university of groningen / faculty of science and engineering / artificial intelligence

# Fish Catch Optimisation using Variational Autoencoder

Bachelor's Project Thesis

Luc Cronin, s4326245, l.cronin@student.rug.nl,
Supervisor: Prof J. D. Cardenas Cartagena

**Abstract:** The fishing industry plays a vital role in the global economy, particularly in Norway. There is currently no accurate methods for fisherman to know the locations of fish, this costs a lot of time, money and resources to find fish. These inefficiencies lead to major negative environmental impact as predicting fish locations on a daily basis remains a significant challenge. This study aims to investigate the effectiveness of a deep learning approach in solving this problem by implementing a Variational Autoencoder (VAE). The nature of the task is very complex due to the spatial and temporal aspects of the data. The model leverages spatio-temporal data, including Sea Surface Salinity (SSS), Sea Surface Temperature (SST), and historical catch data, to capture the complex patterns influencing fish movements. Our approach integrates convolutional and recurrent neural network layers to handle the spatial and temporal dimensions of the data. The results highlight the challenges in using deep learning models for this task, emphasising the need for improved data representation to achieve reliable predictions and support sustainable fishing practices.

## 1 Introduction

Fishing is a mega scale activity on the ocean that produces a large amount of food. For instance, fish caught represent 87% of all vertebrate animals reported to be used for food or animal feed in 2019 (Mood & Brooke, 2024). In addition, fishing is a big part of the Norwegian economy, which is Europe's largest fishing nation. In 2022, they caught 2.6 million tonnes of fish worth €2.8 billion (Jensen, 2024).

Fishing being an important industry, there is a lot of research on fish migration and their behavioural patterns (Leggett, 1977). Despite a lot of research about factors that affect their migration patterns, finding them on a day to day basis can be a challenge. Hence, fishermen spend a lot of time and fuel searching for fish causing millions of tons of Co2 to be released annually (Greer et al., 2019). Investigating and developing methods to reduce the amount of Co2 released by the commercial fishing industry is crucial to reducing its environmental impact.

The FishAI: Sustainable Commercial Fishing Challenge was set up to find new ways to make fishing more efficient using artificial intelligence (AI) and machine learning (ML)(Nordmo et al., 2022). The Sustainable Commercial Fishing Challenge asked for solutions to predict the best places to find fish and help fishermen save time and reduce their impact on the environment. As a result, fishermen would be able to plan optimal routes, save fuel, and support sustainable fishing.

Solutions from the FishAI challenge and their implementation of a variety of classical machine learning methods, such as random forrest regression approaches to solve the problem, were investigated as an initial basis to this paper (Lambon et al., 2022). Instead, a deep learning oriented approach was selected to attempt improving how we predict fish movements. The reasons for selecting this method was to try and handle the complexity of the task brought about by the large temporal and spatial aspects of the data. By definition deep learning approaches can inherently manage more complexity compared to classical machine learning methods. This will be further explored as part of the theoretical framework.

## 1.1 Motivation

Addressing the inefficiencies with current fishing practices is essential for several reasons.
First the fishing industry significantly contributes to carbon emissions due to high fuel consumption by fishing vessels. Estimates suggest that commercial fishing globally emits around 207 million tons of CO2 in 2016, making it a considerable source of marine-related greenhouse gas emissions (Greer et al., 2019). Optimising fishing routes and reducing search times for fish can significantly lower fuel consumption and, consequently, carbon emissions, minimising the ecological footprint of the industry.
Second, fishing is a major economic activity, particularly in Norway. Inefficient fishing practices lead to increased operational costs, reducing the profitability of fishing enterprises. Norway alone in 2022 caught 2.6 million tonnes of fish, to visualise how brobdingnagian a scale that is, it would be equivalent to the weight of 257 Eiffel towers (Jensen, 2024). Improving efficiency through better prediction models can help reduce costs associated with fuel and time, thereby increasing overall profitability for fishermen whilst promoting more sustainable fishing practices.
Finally, overfishing and unsustainable fishing practices have led to the depletion of several fish species, threatening marine biodiversity. Accurate prediction of fish locations can help ensure that fishing efforts are more targeted, reducing the likelihood of overfishing and helping maintain balanced marine ecosystems. Sustainable fishing practices are crucial for the long-term viability of the fishing industry and the health of ocean ecosystems.

## 1.2 State-of-the-art

The problem with predicting fish locations involves understanding and modelling their spatio-temporal behaviour. Fish movement patterns are influenced by various environmental factors such as sea surface temperature (SST) and sea surface salinity (SSS) to predict the locations and movements of fish, incorporating historical catch notes data (Leggett, 1977). Accurate spatio-temporal predictions can significantly enhance the efficiency of fishing operations by reducing resources spent searching for fish. This, in turn, supports more sustainable fishing practices and minimises the environmental impact.

Predicting fish behaviour is not only about where fish are likely to be found but also when they will be there. This spatio-temporal prediction task is complex due to the dynamic nature of marine ecosystems, where multiple interacting variables influence fish movements. Therefore, advanced modelling techniques may be better equipped to handle these complex non-linear relationships.

### 1.2.1 Previous FishAI Challenge solutions

**FishMAZE Project:** The FishMAZE project utilised regression models trained on environmental data, historical catch notes, and coordinate data to predict fish likelihood at specific locations. This model achieved a Root Mean Square Error (RMSE) of 8.6830, demonstrating a significant correlation between environmental variables and fish presence. However, the project faced challenges in data preprocessing and model generalisation.
**Lodestar Fishing Platform:** Lodestar is a web application that integrates historical catch data with environmental data to predict fish locations across the Nordic seas (Brekke et al., 2022). It uses XGBoost regression models to generate probabilities of fish presence and incorporates a route-planning algorithm to optimise fishing routes. The platform, however, struggled with data granularity and true positive data limitations, which affected prediction accuracy.
The solutions proposed for the Fish AI challenge implemented a variety of regression models with a few incorporating ensemble methods to improve generalisation and learning. However, since the data presents challenges in both complexity and sparsity it is worth investigating a deep learning oriented approach, as it could be better suited to handling such complexity and difficulty in a task. Using a deep learning approach offers potential benefits through hierarchical feature extraction. Due to the diverse and complex nature of the data, traditional methods would require extensive domain-specific knowledge to manually create meaningful features. However, deep learning models, such as variational autoencoders (VAE), can perform hierarchical feature extraction, allowing them to automatically learn relevant features from raw data. This capability is particularly useful in handling the intricate relationships in environmental data,

potentially leading to more accurate and insightful predictions.

