ahead forecasts of ocean variables
    Data Preprocessing:
Fill NaNs: temporal interpolation (forward-filling), spatial interpolation (k-d tree)
Aggregate spatial resolution from 1 km to 5 km grid
2D tensor: Time, latitude, longitude, 8 variables
    With masks for land data (-1), winter dates (-2) – no NaNs
for all Modelling approaches **do**
        Training set: 2022-01-27 → 2023-11-17
        Validation set: 2024-01-26 → 2024-07-31
        Test set: 2024-08-01 → 2025-07-21
end for
for all Modelling approaches m **do**
    if m in Baseline models, VAR, K-means + VAR, Edge-aware GNN **then**
        Use the 2D tensor, remove land pixels and winter dates and
    else if m in Factor model, ConvLSTM, TACNN **then**
        Convert 2D tensor to 4D tensor (Time x Ocean Variables x Latitude x Longitude)
        Use 4D tensor, remove winter dates, retain land data
    end if
end for
    Model Training and Evaluation:
for all Modelling approaches **do**
    Produce 1-day-ahead forecasts using rolling window
    Evaluate predictions on validation set using RMSE, MAE, SMAPE
    Pick 1 best performing model from each modelling approach
    Evaluate predictions on unseen test set for 4 models
    Select best-performing model overall

The methodology framework in Algorithm 2 illustrates the end-to-end pipeline for our research task. The dataset selection, interpolation and spatial compression has already been discussed in Sections 3 and 4.

We now proceed with defining the structure of the datasets for our models: namely a 2D dataframe and its equivalent 4D tensor. (1) The 2D dataframe setup includes all the observations across time, latitude, longitude and the 8 variables. Next, the -1 land data and the -2 winter dates have been removed. These were identified using the mask -1 for land data and -2 for winter dates (Section 4). This data structure provides the ideal format for the baseline models, VAR per grid point, k-means + VAR and edge-aware GNN. (2) The 4D tensor setup involves 4 dimensions: time, latitude, longitude and variables; shown as $X \in \mathbb{R}^{1067\times63\times173\times8}$. Here, only the -2 winter dates have been removed. The -1 land data is preserved since we are having a 4D tensor with rectangular satellite images. This is done by creating a new empty 4D tensor with the given dimensions and then values have been

allocated to their respective coordinates and their days. See the visualisation of the 4D tensor in Figure 7. Thus, the land data is masked is zeroed out with the according land mask for all the images. This was used for the CNN models and factor model.

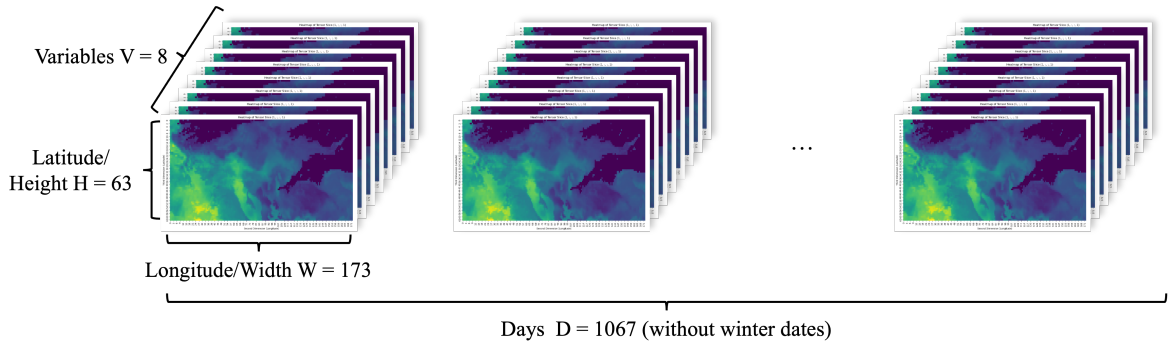


Figure 7: 4D Tensor $X \in \mathbb{R}^{1067\times63\times173\times8}$

After these dataframes and tensors are prepared in accordance with each model's requirements, we consistently follow the same train-validation-test split for all our models. These splits represent all time frames, where the -2 winter dates have been removed. This end up in 593+188+286 = 1067 days:

- **Training Set:** January 27, 2022 – November 17, 2023 (593 days)
- **Validation Set:** January 26, 2024 – July 31, 2024 (188 days)
- **Test Set:** August 01, 2024 – July 21, 2025 (286 days)

This temporal split ensures that models are evaluated on future unseen data, a standard practice in time series forecasting [Bergmeir and Benítez, 2018]. Also note that the train set consists of two blocks of continuously available daily data: block 1 runs from 2022-01-27 → 2022-11-20 and block 2 runs from 2023-01-27 → 2023-11-17, due to the exclusion of winter dates from the dataset. Similarly for the test set, the two continuous blocks are: 2024-08-01 → 2024-11-17 and 2025-01-26 → 2025-07-21.

After training the models on the continuous training blocks, we check the model performance on the validation test by deriving the one-step ahead forecasts, using the observed values for the lags involved in the model to predict the next day. We compute the forecast for each location separately, then compute the loss functions – MSE, MAE, SMAPE, and then average those loss function values over locations to get an overall prediction for each ocean parameter.

We then evaluated the models using three metrics: Symmetric Mean Absolute Percentage Error (SMAPE), Root Mean Squared Error (RMSE), and Mean Absolute Error (MAE), commonly used metrics in forecasting problems [Makridakis et al., 1993, Hyndman and Athanasopoulos, 2008]. SMAPE measures the average percentage difference between predicted and observed values while treating over- and under-predictions equally. RMSE measures the square root of the average squared difference between predictions and actual values, giving more weight to larger errors. MAE measures the average magnitude of errors without considering their direction. We chose these metrics because they cover percentage error

(SMAPE), error magnitude (MAE), and error magnitude with heavier penalty on large mistakes (RMSE), giving a balanced view of model performance. We calculate these metrics per variable, since we are interested in how well we can predict certain ocean variables. The metric formulations applicable to all models is as follows:

$$\mathrm{RMSE}_v \;=\; \sqrt{\frac{1}{N_v}\sum_{n\in\Omega_v}\big(\hat{y}_{v,n}-y_{v,n}\big)^2}, \qquad \mathrm{MAE}_v \;=\; \frac{1}{N_v}\sum_{n\in\Omega_v}\big|\hat{y}_{v,n}-y_{v,n}\big|,$$

$$\mathrm{SMAPE}_v \;=\; \frac{100}{N_v}\sum_{n\in\Omega_v}\frac{\big|\hat{y}_{v,n}-y_{v,n}\big|}{\frac{1}{2}\big(|\hat{y}_{v,n}|+|y_{v,n}|\big)}.$$

$$\Omega_v := \{n \mid \text{validation samples where variable } v \text{ has valid } (y,\hat{y})\}, \qquad N_v := |\Omega_v|.$$

Model performance is compared on the validation set – the top 4 models that achieve the highest forecasting accuracy on the validation set are promoted to a final round of evaluation. The reason for picking 4 models is the basis of their architecture. All 8 models presented are using one of the 4 main architectures and approaches to the forecasting task: baseline/naive-based models, autoregressive models, convolutional-based models, and graph-based models. In order to compare all 4 different approaches, we picked the best out of each category. These 4 models then compete on the test set, with the top performing model being the one that generalises best to unseen data.

6 Modelling I: Baseline models

Before using complex forecasting models like Vector AutoRegression (VAR), neural networks, or tensor-based approaches, it is important to start with a simple baseline model, which acts as a benchmark or reference point. It gives us the lowest level of model performance we can accept, helping us judge whether more advanced models are truly adding value or just increasing complexity and computing time[Reynolds et al., 2007, Wilkin and Arango, 2010]. A baseline model usually involves basic training and forecasting procedures using recent past data, such as averages or simply assigning the last observed values. In ocean modelling, simple methods such as trailing averages or exponential smoothing are often used when there is limited data or computing resources [Nau, 2005, OpenStax, 2020]. They can still give useful short-term forecasts for parameters like sea surface temperature, chlorophyll, or optics. In our study, where we aim to forecast daily changes in several ocean parameters across a wide spatial region, this baseline is essential as it helps us understand how predictable the system is and gives a clear foundation to compare our future, more complex models against.

6.1 Possible baseline approaches

Each of these methods listed below provide a different perspective on the predictability of the data. While none of them can handle very complex ocean behaviors, they give us a starting point to see whether advanced models (like deep learning or VAR) are truly needed and if they are making a meaningful difference.

- **Moving Average (MA):** As one of the simplest forecasting tools, it involves taking the average of the values from the past few days (e.g. last 7 days) and using that average to predict the next value. It helps smooth out some variability in the data and shows the overall trend. Moving averages are commonly used in oceanography to predict variables like wave height or sea surface temperature [Box et al., 1994].

- **Exponential Smoothing:** This method gives more weight to recent observations and less weight to older ones. In simple terms, it works like an average of past values, but instead of treating all days equally, it uses a formula that makes yesterday's value count more than the day before, which counts more than the day before that, and so on. This is done using an exponential formula that steadily reduces the weight for older data points. The result is a forecast that reacts more quickly to recent changes than a plain moving average. Its considered better than the moving average because it smoothly gives more weight to recent data while still considering all past data, making it more responsive and less abrupt Wikipedia [2025]. Exponential smoothing methods have been used to forecast sea levels and biological trends in the ocean [Chang and Su, 2016, Huang and Lin, 2019].

- **Naïve Forecasting:** The most basic approach - it assumes that the next day's value is the same as today's value. For example, if today's chlorophyll value is 0.3, the model predicts 0.3 for tomorrow. If the data has a seasonal pattern (like yearly cycles), the forecast might use last year's value for the same day. This means that if we know the data repeats in a yearly cycle - for example, ocean temperature in June this year is usually similar to June last year — then for forecasting June 15 this year, we would

just take the observed value from June 15 last year and use it as our prediction. Any advanced model should be able to perform better than this to be considered useful. This method is not considered a good approach as it just copies the last observed value as the prediction, without capturing seasonal changes, trends or spacial patterns.[Martinez and Zhou, 2021].

6.2 Chosen baselines

For this study, we selected two simple yet widely used forecasting methods as our baselines: the 7-day Moving Average and Exponential Smoothing. Both methods were applied on a per-grid-point basis after which predictions were stacked and averaged to produce the final forecast for each ocean variable. This means we looked at each small square in our ocean map separately. Each square has its own set of daily measurements over time. We ran the baseline method on each square's data to make a forecast for that location. Then we put all the location forecasts together and worked out the overall result for the whole area. Using separate grid-point processing ensures that local dynamics are captured rather than blending all areas into one bulk average.

The 7-day Moving Average was chosen for its ability to smooth short-term fluctuations and capture the underlying trend in the data using an equal weighting of recent observations [Box et al., 1994]. We used a 7-day window for the moving average as it matches a full weekly cycle, which is common in environmental and ocean data. This length smooths out short-term daily noise while still responding to recent changes, offering a good balance between stability and sensitivity [Makridakis et al., 1993, Hyndman and Athanasopoulos, 2008]. Exponential Smoothing was selected because it gives greater importance to more recent data points, making it more responsive to sudden changes or short-lived trends, while still maintaining a smoothed forecast [Brown, 1956].

Using both approaches allows us to capture two perspectives on predictability: one that is steady and trend-focused (moving average) and another that is more adaptive to recent changes (exponential smoothing). These baselines establish a clear lower bound for performance. Any advanced model—such as VAR, neural networks, or tensor-based approaches—must demonstrate consistent improvement over these baselines in terms of key evaluation metrics to be considered effective [Makarov and Clarke, 2021, Tippett and Anderson, 1995].

6.3 Implementation details

In this study, we used two simple forecasting methods as baselines: the 7-day Moving Average and Simple Exponential Smoothing. Although the forecasting formulas are different, both methods followed the same step-by-step process. This was done for both the validation and test sets so we could compare model behaviour in both phases.

First, we loaded the processed ocean dataset containing daily measurements for eight parameters: $\{\text{KD490}, \text{ZSD}, \text{RRS490}, \text{RRS443}, \text{CHL}, \text{MICRO}, \text{BBP}, \text{CDM}\}$. We removed all land locations by filtering out rows where the `flags` column had a value of 1.0 and excluded any rows in which any parameter took the placeholder value -2.0 (used in our dataset to represent missing observations; see Section 4.2.3). This filtering ensured that only valid ocean

measurements remained, with no missing or placeholder values in the data.

Next, we grouped the data by each location on the map (identified by its latitude and longitude). This meant that each location had its own sequence of daily values. For each location's data, we sorted it by date to ensure the time order was correct before applying the forecasting formulas.

For the 7-day Moving Average method, the forecast for the next day was the average of the previous 7 days for that location. This smooths short-term fluctuations and shows the overall trend. For the Simple Exponential Smoothing method, the forecast used a weighted average where more recent values had higher weight:

$$\hat{y}_{t+1} = \alpha x_t + (1-\alpha)\hat{y}_t, \quad 0 < \alpha < 1$$

where: x_t = actual observation at time t, $\hat{y}_t$ = forecast made for time t, α = smoothing parameter controlling the weight given to the most recent observation.

We fixed the smoothing parameter at $\alpha = 0.3$ because it lies within the commonly recommended range of 0.1–0.3 for Simple Exponential Smoothing Chopra and Meindl [2013]. This value provides a balanced trade-off: it is responsive enough to capture recent changes in ocean variables, while still smoothing out short-term fluctuations caused by noise or measurement errors. A lower α would make the model too slow to react to genuine changes, while a higher α would make it overly sensitive to random spikes.

After the predictions were generated for each location, for both moving average and exponential smoothing, they were then evaluated with the 3 discussed metrics: RMSE, MAE and SMAPE. Running these baselines on both validation and test sets allowed us to check whether the model that performed best during validation also achieved strong performance on the unseen test data. This comparison helps ensure that our model selection in validation was not overfitted to that split.

