Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Engineering Cybernetics

Erik Spieler

Machine learning for detecting the activity of fishing vessels

Master's thesis in Industrial Cybernetics
Supervisor: Morten Omholt Alver
June 2023

NTNU
Norwegian University of
Science and Technology

Erik Spieler

Machine learning for detecting the activity of fishing vessels

Master's thesis in Industrial Cybernetics
Supervisor: Morten Omholt Alver
June 2023

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Engineering Cybernetics

NTNU
Norwegian University of
Science and Technology

Machine learning for detecting the activity of fishing vessels

Erik Spieler

Department of Engineering Cybernetics — NTNU 2023

Summary

This thesis presents a novel approach to predicting the activities of Norwegian vessels, specifically those engaged in fishing Norwegian spring-spawning herring (NSSH). The study sought to classify the behavior of fishing vessels by using two data sources: Automatic Identification System (AIS) data and Electronic Reporting System (ERS) catch data. The challenge lay in creating a useful dataset from the raw data and dealing with the particularities of geographical and temporal data.

This research approach adopted a methodical approach that entailed reducing the size of the data, transforming it, generating images, and extracting relevant features from the dataset. Two main approaches were tested: a semi-supervised and an unsupervised approach. The semi-supervised approach involved using a Convolutional Neural Network (CNN) and a k-means clustering algorithm. The process involved generating images from sectioned, transformed trajectories, after which the CNN was leveraged, utilizing its superior image classification capabilities. The CNN classified the labeled fishing events, while the k-means algorithm clustered the classified non-fishing events into searching and steaming. The unsupervised approach utilized the k-means algorithm on both the feature-based and trajectory-based datasets, clustering the data into their respective classes.

The semi-supervised approach achieved sustainable success in image classification, accurately distinguishing between fishing and non-fishing activities. The k-means algorithm for the semi-supervised approach achieved good results but suffered from the same issues as the unsupervised trajectory-based clustering approach. Despite its heavy reliance on the padding value, the unsupervised approach showed promise in trajectory-based clustering. However, it fell short in terms of clustering compactness and separation. In contrast, feature-based clustering exhibited impressive compactness and separation results, but its overall performance was slightly inferior.

The study used AIS and ERS catch data from the years 2015-2016 only. This relatively small dataset could have limited the generalizability of the models. The models' performance with larger and more recent data still needs to be tested.

Future work should focus on refining feature engineering strategies, optimizing classification boundaries, and adjusting trajectory plotting methods to improve analysis. Moreover, developing an effective post-processing method to transform the raw ML model output into identifiable positive and negative fishing events is critical.

The study demonstrated the potential for ML models to be used in classifying fishing vessel behavior based on AIS and ERS data. Both the semi-supervised and unsupervised approaches exhibited distinct strengths and weaknesses. However, for practical application, these models need further refinement and post-processing.

Sammendrag

Denne oppgaven presenterer en ny tilnærming til å forutsi aktivitetene til norske fartøyer, spesielt de som driver med å fiske norsk vårgytende sild (NSSH). Studien forsøkte å klassifisere atferden til fiskefartøyer ved å bruke to datakilder: Automatic Identification System (AIS) data og Electronic Reporting System (ERS) fangstdata. Utfordringen lå i å lage et nyttig datasett fra rådataene og håndtere de spesielle egenskapene til geografiske og tidsmessige data.

Denne forskningstilnærmingen tok i bruk en metodisk tilnærming som innebar å redusere størrelsen på dataene, transformere dem, generere bilder og trekke ut relevante funksjoner fra datasettet. To hovedtilnærminger ble testet: en semi-overvåket og en ikke-overvåket tilnærming. Den semi-overvåkede tilnærmingen innebar bruk av et Convolutional Neural Network (CNN) og en k-means klyngealgoritme. Prosessen innebar å generere bilder fra seksjonerte, transformerte fartøy baner, hvoretter CNN ble utnyttet, ved å utnytte dens overlegne bildeklassifiseringsevner. CNN klassifiserte de merkede fiskehendelsene, mens k-means-algoritmen grupperte de klassifiserte ikke-fiskehendelsene til søking og damping. Den uovervåkede tilnærmingen brukte k-means-algoritmen på både funksjonsbaserte og banebaserte datasett, og grupperte dataene i deres respektive klasser.

Den semi-overvåkede tilnærmingen oppnådde bærekraftig suksess i bildeklassifisering, og skilte nøyaktig mellom fiske og ikke-fiskeaktiviteter. k-means for den semi-overvåkede tilnærmingen oppnådde gode resultater, men led av de samme problemene som den uovervåkede banebaserte klyngetilnærmingen. Til tross for sin store avhengighet av polstringsverdien, viste den uovervåkede tilnærmingen lovende i banebasert clustering. Den kom imidlertid til kort når det gjelder kompakthet og separasjon. Derimot viste funksjonsbasert clustering imponerende kompaktitet og separasjonsresultater, men dens generelle ytelse var litt dårligere.

Studien brukte kun AIS data og ERS-fangstdata fra årene 2015-2016. Dette relativt lille datasettet kunne ha begrenset generaliserbarheten til modellene. Modellenes ytelse med større og nyere data må fortsatt testes.

Fremtidig arbeid bør fokusere på å avgrense funksjonsingeniørstrategier, optimalisere klassifiseringsgrenser og justere baneplottingsmetoder for å forbedre analysen. Dessuten er det avgjørende å utvikle en effektiv etterbehandlingsmetode for å transformere den rå ML-modellen til identifiserbare positive og negative fiskehendelser.

Studien demonstrerte potensialet for at ML-modeller kan brukes til å klassifisere fiskefartøyets atferd basert på AIS data og ERS-data. Både de semi-veiledede og uovervåkede tilnærmingene viste tydelige styrker og svakheter. Men for praktisk bruk trenger disse modellene ytterligere foredling og etterbehandling.

Preface

This thesis documents the work conducted in connection with my Master´s thesis during the spring semester of 2023. It represents one full semester of work and completes my two-year Master of Science program in Industrial Cybernetics at the Norwegian University of Science and Technology (NTNU), under the Department of Engineering Cybernetics.

I would like to express my sincere gratitude to my supervisor at NTNU, Morten Omholt Alver, for superior guidance and support throughout the semester. Our regular meetings led to fruitful discussions and served as a continuous source of motivation.

Disclaimer: All factual information in this master's thesis has been collected from credible sources. Chat GPT [1] has been exclusively used for grammar and spelling of the author's own generated text.

Contents

Nomenclature

Abbreviations

CNN	Convolutional Neural Network
ANN	Artificial Neural Network
IBM	Individual-Based-Model
EnKF	Ensemble Kalman Filter
AIS	Automatic Identification System
ML	Machine Learning
VHF	Very High Frequency
ERS	Electronic Reporting System
FC	Fully connected
MMSI	Maritime Mobile Service Identity
NSSH	Norwegian Spring Spawning Herring
GCS	Graphical Coordinate System
PCS	Projected Coordinate System
AUC	Area Under The Curve
ROC	Receiver Operating Characteristic
PCA	Principal Component Analysis
DBI	Davies-Bouldin Index
WCSS	Within-Cluster Sum of Squares

1 Introduction

1.1 Sustainable Harvesting of Marine Resources

1.1.1 Norway´s Traditions of Harvesting the Sea

Throughout Norwegian history, the people have made a living from fishing.[2]. Norway has a coastline stretching from the Barents Sea in the north to the North Sea further south. [3] Fish have been the prime marine resource and have traditionally been the basis for life on the coast, and for many of the coastal communities, this is still the case today. Approximately 70 percent of marine traffic in Northern Norway is from fishing fleet activity alone. [3]

Norway controls some of the richest fishing grounds in the world, such as the North Sea, the Norwegian coast, the Barents Sea, and the Polar Front in the Norwegian Sea. These are highly productive areas that provide the basis for one of Norway´s most important export industries, which is also one of the world's largest seafood suppliers.[2] 95 percent of Norwegian seafood is consumed all over the world, in order to meet international demand the catch volume must be high.

1.1.2 Fisheries Management

With the establishment of the international maritime law regime, coastal states gained jurisdiction over vast ocean areas and took on significant management responsibilities. The management task has been developing robust instruments and control systems to fulfill the coastal state's management responsibilities while ensuring good and stable operating conditions for industry stakeholders. [4]. "These include individual vessel quotas, a tradeable quota system, decommissioning schemes, real-time closures of fishing grounds, discard bans, and harvest control rules."[3]

1.1.3 Energy Use

Like all human activities, fishing uses energy, primarily for maritime conveyance and harvesting of aquatic organisms. The energy used in fishing may not be as noticeable as its direct impact on the fish populations and marine ecosystems, but its significance should not be underrated. The abundant energy, especially from fossil fuels, allows many modern fisheries to keep going even when fish numbers are dropping. [5]. Tyedmers [5] analyses the forms and quantities of energy dissipated in fisheries to provide powerful measures of the biophysical scarcity of exploited populations. By implementing data-driven methodologies and new technology, energy usage in the fishing industry can be reduced, thus positively impacting the environment. Furthermore, reducing energy usage is good news for stakeholders as their energy-related costs will be reduced.

1.2 Modern Monitoring of Marine Ecosystems

In the past couple of decades, the international commitment to managing regional ecosystems has emerged. This has consequently resulted in the development of programs devoted to the restoration and conservation of landscape-scale environments. [6] "Possessing advanced knowledge of fish stocks will aid in transitioning to environmentally and economically sustainable harvesting. Thus, there is increasing interest in monitoring fish stocks using data from commercial vessels." [3] "An article by Jones et al. [7] demonstrates how high-resolution data from the US reference fleet has contributed to abundance indices for several stocks, while footprints of fishing vessels can inform the planning of offshore wind projects" [3]. Shahir et al. [8] proposed an algorithmic method for the detection of routine vessel activities for fisheries monitoring. This process included sectioning vessel trips into micro-activities. Shahir et al. [8] method consisted of (1) extracting end-to-end trips, (2) activity detection, (3) end-to-end trip segmentation, and (4) trip type detection.