### 1.3 Contributions

The main contribution of this research is to further understand how to model fish movement patterns in order to promote sustainable fishing practices. It also specifically explores the effectiveness of using a VAE fed with a sequence of spatial maps containing historical catch and environmental data, this being a raw format of data. Through this approach, the aim is to advance this domains' knowledge on how to best model fish movement patterns. Finally, this research provides insights on the feasibility of using a more raw format of data for a deep learning model to extrapolate meaningful features and effectively learn from that.

## 2 Theoretical framework

### 2.1 Problem Outline

The problem of trying to predict where the largest population of fish will be at a given time is a spatio-temporal prediction issue. There is also the challenge of a lot of data with very little substance due to its sparsity. For example, the data is limited by the catch-notes data that comprises of specific pinpoint locations rather than movement trajectories of fish. This issue is exacerbated due to the curse of dimensionality and a lack of density in data points in the catch-notes.

### 2.2 Fish Migration

Salinity is a crucial environmental factor as it impacts fish by influencing their osmotic balance and metabolic costs. Different species also exhibit varying levels of tolerance to salinity changes, which can affect their growth rates, reproductive behaviours, and survival. For example, some fish invest more energy into nest-building when salinity levels change, indicating that even slight environmental shifts can affect their natural behaviours (Lehtonen et al., 2016). Temperature is a vital factor in fish habitat suitability as it directly impacts fish metabolism, reproductive cycles, and migration patterns(Shoji et al., 2011). Understanding the thermal environment helps in predicting fish movements, as different species have specific temperature preferences and thresholds. Warmer temperatures can induce earlier spawning and migration in some species, aligning their reproductive cycles with optimal environmental conditions.

### 2.3 Model Selection

First, the criteria had to be set, that is one capable of modelling data with spatial and temporal aspects. For example, the task of predicting fish locations involves understanding many similar elements to predict future frames in a video sequence. Video data is also comprised of sequential spatial data, which differs in two separate manner. Firstly, rather than using pixels, we are dealing with coordinates. Secondly, rather than using 3 channels of colour, our data consists of 3 channels made up of our historical catch-notes and environmental data.

It was deemed critical when selecting a model that is needed to be able to handle these aspects of the data. If the model had little to no capacity to model relationships in the data both spatially and temporally, the chances of producing a successful model would have been drastically reduced.

As a result of having to handle large amount of complexity as mentioned previously as well as spatio-temporal data, this narrowed down the range of feasible models for practical future use such as a VAE and convolutional long short term memory (ConvLSTM) next frame video prediction (NFVP).

### 2.4 VAE

#### 2.4.1 Fundamentals of Autoencoders

Autoencoders are neural networks that learn to encode data into an efficient compressed latent representation and then decode it back to the original form. This structure can be particularly useful for noise and the dimensionality reduction. In an autoencoder, the encoder part reduces the input data to a lower-dimensional space, capturing the most significant features, while the decoder reconstructs the data from this compact representation. Building upon the autoencoder architecture, VAEs introduce additional capabilities that present potential benefit for our predictive task.

### 2.4.2 Justification for VAE

VAEs are a generative model that extend the autoencoder architecture by incorporating a probabilistic approach to the encoding process. VAEs being a generative model may present benefits to this task as fish movement patterns are not entirely deterministic, VAE's being a generative won't come to deterministic solutions. VAEs have been successfully applied in video prediction tasks, and given the similarities to fish prediction, it is worth exploring their effectiveness in this domain. VAEs' ability to handle high-dimensional data and incorporate both temporal and spatial elements makes them particularly well-suited for our task. By leveraging VAEs, we aim to capture the complex patterns in fish movement and provide accurate and reliable predictions, ultimately supporting more efficient and sustainable fishing practices. VAEs consist of an encoder, a decoder, and a latent space representation.

### 2.4.3 VAE Structure

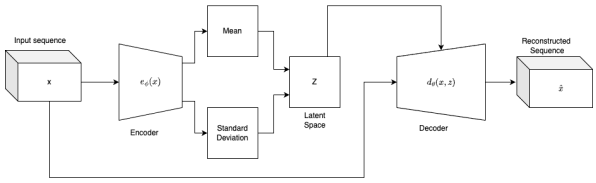


**Figure 2.1: Structure of the VAE used in this study**

**Encoder:** The encoder in a VAE is designed to compress the input data into a lower-dimensional latent space. This process involves several layers and mechanisms to capture essential features of the data. This is done similarly to how it is done in a conventional autoencoder. However, in VAE it generates parameters for a distribution $\mu(x)$ & $\sigma(x)^2$, that capture the essential aspects of the data. This distribution can then be sampled to obtain the latent variable $z$.

The encoder transforms the input $\mathbf{x}$ into latent variables $z$. The output of the encoder is not a single point but a distribution over the latent space, characterised by the mean $\mu$ and standard deviation $\sigma$. This distribution is given by:

$$q(z \mid x) = N(z; \mu(x), \sigma(x)^2)$$

This formulation allows the model to capture variability in the data, providing a more robust representation.

**Latent space:** The latent space in a VAE is a lower-dimensional space where the compressed representation of the input data resides. Instead of mapping each input to a single point, the VAE maps the inputs to a distribution in the latent space. This probabilistic method enables the model to generate new data points by sampling from these distributions. To enable backpropagation through the stochastic layer, the reparameterisation trick is used. This involves expressing the sampling operation in a way that allows gradients to pass through. Specifically, the latent variables $z$ are computed as:

$$z = \mu + e^{0.5 \cdot \sigma} \cdot \epsilon, \text{where } \epsilon \sim \mathcal{N}(0, 1)$$

Epsilon is a random variable drawn from a standard normal distribution. This approach ensures that the sampling operation is differentiable, enabling the model to be trained using gradient descent techniques.