7 Modelling II: Vector AutoRegression models

Given the spatio-temporal nature of the oceanographic dataset, a natural starting point involves considering the statistical models that can capture both temporal and spatial effects, to allow for accurate forecasting of the ocean variables. Time series models such as AutoRegressive (AR) models are well-suited for extracting such temporal dependencies in univariate time series, since they estimate the current value of a variable as a linear function of its own past values [Enders, 2010]. This day-to-day temporal dependence is a reasonable expectation for oceanographic time series, where past conditions can strongly influence current states; we will investigate the degree of this influence in Section 7.2.

Extending this framework, Vector AutoRegressive (VAR) models allow us to jointly model multiple interdependent time series [Yan et al., 2021]. In our case, the eight variables - KD490, RRS443, MICRO, BBP, CDM, RRS490, ZSD, and CHL - can be modelled simultaneously, capturing not only their individual temporal dynamics but also the cross-variable interactions. For example, higher chlorophyll concentration (CHL) due to phytoplanktom booms may result in increased values of KD490, reflecting the quicker rate of light absorption [Morel and Prieur, 1977, Lee et al., 2002]. The VAR model could also incorporate both variable and spatial data, such that we have an flattened vector input for VAR with dimensions (time, spatial points x variables). Considering a time period of 593 days for training (Section 5), 8 variables and 10899 grid points (latitude x longitude), the number of parameters to be estimated explode – statistically infeasible.

To overcome this infeasibility, we can fit these multivariate VAR models individually on each grid point in the ocean. However, this does not explicitly address the spatial dependence. Hence, we consider three levels of spatial complexity in our models:

1. **Grid point-wise VAR modelling** – We fit a separate time series VAR model for each of the 10899 grid points. This method considers the temporal and inter-variable dependencies at each location in isolation, but does not introducing spatial dependencies across locations. It provides a benchmark for assessing whether including spatial correlations improves the forecasting performance.

2. **Clustered-based spatial aggregation: k-means + VAR** – We group grid points with similar temporal behaviour using K-means clustering. We then fit a single VAR model per cluster by averaging the time series within the cluster. While the VAR models the temporal dependencies, each series contains an aggregated and smoothened spatial signal; hence, the model implicitly considers spatial structure. While it preserves shared spatial patterns within the cluster, the per-grid point spatial resolution is lost. Regardless, this model helps observe the effects of reducing the dimensionality (fitting VAR on k clusters instead of 10899 grid points) while still capturing regional spatial structure.

3. **Factor model for tensor time series:** This approach applies Tucker decomposition to factorise the time, latitude and longitude dimensions into a unified low-dimensional space, to jointly learn spatial and temporal structure. The VAR model is only applied to the core tensor, which learns on useful latent signals, rather than noisy, high-dimensional observations (Grid point-wise VAR) or cluster averages (k-means + VAR).

7.1 Formulation and Assumptions of the Vector AutoRegression Model

A standard VAR(p) model with k variables can be written as:

$$\mathbf{y}_t = \mathbf{\Phi}_1 \mathbf{y}_{t-1} + \mathbf{\Phi}_2 \mathbf{y}_{t-2} + \cdots + \mathbf{\Phi}_p \mathbf{y}_{t-p} + \varepsilon_t,$$

where $\mathbf{y}_t$ is a $k \times 1$ vector of variables at time t, $\mathbf{\Phi}_i$ $(i = 1, \ldots, p)$ are $k \times k$ coefficient matrices, p is the number of lags (past values) and ε_t is a $k \times 1$ vector of white noise error terms. This formulation allows each variable to depend on both its own past values and the past values of all other variables in the system. Notably, autocorrelation in this context measures the correlation between a time series and a lagged version of itself at lag k. Partial autocorrelation measures the correlation between $\mathbf{y}_t$ and $\mathbf{y}_{t-k}$ after removing the influence of the intermediate lags $1, \ldots, k$. Note that validity of VAR-based inference and forecasting relies on these standard assumptions:

1. **Stationarity** – The statistical properties (mean, variance, autocorrelation) of the time series should remain constant over time to avoid spurious correlations [Granger and Newbold, 1974]. Stationarity will assessed in Section 7.2 using visual inspection, involving plotting the time series alongside its Autocorrelation and Partial Autocorrelation Functions (ACF, PACF) [Enders, 2010].

2. **Absence of autocorrelation in residuals** – Model residuals should resemble white noise. Significant autocorrelation in residuals indicates that temporal dependencies remain unmodeled, leading to inefficient forecasts and unreliable confidence intervals. This will be tested via residual ACF plots.

3. **Linearity** – VAR assumes a linear relationship between current and lagged variables. Nonlinearity can lead to systematic forecasting errors, indicating the need for nonlinear methods such as neural networks explored in Section 8.

Handling Time Discontinuity in Time Series: Another assumption that traditional time series models make is that there is a consistent time interval between observations; they expect continuously available data at the specified frequency. In our case we have daily data, however, the training set has two continuous blocks of data as seen in Section 5 due to the exclusion of winter dates in between. We are not interested in imputing the winter dates, because the satellite data is truly missing for 67 days. Therefore, from a modelling perspective, we are not interested in modelling and forecasting the variables for the winter period; instead, we would like the model to return ignorance when asked to predict on the winter dates – "*we cannot predict for this period!*".

However, we need to handle this discontinuity in the time series; if we train our VAR model without acknowledging the two-month winter gap, the model will incorrectly assume only a 1-day gap between November and January. This breaks the continuity assumption and leads to an artificial temporal relationships, leading to spurious correlations in ocean dynamics [Box et al., 1994]. Since the VAR model uses lagged observations to predict future values, our proposed solution is to drop the lagged observations bridging the winter gap. We treat each seasonal block, January to November of each year, as separate sequences, fitting 1 joint model across both the blocks, with only the lags present within each block. This process is illustrated in detail in Figure 8 below.

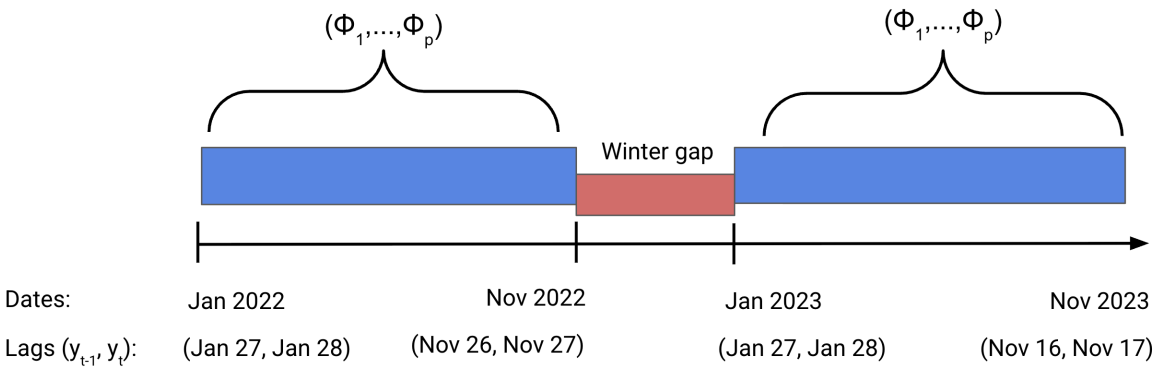


Figure 8: Time discontinuity shown with two continuous blocks and missing winter dates in between. The lag structure indicates that we only fit the lags on time points within each block, removing any cross-block lags. We also fit one joint model across all blocks, as indicated by the same $(\Phi_1, \ldots, \Phi_p)$ coefficient matrix on both continuous periods.

7.2 Per-grid point VAR Model

Before fitting a VAR model on the time series, we need to check if the stationarity assumption is met. In Section 3, we analysed the plots of the univariate, daily-average time series plotted for all 8 variables. Our conclusion was that RRS490, RRS443, BBP and CDM reflected a recurring annual pattern, while MICRO exhibited both annual trends as well as a change in the trend level with high variability in the spring and summer months. These annual trends or seasonality in the time series could be a problem for its stationarity.

Handling seasonality: We considered adding Fourier terms or seasonal dummies [Enders, 2010] as exogenous predictors to capture the annual seasonality pattern. However, before further inspection of the series using ACF plots, it is unclear at this stage whether seasonal effect or the persistent trend component is the dominant source of non-stationarity in our data. If the latter is true, then adding seasonal dummies alone would be insufficient. Another approach is to use seasonal or first differencing, which are transformations to the time series that remove seasonality and trend respectively, by subtracting previous values.

$$\begin{aligned} \text{First differencing:} \quad & \nabla y_t = y_t - y_{t-1} \\ \text{Seasonal differencing:} \quad & \nabla_s y_t = y_t - y_{t-s} \\ \text{First + seasonal differencing:} \quad & \nabla\nabla_s y_t = (y_t - y_{t-s}) - (y_{t-1} - y_{t-s-1}) \end{aligned}$$

If we suspect the existence of a trend, first differencing with lag = 1 can be used to eliminate that trend from the time series. Similarly, if we suspect that there is a seasonal annual trend, we can subtract the annual lag of 365. If both trend and seasonality are present, both transformations may be required to ensure a stationary time series.

Case Study on RRS443: We consider the the time series, ACF and PACF plots for the variable RRS443, where the first panel showcases those plots for the original series in Figure 9. The ACF plot helps identify patterns in the time series and shows how strongly it is correlated with its past lags. The 95% significance bounds (light blue) indicate which lags are statistically significant. For a stationary time series with no trend and seasonal pattern, the ACF plot would show rapidly declining correlation between the lags, towards zero. However, RRS443 has sinusoidal autocorrelations, with a set of significantly correlated lags around lag 300. This indicates seasonality; for our daily data, the set of significant lags

around day 300 imply annual seasonality. Note that we have approximately 2 months of winter data removed every year, causing the annual trend to be around lag 300 instead of 365. Secondly, there is significant autocorrelation up to the first 50 lags. In the presence of a trend, by definition, observations closer in time tend to have similar values, which explains the slowly decaying ACF. Therefore, given the mixture of both oscillating correlations and slowly diminishing lags, we seem to have a non-stationary time series with trend and seasonality.

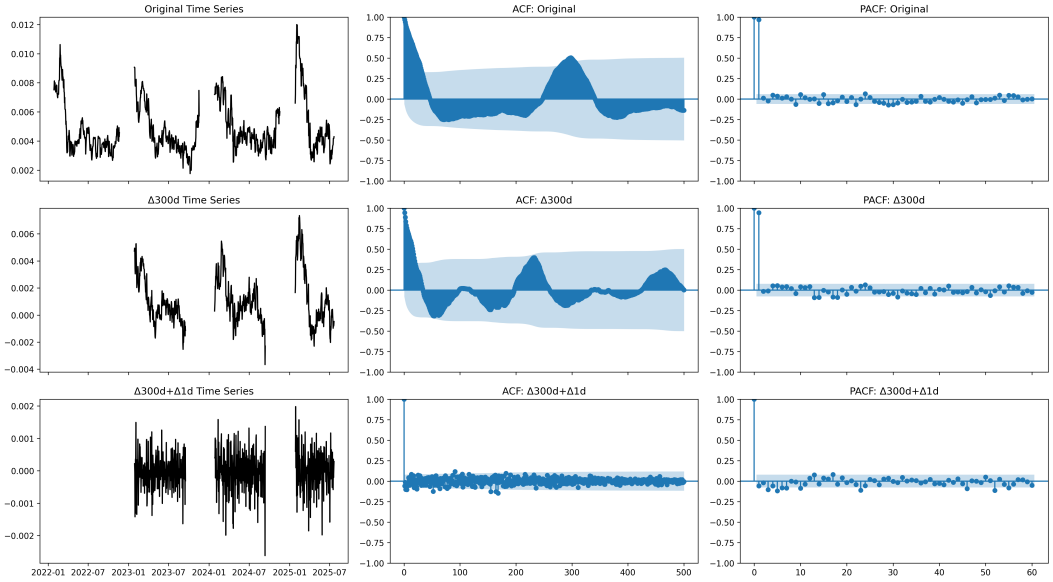


Figure 9: Time series, ACF and PACF plots for RRS443 for three cases: original, seasonally differenced and seasonally + first differenced time series.

Seasonal differencing followed by first differencing: Standard econometrics practice states that in the presence of seasonality, we first apply seasonal differencing [Enders, 2010]; if even after seasonal differencing there remains a slowly decaying or oscillating ACF, that suggests that the series is not yet stationary and we may also require first differencing to de-trend the series. This is precisely what we see in panels 2 and 3 of Figure 9; after seasonal differencing ($\nabla 300d$) we see that the ACF plot is still sinusoidal, and the first few lags still decay slowly. Thus, we apply first differencing on top of seasonal differencing ($\nabla 300d + \nabla 1d$) to see that now we have a stationary looking series with autocorrelations that appear like they are from a random walk in the ACF plot – suggesting that we need both seasonal and first differencing. Moreover, the PACF plot suggests the order of the AR model; we use an AR(1) model with only 1 lag, because that is the cut-off point for significant lags. This implies that there is a strong, significant day-to-day correlations between values today and yesterday, as expected in oceanographic data [Emery and Thomson, 2001].

Commutative order of differencing: An econometrics textbook conducted a similar seasonal + first differencing analysis with airplane passengers time series [Box et al., 1994]. They showed that the order in which we difference the series is commutative. They primarily applied first differencing to see if seasonal trends remain, and then applied seasonal differencing because the oscillating spikes at the seasonal lags remained. We conduct the same sanity check in Figure 10. The surprising result is that in this Figure, doing first differencing already removes the sinusoidal autocorrelations, as well as the dominant autocorrelations at the initial lags — no seasonal pattern is left, achieving a stationary-looking time series. This

most likely occurs because the seasonal component is weak, and thus doing first differencing captures both the trend and seasonal effects. Once stationarity is achieved, further differencing should be avoided because overdifferencing a time series introduces artificial serial correlation and increases model complexity [Box et al., 1994].