1.3 Background and Motivation

NTNU initiated this thesis on behalf of SINTEF [9] and their contribution to the project FishGuider. FishGuider is a project that aims to provide decision support for fishing vessels based on marine ecosystem models and fishery data. FishGuider seeks to increase income and reduce costs related to the catch of pelagic fish species, such as herring, cod, and capelin. The project combines new marine ecological models adapted to data from fishing vessels and satellites. The project has developed a migration model for Norwegian Spring Spawning Herring(NNSH) in connection with the physical, ecological ocean model SINMOD [10]. The system will guide fishermen to locations they can expect to find the right fish species within the given quotas. As a result, the time spent searching for fish or fishing in sub-optimal locations will be reduced. This will lead to more sustainable harvesting of ocean resources. Thus, the innovation will lay a foundation for better cooperation between the fishing industry and research and management communities in countries that harvest in the same ocean area. The fishing industry (IPN) and the MAROFF program in the Research Council of Norway finance the project. [11]

To further strengthen the predictive capacity, the behavior of vessels may be classified [3]. Positive fishing events, which denote times when a vessel is actively engaged in fishing, and negative fishing events, representing times when a vessel searches unsuccessfully for fish, indicating a lack of fish in the area, can provide valuable feedback to enhance the accuracy of the model. While fishing vessels could ideally report these events, this thesis proposes a different approach to gaining this feedback. Historical AIS data from fishing vessels were used to create a ML model that can classify these events. Since AIS is open-source information, it can be used in real-time to update the model and improve its accuracy. "An example of classification of vessel behavior can be found in Souza et al. [12], where fishing activities were detected using vessel speed as an observation input to a Hidden Markov Model."

1.4 Research Objectives and Research Questions

1.4.1 Objectives

Primary Objective: This study aims to investigate if a ML model can be used to classify the behavior of individual fishing vessels based on AIS data and ERS catch data. This data can be used as feedback for the FishGuider model that derives information about herring distribution.

Secondary Objectives:

- To establish a dataset by preprocessing AIS data and ERS catch data.
- To test different ML methods for classification and assess their performance.
- To provide recommendations for further work and improvements.

1.4.2 Research Questions

- Can AIS and ERS catch data be used to classify the behavior of fishing vessels during the annual spawning migration of Norwegian spring-spawning herring?
- What machine learning methods can effectively classify fishing behaviors and detect vessel activity?

1.5 Structure of the Thesis

- Chapter 2 of this thesis will explore crucial theoretical frameworks relevant to the study. The theory includes an introduction to the concepts of ML, highlighting elements such as CNN and Clustering Algorithms. Moreover, the chapter will cover details on the data types used for the study: AIS data and ERS catch data. At last, the different types of behavior of fishing vessels will be discussed. This section sets the foundation for understanding the following material presented in this thesis.
- Chapter 3 of the thesis discusses the methods used to answer the research questions. The method details the preprocessing steps in both the Semi-supervised and Unsupervised approaches. Also, the chapter will explain how the model training and evaluation work for both methods.
- Chapter 4 of the thesis displays the results gathered throughout the research. It includes relevant evaluation metrics and plots to clearly and concisely represent the findings.
- Chapter 5 of the thesis discusses the results presented in Chapter 4. This analysis involves a comparison of the different approaches, as well as a discussion of their respective strengths and weaknesses.
- Chapter 6 concludes the work presented and advises on further work.

2 Theory

This chapter provides an overview of the key concepts utilized in addressing the research question.

2.1 Introduction to Machine Learning

This chapter introduces machine learning, including an overview of its various types and examples of its applications.

2.1.1 Overview of Machine Learning

"Machine Learning(ML) is the field of study that gives computers the ability to learn without being explicitly programmed."[13]. Machine learning aims to build algorithms that can receive input data and use statistical analysis to predict an output while updating outputs as new data becomes available. The input data is used to identify patterns and relationships within the data and then use these patterns to make predictions or decisions on new, unseen data. [14]

2.1.2 Classification and Regression

A computer's ability to "learn" in this context is attaining the ability to perform a task. There are many tasks that can be solved with machine learning, but the two most common "tasks" are Classification and Regression.[15]

- **Classification:** In a classification task, the goal is to specify which of k categories(classes) some input belongs to. In solving this task, a function is created on a set of training data that relates any input to some output class. An example is identifying spam mail.
- **Regression:** In a regression task, the machine learning task is to predict some numerical value given some input. An example is, for example, predicting someone's age based on some input. The task is similar to classification except that the output is different.

2.1.3 Types of Machine learning

In machine learning, there are three main types of algorithms.

Supervised Learning
Supervised learning - the computer receives a set of inputs and desired outputs and aims to find the map between them[16]. In order to utilize a supervised learning algorithm, it is mandatory to have accurate target data; in many applications, this is not the case.

Unsupervised Learning
Unsupervised learning - the computer receives a set of inputs but obtains no supervised target outputs. In unsupervised learning, the algorithm aims to find patterns and structures in the unlabeled data. Two classic examples are clustering and dimensionality reduction.[16].

Reinforcement learning
Reinforcement learning - aims to learn the behavior of software agents or robots based on feedback from the environment[16]. The agent receives feedback through rewards and penalties. This is utilized to improve the agent's knowledge and decision-making.

2.1.4 Applications of Machine Learning

There is a vast quantity of applications that ML is being used for today. A few examples are:

Image Recognition
Image recognition is one of the most common applications of machine learning. It is used to identify objects, persons, places, digital images, an much more. [17]

Speech recognition
Speech recognition is the process of converting spoken instructions into text, also known as "speech-to-text" or "computer speech recognition". Machine learning algorithms are now widely used in various speech recognition applications. Google Assistant, Siri, Cortana, and Alexa use voice recognition technology to follow voice instructions. [17]

Banking
Banks have started implementing ML algorithms to prevent fraud and protect accounts from hackers. The algorithms determine what factors to consider to create a filter to detect illegal activities. Various unauthentic sites are automatically filtered out and restricted from initiating transactions. [18]

2.2 Convolutional Neural Networks

CNNs are a type of Artificial Neural Network (ANN) with a deep feed-forward architecture and good generalizing ability compared to other networks. [19]. CNNs can learn highly abstracted features of objects and identify them efficiently. CNN is most commonly used when there is an unstructured data set, for example, images, and the goal is to extract information from it.[14] It has achieved good results in applications such as image classification, object detection, face detection, speech recognition, and many more. [19]

"Convolutional neural network is composed of multiple building blocks, such as convolution layers, pooling layers, and fully connected layers, and is designed to automatically and adaptively learn spatial hierarchies of features through a backpropagation algorithm." [20]

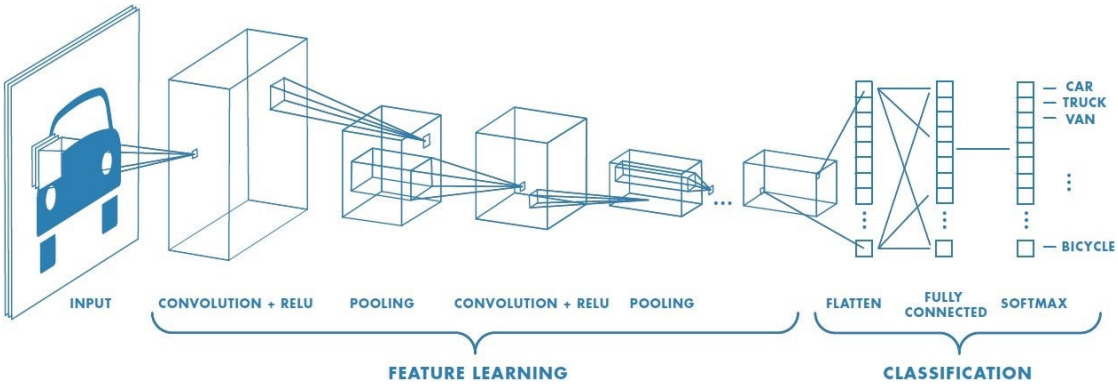


Figure 2.1: Conceptual Model of a CNN Illustrating the Relevant Layers.[21]

2.2.1 Model Architecture

Convolutional Layer

The Convolutional layer is a fundamental building block in CNN. It plays a crucial role in image analysis and is responsible for extracting features from the input image.

A Convolutional layer consists of a set of filters known as kernels. These kernels are matrices that perform a dot product operation on a small region of the input image, known as the receptive field.[22] The output of this dot product operation is a scalar value representing a specific feature in that region of the input image.

During the training process, the weights of these kernels are continuously updated using a process called backpropagation, allowing the Convolutional layer to learn and extract more complex features as the training progresses.

The kernel slides over the input image horizontally and vertically, with the slide size determined by the stride parameter. The choice of kernel size and stride can significantly impact the computational complexity of a CNN, affecting the time required for computation.