**Decoder:** The decoder in a VAE) is responsible for reconstructing the input data from the latent variables, effectively reversing the encoding process. This architecture is critical for generating accurate and meaningful outputs that reflect the original data's spatial and temporal characteristics. The geospatial data is fed as input and is processed by several layers. This intermediate representation of the data is combined with the latent variable $Z$ generated using a sampling layer that implements the reparameterisation trick. The data is then upsampled to the original spatial dimensions of the relevant map of the North sea. By integrating the encoded information from the encoder with the reshaped latent variables and progressively upsampling through multiple layers, the hope is that the decoder can generate outputs that should understand the movement patterns of fish.

## 2.5 ConvLSTM NFVP

ConvLSTM networks combine the strengths of Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks to handle spatio-temporal data effectively. This architecture is particularly well-suited for tasks involving both spatial and temporal dependencies, such as

video prediction, where the goal is to predict future frames in a video sequence. Once again due to the similarities with video prediction and fish prediction the hope is that this architecture can be leveraged to achieve meaningful results and serve as a point of comparison to the VAE.

# 3 Methods

## 3.1 Data

### 3.1.1 Why these datasets

**Historical catch-ntoes:** Historical catch data is essential for training the model as it provides empirical evidence of fish locations and quantities. This data helps the model learn from past fishing activities, identifying patterns and trends about typical locations where fish appear. This can also provide the insights as to how typical fish locations change over time.

**Environmental data:** The catch-notes data is not an ideal dataset however as it only provides pinpoint locations for where fish were caught, this is not accurate to how the fish were caught nor how they were moving prior to being caught. Without movement trajectories of fish, it is difficult to model their movement patterns. Hence, it was important to incorporate environmental factors as we are aware they play a large role in the movement and behavioural patterns. This is why data such as the SST and SSS data were included to try increase the models chances of having a comprehensive understanding of fish behaviour.

### 3.1.2 Data formatting & Missing data

**SSS & SST:** The SSS dataset contains global monthly salinity averages from April 2015 to January 2022, focusing on the 'sss_smap' feature, with values ranging from 0 to 44 parts per thousand. Similarly, the SST dataset provides daily temperature averages from 1981 to 2024. Both datasets are extracted as three-dimensional matrices representing time, latitude, and longitude, containing their respective data. Both the SSS and SST have have a spatial resolution of 0.25 degrees. To maintain consistency between the both environmental datasets, the SSS monthly readings were converted to daily readings using mean imputation. For missing data, linear interpolation was applied to the SSS dataset for July 2019, while any remaining missing values in both datasets were addressed using the griddata interpolation method, ensuring continuity and reliability by interpolating missing points based on known neighbouring values.

**Historical catch notes:** The catch notes dataset spans from 2000 to 2022 and includes 150 features in CSV format. Key features relevant to our model—such as product weight, longitude, latitude, landing date, and species—were extracted for integration with the SSS and SST datasets. Previous solutions to the FishAI challenge mentioned models struggling when attempting to predict the locations of fishes, so it was decided to narrow the field of research and focus on Haddock. This is because it had the most data points out of the most valuable fish for the industry partners (Nordmo et al., 2022). Data points with null values in these selected features were removed to maintain data quality. The preparation process involved aligning the temporal and spatial dimensions of the catch data with those of the environmental datasets, ensuring seamless integration and accurate modelling.

### 3.1.3 Data Alignment

To aggregate all the data into one structure, the range for the longitude and latitude were calculated using the catch-notes data. This was necessary to ensure all catch-notes data points would be included in the final data structure. The SSS and SST data were already spatially aligned so they were simply stacked on top of another. Using the bounds calculated from the catch notes we were then able to create a matrix that would be able to include all the data points from the catch notes. The catch data was aligned with the environmental data by finding the closest coordinate in the dataset for each catch point, ensuring that each data point was within 0.24 degrees of the original location. The temporal range for the data was based off the SSS data as it had the shortest time span of our data.

### 3.1.4 Map Cropping

The initial dataset, which combined catch notes and environmental data, had spatial dimensions that were not conducive to applying convolutions due to their irregular shapes. To address this is-

sue, we created two new datasets with standardised spatial dimensions. The first dataset was designed to encompass all data points from the catch notes, resulting in spatial dimensions of 128x352. This ensured that no data points were omitted while providing a manageable shape for certain convolution operations. The second dataset aimed to create a more compact and dense representation of the data. We selected a spatial dimension of 64x64, focusing on the region with the highest concentration of data points. This smaller, dense dataset allows for more efficient processing and analysis, as it simplifies the convolutional operations by reducing the spatial complexity.

#### 3.1.5 Splitting

To ensure the model can generalise well to unseen data, the dataset was split into training, validation, and test sets in a sequential manner. This method was chosen to simulate real-world scenarios where the model predicts future events without knowledge of future data points. This ensures that there is no data leakage. **Test set:** The most recent 20% of the total dataset was reserved for testing the model. This meant there was a total of 493 test samples. The test set contains the most recent historical data and is used to evaluate the model's performance in predicting future events. **Validation set:** From the remaining 80% of the dataset, the latest 20% was used for validation. This resulted in a total of 393 samples for validation purposes. This set is used to tune hyper-parameters and assess the model during training, ensuring it generalises well to new data. **Training set:** The earliest 80% of the remaining data after the test split (i.e., 64% of the total data) was used for training the model. This left a total of 1594 training samples for the model to learn the necessary relationships and patterns in the data to predict the location of fish.

#### 3.1.6 Data Labelling

After splitting the data, each set was windowed to create input-target pairs for the model. Windowing involves segmenting the time series data into overlapping windows, where each window consists of a sequence of input data points and a corresponding target data point. Each input window comprises 5 consecutive days of spatial data, including environmental measurements and historical catch data. These windows provide the context necessary for the model to make accurate predictions. The target window corresponds to the day immediately following the input window. Thus, the model uses 5 days of data to predict the 6th day. The target window only contains the feature we are trying to predict for, that being the product weight feature of the catch notes data. Using these parameters to generate the input-target pairs, the window was slid across 1 day at a time, resulting in overlapping windows that provide learning examples for the model comprehensively model the dynamics of the data.