Based on Figure 9 and 10, we conclude that since the order of differencing is commutative, it suffices to do only first-differencing to make the time series stationary. Doing both first and seasonal differencing would overdifference the series.

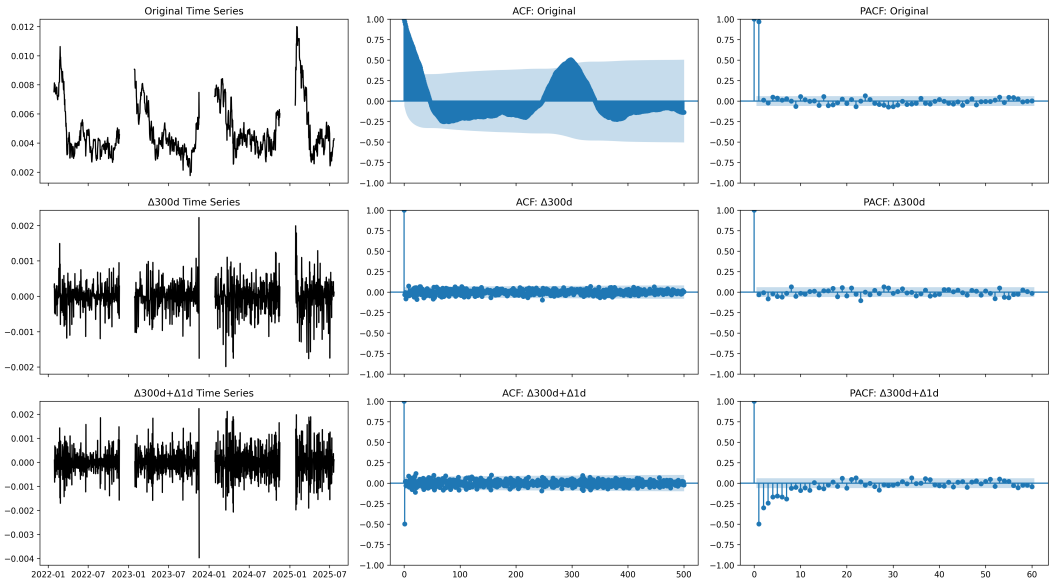


Figure 10: Time series, ACF and PACF plots for RRS443 for three cases: original, first differenced and first + seasonally differenced time series.

VAR with first-differenced series: By fitting the VAR(1) model on a differenced series and then reconstructing it back to the original scale, we gained a model that performed better on the validation set than the VAR(1) model fitted on the original series without first differencing, across all three metrics. This suggests that the non-stationarity in the original series was impairing model performance, and that first differencing removed trend components, producing more stable temporal dynamics – valuable for generalisability. Moreover, the residuals of the VAR(1) model with differencing behave like white noise, as shown in Figure 11. This illustrates that first differencing helped capture the non-stationary trend and seasonal components, since no such patterns remain in the residuals.

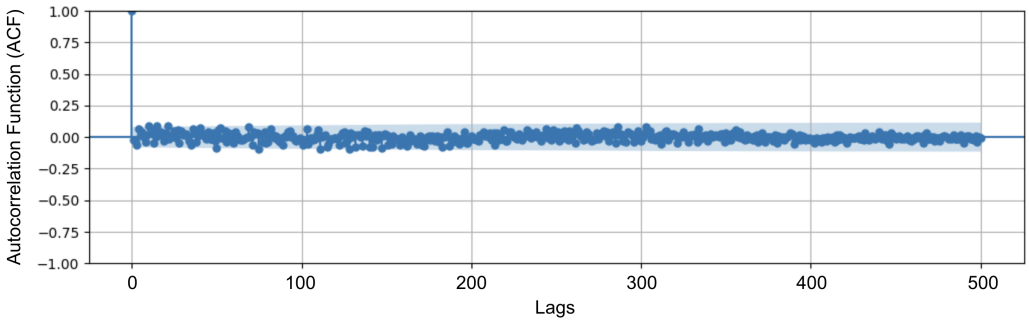


Figure 11: ACF plot for residuals from the VAR(1) model fitted on the original time series with no differencing.

Final per-grid point VAR model implementation: We proceed with fitting the standard VAR(1) model on the original time series with first differencing. We generate one-step ahead "rolling-origin" forecasts using lag-1 coefficients from the VAR(1) model fitted on the training data. We substitute the observed lagged values to predict the next time point.

7.3 k-means + Vector AutoRegression Model

We now consider the case of k-means clustering with the goal to incorporate some spatial information within the purely temporal time series framework so far. k-means is an unsupervised clustering algorithm that assigns each observation in the dataset to a cluster by minimising the within-cluster sum of squares (WCSS), defined as

$$\text{WCSS} = \sum_{i=1}^{k} \sum_{x \in C_i} \|x - \mu_i\|^2,$$

where μ_i is the centroid of cluster C_i. k-means is initialised by specifying a value for k, and then it iteratively allocates the observations to the nearest centroid, updating the centroids and terminating the search when WCSS cannot be reduced further. In our context, k-means will try to find homogeneous regions in the ocean that have similar variable values over time. Once the clusters are found, we can take the daily average across the grid points in each cluster, and then fit a VAR(1) model on each cluster. Finally, we generate one-step ahead forecasts for each grid point individually, using the VAR(1) coefficients from the VAR(1) model of its cluster. Algorithm 3 details the k-means + VAR procedure.

Optimal number of clusters: The number of clusters k to be used in k-means needs to be prespecified and can be selected via an *elbow plot*, as seen in Figure 12. It plots the WCSS versus k and it is evident that as the number of clusters k increases, the WCSS decreases but model complexity increases. Therefore, the elbow point at $k = 5$ or $k = 10$ clusters provides a good trade-off between having highly similar, compact clusters and model complexity; there are only diminishing returns to increasing k beyond that.

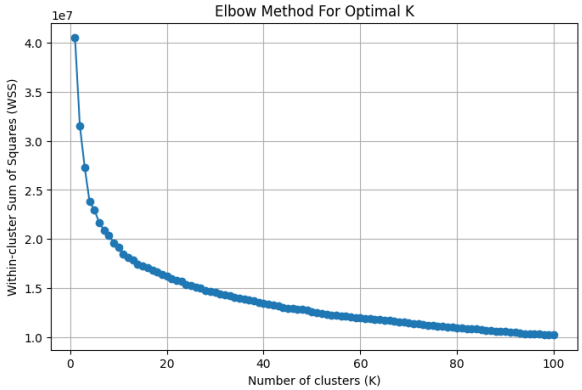
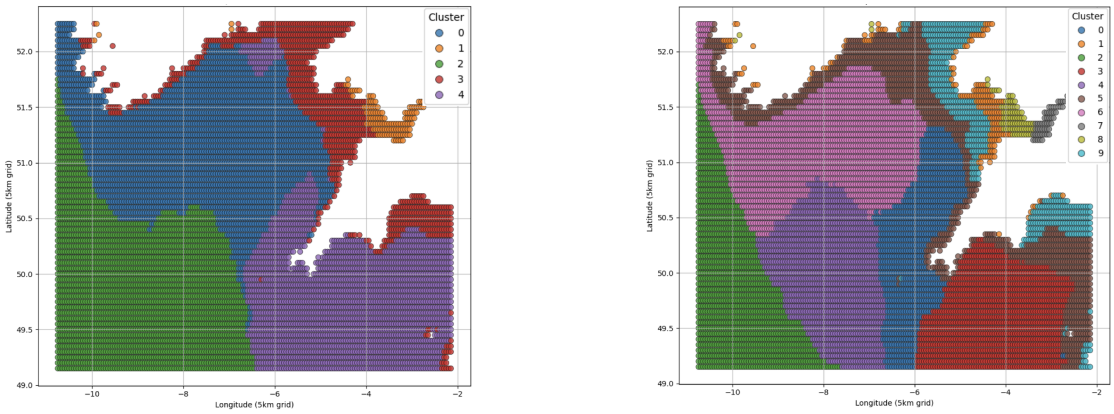


Figure 12: Elbow plot showing the trade-off between compact clusters and model complexity for k-means

The cluster maps shown in 13a for $k = 5$ and in 13b for $k = 10$ demonstrate how the spatial domain is partitioned into homogeneous regions by k-means. While $k = 5$ captures more general spatial trends, increasing k to 10 produces finer partitions capturing more localised variations. We ran k-means + VAR(1) on both $k = 5$ and $k = 10$, as well as

other k leading up to $k = 100$ to see how selecting the number of clusters affected validation performance. Notably, $k = 5$ performed best on the validation set better on the validation set in terms of all metrics, suggesting that increasing the granularity of the spatial clusters can lead to overfitting on the training data and reduced generalisation on the validation set.

Final k-means + VAR implementation: Choosing $k = 5$ offers a good balance between model complexity and prediction accuracy. It groups spatial points into meaningful, larger regions that capture dominant oceanographic patterns. Therefore, we fit the k-means + VAR model on $k = 5$ and 1 lag (Section 7.1).



(a) $k = 5$ (b) $k = 10$

Figure 13: Homogeneous regions identified by k-means clustering for $k = 5$ and $k = 10$

Algorithm 3 k-means Clustering + VAR Forecasting

Require: Dataset $X = \{x_1, x_2, \ldots, x_n\}$ with spatial coordinates $(\text{lat}_j, \text{lon}_j)$, time series data, number of clusters k

Ensure: Forecasts for each grid point

Clustering step:

Initialize k centroids $\mu_1, \ldots, \mu_k$ randomly
repeat
  for each observation x_j **do**
    Assign x_j to cluster C_i minimizing $\|x_j - \mu_i\|^2$
  end for
  for each cluster C_i **do**
    Update centroid $\mu_i = \frac{1}{|C_i|} \sum_{x_j \in C_i} x_j$
  end for
until centroids μ_i stabilize

VAR training per cluster:

for each cluster C_i **do**
  Identify grid points $G_i = \{(\text{lat}_j, \text{lon}_j) : x_j \in C_i\}$
  Aggregate time series within cluster by averaging across grid points: $\bar{X}_t^{(i)} = \frac{1}{|G_i|} \sum_{g \in G_i} X_t^{(g)}$
  Fit VAR model on $\{\bar{X}_t^{(i)}\}$ to estimate coefficients Φ_i
end for

Forecasting per grid point:

for each cluster C_i **do**
  for each grid point $g \in G_i$ **do**
    For each forecast time t:
    Use VAR coefficients Φ_i with observed lagged values from grid point g $\hat{X}_t^{(g)} = \Phi_i \cdot X_{t-1}^{(g)}$
  end for
end for
return Forecasts $\hat{X}_t^{(g)}$ for all grid points =0

7.4 Factor models for tensor time series

Factor models further extend the time series framework by integrating spatial, variable, and temporal structure into a low-dimensional latent space via low-rank approximation. This provides dimensionality reduction and allows VAR to be fitted on meaningful signals rather than high-dimensional noisy data. Unlike the k-means + VAR model where we decoupled space and time, this approach jointly models the spatial and temporal ocean dynamics.

Tucker Decomposition: The factor model assumes that our observed ocean data tensor $X \in \mathbb{R}^{d_1 \times d_2 \times d_3 \times d_4}$ (Section 5) can be decomposed into a smaller core tensor and four factor matrices, one for each mode (time, variables, latitude, longitude) via Tucker decomposition:

$$X = \mathcal{M} + \mathcal{E} = \mathcal{F} \times_1 U_{\text{time}} \times_2 U_{\text{var}} \times_3 U_{\text{lat}} \times_4 U_{\text{lon}} + \mathcal{E},$$

where $\mathcal{F} \in \mathbb{R}^{r_1 \times r_2 \times r_3 \times r_4}$ is the time-dependent core tensor, capturing the compressed latent dynamics of the ocean, which we aim to model over time. $U_{\text{time}} \in \mathbb{R}^{r_1 \times d_1}$, $U_{\text{var}} \in \mathbb{R}^{r_2 \times d_2}$, $U_{\text{lat}} \in \mathbb{R}^{r_3 \times d_3}$ and $U_{\text{lon}} \in \mathbb{R}^{r_4 \times d_4}$ are the factor matrices that capture the static structure – spatial, variable and temporal patterns that we assume remain constant across the entire dataset. Note: r_1, r_2, r_3, r_4 as the ranks that each matrix needs to be compressed to.

NaNs in the Tucker framework: While all winter date observations have been excluded from X, the observations at the land coordinates within the 63×173 spatial grid are NaNs imputed as -1 (Section 4). Tucker decomposition typically requires a dense tensor, but the TensorLy package allows for an in-built masking function – we can input an array of booleans with the same shape as tensor, that should be set to 0 where NaNs are present, and 1 everywhere else [user guide, 2024]. Tucker decomposition then incorporates this mask and effectively ignores the NaNs in optimisation, with no need to impute them. This way, we ensure that we do not have an arbitrary values set for land since that would make the Tucker decomposition unreliable.