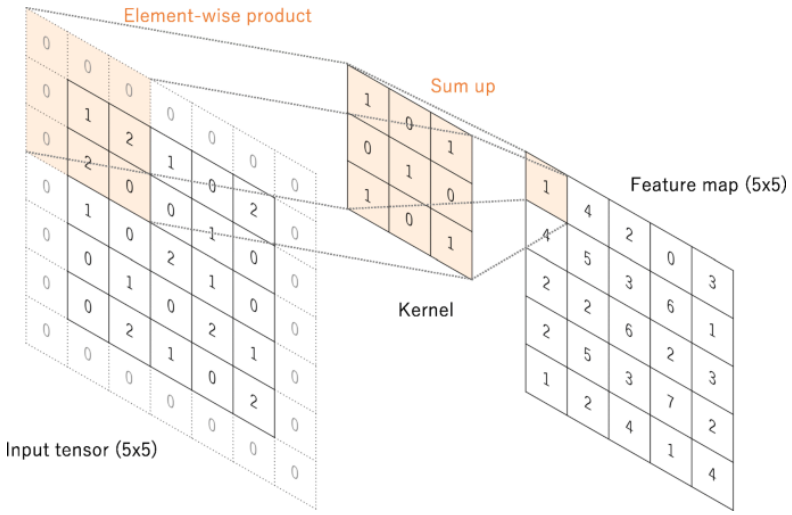


Figure 2.2: Illustration of a Convolutional Operation in a CNN.[20]

Figure 2.2 illustrates a convolutional operation with zero padding. The figure shows the kernel size is (3,3) and the stride is 1.

Pooling Layer

A feature map is the output of one filter applied to the previous layer. Pooling layers are used to reduce the size of the feature maps. Pooling is achieved while preserving the most important information in the feature maps.

As for the convolutional layer, the poling layer slides a "filter" across the image horizontally and vertically, except that the operation differs depending on the chosen technique. The techniques are max pooling, min pooling, average pooling, gated pooling, and tree pooling. Max pooling chooses the largest element in the sub-region in the feature map and discards all other values. Min pooling chooses the lowest value; average pooling calculates the average of all the values. Among them, Max Pooling is the most popular and widely used technique. The dimension and step size of the "filter" is defined by the region size and stride of the operation, similar to the convolution operation.[19]

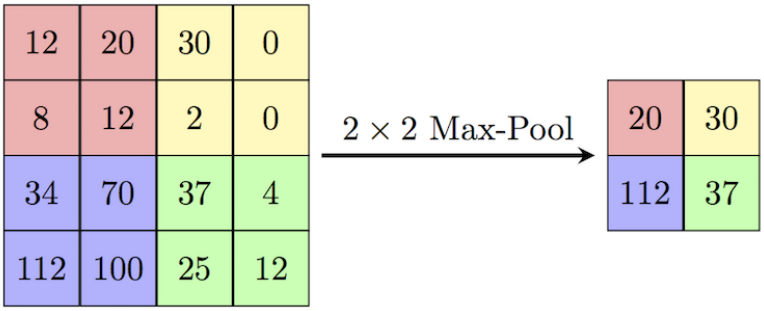


Figure 2.3: Illustration of a Pooling Operation in a CNN.[23]

Figure 2.3 depicts applying a max pooling operation to a feature map of size (4,4). By employing a pooling layer with size (2,2) and stride (2,2), the feature map is reduced to a size of (2,2).

Activation Function

The activation function applies a transformation to the output of each neuron from a linear operation, such as a convolutional layer, by applying a nonlinear activation function. [22] Without the activation function, the network struggles to learn complex and nonlinear relationships between the input and output data. The most commonly used activation function used in CNNs are:

- Sigmoid: The Sigmoid activation function bounds the input to the range [0,1] and is defined as[19]:

$$\sigma(z) = \frac{1}{1 + e^{-z}} \tag{2.1}$$

- Rectified Linear Unit(ReLU): The ReLU activation function converts the input into positive values and is defined as:

$$\text{ReLU}(z) = \max(0, z) \tag{2.2}$$

- Hyperbolic Tangent(Tanh): The Tanh activation function bounds the input to the range [-1,1] and is defined as:

$$\tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}} \tag{2.3}$$

All three activation functions are widely used due to their computational efficiency. However, the ReLU function is the most commonly used due to its minimal computational load, which is important for the efficient training of large models. [24]

Fully Connected(FC) Layer

A fully connected layer connects each neuron from the previous layer to a specified number of neurons in the fully connected layer. FC layers are used trough-out the CNN architecture but have an important purpose as the last layer in a CNN. The FC layer works as a classifier. Hence, the amount of neurons in the last FC layer corresponds to the number of classes the model predicts. [19]

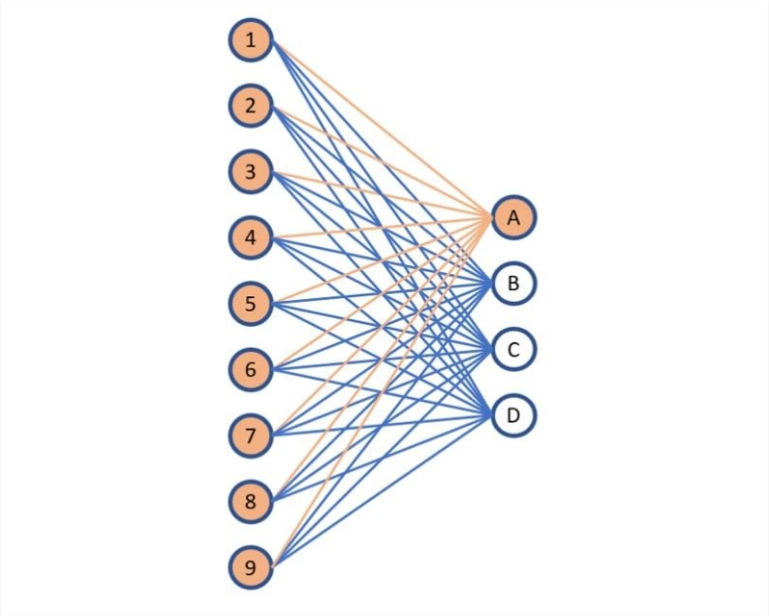


Figure 2.4: Illustration of a Fully Connected Layer.[25]

Loss Function

After computing all layers within the CNN, the final FC layer yields a probability ranging from 0 to 1 for each potential class. The CNN's prediction corresponds to the class associated with the neuron exhibiting the highest probability. To measure the difference between the predicted labels (output neuron probabilities) and the ground truth, a loss function is used. The error is used to update the model weights to predict new data better. "When training, the aim is to minimize this loss between the predicted and target outputs." [26]Various loss functions are used for distinct problem types, with the most widely used being Cross-Entropy, Euclidean, and Hinge loss functions.[19]

- **Cross-Entropy Loss:** Also called Log Loss, is typically used in binary and multi-class classification problems. The formula for Cross-Entropy Loss is:

$$H(p, y) = -\sum_{i=1}^{N} y_i \log(p_i) \quad \text{where } i \in [1, N] \tag{2.4}$$

  where y is the true label, and p is the predicted probability for each category, and N is the number of samples.[19]

- **Euclidean Loss:** Also called Mean Squared Error, is commonly used for regression problems. It calculates the distance between the predicted and the actual value. The formula is:

$$H(p, y) = \frac{1}{2N} \sum_{i=1}^{N} (p_i - y_i)^2 \quad \text{where } i \in [1, N] \tag{2.5}$$

  where y_i is the true label, $\hat{y}_i$ is the predicted value, and N is the number of samples.[19]

- **Hinge Loss:** The Hinge Loss function is widely used in binary classification problems, typically with Support Vector Machine (SVM) classifiers. It measures the error for a given data point by considering the margin between that point and the decision boundary. The formula is:

$$H(p, y) = \sum_{i=1}^{N} \max(0, m - (2y_i - 1)p_i) \quad \text{where } i \in [1, N] \tag{2.6}$$

  where y is the true label, and $\hat{y}_i$ is the predicted label, m is the margin which is normally set equal to 1, and N is the number of samples.[19]

2.2.2 Regularization

One of the significant challenges of deep learning algorithms is the problem of over-fitting and under-fitting. Overfitting arises when a model demonstrates good performance on training data but underperforms when presented with new data. In contrast, under-fitting occurs when a model fails to derive meaningful insights from the data, potentially due to an insufficiently large dataset. "The capacity of a model to effectively adapt to new, previously unseen data, originating from the same distribution as the data employed during its training, is referred to as the model's generalization ability." [19] Regularization aims to tackle this problem with several different techniques.

Dropout Regularization

Dropout regularization is a technique that prevents overfitting and is the technique that is most frequently used in deep learning. It has proven very effective and was first introduced by Hinton et al[27]. The term "dropout" refers to dropping out neurons(hidden

and visible) in a neural network. By dropping a neuron, it means temporarily removing it from the network, along with all its incoming and outgoing connections." [28] This introduces randomness into the model, which forces the remaining active neurons to learn more robust and generalized features instead of relying on specific neurons.

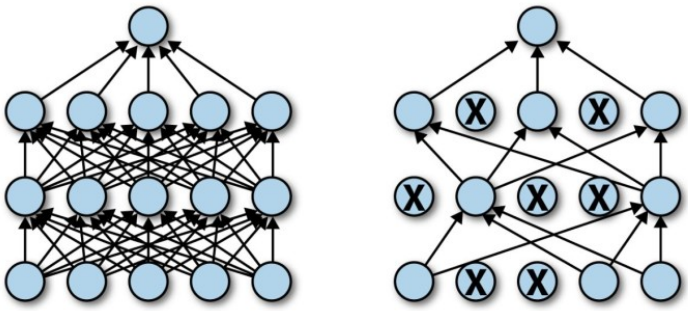


Figure 2.5: Illustration of how Dropout Regularization is Applied to a CNN.[29]

Early Stopping

Early stopping is a method that stops the training process if the model exhibits indications of overfitting. Cross-validation is used to determine if a model is overfitting, which involves examining the performance of both the training set and the test set for each epoch (a single iteration of the entire training data). If the training error continues to improve while the test error worsens, this indicates overfitting. Hence, the training process is stopped.

Data Augmentation

Data augmentation reduces overfitting and improves generalizability by changing the training dataset. The type of augmentations is numerous, including geometric and color transformation, random erasing, flipping, rotating, cropping, etc. "Data Augmentation prevents overfitting by modifying limited datasets to possess the characteristics of big data".[30]. Data augmentation transforms the data available to new data without changing the labels.