#### 3.1.7 Normalisation

To ensure all features contributed equally during model training and to encourage stability in training, Min-Max Scaling was applied using scikit-learn's MinMaxScaler. This technique normalised each feature independently to the range $[0, 1]$, based on its minimum and maximum values. All features were normalised separately, this step was crucial as the features had significantly different ranges, which could have led to instability in the model learning process.

## 4 Model Design

#### 4.0.1 VAE

**Encoder:** The encoder network compresses the input geo-spatial data into a latent space, capturing essential features for reconstruction. The input shape is (time-steps, height, width, channels). The encoder comprises three ConvLSTM2D layers with ReLU activations, filter sizes of [32, 64, 128], and HeNormal weight initialisation. Each ConvLSTM2D layer includes L1 regularization with a factor of 0.0001. The output is flattened and fed into two dense layers that produce the mean and log variance to generate the latent distribution. A custom sampling layer implements the reparameterisation trick, allowing gradients to flow through the network.
**Decoder:** The decoder network reconstructs the input data from the latent space, effectively reversing the encoding process. It takes geo-spatial data and the latent variable z as inputs. The geo-spatial input passes through three ConvLSTM2D layers

with filter sizes of [32, 32, 64] and ReLU activations. The latent variable z is processed through three dense layers and a reshaping layer to match the dimensions for concatenation. The combined data is processed by a ConvLSTM2D layer with 64 filters and reduces the time dimension to one time-step. The up-sampling phase restores the necessary spatial dimensions using four Conv3DTranspose layers with filter sizes of [128, 64, 32, 1]. All layers use ReLU activation except for the final output layer, which uses a sigmoid activation.

**Loss Calculation** The loss for the VAE model is computed using a combination of reconstruction loss and Kullback-Leibler (KL) divergence loss. This approach ensures that the model generates outputs similar to the input data while also encouraging the latent space to follow a normal distribution.

The reconstruction loss measures how well the model's output resembles the target data. Specifically, the mean absolute error (MAE) is used for this purpose due to its robustness against sparse data. The MAE is calculated using the following equation:

$$reconstruction = \frac{1}{N} \cdot \sum_{i=1}^{N} |y_{true} - y_{pred}|$$

The MAE is calculated per element between the target data and the reconstruction, and then summed over the spatial dimensions to obtain a single reconstruction loss per time step. Finally, this reconstruction loss is averaged over the batch to get the final reconstruction loss.

The MAE is calculated per element wise between the target data and the reconstruction. The MAE values are then summed over the spatial dimensions to obtain a single reconstruction loss per time step. The reconstruction is then averaged over the batch to get the final reconstruction loss.

The KL divergence loss encourages the latent variables to follow a standard normal distribution. It is computed as:

$$KL = -0.5 \sum_{j=1}^{D_z} \left(1 + \log(\sigma_j^2) - \mu_j^2 - \sigma_j^2\right)$$

Where, $\mu$ and $\sigma$ are the mean and standard deviation of the latent variables, respectively, $D_z$ is the dimensionality of the latent space.

To prevent the KL loss from dominating the model and preventing it form learning the KL divergence is multiplied by a weight to control its influence. Hence the total loss function is calculated as:

$$Loss_{total} = reconstruction + Wight_{KL} \cdot KL$$

#### 4.0.2 ConvLSTM NFVP

The ConvLSTM Next Frame Video Prediction (ConvLSTM NFVP) model is designed based on a model from a Keras 3 code examples (Team, n.d.). The shape of the input follows the same dimensions as mentioned previously for the VAE.

The model comprises three ConvLSTM2D layers, each configured with ReLU activation functions and followed by batch normalisation layers. The ConvLSTM2D layers are designed with 64 filters and implement L1 and L2 regularisation with factors both set to 0.1 to prevent overfitting. The final ConvLSTM2D layer reduced the temporal dimension to one time-step to align the data with the temporal dimension of the target data. After reducing the temporal dimension, the data is reshaped to add the channel dimension necessary for Conv3D processing. The final Conv3D layer has a filter size of 1 and uses a sigmoid activation function to generate the predicted frame

### 4.1 Training and Evaluation

#### 4.1.1 Training details

The VAE and ConvLSTM NFVP models were trained over 20 epochs with a batch size of 16, used for both training and validation. The Adam optimiser was employed with a learning rate of 0.0001 to adjust the model's gradients. The VAE was trained using the data with larger spatial dimensions of 128x352, while the ConvLSTM NFVP model was trained using the dataset with spatial dimensions of 64x64. The VAE incorporated the 'clipnorm' parameter in the Adam optimiser to clip gradients, preventing them from exceeding a specified norm value set to 1. This technique helps stabilise training by mitigating the impact of large gradient updates, which can cause instability, especially in deep networks. The weight the KL divergence is

initially set to 0 and is incremented each epoch by a factor of 0.01.

#### 4.1.2 Evaluation Metrics

To evaluate the models performance, a variety of metrics were tracked during the training and evaluation phases of the model. The MAE was selected as the data is very sparse. The MAE provides a straightforward measure of the average absolute differences between predicted and actual values, making it less sensitive to outliers and missing data points. The KL divergence loss was tracked as a metric for the VAE to monitor the performance of the encoder. This is crucial for ensuring the parameters the encoder output are close to that of a normal distribution. Precision measures the proportion of correct positive predictions (i.e., correctly predicted locations with fish) among all positive predictions, offering insights into the model's accuracy. The Recall measures the proportion of correct positive predictions (i.e., correctly predicted locations with fish) among all actual positive instances, reflecting the model's sensitivity in detecting relevant events. These metrics provide a comprehensive method for monitoring and evaluating the models' learning and performance. The ConvLSTM NFVP model tracked the same metrics except for the KL divergence.

#### 4.1.3 Validation Strategy

Validation was performed after each epoch to monitor the model's performance on unseen data. Early stopping callback was implemented to prevent the model from training needlessly without the chance of converging. The criteria set to a patience of 10 epochs for the KL divergence of the VAE not improving and 12 epochs for the total loss not improving for both models. The ConvLSTM NFVP model implemented an additional callback to reduce the learning rate after 10 epochs if there was no improvement.