Vector AutoRegressive Model on $\mathcal{F}$: We only perform Tucker decomposition on the training set, yielding $\mathcal{F} \in \mathbb{R}^{593 \times 8 \times 63 \times 173}$. Since $\mathcal{F}$ is much smaller than X, we can model its temporal evolution using a VAR model. For each time step we get $\mathcal{F}_t \in \mathbb{R}^{r_2 \times r_3 \times r_4}$, vectorise it to $f_t = vec(\mathcal{F}_t) \in \mathbb{R}^{r_2 r_3 r_4}$. We then stack these f_t over $t = 1, \ldots, r_1$ to form the input VAR matrix. After training the VAR(1) model, we can run predictions for $\hat{f}_{t+1}$ and reshape it to a tensor. Lastly, we derive the forecasted tensor $\hat{X}_{t+1}$ by using mode-k multiplication.

$$\begin{aligned}
\hat{f}_{t+1} &= A_1 f_t + A_2 f_{t-1} + \cdots + A_p f_{t-p+1} + \epsilon_{t+1} \\
\hat{\mathcal{F}}_{t+1} &= \text{reshape}(\hat{f}_{t+1}, (r_1, r_2, r_3)) \\
\hat{X}_{t+1} &= \hat{\mathcal{F}}_{t+1} \times_1 U_{\text{time}} \times_2 U_{\text{var}} \times_3 U_{\text{lat}} \times_4 U_{\text{lon}}
\end{aligned}$$

The rank selection for r_1, r_2, r_3, r_4 is critical and depends on the specific dataset. We want to retain information on all variables and thus $r_2 = d_2 = 8$. The time mode r_1 can be compressed from 593 to 150-200, latitude mode r_3 reduced from 63 to 5-15 and longitude mode r_4 compressed from 163 to 10-25. This helps reduce dimensionality, while also keeping sufficient data for the model to learn the patterns. Overall, we adopt factor models for time series with tucker decomposition, fit a VAR(1) model on the core tensor, do one-step ahead forecasting, and reconstruct the forecasted tensor to compute evaluation metrics. We simulate this process for various rank combinations.

8 Modelling III: Deep learning models

The reason why we explore neural networks applicability in the area of ocean variable forecasting is its great ability of generalization which lies in its core architecture. Neural networks consist of many layers which each have a set of neurons which can be represented by $ReLU(XW+b)$ where $X \in \mathbb{R}^{n \times d}$ is the data matrix and $W \in \mathbb{R}^{d \times k}$ and $b \in \mathbb{R}^k$ are the learnable weights and biases, respectively. Here n, d, and k denote the sample, features/dimensions and output features/classes respectively. Together, the matrix multiplication put through the non-linear activation function $ReLU = max(0, x)$ have the ability – with enough neurons per layer and enough layers – to approximate any non-linear function which is based on the Universal Approximation Theorem [Kidger and Lyons, 2020]. The workhorse here to optimize the W and b to approach any function is gradient descent using backpropagation and its chain rule [Rumelhart et al., 1986]. The ability to be applied in various tasks forms the hypothesis of testing these deep learning models in our ocean variable forecasting scenario.

8.1 ConvLSTM Network

8.1.1 ConvLSTM Deep Dive

Convolutional neural networks (CNNs) are able to build predictive models with neural networks when the data has the format of an image [LeCun, 1989, LeCun et al., 1998]. In the simplest form, CNNs use learnable filters (small images of 3x3 or 5x5 pixel size) which are moved, step-by-step, over all pixels from the input image. The filter takes then the dot product (similarity measure) of each covered area to summarize patterns of the image in a feature map [LeCun et al., 1998]. After training, the filters specialize to oceanographic signatures of all the images. Choosing smaller filters (e.g., 3×3) emphasizes fine structure, while larger ones (e.g., 5×5) capture broader trends. Pooling then summarizes the feature maps even more, before a task-specific head like an LSTM integrates the features [LeCun et al., 1998, 1989b]. Originally popularized for classification (handwritten digits, zip codes), the same convolutional backbone transfers directly to spatio-temporal forecasting: We keep the convolution–pooling feature extractor, replace the classifier with a regression head (e.g., linear outputs with a forecasting loss), and train end-to-end to predict the next satellite image [LeCun et al., 1998, 1989a].

Long Short–Term Memory (LSTM): The logical backbone of LSTMs are recurrent neural networks (RNNs) which model sequences by trying to learn the conditional probability distribution $p(x_t \mid x_{t-1}, \ldots, x_1)$ through a hidden state h_t that summarizes the past events and trends of the sequence. However, classic RNNs are not used in practice due to their vanishing/exploding gradient problem which arises due to the backpropagation through time and the recurrent multiplication of many gradients [Bengio et al., 1994, Pascanu et al., 2013]. LSTMs are the solution for this phenomenon. Refer to the architecture of a single ConvLSTM/LSTM cell in Figure 14. In order to save past information, LSTMs address this by adding a cell state C_t that is updated additively and regulated by different gates. By means of this, the C_{t-1} acts as long term memory and the input data X_t acts as short term memory. Taking both the long term and short term memory as inputs, we are able to: Forget past information in the long term memory, input new information of current events into the long term memory, and output a mix of both long term memory C_{t-1} and short term memory X_t.

The output, next image prediction, is the latent variable H_t. These functions are provided by the forget gate (F_t), input gate (I_t), and output gate (O_t) respectively [Hochreiter and Schmidhuber, 1997, Graves, 2012]. Figure 14 is depicting a LSTM cell which takes in spatial training data as $X_t \in \mathbb{R}^{Lag \times 63 \times 173 \times 8}$, making it a ConvLSTM. GRUs could have also been used as a gated RNN, however the usage of LSTMs is widely spread across hybrid models, which enabled us to build a ConvLSTM and is explained in the following [Chung et al., 2014, Cho et al., 2014].

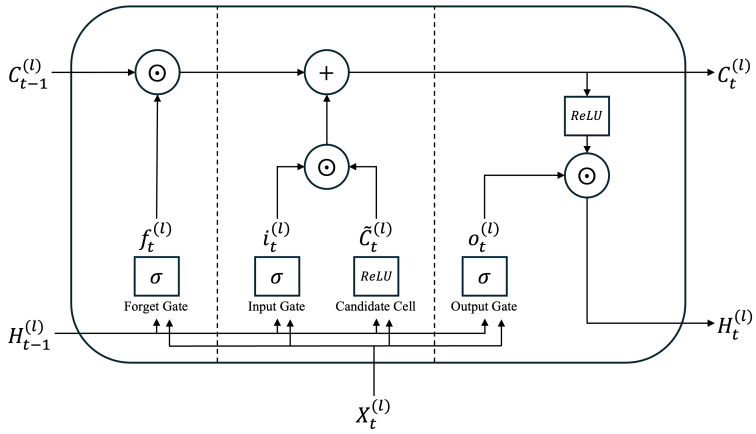


Figure 14: Single ConvLSTM Layer

$$
\begin{aligned}
f_t^{(\ell)} &= \sigma\Big(W_{xf}^{(\ell)} * X_t^{(\ell)} + W_{hf}^{(\ell)} * H_{t-1}^{(\ell)} + b_f^{(\ell)}\Big) && \text{(Forget Gate)} \\
i_t^{(\ell)} &= \sigma\Big(W_{xi}^{(\ell)} * X_t^{(\ell)} + W_{hi}^{(\ell)} * H_{t-1}^{(\ell)} + b_i^{(\ell)}\Big) && \text{(Input gate)} \\
\tilde{C}_t^{(\ell)} &= \mathrm{ReLU}\Big(W_{xg}^{(\ell)} * X_t^{(\ell)} + W_{hg}^{(\ell)} * H_{t-1}^{(\ell)} + b_g^{(\ell)}\Big) && \text{(Candidate Cell)} \\
o_t^{(\ell)} &= \sigma\Big(W_{xo}^{(\ell)} * X_t^{(\ell)} + W_{ho}^{(\ell)} * H_{t-1}^{(\ell)} + b_o^{(\ell)}\Big) && \text{(Output Gate)} \\
C_t^{(\ell)} &= f_t^{(\ell)} \odot C_{t-1}^{(\ell)} + i_t^{(\ell)} \odot \tilde{C}_t^{(\ell)} && \text{(Memory Cell)} \\
H_t^{(\ell)} &= o_t^{(\ell)} \odot \mathrm{ReLU}\big(C_t^{(\ell)}\big) && \text{(Latent Variable)}
\end{aligned}
$$

Here $*$ denotes 2D spatial convolution (stride 1, padding "same"), $\odot$ is elementwise product, σ as the sigmoidal function for the recurrent activation, and ReLU is the layer activation for the Candidate Cell. W and b depict respectively the learnable parameters, the weights and biases, for the given connections in the cell. Initial states of the latent variable and memory cell are H_0=0, C_0=0.

8.1.2 ConvLSTM Model

We want to explore a different approach, an alternative to autoregressive models, by applying CNN techniques to test whether these are feasible in the context of our oceanographic forecasting. Our first deep learning model is the Convolutional LSTM (ConvLSTM). This model was originally introduced for precipitation nowcasting [Shi et al., 2015] which showed that stacking ConvLSTM layers in an encoder–forecasting architecture consistently outperformed fully connected LSTMs, proving the point that it can capture spatio-temporal motion patterns. This acts as the initial hypothesis why that hybrid "convolution + memory" design could be well matched to geophysical sequence prediction like ocean variable forecasting

[Shi et al., 2015]. Following the dataset structure as mentioned in our Section 5, for the convolutional-based ConvLSTM, our input will be $X_t \in \mathbb{R}^{Lag\times 63\times 173\times 8}$ which resembles the satellite images (latitude=63 $\times$ longitude=173) for each 8 variables, for a sequence of past days $X_{t-1,\ldots,t-Lag}$. The output will be $\hat{Y} \in \mathbb{R}^{8\times 63\times 173}$ which tries to approximate $Y_t = X_{t+1}$. Regarding the correlation of previous days $T = t-1,\ldots,1$ to today $T = t$ which was explained in Section 7.3, we are choosing a $Lag = 1$ for our ConvLSTM model, since the single previous day $t-1$ correlates most with the next day t compared to all other previous days. According to the Algorithm Box 4 one can follow the structure of the ConvLSTM model.

(1) ConvLSTM layer $\ell \in \{1,2,3\}$: At time t, define the (per-layer) input X_t, the hidden state $H_t^{(\ell)} \in \mathbb{R}^{63\times 173\times C_\ell}$ and cell state $C_t^{(\ell)} \in \mathbb{R}^{63\times 173\times C_\ell}$ as shown in Figure 14. Convolutional recurrences preserve spatial resolution at every step, so locality is maintained while temporal dependencies are integrated by the LSTM dynamics. Hence, we stack three ConvLSTM layers (filters C_1=32, C_2=64, C_3=32) with filter/kernel sizes $k_3 = 5\times 5$, $k_2 = 3\times 3$, k_3=3×3 and same padding to preserve the spatial dimensionality of the same latitude and longitude. The first 5×5 filters captures broader structures and subsequent 3×3 filters refine local detail.

(2) Batch normalization and temporal outputs: Batch Normalization is a technique that normalizes the inputs to each layer in a neural network by adjusting and scaling them by their mean and variance to have a mean of zero and variance of one. It standardizes the inputs to each layer for every mini-batch, similar to how we might standardize raw data before feeding it into a model. It essentially ensures that regardless of how the previous layer's weights change during training, the next layer always receives inputs with a consistent distribution [Ioffe and Szegedy, 2015]. This stabilization ensures faster training with higher learning rates, reduces the sensitivity to weight initialization, and speeds the convergence of gradient descent. It is applied after each ConvLSTM layer:

$$\tilde{H}_t^{(\ell)} \;=\; \mathrm{BN}^{(\ell)}\big(H_t^{(\ell)}\big) \quad \Rightarrow \quad \text{the next layer uses } X_t^{(\ell+1)} = \tilde{H}_t^{(\ell)}.$$

Layers 1 and 2 emit $\{\tilde{H}_t^{(1)}\}_{t=1}^{T}$ and $\{\tilde{H}_t^{(2)}\}_{t=1}^{T}$. Layer 3 emits only the last step T after BN:

$$\tilde{H}^{(3)} \;=\; \mathrm{BN}^{(3)}\big(H_T^{(3)}\big) \in \mathbb{R}^{63\times 173\times 32}.$$

(3) Readout (1×1 convolution) Layer 3 returns only the last time step for the final readout. A 1×1 Conv2D head (linear) maps the final hidden state to $\hat{Y} \in \mathbb{R}^{8\times 63\times 173}$. The final prediction uses a channel-wise linear projection:

$$\hat{Y} \;=\; W_{\text{out}} * \tilde{H}^{(3)} + b_{\text{out}} \;\in\; \mathbb{R}^{8\times 63\times 173}.$$

Algorithm 4 ConvLSTM Network Model

Require: Input tensor $X_t^{(1)} \in \mathbb{R}^{Lag \times 63 \times 173 \times 8}$ with the amount of Lag days for day t, spatial dimensions $(63, 173)$, and 8 channels
Ensure: Predicted ocean variable $\hat{Y}_t \in \mathbb{R}^{63 \times 173 \times 8}$
3 ConvLSTM layers:
$H_t^{(1)} \leftarrow \text{ConvLSTM}(X_t^{(1)}, \text{filters} = 32, \text{kernel} = 5 \times 5, \text{padding} = \text{same},$
    $\text{activation} = \text{ReLU}, \text{return\_sequences} = \text{True})$ // Output: $\mathbb{R}^{Lag \times 63 \times 173 \times 32}$
$\tilde{H}_t^{(1)} \leftarrow \text{BatchNorm}(H_t^{(1)})$
$H_t^{(2)} \leftarrow \text{ConvLSTM}(\tilde{H}_t^{(1)}, \text{filters} = 64, \text{kernel} = 3 \times 3, \text{padding} = \text{same},$
    $\text{activation} = \text{ReLU}, \text{return\_sequences} = \text{True})$ // Output: $\mathbb{R}^{Lag \times 63 \times 173 \times 64}$
$\tilde{H}_t^{(2)} \leftarrow \text{BatchNorm}(H_t^{(2)})$
$H_t^{(3)} \leftarrow \text{ConvLSTM}(\tilde{H}_t^{(2)}, \text{filters} = 32, \text{kernel} = 3 \times 3, \text{padding} = \text{same},$
    $\text{activation} = \text{ReLU}, \text{return\_sequences} = \text{False})$ // Output: $\mathbb{R}^{63 \times 173 \times 32}$
(returns only last timestep)
$\tilde{H}_t^{(3)} \leftarrow \text{BatchNorm}(H_t^{(3)})$
Output layer: final convolution for channel/variable mapping
$\hat{Y}_t \leftarrow \text{Conv2D}(\tilde{H}_t^{(3)}, \text{filters} = 8, \text{kernel} = 1 \times 1, \text{padding} = \text{same},$
    $\text{activation} = \text{linear})$ // Output: $\mathbb{R}^{63 \times 173 \times 8}$
return $\hat{Y}_t$

Training: We trained the ConvLSTM with standard deep-learning practices: mini-batches, sensible initialization, and normalization to keep activations well-behaved. Among optimizers, we considered AdaGrad and RMSProp, but defaulted to Adam because it combines per-parameter learning-rate scaling (like AdaGrad) and exponential average of squared gradients (like RMSProp). In effect, when a parameter's gradient keeps the same sign, Adam maintains steady updates. When signs change constantly, it naturally shrinks the step size, which helps the model settle into a minimum of the loss function [Duchi et al., 2011, Hinton et al., 2012, Kingma and Ba, 2015]. For initialization of the weights and biases we used He (Kaiming) initialization, sampling each layer's weights from a normal distribution with variance $2/n_{l-1}$ where n_l is the number of neurons of the first layer in a neural network [He et al., 2015]. To stabilize optimization, we normalize hidden activations. Batch Normalization keeps each mini-batch's pre-activations in a comparable range and typically permits larger learning rates. For the loss function, the classic mean squared error (MSE) has been used.