Batch Normalization

Batch normalization is a regularization technique that normalizes a layer's activation values (output of the neurons). It does this by subtracting the batch mean from the set of activations in a layer and then dividing by the batch's standard deviation. This technique, together with standardization, is often used when reprocessing pixel values. [30]

2.2.3 Evaluation Metrics

A range of evaluation metrics are used to evaluate a CNN's performance. The building blocks of these metrics are the samples referred to as true positive, true negative, false positive, and false negative.

- **True Positives (TP):** These are the positive instances that the classifier correctly identified.
- **True Negatives (TN):** These are the negative instances that the classifier correctly identified.
- **False Positives (FP):** These are positive instances that the classifier incorrectly identified.
- **False Negatives (FN):** These are negative instances that the classifier incorrectly identified.

These instances are comprised into a confusion matrix, a commonly used tool in machine learning for illustrating a model's performance. The matrix dimensions are $(n \times n)$, with 'n' representing the total number of classes. [31]

Some of the popular evaluation metrics are: **Accuracy**
This metric is used to represent the effectiveness of the classifier. It is computed by dividing the total of correctly classified data by the sum of all data. [32]

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.7}$$

Precision
Precision is a metric that shows the percentage of data correctly predicted as positive out of all positive predictions. In simpler terms, a higher precision corresponds to fewer false positives.[32]

$$\text{Precision} = \frac{TP}{TP + FP} \tag{2.8}$$

Recall
Recall is a metric used to measure the classifier's completeness. Higher recall corresponds to fewer false negatives, while lower recall corresponds to more false negatives. Generally, an increase in recall may lead to a decrease in precision.[32]

$$\text{Recall} = \frac{TP}{TP + FN} \tag{2.9}$$

F1-Score
The F1-score is calculated by dividing twice the product of recall and precision by the sum of recall and precision.[32]

$$\text{F1-Score} = \frac{2 \cdot (\text{Recall} \cdot \text{Precision})}{\text{Recall} + \text{Precision}} \tag{2.10}$$

AUC - ROC curve

ROC(Receiver Operating Characteristic) is a probability curve displaying the performance of a classification model at all classification thresholds. The ROC curve is created by plotting the true positive rate(TPR) against the flase positive rate(FPR) for various thresholds. The threshold is a set value that determines the decision boundary for the classification. [33] The TPR and FPR are defined as follows:

True Positive Rate:

$$TPR = \frac{TP}{TP + FN} \tag{2.11}$$

True Negative Rate:

$$FPR = \frac{FP}{FP + TN} \tag{2.12}$$

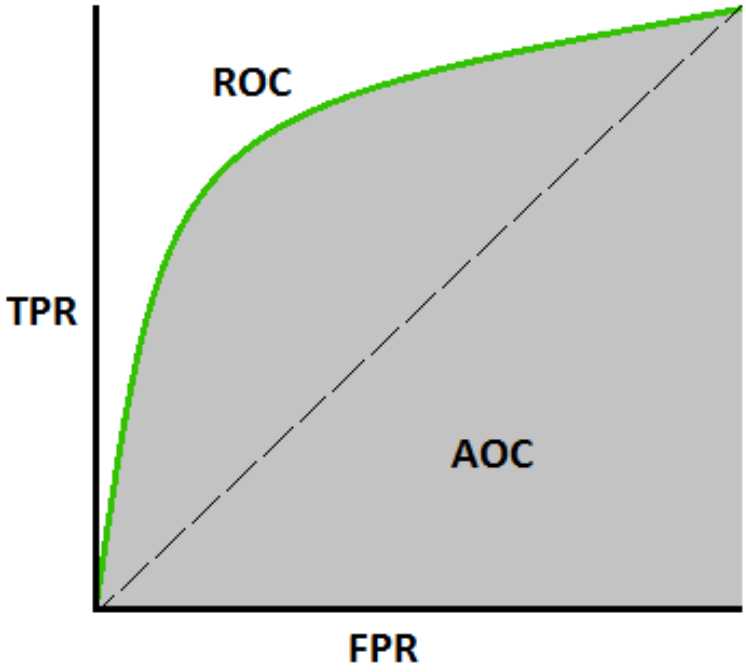


Figure 2.6: Example of a AUC-ROC curve [33]. AOC in the figure represents AUC.

AUC(Area Under the Curve) represents the degree of how well the model assigns a higher probability to a random positive example than to a random negative example. The AUC value, as the name implies, is the area under the curve of the ROC curve. The value ranges from 0 to 1, with higher values indicating that the model is better at distinguishing between positive and negative classes[33].

2.3 Clustering

Clustering is often regarded as the most important unsupervised learning method. Aggarwal and Reddy [34] defines the problem of clustering as: "Given a set of data points, partition them into a set of groups which are as similar as possible." In other words, clustering aims to identify a structure within a set of unlabeled data. A cluster is a collection of objects that exhibit similarity among themselves based on some similarity measure and dissimilarity with objects belonging to other clusters [35]. Clustering algorithms are used in many different applications, such as image segmentation, data mining, compression, and vector and color image quantization [36].

2.3.1 Clustering Techniques

In the context of clustering, two popular techniques are hierarchical and partitional clustering.

Hierarchical clustering

Hierarchical clustering algorithms generate a tree-like structure(dendrogram) at varying levels of granularity. [34] Depending on whether the hierarchical representation is created in a top-down or bottom-up fashion, the techniques are called divisive and agglomerative, respectively. [36]

Divisive
The divisive hierarchical algorithm starts by assigning the entire dataset as a single cluster. It then proceeds to iteratively divide one large cluster into smaller subclusters based on some similarity measure. This process continues until every data point is its own cluster; the result is a dendrogram. This data representation depicts the nested relationships between the clusters at various levels.[36]

Agglomerative
The agglomerative hierarchical algorithm begins by designating each data point as its own cluster. It subsequently proceeds to iteratively merge the two most similar clusters based on some similarity measure. This process continues until the entire dataset forms a single large cluster.

Partitional Clustering

With partitional clustering, the data is segmented into multiple groups simultaneously, typically by partitioning representatives. Partitioning representatives is often referred to as "centroids" or the center of the individual clusters. These centroids are used to divide the data into distinct clusters. Selecting the centroid and the distance function is essential as it controls the main operation of the algorithm. In every iteration, the data points are grouped with the centroid that's closest to them. Then, this centroid is modified according to the data points that are assigned to its respective cluster. [34]. The most popular partitional clustering method is the k-means method:

k-Means
The k-Means method is one of the most used methods due to its simplicity. The method uses the mean of each cluster as the partitioning representative and the Euclidean distance as the distance function. The pseudo-code is as follows:

1. Choose the number of clusters and obtain the data points
2. Initialize k centroids randomly: $C = \{c_1, c_2, \ldots, c_k\}$
3. Repeat until convergence:
   a) For each data point x_i, assign it to the nearest cluster centroid:
   $$S_j = \{x_i : ||x_i - c_j|| \leq ||x_i - c_k|| \text{ for all } k, i = 1, \ldots, n\}$$
   b) Update each centroid position to be the mean of the data points assigned to it in the previous step:
   $$c_j = \frac{1}{|S_j|} \sum_{x_i \in S_j} x_i$$

Here, $||x_i - c_j||$ denotes the Euclidean distance between data point x_i and centroid c_j, and $\sum_{x_i \in S_j} x_i$ denotes the sum of all data points x_i that are in cluster S_j.

2.3.2 Evaluation Methods

"Determining the quality of the results obtained by clustering techniques is a key issue in unsupervised machine learning" [37] This is due to the nature of the problem; since the data is unlabeled, it is not possible to use the same evaluation metrics as for a supervised problem. Evaluating the results of a clustering algorithm is a critical step in assessing its performance. Some metrics can be used to evaluate the performance of clustering algorithms, such as the Silhouette Score, David-Bouldin Index, and Inertia.

Silhouette score: The Silhouette score is a metric used to calculate the quality of a clustering technique. The score is calculated using each sample's average intra-cluster distance (average distance between each point within clusters) and the average inter-cluster distance (average distance between all clusters). Its value ranges from -1 to 1. Where 1 means the clusters are well separated, 0 indicates that the clusters overlap, and -1 indicates potential misclassifications [38].

$$SilhouetteScore = \frac{d_{ij} - \sigma_i}{\max\{\sigma_i, d_{ij}\}} \tag{2.13}$$

Where:

- σ_i is the mean intra-cluster distance, the average distance from the i^{th} sample to all other points within the same cluster.
- d_{ij} is the mean inter-cluster distance, the average distance from the i^{th} sample to all points in the nearest cluster.

Davies-Bouldin Index(DBI): DBI is a metric that assesses the performance of a clustering algorithm. It evaluates the separation between clusters, aiming for maximal inter-cluster distance and minimal intra-cluster distance. In terms of correlation, DBI positively correlates with the intra-cluster scenario and negatively correlates with the inter-cluster scenario.[39]

$$DBI = \frac{1}{n} \sum_{i=1}^{n} \max_{j \neq i} \left(\frac{\sigma_i + \sigma_j}{d_{ij}} \right) \tag{2.14}$$

Where:

- n is the number of clusters.
- σ_i and σ_j represent the average distances from each point in clusters i and j to their respective cluster centroids (intra-cluster distances).
- d_{ij} represents the distance between the centroids of clusters i and j (inter-cluster distance).
- The max function is taken over all $j \neq i$ for each cluster i.

Inertia or Within-Cluster Sum of Squares (WCSS): Inertia, also known as the within-cluster sum of squares, is a metric used to evaluate the quality of a clustering algorithm. It calculates the sum of the squared distances from each sample to the nearest cluster center, which it attempts to minimize. A lower inertia value indicates better clustering because each object is closer to its respective centroid. However, inertia can be biased towards clusters of similar sizes and may not yield meaningful interpretations when dealing with clusters of various shapes and densities [40].

$$Inertia = \sum_{i=1}^{n} \sum_{x \in C_i} (\|x - \mu_i\|^2) \tag{2.15}$$

Where:

- n is the number of clusters.
- x represents each data point in cluster C_i.
- μ_i is the centroid of cluster C_i.