Validation results were used to tune the model by adjusting hyper-parameters, exploring different architectures, and incorporating regularisation techniques.

#### 4.1.4 Testing

Finally the models were evaluated on the test set, which comprised of the most recent 20% of the dataset. The same metrics mentioned previously were reported for the test data to provide an unbiased assessment of the model's performance.

# 5 Results

The primary objective of this study was to develop a robust model to predict the locations of a specific fish species using spatio-temporal data. Despite the efforts to create a VAE model tailored for this task, the results indicate several challenges and limitations that affected the model's performance. This section details the performance metrics, example outputs, and a comprehensive analysis of the results, highlighting the areas where the model fell short and potential directions for future improvements.

## 5.1 VAE

### 5.1.1 Train & Validation

To monitor the learning process of the VAE model, we tracked the training and validation loss and metrics over the 20 epochs. The training loss represents how well the model fits the training data, while the validation loss indicates how well the model generalises to unseen data.

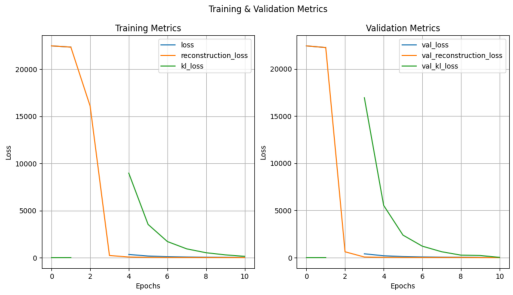


**Figure 5.1: Training and validation loss curves showing total loss, reconstruction loss, and KL loss over epochs.**

The training loss for visible in figure 5.1 begins at an extremely high value, initially registering as

infinity. This anomaly is why the blue line indicating the total loss is not visible until the fourth epoch. The spike in the initial loss is attributed to the encoder struggling to fit its output parameters to a normal distribution, resulting in infinite values. Following this phase, the total loss decreases gradually, with the early stopping callback in Keras preventing further overfitting.

The total loss consists of both the reconstruction loss and the KL divergence loss. Examining these individual loss components provides more detailed insights into the model's learning process. The reconstruction loss drops sharply, which suggests that the model quickly adapts to produce outputs that minimize this loss. However, a low reconstruction loss on its own does not guarantee a well-performing model, as it could simply indicate that the model has learned to output trivial solutions, such as predicting zeros.

Upon further analysis of the final training and validation loss values that are visible in table 5.1, it is apparent that the model's performance is suboptimal. The model does not make the best approximation of the data, indicating potential issues with overfitting or model complexity. These observations highlight areas for improvement and suggest alternative approaches to enhance the model's ability to predict spatio-temporal patterns more accurately.

In addition to the loss curves, the precision and recall metrics obtained from both training and validation further illustrate the model's limitations. The poor precision indicates that the model has a high rate of false positives, meaning it frequently predicts fish locations where there are none. Similarly, the low recall suggests that the model misses a significant number of actual fish locations, indicating a high rate of false negatives. These metrics reflect the model's difficulty in accurately identifying and predicting the presence of fish, which is crucial for its intended application.

### 5.1.2 Evaluation on Test Data

The performance of the VAE model was evaluated on a separate test set, comprising of the most recent 20% of the dataset. The metrics used to evaluate the model can be seen in the table 5.2.

The MAE of 2.18 indicates that the model's predictions had an average error of 2.18 units from the actual values. Given that the values are normalised, this error is quite substantial. This suggests that the model has difficulties in accurately capturing the underlying movement patterns of fish in order to predict their locations

| Metric | Train | Validation |
|---|---|---|
| MAE | 6.000 | 1.289 |
| KL Divergence | 139.829 | 29.369 |
| Total loss | 18.589 | 3.933 |
| Precision | 0 | 0 |
| Recall | 0 | 0 |

**Table 5.1: Performance Metrics for Train and Validation Sets**

| Metric | Value |
|---|---|
| MAE | 2.18 |
| KL Divergence | 22,432 |
| Precision | 0 |
| Recall | 0 |

**Table 5.2: Final Performance Metrics on the Test Set**

The KL Divergence value of 22,432 is extremely high, indicating that the latent variables do not follow the intended standard normal distribution. This high value suggests that the encoder is struggling to generate a latent space that effectively captures the crucial aspects of the data.

The precision and recall metrics both being zero highlight the model's complete failure to identify true fish locations, either predicting false positives or missing all relevant instances. These metrics collectively demonstrate that the model struggled significantly in achieving its prediction goals, indicating substantial room for improvement.

### 5.1.3 Example Outputs

To better assess and investigate the model's predictions, we compared the reconstructed outputs with the target data using a portion of the test data the model was evaluated with. This visualisation can be seen in figure A.1 in the appendix. This was done through the creation of a heat-map of the fish locations laid over a map of the region we are predicting.

The example outputs of the VAE model highlight significant discrepancies between the true and pre-

dicted fish locations. The true data shows a distinct and isolated fish location, represented by a yellow dot, which is clearly visible and shifts position over time. In contrast, the predicted data exhibits extensive noise and lacks a focused fish location, indicating that the model predictions are not aligned with the actual data. The model appears to simply try output values around zero to reduce the loss as the data consists mostly of zeros. This explains however why there is an initial large drop in the reconstruction loss that begins to stagnate as can be seen in 5.1.

### 5.1.4 Training & Inference Time

Discussing the computational efficiency of the model is important as it provides insights into its practicality for real-world applications. It is an important aspect in how feasible such a tool could be for fisherman.

The total training time for the model was approximately 1 hour and 20 minutes using the GPU of a Macbook pro with an M1 pro chip. The training time indicates the computational resources required to train the VAE model, which may be significant depending on the size of the dataset and the complexity of the model.

The average inference time per sample was 2 seconds, along with the total time to evaluate the mode being approximately 1 minute. This metric is crucial for understanding how quickly the model can make predictions in a real-time or operational setting. The inference time reflects the model's efficiency and feasibility for practical use in predicting fish locations.

## 5.2 ConvLSTM NFVP

The ConvLSTM Next Frame Video Prediction (ConvLSTM NFVP) model was developed to serve as a baseline for predicting fish locations. This section evaluates the performance of the ConvLSTM NFVP model using various metrics and compares it to the VAE model. Despite the efforts to develop this model, the results indicate significant challenges and limitations in its ability to predict fish locations accurately much like the VAE.