8.2 TACNN (CNN + Temporal Attention)

8.2.1 TACNN Deep Dive

After testing the gated approach of a ConvLSTM we want to explore another temporal information preservation technique, widely used in sequential modelling, being attention. We want additionally to explore the Temporal Attention + CNN (TACNN) model. The CNN first encodes today's image, then the attention head learns where to look in those filters to assemble the forecast $\hat{Y}_t$. Instead of pushing everything through fixed convolutional layers and recurrent layers like in the ConvLSTM, attention assigns weights to the most predictive spatio-temporal regions and shows what it relied on via attention maps. The CNN handles spatial structure, but it doesn't decide which days matter most. Coming to the temporal attention, it looks across the whole row of all filters and learns a set of weights. Important

days get high weight, routine or noisy ones get low weight. The model then forms a weighted summary of the sequence. In effect, attention is a learnable spotlight over time, since it highlights the few timesteps that drive the target and lets gradients flow through them, so the CNN sharpens those specific patterns during training [Vaswani et al., 2017, Wang et al., 2021a, Liu et al., 2020, Wang et al., 2021b]. The intuition why a TACNN could be applicable in our ocean forecasting environment is that the CNN gives strong, spatially aware features at each step, while temporal attention preserves temporal information. It can be run causally for forecasting, and it scales well because attention can see the whole window at once [Lin et al., 2021, Zhang et al., 2024].

8.2.2 TACNN Model

The Algorithm Box 5 gives a step-by-step description of the TACNN model. Given a window of $Lag = 1$ days with $C = 8$ channels each, $X_t \in \mathbb{R}^{Lag\times 8\times 63\times 173}$, the model predicts the next frame's $C = 8$ channels, $\hat{Y} \in \mathbb{R}^{63\times 173\times 8}$, which tries to come close to $Y_t = X_{t+1}$. Our TACNN processes ocean data in three steps. First, it looks at each image from all days and compresses them into smaller, meaningful representations using the same processing rules for each image - ensuring all images are comparable. The compression starts at the initial dimensionality of $X_t \in \mathbb{R}^{Lag\times 63\times 173\times C}$ and the halves longitude and latitude per layer to $U_t^{(1)} \in \mathbb{R}^{Lag\times 31\times 86\times C}$ and then $U_t^{(2)} \in \mathbb{R}^{Lag\times 15\times 43\times C}$, where division without a remainder is applied. Each image/time step is then flattened to a $F_t \in \mathbb{R}^{Lag\times 82560}$ (where $15 * 43 * *128 = 82560$, 128 being the embedding dimension) so the attention mechanism can compare each of those compressed image parts. Second, an attention mechanism examines all these compressed images together to figure out which parts of past days are most important for making predictions - it's like the model asking "which previous ocean days should I pay most attention to?". The model keeps the final processed information as a summary of everything it learned. Third, a decoder takes this compact summary of important days and expands it back to the original dimension size ($\mathbb{R}^{63\times 173\times C}$), then converts it to predict our eight ocean variables. According to the Algorithm Box 5 the model can be described as follows:

(1) Encoder (spatio-temporal feature extraction): The encoder progressively compresses the spatio-temporal input $X_t^{(1)} \in \mathbb{R}^{Lag\times 63\times 173\times 8}$ into a compact representation. Two Conv2D layers with 1×1 and 3×3 filter sizes extract hierarchical features while MaxPool2D operations halve the spatial dimensions by factors of 2. The first layer uses 1×1 convolutions with 64 filters to perform channel-wise feature mixing, while the second layer's 128 filters of size 3×3 capture local spatial patterns. After each pooling operation, batch normalization stabilizes the feature distributions:

$$\tilde{U}_t^{(1/2)} = \mathrm{BN}^{(1)}\big(U_t^{(1/2)}\big) \in \mathbb{R}^{Lag\times 31\times 86\times 64}, \quad \tilde{U}_t^{(2/4)} = \mathrm{BN}^{(2)}\big(U_t^{(2/4)}\big) \in \mathbb{R}^{Lag\times 15\times 43\times 128}.$$

The spatial features are then flattened to $F_t \in \mathbb{R}^{Lag\times 82560}$ and compressed through a dense layer to create temporal embeddings $E_t \in \mathbb{R}^{Lag\times 128}$, preserving the temporal sequence while encoding spatial information into a fixed-size representation.

(2) Multi-Head Attention Mechanism: The attention module looks at all the time steps at once (rather than one after another) to figure out which past moments matter most for prediction. Think of it as having 4 different "experts" (the 4 attention heads) simultaneously

examining the temporal data E_t, each looking for different patterns - one might focus on recent changes, another on long-term trends. Instead of treating all past time points equally, the model learns to give more weight to the important ones and less to irrelevant ones. It's like the model asking: "Which past ocean conditions best help me predict what happens next?" The final output is a single vector $c \in \mathbb{R}^{128}$ that summarizes all the important information:

$$c = \text{MultiHeadAttention}(E_t, \text{heads} = 4) \in \mathbb{R}^{128}.$$

This selective focus allows the model to automatically identify which historical ocean states are most useful for forecasting, rather than treating all past observations as equally important.

(3) Decoder (Spatial Reconstruction): The decoder reconstructs the full spatial resolution from the compressed representation c. First, a dense, fully connected neural network layer projects the context vector to match the now compressed spatial dimensions: $z = \text{Dense}(c, H/4 \times W/4 \times 64)$, which is reshaped to $Z \in \mathbb{R}^{15\times43\times64}$. Two transposed convolution layers (Conv2DTranspose) with stride 2 progressively upsample (or "decompress") the spatial dimensions, using 32 and 16 filters respectively with 3×3 filter sizes:

$$Z_t^{(1)} \in \mathbb{R}^{31\times86\times128} \xrightarrow{\text{Conv2DTranspose}} Z_t^{(2)} \in \mathbb{R}^{60\times172\times16}.$$

A padding adjustment layer with 4×2 filter size ensures exact spatial alignment to the target dimensions $\mathbb{R}^{63\times173\times8}$. Finally, a 1×1 convolution with linear activation produces the predicted ocean variables:

$$\hat{Y}_t = W_{\text{out}} * \tilde{Z}_t^{(2)} + b_{\text{out}} \in \mathbb{R}^{63\times173\times8}.$$

Algorithm 5 TACNN (Temporal Attention Convolutional Neural Network) Architecture

Require: Input tensor $X_t^{(1)} \in \mathbb{R}^{Lag \times 63 \times 173 \times 8}$ with the amount of Lag days for day t, spatial dimensions $(63, 173)$, and 8 channels
Ensure: Predicted ocean variable $\hat{Y}_t \in \mathbb{R}^{63 \times 173 \times 8}$
Encoder:
  $U_t^{(1)} \leftarrow \text{Conv2D}(X_t^{(1)}, \text{filters} = 64, \text{filter size} = 1 \times 1, \text{activation} = \text{ReLU})$
  $U_t^{(1/2)} \leftarrow \text{MaxPool2D}(U_t^{(1)}, \text{pool\_size} = 2 \times 2)$ // Output: $\mathbb{R}^{Lag \times 31 \times 86 \times 64}$
  $\tilde{U}_t^{(1/2)} \leftarrow \text{BatchNorm}(U_t^{(1/2)})$
  $U_t^{(2)} \leftarrow \text{Conv2D}(\tilde{U}_t^{(1/2)}, \text{filters} = 128, \text{filter size} = 3 \times 3, \text{activation} = \text{ReLU})$
  $U_t^{(2/4)} \leftarrow \text{MaxPool2D}(U_t^{(2)}, \text{pool\_size} = 2 \times 2)$ // Output: $\mathbb{R}^{Lag \times 15 \times 43 \times 128}$
  $\tilde{U}_t^{(2/4)} \leftarrow \text{BatchNorm}(U_t^{(2/4)})$
  $F_t \leftarrow \text{Flatten}(\tilde{U}_t^{(2/4)})$ // Output: $\mathbb{R}^{Lag \times 82560}$
  $E_t \leftarrow \text{Dense}(F_t, \text{units} = 128)$ // Output: $\mathbb{R}^{Lag \times 128}$
Attention Mechanism:
  $c \leftarrow \text{MultiHeadAttention}(E_t, \text{heads} = 4)$ // Take last timestep, Output: $\mathbb{R}^{128}$
Decoder:
  $z \leftarrow$ Dense, fully connected NN$(c, \text{units} = H/4 \times W/4 \times 64)$ // $H = 63, W = 173$
  $Z \leftarrow \text{Reshape}(z, \text{shape} = (15, 43, 64))$ // Output: $\mathbb{R}^{15 \times 43 \times 64}$
  $Z_t^{(1)} \leftarrow \text{Conv2DTranspose}(Z, \text{filters} = 32, \text{filter size} = 3 \times 3, \text{stride} = 2, \text{activation} = \text{ReLU})$ // Output: $\mathbb{R}^{31 \times 86 \times 128}$
  $Z_t^{(2)} \leftarrow \text{Conv2DTranspose}(Z_t^{(1)}, \text{filters} = 16, \text{filter size} = 3 \times 3, \text{stride} = 2, \text{activation} = \text{ReLU})$ // Output: $\mathbb{R}^{60 \times 172 \times 16}$
  $\tilde{Z}_t^{(2)} \leftarrow \text{Conv2D}(Z_t^{(2)}, \text{filters} = 16, \text{filter size} = 4 \times 2, \text{padding} = \text{valid}, \text{activation} = \text{ReLU})$ // Output: $\mathbb{R}^{63 \times 173 \times 16}$
  $\hat{Y}_t \leftarrow \text{Conv2D}(\tilde{Z}_t^{(2)}, \text{filters} = 8, \text{filter size} = 1 \times 1, \text{padding} = \text{same}, \text{activation} = \text{linear})$
return $\hat{Y}_t$

Training: For training, we used again the classical MSE as the loss function and the Adam optimizer in order to improve the gradient descent algorithm by letting the algorithm behave more stable and focused in its process of convergence.

8.3 Edge-Aware GNN + LSTM

8.3.1 Edge-Aware GNN + LSTM Deep Dive

Our methodology extends spatio-temporal modelling by introducing the Edge-Aware GNN + LSTM for Multi-Variable Ocean Forecasting model, designed to address the challenges of high-dimensional spatio-temporal oceanographic data, including computational burdens and missing values. While traditional statistical approaches such as Autoregressive (AR) models have limited applicability to high-dimensional spatial data, and Convolutional LSTMs (ConvLSTMs) struggle with irregular grids containing land or islands, Graph Neural Networks (GNNs) excel in these scenarios by naturally handling irregular structures and strong interdependencies. This work builds upon the foundation laid by [Liang et al., 2023] , who proposed a Graph Memory Neural Network (GMNN) for sea surface temperature (SST) forecasting, notable for its ability to encode irregular data using both node and edge information.

Tailoring the GMNN for Multi-Variable Forecasting While our approach draws inspiration from the GMNN framework introduced by Liang et al. [2023], it has been adapted to better suit our dataset, forecasting objectives, and computational constraints.

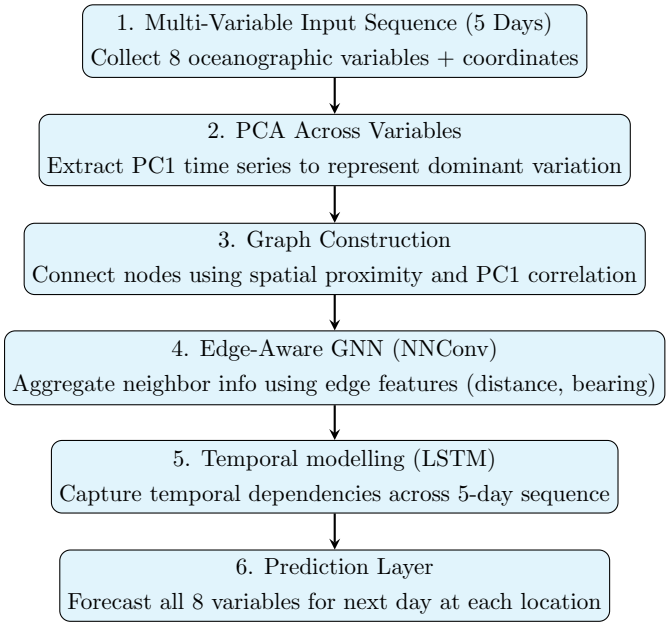
- **Multi-Variable Forecasting:** The original GMNN was designed for single-variable SST prediction. In contrast, our model predicts eight oceanographic variables simultaneously (KD490, ZSD, RRS490, RRS443, CHL, MICRO, BBP, and CDM), enabling a broader ecological and physical interpretation of ocean conditions.