Principle Component Analysis (PCA)
"PCA is a standard tool in modern data analysis and is used by almost all scientific disciplines. The goal of PCA is to identify the most meaningful basis to reexpress a given data set. It is expected that this new basis will reveal hidden structure in the data set and filter out the noise." [41] "This is accomplished by linearly transforming the data into a new coordinate system where (most of) the variation in the data can be described with fewer dimensions than the initial data." [42]

2.4 AIS Data

2.4.1 Introduction to Automatic Identification System(AIS)

AIS is an important surveillance tool for the coastal authority and the national preparedness along the coast. The AIS was first established as an anti-collision tool for the shipping industry and has been used in maritime transportation for over two decades.[43]. The AIS network consists of land-based and satellite-based AIS and is operated by the coastal authority. Ships with an AIS transceiver on board transmit dynamic information about their identity, speed, and course to nearby ships and shore-based stations via the Very High Frequency (VHF) band. The range at sea is around 20 nautical miles. There are approximately 5000 AIS-equipped ships in Norwegian waters at all times. For ships, the system is a supplement for radar-based information. [44] AIS transmits two main types of data: static and dynamic.

Static data

- MMSI number
- IMO number
- Name and Call Sign
- Length and Beam
- Type of ship
- Location of position fixing antenna

[45]

Dynamic Information

- Ship´s position with accuracy indication
- Position timestamp(in UTC)
- Course Over ground(COG)

2.4.2 Applications of AIS

As the quality of AIS data improves, its application expands to cover various other applications. The increasing number of publications and the wide range of topics related to AIS applications show how AIS has grown in recent years.

Vessel Tracking and Monitoring

AIS data can be utilized to track and monitor individual vessel movements. Shipping authorities, port operators, and other stakeholders can use AIS data to follow and monitor individual vessel movements in real-time and gain vital information. The data can improve the efficiency of port operations, control marine traffic, and help enforce legal compliance. Kaluza et al. [46] presented a successful study of ship movements based on AIS records.

Maritime Security

AIS data can be used to improve maritime security by detecting potential security threats, such as piracy, illegal fishing, and smuggling. Authorities can identify suspicious activities by analyzing vessel movements and behavior and taking the appropriate actions to protect maritime assets and resources. [47]

AIS improves maritime domain awareness and allows for heightened security and control. AIS enables authorities to identify individual vessels and their activity within or near a nation's Exclusive Economic Zone. In order to make better use of security resources,

possible dangers can be highlighted by automatically processing AIS data to produce normalized activity patterns for individual vessels. When these patterns are breached, alerts can be generated.[45]

Environmental Impact Assessments

Analyzing AIS data makes it possible to assess the environmental impact of shipping activities. This includes evaluating the emission levels of greenhouse gases, monitoring oil spills, and identifying areas prone to marine pollution. Winther et al. [48] used AIS data to present a detailed emission inventory for ships in the Arctic area.

Fisheries Management

Overfishing significantly threatens the marine environment because it depletes resources and destabilizes ecosystems. Furthermore, overfishing causes harm to the seabed, mainly through the use of vast nets towed by ships along the sea floor. AIS data can be used to monitor illegal fishing and help protect the marine environment.

2.4.3 Limitations and Challenges of AIS Data

AIS data is valuable and has many applications but it has some challenges and limitations.

Data Quality and Reliability

The accuracy of AIS information received is as good as the accuracy of the AIS transmitted. [49] The AIS data is self-transmitted by every vessel, which leads to inconsistencies and errors in the transmissions.

The IMO Convention for the Safety Of Life At Sea (SOLAS) Regulation V/19.2. 4 requires all vessels of 300 GT and above engaged on international voyages and all passenger ships, irrespective of size, to carry AIS onboard. [45]

AIS cannot be switched off, except for very few exceptions. According to IMO guidelines provided by Resolution A. 917(22), AIS should continuously operate when ships are underway or at anchor. A ship's crew, in singular circumstances, may turn off its AIS broadcast for various legitimate reasons. However, this behavior may indicate that a vessel hides its location and identity to conceal illegal activities. [45]

These include illegal fishing, smuggling, and unsanctioned trading. Furthermore, the AIS data quality is limited by the VHF range and can be affected by the curvature of the earth, atmospheric condition, and interference from other radio signals. This interference can lead to inconsistencies and inaccurate data.

Data Volume and Processing

As mentioned previously, AIS data is transmitted with a 10-second resolution, resulting in a significant amount of data due to the continuous transmission by numerous vessels. With each ship transmitting 86,400 data points daily, it can pose a challenge for data processing and analysis when analyzing a large number of vessels over an extended

period of time. Therefore, to derive meaningful insights, it is crucial to develop efficient algorithms capable of handling uncertainties in the data and supporting real-time decision-making processes.

Privacy and Security

The privacy and security of AIS data is a concern as AIS broadcasts are not encrypted or authenticated. This makes AIS vulnerable to unauthorized receivers within range, who can read and forward transmitted data to internet sites, leading to threats against privacy. Furthermore, the absence of source authentication makes AIS vulnerable to vessel spoofing, where false data can be broadcasted, giving rise to attack scenarios such as false alarms, traffic information, and maneuvering information. These vulnerabilities present a security issue regarding AIS data's confidentiality, integrity, and availability. [50]

2.5 Fishing Vessel Behavior

2.5.1 Types of Fishing Vessels

Various types of vessels are used for fishing. Vessels primarily used for fishing NSSH are Purse seines and trawlers.

Purse seines:
Purse Seines are large fishing vessels that deploy long nets that hang vertically from floats around the schooling of fish. To prevent fish from escaping, the school must be encircled swiftly. When the net surrounds the school, the bottom is sealed by drawing a purse line that is threaded through a series of rings along the base. The boat drifts with the attached net and hauls in to retrieve the fish. The overall time required for a single catch varies depending on the volume but can range between one and several hours. [12]

Trawlers:
Fishing trawlers are large marine vessels that use sizable trawl nets, systematically dragged along the ocean floor or within the water column. The ship maintains a steady and moderate speed, thus ensuring a consistent strain on the trailing net so aquatic species become entrapped within. Generally, one trawling session endures between 3 to 5 hours, the duration largely dependent on the concentration of the targeted species.[12]

2.5.2 Fishing Vessel Behaviour and Monitoring

In the context of this work, a fishing vessel is engaged in five primary activities:

- **Fishing:** This is defined as the period during which a vessel is actively involved in catching fish.
- **Steaming:** Characterized by a vessel moving along a virtually straight path at a relatively high speed.
- **Searching:** A state when the vessel travels at a slower speed and moves in multiple directions.
- **Stationary:** Occurs when the speed of the vessel is virtually zero.

- **Dark shipping:** A situation when a ship has switched off its AIS.

With dark shipping as an exception, these activities will be the classes that the ML models will attempt to predict.

2.5.3 Factors Influencing Fishing Vessel Behavior

Fish Distribution
The spatial distribution of fish significantly influences the navigational paths of fishing vessels. High concentrations of fish in a specific geographic region attract fishermen, who strategically seek out these areas to meet their catch quotas effectively.

Weather Conditions
Weather conditions such as wind, waves, and visibility can significantly affect the behavior of fishing vessels. Bad weather can force a vessel to change course, speed, or fishing activity to ensure crew and vessel safety. [51]

Fishing Regulations
Fishery regulations, such as catch limits, restricted areas, and gear restrictions, can affect the behavior of fishing vessels by determining where, when, and how fishing activities can be conducted. Vessels must comply with these regulations and adapt their behavior to optimize catches within the limits imposed by management measures.

Marked Demand
Market demand for seafood can also affect the behavior of fishing vessels. Changes in market demand can influence the target species, fishing methods, and operational strategies of fishing vessels seeking to maximize profits and meet consumer preferences.

2.6 Electronic Reporting System(ERS)

When fishing in other countries' waters and international waters, Norwegian vessels have been required to send various catch and activity reports for many years, in 2005 the Directorate of Fisheries facilitated the electronic system that enabled Norwegian fishing vessels to fulfill the reporting requirements by sending such reports electronically and replacing the manual system that existed by telefax and telex. Norwegian vessels can report their fishing activity using a single reporting system, whether operating in national waters, other countries' economic zones, or international waters. "All relevant information, including the final recipient of the report, is registered in the software on board the vessel, and the reports are sent electronically to the Directorate of Fisheries."[52] The regulation applies to all Norwegian fishing vessels over 15 meters, regardless of their location. When fishing, fishers must submit four reports: port departure, catch, port call, and transshipment. The data contains important environmental information about the ocean resources owned by Norway and is, therefor, open source information. [53]

2.7 Herring Distribution

Herring have similar migration patterns year after year. The herring learn migration patterns from older individuals in their school such that there tends to be a clear pattern that the schools follow each year. "Changes in migration pattern of Norwegian spring-spawning herring co-occur with the recruitment of abundant year classes to the spawning stock."[54] Huse et al. [54] models the changes in herring migration patterns according to the adopted-migrant hypothesis [55]. The herring pattern is feeding, overwintering then spawning migration.

2.7.1 Feeding

During the summer months, adult herring move to areas in the Norwegian Sea with a high zooplankton concentration. The amount of zooplankton can change a lot yearly, but the typical locations are the same year after year. Because of this, the herring can adjust and know where to find their food. [54]

2.7.2 Overwintering

Since 1987, a majority of the NSSH have been spending their winters in two fjords in northern Norway, namely Ofotfjord and Tysfjord[56]. They move to these specific places when winter starts, usually in October. This move when their main food, a type of plankton called Calanus finmarchicus, disappears from the top layers of their feeding grounds after the summer bloom.