### 5.2.1 Training & Evaluation

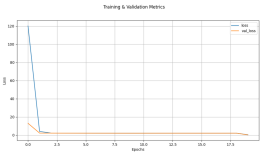


**Figure 5.2: Training and validation loss over 20 epochs of training**

Figure 5.2 displays the training and validation loss curves for the ConvLSTM NFVP model over 20 epochs. The training loss starts high, rapidly decreasing and stabilising after the second epoch. However, despite the low loss values, the precision and recall metrics were both zero, indicating a significant shortfall in predictive performance. This resembles the shape of the reconstruction loss of the VAE but appears to perform better. This is likely due to there being no influence of a latent space, which in the VAEs case was not able to converge to any meaningful latent representation of the data which impeded it's performance. The MAE for training and validation shown in table 5.3 indicate low absolute errors, which can be deceiving as they represent differences in the normalised data. So the fact that the differences are much larger is an important consideration. The precision and recall being zero highlight the model's complete failure to accurately predict fish locations, generating either false positives or missing all relevant instances.

| Metric | Train | Validation |
|---|---|---|
| **MAE Loss** | 0.411 | 0.222 |
| **Precision** | 0 | 0 |
| **Recall** | 0 | 0 |

**Table 5.3: Training and Validation Metrics for ConvLSTM NFVP Model**

### 5.2.2 Evaluation on test data

The performance on the test set, which is comprised of the most recent 20% of the dataset, further illustrates the limitations of the ConvLSTM NFVP model. The final loss of the model shown in table 5.4 was 0.222, indicating that while the model learned to some extent, it struggled to capture the

| Metric | Value |
|---|---|
| **MAE Loss** | 0.222 |
| **Precision** | 0 |
| **Recall** | 0 |

**Table 5.4: Test Metrics for ConvLSTM NFVP Model**

critical relationships in the data. The precision and recall metrics reaffirm the model's severe difficulty in identifying true fish locations and avoiding false positives. Though the ConvLSTM NFVP model achieves better results, it still does not achieve adequate results. This difference in performances between the models can largely be attributed to the lack of a latent representation in the ConvLSTM NFVP model. This assessment can be made as the structures of the decoder is very similar and function very similarly.

#### 5.2.3 Example Outputs

The example outputs of the ConvLSTM NFVP model, as shown in Figure A.4 in the appendix, illustrate the significant discrepancies between the predicted and actual fish locations. The model's predictions are essentially zeroes, failing to capture any meaningful patterns in the data. This issue arises because the product weight data is sparse, leading the model to minimize the loss by predicting zeroes. This behaviour results in a deceptively low loss value and poor predictive performance. Both models exhibit substantial discrepancies between the true and predicted fish locations. The VAE model's predictions are just noise, while the ConvLSTM NFVP model's predictions are overly simplified, indicating a failure to learn from the sparse data.

#### 5.2.4 Training & inference time

The model took approximate 54 minutes and evaluation time took approximately 26 seconds. The training and evaluation of this model was done using the same device as for the VAE. Overall the ConvLSTM NFVP model is more computationally efficient, with shorter training and inference times compared to the VAE model. This efficiency, however, does not translate to better predictive performance. The difference between the two models operating in inference mode is not significant, showing if these models are trained successfully they could be a feasible option for practical use.

# 6 Conclusions

## 6.1 Discussion of Results

The results of the study highlight significant challenges in developing robust models for predicting fish locations using spatio-temporal data. Both the VAE and ConvLSTM NFVP models exhibited limitations in accurately capturing the complex patterns necessary for reliable predictions.

The training and validation loss curves for the VAE model showed an initial spike, indicating difficulties in the encoder's ability to fit its parameters to a normal distribution. Despite a subsequent decrease, the model's overall performance remained suboptimal, with a high KL divergence suggesting that the latent space did not effectively capture the underlying data structure. Additionally, the precision and recall metrics were zero, indicating a complete failure to identify true fish locations. This poor performance is further confirmed by looking at the models' prediction, which appear to simply predict noise rather than provide anything informative.

Similarly, the ConvLSTM NFVP model, based on the Keras tutorial, showed low MAE values due to normalisation but failed to produce meaningful predictions. The precision and recall metrics were also zero, reflecting the model's inability to distinguish between fish and non-fish locations. The example outputs further illustrated this, with predicted maps showing no clear patterns or distinct fish locations, unlike the true data.

Both models unfortunately severely underfit the data, resulting in the VAE producing noise and the ConvLSTM NFVP developing a quick bias to outputting zeros. This indicates fundamentally the models were not able to do an adequate job and require more work on the limiting factors of the project.

## 6.2 Limitations

The performance of the developed models in this study was significantly impacted by various limita-

tions, primarily related to the data preprocessing and representation. Several key issues were identified that hindered the models' ability to learn and generalise effectively.

One major limitation is the nature of the catch notes data, which does not provide information on the movement trajectories of fish. This lack of temporal continuity means the data suggests that fish remain stationary for a day and then suddenly appear in different regions the next day. Such abrupt changes create an unrealistic representation of fish behaviour, making it challenging for the model to discern any clear direction or movement patterns. The absence of sequential movement data prevents the model from learning the natural flow and migration patterns of fish, which are crucial for accurate predictions.

Moreover, the sparsity of the catch notes data exacerbates this issue. The data often indicates large quantities of fish in specific regions while showing no presence in other areas. This binary-like distribution does not reflect the more gradual and diffuse nature of fish populations in reality. The model, therefore, struggles to generalise from such sparse data, leading to poor performance in predicting fish locations.

The preprocessing steps, while necessary to align the data for model input, may also have introduced further complications. The conversion of monthly salinity readings to daily averages and the handling of missing data through interpolation, although essential, could have led to a loss of important temporal nuances. These nuances might have been critical for the model to understand the subtle environmental factors influencing fish movements.

These limitations suggest that future work should focus on improving the data representation by incorporating more continuous and detailed movement data, potentially from tagging studies or higher resolution spatio-temporal data sources. Additionally, refining preprocessing techniques to better preserve the inherent patterns in the data could enhance model performance. Addressing these data-related challenges is crucial for developing more accurate and reliable models for predicting fish locations.