- **Graph Construction Tailored to Multi-Variable Data:** Instead of computing correlations on a single variable, we first apply Principal Component Analysis (PCA) across all variables for each location and use the first principal component (PC1) time series to calculate Pearson correlation coefficients (PCC). This provides a more representative measure of similarity between locations in a multi-variable setting. Unlike our earlier VAR-based experiments that used Tucker decomposition for multi-way temporal–spatial–feature structure, here PCA was preferred as we only needed a single dominant component per location; PCA provides a stable closed-form solution with fewer hyperparameters and lower computational cost, whereas Tucker requires multi-mode rank tuning

- **Simplified Architecture:** We omit the GMNN memory module to reduce complexity and training time. This design choice prioritises interpretability and computational efficiency over additional long-term context storage.

- **Edge-Aware Message Passing:** Spatial relationships are modelled with NNConv layers, which incorporate edge features such as distance and bearing when aggregating information from neighbouring nodes.

These adjustments are motivated by the characteristics of our dataset irregular spatial coverage, multiple interacting variables, and the need for efficient training on large time series rather than by an attempt to outperform the original GMNN in all respects. .



8.3.2 Edge-Aware GNN + LSTM Model

The architecture consists of a graph encoder for spatial dependencies, followed by an LSTM for temporal modelling, and a fully connected output layer for multi-variable forecasting.

1. Graph Encoder (Edge-Aware GNN / NNConv Layers) At each time step t, node features $X_t \in \mathbb{R}^{N \times \text{input\_dim}}$ (with input_dim = 10) are processed alongside edge features e_{ij} (bearing and distance, normalised). The NNConv layer updates node representations as:

$$h_v^{(l+1)} = \sigma\left(h_v^{(l)} + \sum_{u \in N(v)} \text{MLP}(e_{vu}) \cdot h_u^{(l)}\right)$$

where $h_v^{(l)}$ is the feature vector for node v at layer l, $N(v)$ is its neighborhood, $\text{MLP}(e_{vu})$ maps edge attributes to a weight matrix, and σ is a ReLU activation. Two NNConv layers are used, with multiple iterations for enhanced spatial feature extraction.

2. Temporal Encoder (LSTM) The sequence of graph-encoded features $g_{seq} \in \mathbb{R}^{T \times N \times \text{hidden\_dim}}$ is permuted to $\mathbb{R}^{N \times T \times \text{hidden\_dim}}$ and passed to an LSTM to model temporal dependencies. The output from the last time step h_{last} is retained for prediction.

3. Output Layer A fully connected layer maps h_{last} to the prediction $\hat{Y} \in \mathbb{R}^{N \times 8}$ for all oceanographic variables:

$$\hat{Y} = \text{FC}(h_{\text{last}})$$

4. Loss Function Training minimises the mean squared error (MSE) over all spatial locations (i, j) and variables c:

$$L = \frac{1}{HWC} \sum_{i=1}^{H} \sum_{j=1}^{W} \sum_{c=1}^{C} \left(\hat{X}_{T+1,i,j,c} - X_{T+1,i,j,c}\right)^2$$

where H, W, C are height, width, and channel count (variables) respectively.

Training Setup and Evaluation

The model is trained with the Adam optimizer (learning rate 1×10^{-3}) using sequences of length `seq_len` = 5 days to predict the next day's variables. Performance is evaluated with SMAPE, RMSE, and MAE for individual grid points and then aggregated per variable. Memory optimisation strategies, such as memory-mapped arrays and chunked processing, are used to handle the dataset efficiently.

9 Model comparison

9.1 Validation set evaluation

We explored four categories of models: baseline, time series, convolutional-based and GNN-based models. Each category contains several variants; baseline models include moving average and exponential smoothing, time series models such as VAR, k-means + VAR and factor model, convolutional-based include ConvLSTM and TACNN, and finally edge-aware GNN. We now evaluate these models on the validation set to determine which four models, one from each category, would serve as the final competing models for the best generalisable performance on the test set. The comparison was made using three key metrics: SMAPE, RMSE and MAE as mentioned in Section 5.

Table 1: Validation-set metrics for all models. Bold indicates the best value within each comparable group.

Variable	Metric	Method						
		MA	Exp. Smooth	VAR	k-means+VAR	ConvLSTM	TACNN	Edge-Aware GNN
KD490	SMAPE	13.21%	**10.81%**	9.39%	**2.47%**	**17.19%**	24.78%	**4.58%**
	RMSE	0.0301	**0.0258**	0.0290	**0.0123**	**0.0224**	0.0669	**0.0242**
	MAE	0.0144	**0.0117**	0.0100	**0.0025**	**0.0106**	0.0287	**0.0102**
ZSD	SMAPE	15.10%	**12.33%**	9.88%	**1.97%**	**18.42%**	26.46%	**5.33%**
	RMSE	2.0983	**1.7683**	1.7910	**0.4644**	**1.4889**	3.4596	**1.4768**
	MAE	1.4666	**1.1878**	0.8680	**0.1445**	**1.0641**	2.6205	**0.9834**
RRS490	SMAPE	21.67%	**17.89%**	14.12%	**3.94%**	**17.89%**	27.37%	**8.62%**
	RMSE	0.0019	**0.0016**	0.0020	**0.0005**	**0.0011**	0.3894	**0.0013**
	MAE	0.0012	**0.0010**	0.0010	**0.0002**	**0.0007**	0.1227	**0.0009**
RRS443	SMAPE	25.77%	**21.95%**	18.25%	**5.31%**	**21.41%**	28.33%	**10.47%**
	RMSE	0.0018	**0.0016**	0.0020	**0.0005**	**0.0012**	0.3904	**0.0014**
	MAE	0.0013	**0.0010**	0.0010	**0.0002**	**0.0008**	0.1282	**0.0010**
CHL	SMAPE	24.04%	**19.93%**	17.54%	**6.80%**	**21.92%**	31.25%	**10.01%**
	RMSE	0.9646	**0.8381**	0.9420	**0.3957**	**0.7588**	1.2206	**0.8304**
	MAE	0.3398	**0.2785**	0.2410	**0.0726**	**0.2526**	0.4645	**0.2616**
MICRO	SMAPE	65.14%	**60.36%**	45.27%	**31.49%**	**52.32%**	59.90%	**43.07%**
	RMSE	4.4576	**3.9746**	3.8830	**1.7502**	**3.3434**	5.9750	**4.2726**
	MAE	1.4209	**1.2130**	0.8870	**0.3630**	**1.2376**	2.6757	**1.3977**
BBP	SMAPE	31.04%	**25.46%**	19.67%	**7.53%**	**27.38%**	34.75%	**12.95%**
	RMSE	0.0056	**0.0047**	0.0050	**0.0019**	**0.0044**	0.3956	**0.0046**
	MAE	0.0026	**0.0021**	0.0020	**0.0004**	**0.0021**	0.1272	**0.0020**
CDM	SMAPE	27.79%	**23.80%**	22.58%	**12.93%**	**27.35%**	32.62%	**14.47%**
	RMSE	0.0900	**0.0768**	0.0800	**0.0340**	**0.0747**	0.3909	**0.0820**
	MAE	0.0222	**0.0184**	0.0170	**0.0079**	**0.0210**	0.1516	**0.0209**

(1) Baseline approaches: We evaluated two baseline models moving average and exponential smoothing. The results, summarised in Table 1, show that exponential smoothing consistently outperformed moving average across most variables for SMAPE, RMSE and MAE. In particular, it achieved lower SMAPE for 7 out of 8 variables, and had smaller RMSE and MAE in all cases. Its ability to assign higher weight to recent data allowed it to better capture underlying trends without over-smoothing short-term variations. This is

validated by our findings in VAR modelling in Section 7.2, where the PACF plots showed strong correlation between the value of a variable today and its value yesterday. Hence, we select exponential smoothing as the preferred baseline model for the test set evaluation.

(2) Autoregressive approaches: Regarding the VAR and k-means + VAR models, Table 1 shows that k-means + VAR has consistently outperformed VAR performed on each grid point, across all metrics. Even compared to other models, it offers the lowest reduction in SMAPE not just for MICRO, but for RRS443, BBP and CDM as well. These results can be explained by the fact that k-means most likely benefitted from the noise reduction via pooling the grid points. Individual grid points have noisy time series, and thus fitting a VAR to each point could overfit to local noise. On the other hand, clustering smooths the data by grouping points with similar temporal behaviour, leading to more stable parameter estimation. Additionally, a cluster-level VAR better captures the shared spatial patterns than the independent pointwise VAR model. Therefore, we choose k-means + VAR as the best VAR model, and will thus be used for the test set evaluation.

Considering the factor model for tensor time series, we used tucker decomposition to extract a smaller, latent representation for the ocean data tensor, suitable for fitting a VAR model and forecasting. However, we observed that while Tucker decomposition successfully enabled a low-rank approximation for various combinations of ranks across the four dimensions (time, variables, latitude, longitude), we were unable to fit meaningful VAR models given the limited timeframe of 593 training days. We mentioned in Section 7.4 that to apply VAR, we must flatten the core tensor into a vector of shape (time, features), where the number of features is the product of the remaining ranks: r_{var} x r_{lat} x r_{long}. For instance, with $r_{\text{time}} = 50$, $r_{\text{var}} = 8$, $r_{\text{lat}} = 5$ and $r_{\text{lon}} = 10$, the flattened matrix becomes $(50, 400)$. Fitting a VAR on this vector fails because we only have 50 time observations to fit a model with 400 variables — the residual covariance matrix not positive definite and thus we cannot find a numerically stable solution for that many components. We analysed various ranks and observed that we need to have at least 3 times the amount of observations than features for VAR to work. If we consider the extreme case of setting the time rank to 593, the product or spatial ranks should be at most 24 based on our rule of thumb (593, 8x24). We tried 10 variations of spatial ranks under these constraints, and none outperformed our baseline model. Therefore, the Tuker-based factor modelling, while promising in theory, is not a feasible approach for building a competitive forecasting model, due to our high-dimensional dataset and limited training timeframe bottlenecks.

(3) Convolutional-based approaches: The ConvLSTM performed better when compared to the TACNN in all the metrics, SMAPE, RMSE, and MAE out of the 2 convolutional based approaches which leads to the choice of the ConvLSTM which will be chosen as the convolutional model used for the test set, as shown in the validation metrics in Table 1. Although attention is powerful and faster in training due to its parallelism, our TACNN model carries inductive biases and bottlenecks that seem misaligned with spatio-temporal ocean forecasting, which helps explain why its model performance trails behind ConvLSTM. By flattening each frame (15×43×128) into a 128-dimensional vector, the encoder compresses away fine-scale spatial structure that the ConvLSTM preserves through its convolutional hidden state. Temporal self-attention then operates over these per-frame vectors (not spatial tokens), so it can decide when something matters but not where it moved. In contrast, the

ConvLSTM's state-to-state convolutions explicitly track motion on the grid. Because we also omit temporal positional encodings, attention becomes nearly permutation-invariant, blurring recency and weakening purely causal forecasting. These choices compound in the decoder. We reconstruct from a single 128-d "last token" asking one vector to regenerate an entire 63×173×8 field. The dense to deconvolution pathway on odd image sizes (plus a pad-fix step) can introduce small spatial artifacts, and the lack of skip connections means high-frequency details from the encoder never reach the decoder. By comparison, the ConvLSTM implicitly carries multi-scale, location-aware information forward in time, so less must be "re-invented" at the end. Moreover, especially for ConvLSTM, the observed learning pattern of the model is, that the filters are not trying to separately predict each variable precisely, but rather to approximate each variable with the same amount of precision, at the same time. The filters together form a combination to predict these variables. Therefore, with more compute, one could further investigate an even deeper ConvLSTM in order to better train each filter so that they each are able to represent one ocean variable.

(4) GNN-based approaches: Our work initially employed a conventional GNN-LSTM architecture with static k-nearest neighbor (k=8) graph connectivity. However, recognising the need for physics-informed spatial relationships, we developed an edge-aware GNN that incorporates physics-informed graph construction ($|r| > 0.8$ correlation filtering) and directional edge attributes (bearing, distance). Through comparing models on the validation set, we finalized an architecture with hidden-dim=64, lstm-hidden=32, learning-rate=0.001 and num-iters=2 message-passing steps. We selected the final parameters by testing different options and choosing the best balance between accuracy and computing time.

9.2 Test set evaluation

The performance of all developed models was rigorously evaluated on a held-out test set to determine the most accurate and generalisable approach for forecasting the eight oceanographic variables. The results of this final evaluation are presented in Table 2. SMAPE, being scale-independent, is particularly valuable for comparing proportional accuracy across variables with different magnitude ranges, while RMSE and MAE capture absolute deviations, with RMSE penalising larger errors more heavily. Across all variables, the new k-means + VAR method consistently achieves the lowest SMAPE, RMSE, and MAE, indicating superior predictive accuracy and stability. For example, in KD490, it reduces SMAPE to 3.17% compared to 11.90% for exponential smoothing and 20.11% for ConvLSTM, and it shows similar dominance in variables with both stable dynamics (ZSD, RRS bands) and high variability (MICRO, CHL). The edge-aware GNN is competitive in most cases, especially where spatial context is important. However, it tends to have slightly higher error magnitudes. We observed a similar ranking between k-means + VAR and GNN in the validation set results in Section 9.1. This suggests that for this dataset, the temporal–cluster structure exploited by k-means + VAR is more effective than deep spatial–temporal graph modelling – GNN overfits the data relative to k-means + VAR. This critical, metric-driven comparison highlights that optimal model choice depends on balancing proportional accuracy and absolute error control, and in our case, the clustering-based autoregressive approach emerges as the most reliable and generalisable solution across diverse ocean characteristics.

Table 2: Test-set metrics (lower is better). Best result per row in **bold**.