During the winter months, the herring save energy and eat very little. A lot of their time in winter is spent avoiding predators. So, during the day, the herring stay deep in the water in schools to avoid some of their main predators. At night, some groups usually move to the upper parts of the water. Their main predators are cod, saithe, and killer whales. They stay in these fjords until the end of January. After this, they start their journey to their spawning grounds. [56].

2.7.3 Spawning Migration

After the herring's overwintering period, typically from January to March, the herring migrate to their spawning grounds with the goal of reproduction. These spawning migrations represent one of the most impressive examples of stability in herring migration. Every year, they return to the same areas to lay their eggs on a bed of gravel or stones. [57] In these areas, they can spawn and ensure the survival of their offspring. This period is a vital catch period for fishers. Kelly [3] developed an Individual-Based-Model(IBM) in combination with an estimation procedure commonly used in cybernetics called an Ensemble Kalman Filter(EnKF) to forecast this specific spawning migration of NSSH[3]. This spawning migration period will be the focus of this report.

3 Method

As outlined in chapter 1, the objective of this thesis was to develop a ML model capable of predicting specific vessel activity based on AIS data and ERS catch data. In order to determine the activity within a given time period, the AIS data were segmented into 30-minute intervals and then classified by applying the ML methods. Two distinct methods were explored: a semi-supervised approach and an unsupervised approach.

The dataset for this study consisted of AIS data collected from Norwegian fishing vessels during 2015-2016. The months analyzed were January, February, and March, as these are the months of highest interest with regard to the NSSH spawning migration. This research exclusively analyzed data obtained from vessels engaged in fishing for NSSH.

Semi-Supervised Approach
The semi-supervised approach aimed to leverage the labels derived from the ERS dataset, which could potentially enhance the performance and robustness of the overall model. This approach involved creating images of the segmented trajectories and utilizing a CNN for image classification of the vessel trajectories. The CNN was only used to classify fishing events in conjunction with the derived fish labels from the ERS dataset. Upon the classification of fishing events by the CNN, the negative classified class was clustered into three additional classes using the k-means clustering algorithm. The clustering algorithm used the preprocessed, segmented AIS data, as opposed to the generated images utlized for the CNN.

Unsupervised Approach
The unsupervised approach applied the k-means clustering algorithm to unlabeled, segmented, and preprocessed AIS data. This method sought to identify patterns and relationships within the data without the need for prior information from the ERS dataset. The unsupervised approach explored two different preprocessing techniques: trajectory-based and feature based, these will be explained in depth in this chapter.

3.1 Data Collection

Two primary datasets were utilized in this thesis project: AIS data and ERS catch data. The AIS dataset was provided by Norwegian Coastal Administration and the ERS catch data was obtained from the Directory of Fishers website. Both of these datasets provide distinct but complementary insights into the fishing activities of Norwegian vessels and were appropriately filtered and transformed to meet the specific needs of this study.

AIS Data
The AIS dataset contains daily logs from 2015 to 2016, each stored in individual CSV files. Each day's data provides roughly 800,000 entries that detail the position, speed, and identification of vessels. This dataset includes all commercial vessels over 15 meters operating in Norwegian waters, where each day of AIS data is stored in separate CSV files.

Catch Reports
As outlined in chapter 2, the ERS includes several reports from vessels including port departure, catch, port call, and transshipment. The focus of this study was primarily on the catch report. The catch report contains a great amount of information, the following list outlines the most relevant features of this study:

- **Reporting Year**: Year the report was sent
- **Message Type**: Type of report
- **Message ID**: Unique ERS number
- **Message Timestamp**: Date and time of the report
- **Vessel Name**: Vessel name
- **Vessel Nationality**: Flag state sender
- **Quota Type**: Quota type
- **Activity**: Fishing activity
- **Port**: Port code for call
- **Start Timestamp**: Date and time of the start of fishing
- **Start Position**: Latitude and Longitude position
- **Main Area Start**: Location where the fishing operation started
- **Zone**: Zone
- **Sea Depth Start**: Start zone
- **Gear Type**: Fishing gear used
- **Main Species**: Main species caught
- **Weight**: Weight in kilos

The report includes catch data for all species captured in Norwegian waters, therefore for the purpose of this study the data had to be masked to only contain vessels fishing for NSSH.

Translation Document
The AIS dataset and catch report use different identifiers for vessels: the AIS dataset uses a nine-digit Maritime Mobile Service Identity (MMSI) number, while the ERS dataset uses a four-letter callsign. This difference in identification systems posed a challenge for data integration, thus a translation document was used. The translation document serves as a bridge between the two datasets, facilitating the conversion of MMSI values into corresponding ERS callsigns. This conversion process is crucial for data alignment and ensures that the filtered ERS data can be appropriately applied as a mask for the AIS data.

3.2 Data Preprocessing

The process of data preprocessing included a series of well-structured stages to construct a valuable dataset that a ML model could effectively use. To manage the transformation of the entire dataset, a code capable of cycling through numerous directories was developed. The following preprocessing steps are the same for both of the approaches explored. The unique preprocessing steps for the approaches will be discussed in separate subchapters.

3.2.1 Data Reduction

Given the extensive amount of data provided in the AIS dataset, it was crucial to develop a method for efficiently reducing the size of this data and removing irrelevancies. The aim was to exclude any vessels not involved in NSSH fishing from the daily AIS dataset. This was done through three steps:

Filter ERS data
The ERS catch reports contain catch data for all fish species captured in Norwegian waters. Consequently, it was necessary to filter the ERS dataset to only include entries in which herring was the species being caught. Notably, although the ERS dataset is intended to represent catch data, it also contains entries where the vessel is not engaged in a fishing activity. As a result, the data was filtered to exclusively feature entries where vessels were actively fishing.

Translation from ERS Callsign to MMSI
As mentioned the AIS dataset did not contain the ERS callsigns that are used in the ERS dataset. The translation document was used to translate the callsign to the MMSI value that was used in the AIS dataset.

Filter AIS Data
Lastly, the filtered and translated ERS dataset was used as a mask on the AIS dataset to only include entries with vessels engaged in fishing activity.

This resulted in a dimensionality reduction of approximately 95%

3.2.2 Trajectory Segmentation

Trajectory segmentation involved dividing each vessel's daily AIS data into 30-minute segments. This process is carried out by processing a day's worth of masked AIS data and breaking down the data for each unique MMSI value into 30-minute intervals. Following this, all intervals are combined and stored as NumPy arrays. Due to the variable quality and resolution of AIS data, the 30-minute intervals may exhibit significant size differences. To address this issue, the arrays are padded to a fixed length in order to be grouped together. Ideally, a vessel transmits AIS data every 10 seconds, meaning that 30 minutes of data would result in arrays of length 180. However, to allow for higher resolutions, the fixed array size is set to 300. For arrays with a length shorter than 300, NaN values are used for padding. These 'NaN' values are temporarily used before being changed later in the preprocessing.

3.3 Preprocessing for Semi-Supervised Approach

This section describes the preprocessing steps taken for the Semi-Supervised approach.

3.3.1 Plot Sectioned Data

To utilize image classification with a CNN, the AIS data had to be converted into images. Consequently, the sectioned AIS data were plotted on a blank canvas, eliminating any geographical bias it may have had. The x and y axes of the plot were set to a fixed size of 7,000 meters by 7,000 meters to maintain a consistent trajectory size across all images.

Including Speed in the Plot
According to Shahir et al. [8] vessel activity is strongly correlated with speed. Several methods can be used to incorporate a vessel's speed into an image, such as adjusting the line thickness, altering the line color, or using velocity vectors based on the speed. The chosen approach involves plotting a dot for every 10-second interval. As a result, vessels moving at higher speeds have dots spaced further apart, while those moving slowly have dots positioned closer together. One challenge to consider is the resolution issue with AIS data, as it does not always provide data for every 10-second interval. This challenge is addressed by interpolating the sectioned data, ensuring that dots can be placed at 10-second intervals. RGB images contain 3 channels, hence resulting in more data per image. Opting for a grayscale image instead of color-coding reduces the image's dimensionality by a factor of three.

Trajectory Rotation
Since all the vessel trajectories in the dataset originate from the coast of Norway, there is a potential for a directional bias in the images. There are multiple ways to handle this, a popular method is to randomly flip and rotate the images to remove the bias. The method chosen for the report was to align all the trajectories to a common axis.

Image Resolution
When saving the images, it was crucial to select an appropriate resolution. A higher resolution offers greater detail but increases the training time for a ML model, whereas a lower resolution provides less detail but reduces training time. The resulting resolution was determined through experimentation and visual inspection of the detail within the image. The final resolution chosen was (693, 693).

3.3.2 Label Data and Convert Images

To implement a supervised ML model, it was necessary to label the images. This was achieved using the ERS data and MMSI translation, which converted Callsigns in the ERS data to MMSI numbers. Accurate labeling was vital for obtaining reliable results. To ensure the quality of labels, a minimum fishing threshold of 10 minutes was set. Consequently, if an image depicts fishing activity for less than 10 minutes, it was labeled as a non-fishing activity. To assign labels to the images, it was essential to include a unique identifier for each image. Consequently, the name of each image was constructed using the start date, end date, and MMSI number. The images were stored in a PNG format where the filename of each image consisted of a unique identifier. The image identifier and the corresponding labels were stored in a CSV file for each day of fishing.

A CNN requires input in the form of arrays rather than images stored in PNG format. As a result, the images were converted into arrays. Additionally, at this point, the labels were assigned to the arrays.