## 6.3 Statement of impact

The findings indicate that data preparation and representation are the most critical areas that need improvement for machine learning to effectively aid in optimal and sustainable fishing practices. While deep learning models has potential, the current study highlights that without substantial investment in improving data representation for these models, achieving meaningful success remains unlikely. This research underscores the necessity for refined data strategies to harness the full capabilities of machine learning in fisheries management.

## 6.4 Future work

### 6.4.1 Incorporating fishing vessel data

A promising direction for future work is to incorporate fishing vessel data to capture movement trajectories. This data can provide valuable context, helping the model to better understand fish movement patterns. By leveraging this additional data, convolutional layers can be utilised more effectively, allowing the model to capture spatio-temporal dependencies with greater accuracy. Integrating such trajectory data can enhance the model's ability to predict fish locations by learning from the paths and behaviours of fishing vessels, which are often correlated with fish presence.

### 6.4.2 Using statistical movement domain

Another significant improvement can be achieved by encoding spatial information through statistical movement domains, as described in Dong et al. (2016). This approach can help mitigate the issue of data sparsity by transforming raw spatial maps into a more informative and compact representation. By summarising the movement patterns statistically, the model can then focus on the most relevant features, potentially improving predictive performance. This method could provide a richer and more informative representation of the data, facilitating better learning by the model.

### 6.4.3 Altering the loss function

Modifying the loss function to penalise errors differently based on the type of prediction error (incorrect zero location vs. incorrect fish location) can

also lead to substantial improvements. This tailored loss function can guide the model to place greater emphasis on accurately predicting fish locations, which are more critical for the application. By distinguishing between types of errors, the model can be trained more effectively to prioritise reducing the most impactful mistakes, enhancing overall predictive capability.

#### 6.4.4 Additional considerations

In addition to these primary directions, future work could also explore applying a mask to the output layer of the models to force lower values to zero. This approach can help the model focus on predicting the stochastic nature of fish locations rather than noise. Furthermore, utilising t-Distributed Stochastic Neighbour Embedding (TSNE) to visualise the latent space can provide insights into the encoder's performance, helping to diagnose issues and refine the model.

## References


Brekke, Å., Dammen, J., Hole, K. A., Løddesøl, L., Ortheden, J., & Roaldsnes, T. (2022). Fishai: The lodestar fishing platform. *Nordic Machine Intelligence*, *2*(2), 10–12.

Dong, W., Li, J., Yao, R., Li, C., Yuan, T., & Wang, L. (2016). Characterizing driving styles with deep learning. *arXiv preprint arXiv:1607.03611*.

Greer, K., Zeller, D., Woroniak, J., Coulter, A., Winchester, M., Palomares, M. D., & Pauly, D. (2019). Global trends in carbon dioxide (co2) emissions from fuel combustion in marine fisheries from 1950 to 2016. *Marine Policy*, *107*, 103382.

Jensen, T. (2024, Jan). *Norway.* Retrieved from `https://eurofish.dk/member-countries/norway/`

Lambon, A., Sagun, E. F., Saet, M., Maranon, Z., & Berlin, S. (2022). Fishmaze: Fish monitoring and ai-based zone evaluation. *Nordic Machine Intelligence*, *2*(2).

Leggett, W. C. (1977). The ecology of fish migrations. *Annual Review of Ecology and Systematics*, 285–308.

Lehtonen, T. K., Wong, B. B., & Kvarnemo, C. (2016). Effects of salinity on nest-building behaviour in a marine fish. *BMC ecology*, *16*, 1–9.

Mood, A., & Brooke, P. (2024). Estimating global numbers of fishes caught from the wild annually from 2000 to 2019. *Animal Welfare*, *33*, e6.

Nordmo, T.-A. S., Kvalsvik, O., Kvalsund, S. O., Hansen, B., Halvorsen, P., Hicks, S., ... Riegler, M. A. (2022). Fish ai: Sustainable commercial fishing challenge. *Nordmo, TAS (2023). Dutkat: A Privacy-Preserving System for Automatic Catch Documentation and Illegal Activity Detection in the Fishing Industry.(Doctoral thesis). https://hdl. handle. net/10037/29768.*.

Shoji, J., Toshito, S.-i., Mizuno, K.-i., Kamimura, Y., Hori, M., & Hirakawa, K. (2011). Possible effects of global warming on fish recruitment: shifts in spawning season and latitudinal distribution can alter growth of fish early life stages through changes in daylength. *ICES Journal of Marine Science*, *68*(6), 1165–1169.

Team, K. (n.d.). *Keras documentation: Next-frame video prediction with convolutional lstms.* Retrieved from `https://keras.io/examples/vision/conv`$_l$*stm*/