Variable	Metric	Method			
		Exponential Smoothing	k-means + VAR	ConvLSTM	Edge-Aware GNN
KD490	SMAPE	11.90 %	**3.17 %**	20.11 %	7.60 %
	RMSE	0.028	**0.0146**	0.0486	0.0288
	MAE	0.0128	**0.0032**	0.0188	0.0154
ZSD	SMAPE	13.48 %	**2.27 %**	19.43 %	8.53 %
	RMSE	2.0386	**0.5500**	1.9839	2.407
	MAE	1.3523	**0.1673**	1.4028	1.7283
RRS490	SMAPE	18.35 %	**6.02 %**	20.46 %	11.60 %
	RMSE	0.0016	**0.0008**	0.1149	0.0017
	MAE	0.0009	**0.0002**	0.0309	0.0011
RRS443	SMAPE	23.62 %	**7.67 %**	22.89 %	13.13 %
	RMSE	0.0016	**0.0007**	0.11	0.0017
	MAE	0.001	**0.0003**	0.0253	0.0012
CHL	SMAPE	22.10 %	**8.99 %**	24.10 %	16.80 %
	RMSE	0.9259	**0.4612**	0.8655	0.9585
	MAE	0.3062	**0.0930**	0.2943	0.3801
MICRO	SMAPE	84.60 %	**43.67 %**	54.40 %	51.89 %
	RMSE	6.0187	**2.6297**	4.4918	7.4791
	MAE	2.274	**0.6500**	1.7836	2.398
BBP	SMAPE	27.26 %	**11.83 %**	32.28 %	21.19 %
	RMSE	0.0049	**0.0024**	0.1537	0.0055
	MAE	0.002	**0.0005**	0.0426	0.0027
CDM	SMAPE	27.63 %	**16.49 %**	28.73 %	19.08 %
	RMSE	0.0741	**0.0454**	0.1745	0.0858
	MAE	0.0219	**0.0116**	0.0749	0.0291

Compared to our baseline model, exponential smoothing, k-means + VAR performed better. This indicates that explicitly incorporating the spatio-temporal dependencies in k-means + VAR resulted in more accurate forecasts than a simple time-based smoothing approach. The ocean data consists of meaningful patterns beyond what exponential smoothing could capture. Nonetheless, it is an efficient, reliable and interpretable benchmark model.

K-Means + VAR outperformed the ConvLSTM and GNN frameworks, because of two key advantages: working with a stationary time series and learning from spatial homogeneity. Firstly, conducting first differencing on the original series transformed it into a stationary series, removing any trends or seasonal components, such that the VAR could model stable cross-variable dependencies with consistent estimate of coefficients across time. Furthermore, k-means clustering grouped the grid points based on similar variable characteristics, reflecting shared ocean dynamics. This not only reduced the dimensionality, but also averaged out the local, per-grid point noise, and ensure that each VAR model was trained only one of the 5 clusters. In contrast, the ConvLSTM and GNN models were fitted directly on the original, non-stationary series with the expectation that they would learn the variations consisting of low and high frequency trends across time. However, this was a harder task given that our training set was limited to 593 days with irregular data (winter dates removed). Also, this combined with the large parameter space in the neural network models and the need to

capture both spatial and temporal correlations made the models more prone to overfitting, reducing their forecasting accuracy.

Our key takeaway from evaluating the models on the test set is that in our case, a simpler time series model was way more effective in capturing the trends in ocean variables than complex neural network models, reminding us that in certain forecasting tasks, model parsimony can outperform complexity. The actual (blue) versus predicted (red) daily-averaged time series per variable is visible in Figure 15, with grey masks for the winter gaps. This provides evidence for how well the model captured generalised to the test set, capturing the fluctuations for all variables quite well, including MICRO.

Figure 15: Actual (blue) versus predicted (red) time series per variable for the k-means + VAR model, with grey masks for winter gaps

10 Conclusion and extensions

Our goal was to find the best way to predict ocean conditions using satellite data, even when that data has large gaps from clouds or is very complex. We tested several methods, from simple averages to advanced deep learning models, on data from the coast of England. Throughout this investigation, we addressed major oceanographic data challenges. For that we used a multi-stage pipeline that applied hybrid imputation, capped (up to 15 days) temporal forward-filling, k-d tree spatial filling, and resolution reduction. The evaluation process utilized a held-out test dataset covering the most recent 12-month period (2024, Aug to 2025, July), with model performance assessed using three standard metrics: Symmetric Mean Absolute Percentage Error (SMAPE), Root Mean Square Error (RMSE), and Mean Absolute Error (MAE).

We found that a two-step method, k-means + AutoRegression (VAR) model consistently demonstrated superior performance across all eight predicted variables, including KD490, ZSD, RRS490, RRS443, CHL, MICRO, BBP, and CDM. The model's exceptional performance can be attributed to its effective noise reduction capabilities achieved through clustering homogeneous ocean regions, which facilitated more stable parameter estimation compared to fitting VAR models on individual, noisy grid points. The cluster-level VAR approach successfully captured shared spatial patterns, contributing to its overall effectiveness.

While the Edge-Aware GNN + LSTM model emerged as the second-best performer, demonstrating strong capabilities in capturing complex spatial dependencies and temporal dynamics through its graph-based architecture and LSTM components, it tended to have slightly higher error magnitudes compared to k-means + VAR in the test set evaluation. This suggests that, for this specific dataset, the temporal–cluster structure exploited by k-means + VAR was more effective than deep spatial–temporal graph modelling. Additionally, deep learning models, including the Edge-Aware GNN and ConvLSTM, did not generalise as well as the k-means + VAR model on the final test data, indicating that their complexity may have led to overfitting or that they require more extensive tuning, deeper layer-architecture and larger datasets to unlock their full potential.

For end-users who are interested in operational, day-to-day nowcasting of ocean state, the k-means + Vector AutoRegression (VAR) model is recommended for use as the primary nowcaster in the specific region, namely at 5km resolution. It has shown high predictive skill and stability for variables that are critical for sustainable utilization of the ocean, ensuring livelihood security, as well as the health of marine ecosystems. It can be effectively applied to the understanding and prediction of dynamic ocean phenomena that are related to disaster risk reduction and response. It also supports the optimisation of operational planning in sectors like shipping, fishery, ocean resource management, and coastal engineering, where precise high-resolution local prediction is of utmost priority. Operationally, the predictions must be updated daily as new Level-3 (L3) satellite data is received. A consideration for practical use is handling missing data: the model was designed to intentionally leave large winter gaps missing for most variables due to reduced solar illumination and prolonged cloud cover and not to estimate them, particularly over the North Atlantic region where the research was conducted. This method maintains scientific validity by avoiding the introduction of artificial values for temporarily absent data. Instead of creating estimates, the model produces no predictions ("ignorance") when evaluated on such masked periods. Thus, it is important to ensure that

these prolonged winter outages are concealed during training and forecasting. While a robust hybrid imputation technique (15-day forward-fill capped, then nearest-neighbour spatial fill) was employed to enable higher data completeness for other missing intervals, the end-users must keep in mind that data characteristics, including the density of good observations and the pattern of missingness, can vary significantly by geographic areas. This means that the underlying imputation logic is sound, but its behaviour can differ in regions of substantially different patterns of missingness or sparseness of data than the south-west coast of England for which it has been implemented. Finally, the Exponential Smoothing model ought to be retained as an explicit, simple baseline and a known, reliable rollback point, offering a straightforward but interpretable comparison point. Observe that the model at hand has been trained for one specific coastal area only, which suggests that its direct usability and operation may need to be manually tested or re-trained separately for other diverse geographic areas.

Further improvements and research would firstly resolve in scaling the models by scaling power and compute since we have been limited by both factors as mentioned in section 4.1. Further work can be done on the systems side. We recommend an online prediction service that accepts coordinates (and a bounded area) as the desired ocean area which needs to be predicted. Hyperparameters the user could chose from would be the number of past days to base the model on, and the forecast horizon. A fully connected data pipeline could be connected via an API to Copernicus datasets in order to speed up the prediction task from start to end. Bounding the selection window of spatiality keeps latency and data movement predictable, while a rolling cache of recent inputs accelerates repeated queries. While we employed rigorous criteria to identify the most suitable evaluation approach, future work should expand training across broader regional and temporal scales to enhance model scalability and robustness. Additionally, developing specialized evaluation metrics specifically designed for high-dimensional, multivariate oceanographic data better represent the spatiotemporal complexities that are representative of marine ecosystems. Ultimately, as an application extension, we suggest coupling the forecaster to policy optimization from the area of reinforcement learning for fisheries: Treat chlorophyll-related indicators (e.g., CHL and MICRO as proxies for plankton/biomass) and operational costs as a reward signal, and use reinforcement learning (e.g., policy-gradient methods) to plan routes that maximize expected catch potential subject to fuel, ocean variable prediction values, and regulatory constraints. In sum, the next phase is to enhance the compute and amount of data used for the forecasters, expose a low-latency online service backed by Copernicus data, and explore decision-making on top of the forecasts where it delivers clear value.

(Additional note: Generative AI was used to paraphrase the content and gain valuable knowledge during the preparation of this work. For coding tasks, generative AI has been used to support syntax and structure [AI Acknowledgement, 2025].)

References

Acar, E., Kolda, T. G., and Bader, B. W. (2011). Sparse tensor decompositions in data mining applications. In *SIAM International Conference on Data Mining*.

AI Acknowledgement (2025). Generative ai was used to paraphrase the content and gain valuable knowledge during the preparation of this work. for coding tasks, generative ai has been used to support syntax and structure. AI usage acknowledgement.

Alvera-Azcárate, A., Barth, A., and Backeljauw, P. (2005). Reconstruction of sea surface temperature maps using empirical orthogonal functions. *Journal of Geophysical Research: Oceans*, 110(C3).

Baltagi, B. H. (2013). *Econometric Analysis of Panel Data*. Wiley, 5th edition.

Beckers, J. M. and Rixen, M. (2003). EOF calculations and data filling from incomplete observations. *Journal of Atmospheric and Oceanic Technology*, 20(12):1839–1856.

Behrenfeld, M. J. et al. (2006). Climate-driven trends in contemporary ocean productivity. *Nature*, 444(7120):752–755.

Bengio, Y., Simard, P., and Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, 5(2):157–166.

Bergmeir, C. and Benítez, J. M. (2018). A note on the validity of cross-validation for evaluating time series prediction. *Information Sciences*, 406:1–4.

Bonino, G., Galimberti, G., Masina, S., McAdam, R., and Clementi, E. (2024). Machine learning methods to predict sea surface temperature and marine heatwave occurrence: a case study of the Mediterranean Sea. *Ocean Science*, 20:417–432.

Box, G. E. P., Jenkins, G. M., and Reinsel, G. C. (1994). *Time series analysis: Forecasting and control*. Prentice Hall.

Brown, R. G. (1956). *Smoothing, Forecasting and Prediction of Discrete Time Series*. Prentice-Hall.

Campbell, J. W. (1995). The ocean color science and the CZCS: A retrospect and a prospect. *Journal of Geophysical Research: Oceans*, 100(C7):13217–13227.

Canova, F. and Ciccarelli, M. (2013). Panel vector autoregressive models: A survey. Technical Report Working Paper No. 1507, European Central Bank.

Chaigneau, A. A., Law-Chune, S., Melet, A., Voldoire, A., Reffray, G., and Aouf, L. (2023). Impact of sea level changes on future wave conditions along the coasts of western Europe. *Ocean Science*, 19:1123–1143.

Chang, Y. and Su, M. (2016). Application of exponential smoothing in ocean forecasting. *Quarterly Journal of the Royal Meteorological Society*, 142(694):2011–2023.

Chidean, M. I., Caamaño, A. J., Ramiro-Bargueño, J., Casanova-Mateo, C., and Salcedo-Sanz, S. (2018). Spatio-temporal analysis of wind resource in the Iberian Peninsula with data-coupled clustering. *Renewable and Sustainable Energy Reviews*, 81(2):2684–2694.

Cho, K., van Merriënboer, B., Bahdanau, D., and Bengio, Y. (2014). On the properties of neural machine translation: Encoder–decoder approaches. In *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, pages 103–111. Association for Computational Linguistics.

Chopra, S. and Meindl, P. (2013). *Supply Chain Management: Strategy, Planning, and Operation*. Pearson. Recommends smoothing parameter α no larger than 0.20; typical α in 0.1–0.3 range.

Chu, S., Keogh, E., Hart, D., and Pazzani, M. (2002). Iterative deepening dynamic time warping for time series. In *Proceedings of the 2002 SIAM International Conference on Data Mining*, pages 195–212. SIAM.

Chung, J., Gulcehre, C., Cho, K., and Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modelling. arXiv preprint arXiv:1412.3555.

Ciliberti, S. A., Stips, A., Gualdi, S., et al. (2021). Monitoring and forecasting the ocean state and biogeochemical processes in the Black Sea: Recent developments in the Copernicus Marine Service. *Journal of Marine Science and Engineering*, 9(10):1146.

CMEMS (2023). *Product User Manual*.

Copernicus Marine Environment Monitoring Service (2023a). *Product User Manual: Level-3 and Level-4 Data Products*.

Copernicus Marine Environment Monitoring Service (2023b). *Product User Manual: Sentinel-3 OLCI L3 Products*.

Copernicus Marine Service (n.d.). *Product User Manual for Global Ocean Products (CMEMS-GLO-PUM-001-028)*.

Cucco, A., Simeone, S., Quattrocchi, G., Sorgente, R., Pes, A., Satta, A., et al. (2024). Operational oceanography in ports and coastal areas, applications for the management of pollution events. *Journal of Marine Science and Engineering*, 12:380.

de Bodas Terassi, P. M., Galvani, E., Sobral, B. S., et al. (2023). Application of the vector autoregressive model and the association between ocean indicators and rainfall anomalies in eastern Paraná State, Brazil. *Theoretical and Applied Climatology*, 154:925–943.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. (2021). An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations (ICLR)*.