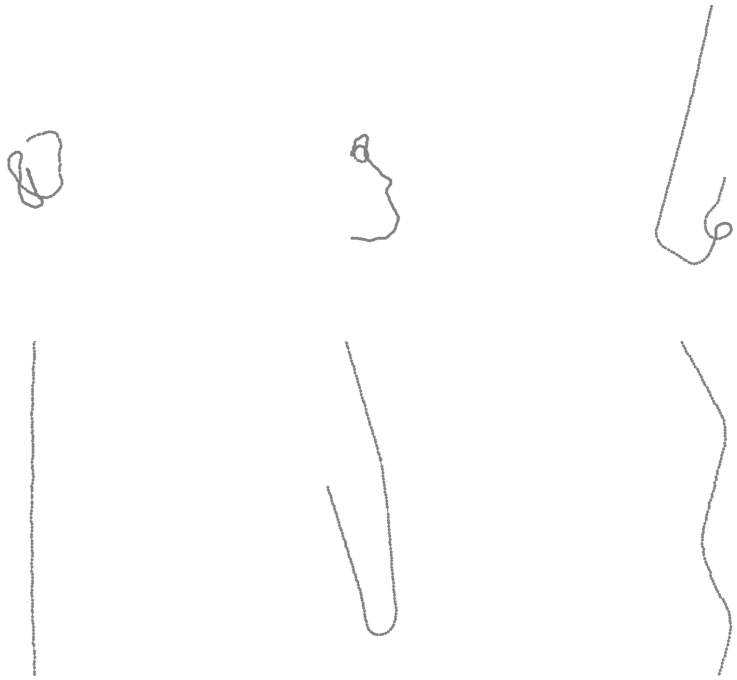


Figure 3.1: Example Images of the Generated Images Used For Image Classification.

Figure 3.1 displays a random selection of the images produced through the explained preprocessing technique.

3.4 Preprocessing for Unsupervised Approach

This section describes the preprocessing steps that were taken for the unsupervised approach. The first section is the same for both the trajectory-based and feature-based clustering strategies. The unique preprocessing steps of the two strategies are covered separately in the following two sections.

3.4.1 Resolution-Demand-Based Trajectory Segmentation

As part of the data preprocessing for the unsupervised approach, an additional step was taken to refine the trajectory data. While trajectory segmentation into 30-minute intervals was performed across all AIS data, a critical variation was introduced in this approach due to the inconsistent quality of AIS transmissions. There were occasions where AIS data could be missing for several minutes or even hours. Consequently, a 30-minute segment could contain fewer data points than expected, which could potentially bias the data and the subsequent ML model. To tackle this challenge, a resolution demand was implemented during trajectory segmentation in this approach. This resolution demand ensured a minimum number of data points per segment to guarantee a level of data consistency across all segments. If a segment didn't meet the resolution demand due to gaps in AIS transmission, it would be excluded from the dataset. This

preprocessing step enhanced the robustness of the data and improved the performance of the k-means algorithm.

3.4.2 Preprocessing for Trajectory Based Clustering

Coordinate System Transformation

The AIS vessel position coordinates were received in a Geographic Coordinate System(GCS). After the trajectory segmentation, the coordinates in the segmented data were transformed into a Projected Coordinate System (PCS). A PCS is beneficial as opposed to a GCS when geospatial data were used in clustering algorithms. The main reason for this transformation in this specific case was distance calculations. For the k-means clustering algorithm, as well as other machine learning methods, the Euclidean distance plays a significant role as it is used to calculate the distance between data points and centroids. In GCS, coordinates are expressed as latitude and longitude, which are not well suited to Euclidean distance computations. Thus, by transforming the coordinates to a PCS, which operates on a flat, 2D plane, it was possible to achieve more accurate distance calculations.

Data Normalization for Geographical Bias Mitigation

To ensure that equal significance was given to each feature (the transformed latitude and longitude coordinates) during the clustering process, it was essential to normalize the data. The feature scaling technique Min-Max Scaling was used to achieve this. Additionally, this scaling technique removed any potential geographical bias the data might have had.

Padding Adjustment

For most clustering algorithms, including the k-means method, the algorithm requires the absence of 'NaN' values in the dataset. The reason for this is because the k-means relies on calculating the Euclidean distance between the data points and the centroids, this is not possible for 'NaN' values. Consequently, these had to be addressed in the segmented arrays. One approach to handle this issue involves substituting the 'NaN' values with a fixed value. However, the selection of this value should be carefully considered, as it can significantly influence the outcome of the dataset analysis. This thesis explored two different padding values, 0 and -999.

3.4.3 Preprocessing for Feature-Based Clustering

In the feature-based clustering approach, a range of potentially significant features was computed from the sectioned AIS arrays. Then a feature selection method was applied to remove redundant features.

Feature Calculation

The following potentially significant metrics were computed:

- **Average Speed:** This is the mean speed of the ship across all data points. It is calculated using the 'SOG' column which stands for 'Speed over Ground'.
- **Min Speed:** This is the minimum speed the ship has traveled. It is calculated using the `min()` function on the 'SOG' column.
- **Max Speed:** This is the maximum speed the ship has traveled. It is calculated using the `max()` function on the 'SOG' column.
- **10th Percentile Speed:** This is the speed below which 10% of the speed measurements fall. It's calculated using the `quantile(0.10)` function on the 'SOG' column.
- **90th Percentile Speed:** This is the speed below which 90% of the speed measurements fall. It's calculated using the `quantile(0.90)` function on the 'SOG' column.
- **Number of Sectors Visited:** This calculates how many distinct sectors ('North', 'South', 'East', 'West') the ship has visited.
- **Standard Deviation of Speed:** This measures the amount of variation in the ship's speed. It's calculated using the `std()` function on the 'SOG' column.
- **Time Spent Stationary (%):** This is the percentage of time the ship's speed was less than or equal to 1. It's computed by finding the proportion of the 'SOG' values that are less than or equal to 1.
- **Average Change in Direction:** This is the average change in direction of the ship's heading. It's computed by first calculating the difference between all 'True_Heading' values, then taking the absolute value, and finally calculating the mean.
- **Distance Travelled:** This is the total distance traveled by ship. It's calculated by using the 'geodesic' function from the geopy. distance package to calculate the distance (in kilometers) between successive locations of the ship, and then sum up these distances.
- **Average Heading:** This is the average heading of the ship. It's calculated using the 'True_Heading' column which represents the ship's heading in degrees.
- **Heading Variance:** This is the variance of the ship's heading, which is a measure of how much the heading varies. It's calculated using the `var()` function on the 'True_Heading' column.
- **Bounding Box Area:** This is the area of the smallest box that contains all the ship's locations. The box is defined by the minimum and maximum latitude and longitude values.
- **Idle Time:** This is the total time the ship was idle, defined as when the 'SOG' is less than 1. It's calculated by summing up all the instances when 'SOG' is less than 1.

Correlation Matrix

The correlation between all features was calculated in order to determine if there were any redundant features in the feature set.

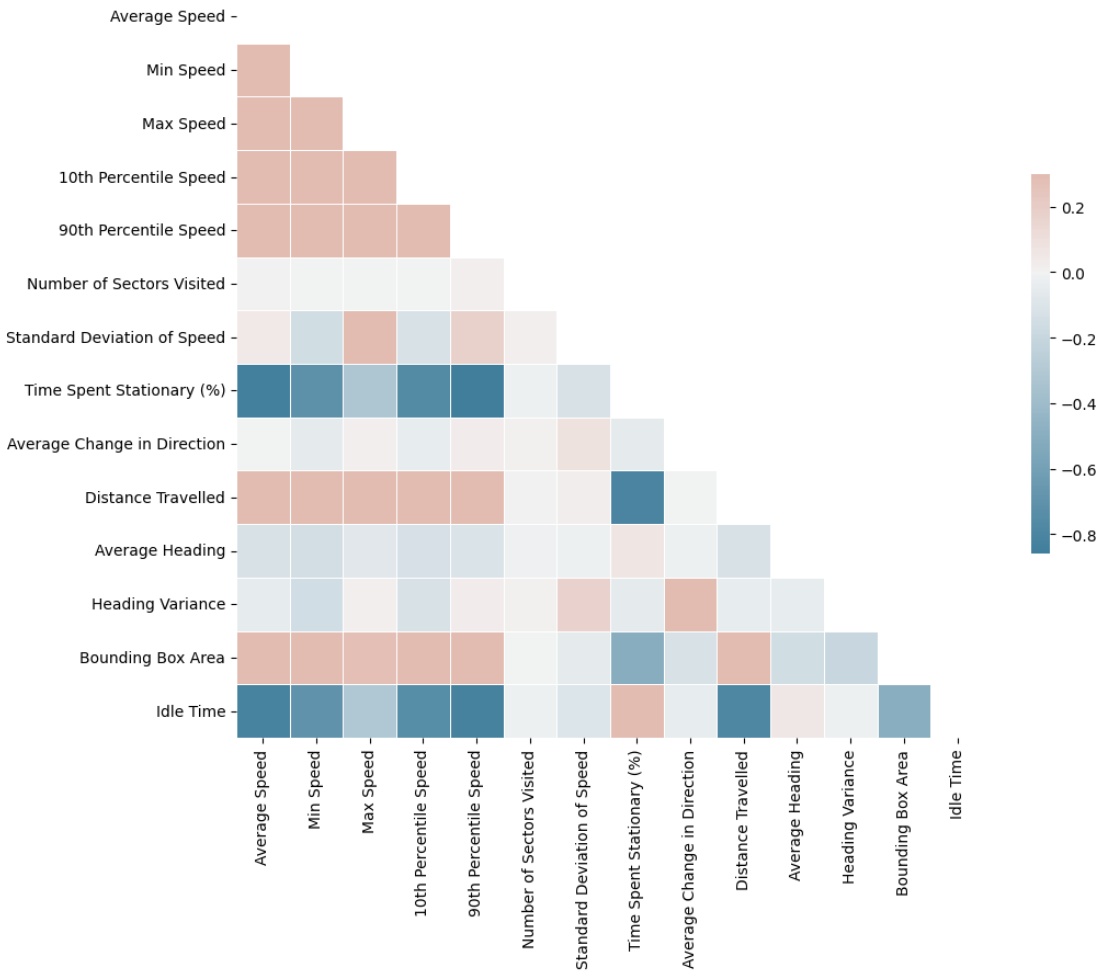


Figure 3.2: Correlation matrix of feature set. This matrix visualizes the pairwise correlation coefficients between all features used in the study. Darker cells indicate a stronger correlation.