# A Appendix

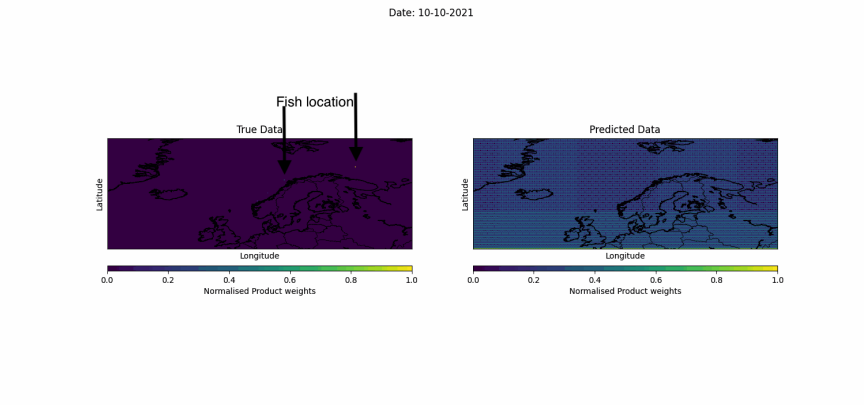


**Figure A.1: Frame 1 from VAE example outputs**

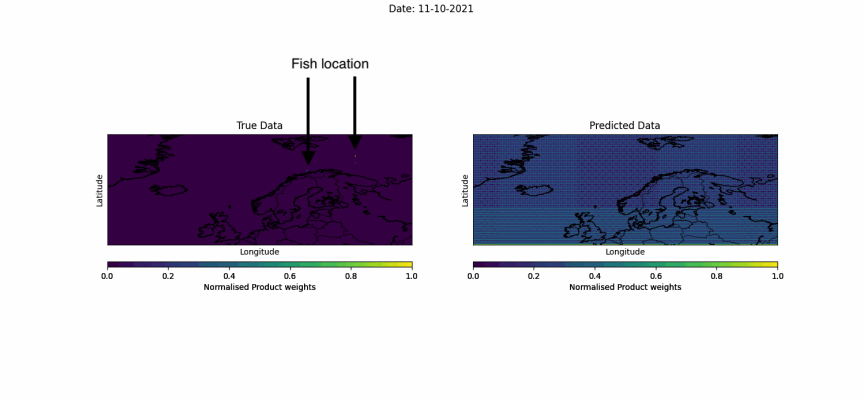


**Figure A.2: Frame 2 from VAE example outputs**

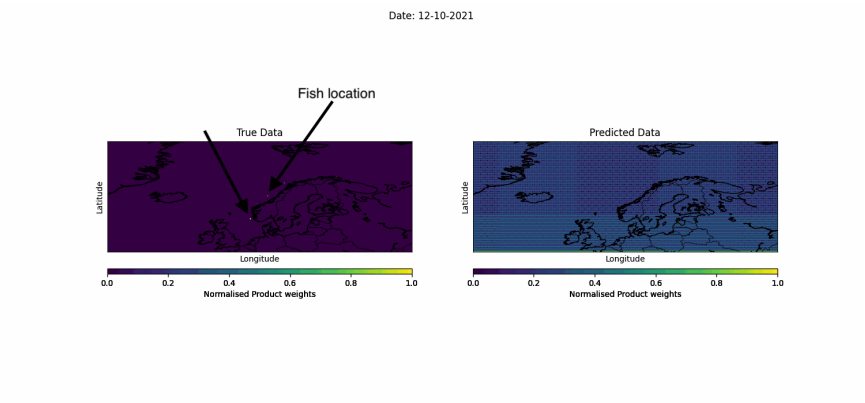


**Figure A.3: Frame 3 from VAE example outputs**

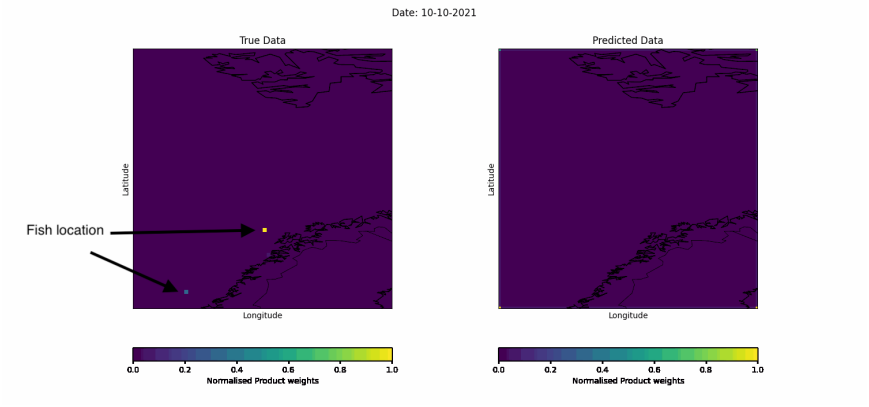


**Figure A.4: Frame 1 from ConvLSTM NFVP example outputs**

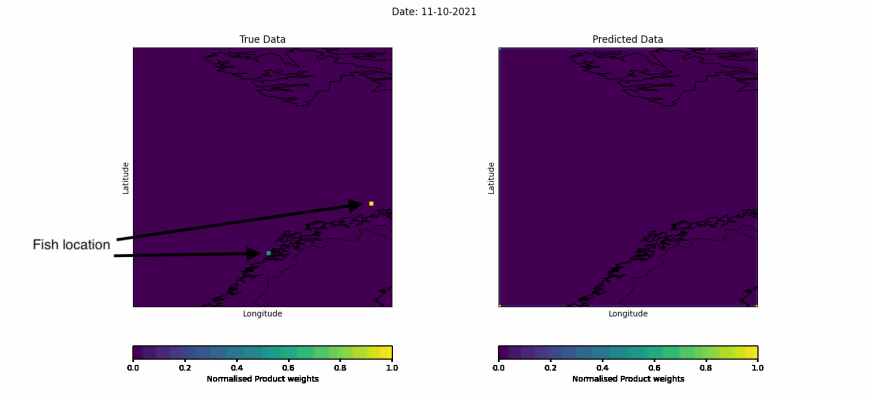


**Figure A.5: Frame 2 from ConvLSTM NFVP example outputs**

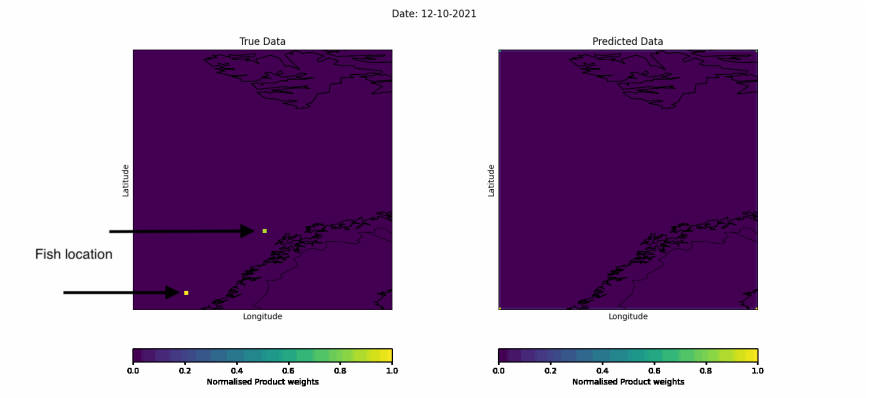


**Figure A.6: Frame 3 from ConvLSTM NFVP example outputs**