Duchi, J., Hazan, E., and Singer, Y. (2011). Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12:2121–2159.

Emery, W. J. and Thomson, R. E. (2001). *Data analysis methods in physical oceanography*. Elsevier.

Enders, W. (2010). *Applied econometric time series*. John Wiley & Sons.

European Space Agency (2023). 25 times Copernicus made the headlines.

Franz, B. A., Werdell, P. J., and Meister, G. (2015). The SeaWiFS and MODIS ocean color data sets. *Frontiers in Marine Science*, 2:75.

Garnesson, P., Mangin, A., Fanton d'Andon, O., Demarcq, H., and d'Ortenzio, F. (2019). The CMEMS GlobColour chlorophyll a product based on satellite observation: multi-sensor merging and flagging strategies. *Ocean Science*, 15:819–830.

Granger, C. W. J. and Newbold, P. (1974). Spurious regressions in econometrics. *Journal of Econometrics*, 2(2):111–120.

Graves, A. (2012). *Supervised sequence labelling with recurrent neural networks*, volume 385 of *Studies in Computational Intelligence*. Springer.

Gregg, W. W. and Casey, N. W. (2007). Global patterns of ocean chlorophyll. *Remote Sensing of Environment*, 102(3–4):351–373.

Groom, S., Sathyendranath, S., Ban, Y., Bernard, S., Brewin, R., Brotas, V., et al. (2009). Satellite ocean colour: current status and future perspective. *Frontiers in Marine Science*, 6:485.

Groom, S., Tilstone, G., Aiken, J., et al. (2019). Regional assessment of chlorophyll dynamics in UK waters using satellite data. *Earth System Science Data*, 11(3):1367–1385.

Hardman-Mountford, N. J., Richardson, A. J., Boyer, D. C., Kreiner, A., and Boyer, H. J. (2020). Seasonal phytoplankton cycles in shelf waters of the northeast Atlantic. *Journal of Marine Systems*, 207:103116.

Hartog, J. R., Spillman, C. M., Smith, G., and Hobday, A. J. (2023). Forecasts of marine heatwaves for marine industries: Reducing risk, building resilience and enhancing management responses. *Deep Sea Research Part II: Topical Studies in Oceanography*, 209:105276.

He, K., Zhang, X., Ren, S., and Sun, J. (2015). Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 1026–1034.

Hinton, G., Srivastava, N., and Swersky, K. (2012). Overview of mini-batch gradient descent. Coursera: Neural Networks for Machine Learning, Lecture 6.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Holmberg, D., Clementi, E., Epicoco, I., and Roos, T. (2025). Accurate Mediterranean Sea forecasting via graph-based deep learning. arXiv preprint arXiv:2506.23900.

Holtz-Eakin, D., Newey, W., and Rosen, H. S. (1988). Estimating vector autoregressions with panel data. *Econometrica*, pages 1371–1395.

Huang, J. and Lin, C. (2019). Short-term sea level prediction using Holt-Winters. *IOP Conference Series: Earth and Environmental Science*, 252:012034.

Hyndman, R. J. and Athanasopoulos, G. (2008). *Forecasting: Principles and Practice*. OTexts.

International Ocean Colour Coordinating Group (2018). Ocean colour algorithm development and validation. IOCCG Report 18, IOCCG.

Ioffe, S. and Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, pages 448–456.

Isaaks, E. H. and Srivastava, R. M. (1989). *An Introduction to Applied Geostatistics*. Oxford University Press.

Kamalov, F. and Sulieman, H. (2021). Machine learning applications in ocean engineering. arXiv preprint arXiv:2110.12631.

Kärnä, T., Alenius, P., Tuomi, L., et al. (2021). Nemo-nordic 2.0: Operational marine forecast model for the Baltic Sea. *Geoscientific Model Development*, 14(9):5731–5749.

Keramea, P., Kokkos, N., Zodiatis, G., and Sylaios, G. (2023). Modes of operation and forcing in oil spill modelling: state-of-art, deficiencies and challenges. *Journal of Marine Science and Engineering*, 11:1165.

Kidger, P. and Lyons, T. (2020). Universal approximation with deep narrow networks. Proceedings of Machine Learning Research.

Kim, J., Kim, H., and Lee, K. S. (2019). ConvLSTM-based deep learning model for sea surface temperature prediction. *Remote Sensing*, 11(11):1310.

Kingma, D. P. and Ba, J. L. (2015). Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*.

Kolda, T. G. and Bader, B. W. (2009). Tensor decompositions and applications. *SIAM Review*, 51(3):455–500.

Laizure, S. C. (2024). Caution: ChatGPT doesn't know what you are asking and doesn't know what it is saying. *Journal of Pediatric Pharmacology and Therapeutics*, 29(5):558–560.

LeCun, Y. (1989). Generalization and network design strategies. Technical Report CRG-TR-89-4, University of Toronto Connectionist Research Group. A shorter version was published in Pfeifer, R., Schreter, Z., Fogelman, F., & Steels, L. (Eds.), Connectionism in Perspective, Elsevier.

LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., and Jackel, L. D. (1989a). Back-propagation applied to handwritten zip code recognition. *Neural Computation*, 1(4):541–551.

LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., and Jackel, L. D. (1989b). Handwritten digit recognition with a back-propagation network. In *Advances in Neural Information Processing Systems*, pages 396–404. Morgan Kaufmann.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

Lee, Z., Carder, K. L., and Arnone, R. (2002). Deriving inherent optical properties from water color: a multiband quasi-analytical algorithm for optically deep waters. *Applied Optics*, 41(27):5755–5772.

Lee, Z., Carder, K. L., and Arnone, R. (2005). Deriving inherent optical properties from water color: A multiband quasi-analytical algorithm for optically deep waters. *Remote Sensing of Environment*, 97(3):326–336.

Lee, Z. et al. (2015). Secchi disk depth from satellite ocean color products. *Remote Sensing of Environment*, 169:152–161.

Li, H., Chen, X., Li, X., Wu, X., and Li, R. (2021). Spatiotemporal forecasting of ocean currents using a ConvLSTM neural network. *Remote Sensing*, 13(12):2411.

Liang, S., Zhao, A., Qin, M., Hu, L., Wu, S., Du, Z., and Liu, R. (2023). A graph memory neural network for sea surface temperature prediction. *Remote Sensing*, 15(14):3539.

Lin, Y., Koprinska, I., and Rana, M. (2021). Temporal convolutional attention neural networks for time series forecasting. In *Proceedings of the 30th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 2979–2985.

Link, J. S., Thur, S., Matlock, G., and Grasso, M. (2023). Why we need weather forecast analogues for marine ecosystems. *ICES Journal of Marine Science*, 80:2087–2098.

Liu, J., Wang, G., Duan, L.-Y., Abdiyeva, K., and Kot, A. C. (2020). Skeleton-based human action recognition with convolutional neural network. *IEEE Transactions on Multimedia*, 22(2):494–505.

Liu, Y., Li, S., and Han, K. (2017). Missing data imputation in remote sensing images using deep learning. *Remote Sensing Letters*, 8(6):570–579.

Loisel, H. and Morel, A. (1998). Light scattering and chlorophyll concentration in case 1 waters: A reexamination. *Limnology and Oceanography*, 43(5):847–858.

Makarov, A. and Clarke, R. (2021). Regional ocean forecasting using simple statistical models. *Journal of Marine Systems*, 220:103451.

Makridakis, S., Wheelwright, S. C., and Hyndman, R. J. (1993). *Forecasting: Methods and Applications*. Wiley, 3rd edition.

Martinez, A. and Zhou, L. (2021). Naive forecasting and its role in environmental baselines. *Remote Sensing*, 13(5):1040.

Mobley, C. D. (1999). Estimation of remote-sensing reflectance from above-surface measurements. *Applied Optics*, 38(36):7442–7455.

Morel, A. and Prieur, L. (1977). Analysis of variations in ocean color. *Limnology and Oceanography*, 22(4):709–722.

Morim, J., Wahl, T., Vitousek, S., Santamaria-Aguilar, S., Young, I., and Hemer, M. (2023). Understanding uncertainties in contemporary and future extreme wave events for broad-scale impact and adaptation planning. *Science Advances*, 9:eade3170.

Nau, R. (2005). Moving averages and smoothing models. Statistical Forecasting: Notes on Regression and Time Series Analysis, Duke University.

Oliver, M. A. and Webster, R. (1990). Kriging: a method of interpolation for geographical information systems. *International Journal of Geographical Information Systems*, 4(3):313–332.

OpenStax (2020). Time series models: Exponential smoothing. In Fundamentals of Operations Management, eCampus Ontario.

Palmer, S. C., Hunter, P. D., Lankester, T., Hubbard, S., Spyrakos, E., and Tyler, A. N. (2015). Validation of Copernicus Sentinel-3 OLCI for coastal monitoring. *Remote Sensing of Environment*, 160:32–45.

Paparrizos, J. and Gravano, L. (2015). k-shape: Efficient and accurate clustering of time series. In *Proceedings of the 2015 ACM SIGMOD international conference on management of data*, pages 1855–1870.

Pascanu, R., Mikolov, T., and Bengio, Y. (2013). On the difficulty of training recurrent neural networks. In *Proceedings of the 30th International Conference on Machine Learning (ICML)*, pages 1310–1318.

Pérez Gómez, B., Vilibić, I., Šepić, J., Međugorac, I., Ličer, M., Testut, L., et al. (2022). Coastal sea level monitoring in the Mediterranean and Black seas. *Ocean Science*, 18:997–1053.

Quante, M. and Colijn, F. (2016). *North Sea Region Climate Change Assessment*. Springer.

Rabus, P., Kolden, T., and Skar, B. (2019). Tensor-based methods in oceanography and remote sensing: A review. *Remote Sensing*, 11(17):2028.

Reichstein, M., Camps-Valls, G., Stevens, B., Jung, M., Denzler, J., Carvalhais, N., and Prabhat (2019). Deep learning and process understanding for data-driven earth system science. *Nature*, 566:195–204.

Reynolds, R. W., Smith, T. M., Liu, C., Chelton, D. B., Casey, K. S., and Schlax, M. G. (2007). Daily high-resolution-blended analyses for sea surface temperature. *Journal of Climate*, 20(22):5473–5496.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, 323:533–536.

Sakamoto, K., Takamura, T., Ueno, H., et al. (2019). Development of a 2-km resolution ocean model covering the coastal seas around Japan for operational application. *Ocean Dynamics*, 69(10):1181–1202.

Shi, X., Chen, Z., Wang, H., Yeung, D. Y., Wong, W. K., and Woo, W. C. (2015). Convolutional LSTM network: A machine learning approach for precipitation nowcasting. In *Advances in Neural Information Processing Systems*, volume 28.

Siegel, D. A., Nelson, N. B., and Carlson, C. A. (2002). Seasonal variability of colored dissolved organic matter in the Sargasso Sea. *Journal of Geophysical Research: Oceans*, 107(C8):3219.

Skrodzki, M. (2019). The k-d tree data structure and a proof for neighborhood computation in expected logarithmic time. arXiv preprint.

Sugawara, D. (2021). Numerical modelling of tsunami: advances and future challenges after the 2011 Tohoku earthquake and tsunami. *Earth-Science Reviews*, 214:103498.

Sun, Q., Little, C. M., Barthel, A. M., and Padman, L. (2021). A clustering-based approach to ocean model–data comparison around Antarctica. *Ocean Science*, 17(1):131–145.

Sun, R., Li, S., and Wang, Y. (2020). Missing data imputation for satellite images using generative adversarial networks. *Remote Sensing*, 12(10):1686.

Tippett, M. and Anderson, J. (1995). Benchmarking forecasts: The need for a realistic baseline in atmospheric prediction. *Monthly Weather Review*, 123(10):2811–2816.

Tsushima, H. and Ohta, Y. (2014). Review on near-field tsunami forecasting from offshore tsunami data and onshore GNSS data for tsunami early warning. *Journal of Disaster Research*, 9:3.

Uitz, J., Claustre, H., Morel, A., and Hooker, S. B. (2006). Vertical distribution of phytoplankton communities in open ocean: An assessment based on surface chlorophyll. *Journal of Geophysical Research*, 111:C08005.

user guide, T. (2024). tensorly.decomposition.tucker. Accessed: 2025-08-14.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 30, pages 5998–6008.

Virtanen, P., Gommers, R., Oliphant, T. E., et al. (2020). SciPy 1.0: Fundamental algorithms for scientific computing in Python — `scipy.spatial.KDTree` documentation. *Nature Methods*, 17:261–272.

Visbeck, M. (2018). Ocean science research is key for a sustainable future. *Nature Communications*, 9:690.

Wang, Y., Feng, C., and Anderson, D. V. (2021a). CNN with temporal attention for environmental sound classification. In *2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 930–934. IEEE.

Wang, Y., Feng, C., and Anderson, D. V. (2021b). A multi-channel temporal attention convolutional neural network model for environmental sound classification. In *2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 930–934. IEEE.

Wikipedia (2025). Exponential smoothing. https://en.wikipedia.org/wiki/Exponential_smoothing. Explains that simple exponential smoothing weights recent data more than moving average.

Wilkin, J. and Arango, H. (2010). Ocean forecasting and monitoring using real-time data assimilation. *Bulletin of the American Meteorological Society*, 91(1):107–109.

Yan, Y., Huang, H. C., and Genton, M. G. (2021). Vector autoregressive models with spatially structured coefficients for time series on a spatial grid. arXiv preprint arXiv:2103.00160.

Zhang, Q., Liu, Q., and Ye, Q. (2024). An attention-based temporal convolutional network method for predicting remaining useful life of aero-engine. *Engineering Applications of Artificial Intelligence*, 127:107241.

Zheng, J., Huang, X., Sangondimath, S., Wang, J., and Zhang, Z. (2021). Efficient and flexible aggregation and distribution of MODIS atmospheric products based on climate analytics as a service framework. *Remote Sensing*, 13(17):3541.