Figure 3.2 represents the correlation matrix of the calculated features, visualized as a heatmap. Each square in the matrix corresponds to a correlation coefficient between two features, represented by the intersection of the respective row and column. The correlation coefficient measures the linear relationship between these two features.

The color palette ranging from dark blue to dark red tones signifies the strength and direction of the correlation. Dark blue indicates a strong negative correlation (close to -1), meaning that as one feature increases, the other tends to decrease. On the other hand, dark red tones indicate a strong positive correlation (close to 1), which implies that both variables tend to increase or decrease together.

The diagonal line from the top left to the bottom right represents the correlation of each variable with itself, which is always 1, signified by a neutral color. Areas with darker

blues and deeper pinks represent features that are highly correlated. Such features carry similar information, potentially introducing redundancy into the model. This issue is addressed by setting a threshold and removing features with a correlation above this threshold. The threshold is set to 0.7, meaning that if two features have a positive or a negative correlation above 0.7, one of the features is removed for the clustering analysis.

Filtered Correlation Analysis

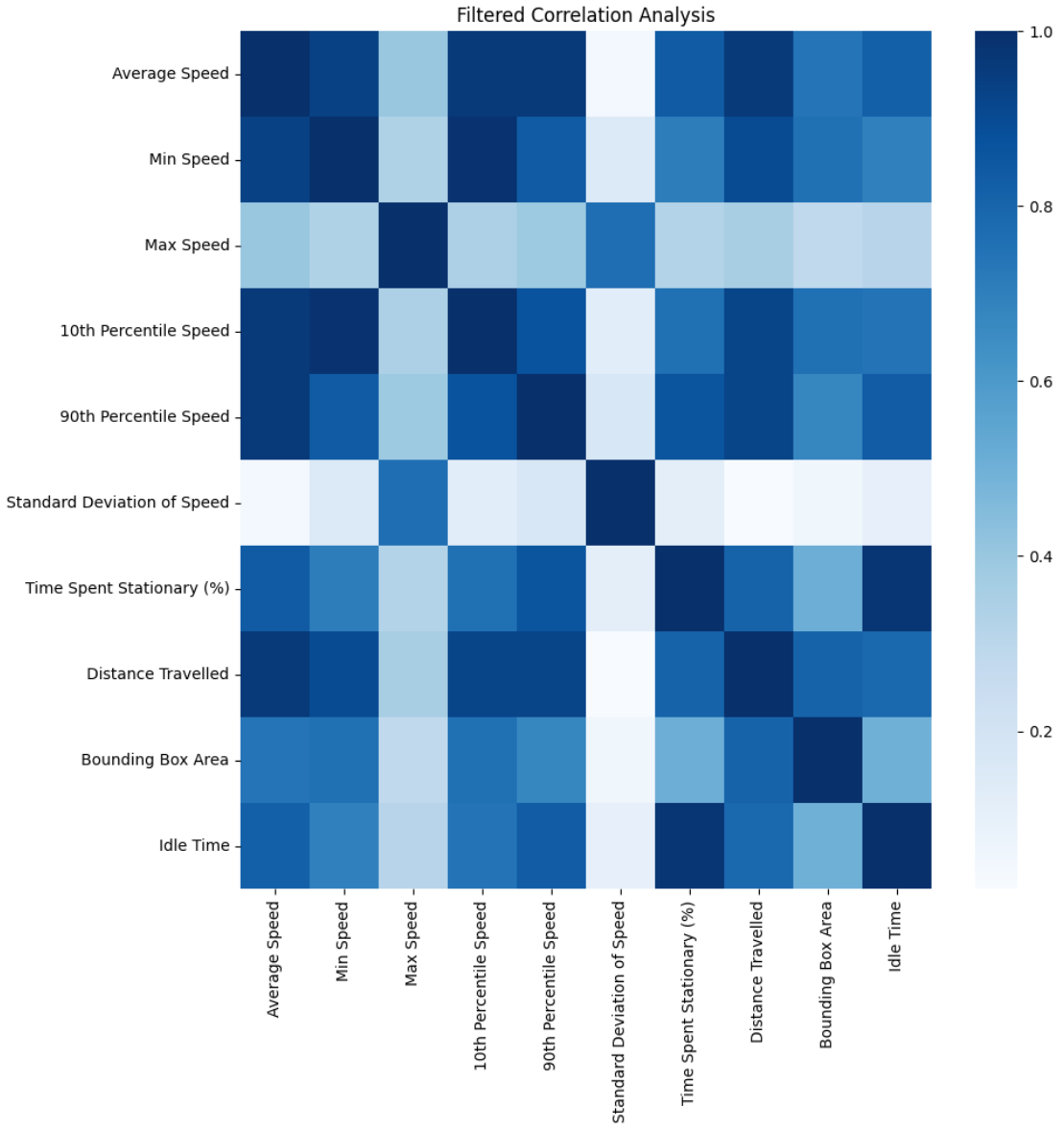


Figure 3.3: Heatmap of Filtered Correlation Matrix. Darker cells indicate a stronger correlation.

Figure 3.3 is a heatmap that represents the pairwise correlation coefficients of the features that exhibit a correlation above the previously mentioned threshold. The intensity of the colors indicates the strength of the correlation; darker blue signifies a stronger correlation.

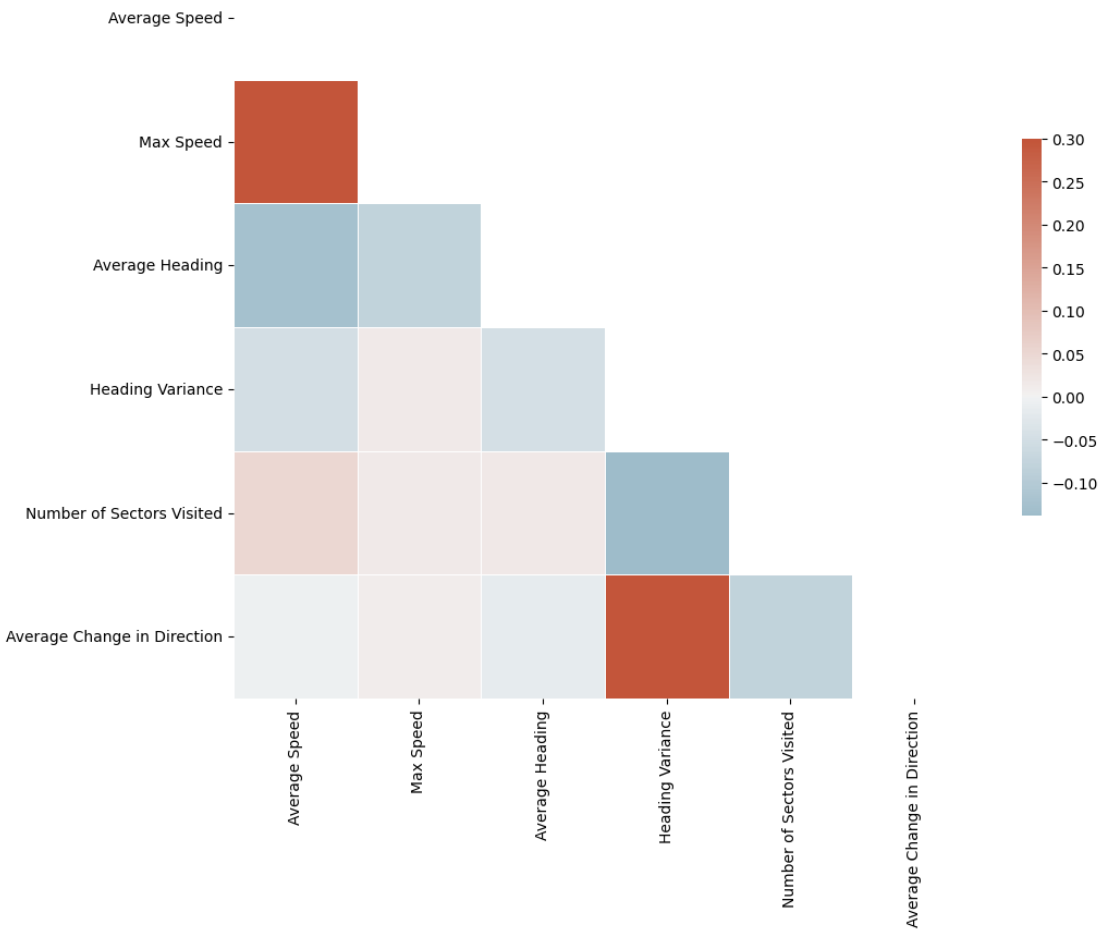


Figure 3.4: Correlation matrix of remaining feature set. Darker cells indicate a stronger correlation.

Figure 3.4 represents the correlation matrix of the remaining features after removing the redundant features. This is visually seen as the highest correlation between the remaining features is 0.3. These remaining features were used as input for the k-means clustering algorithm.

3.5 Model Training and Evaluation for Semi-Supervised Approach

As previously mentioned, the semi-supervised approach consists of two machine learning models, one supervised and one unsupervised. Hence, two separate machine learning models were trained and evaluated. For the supervised part, a CNN was utilized, and for the unsupervised part, a k-means clustering algorithm was utilized.

3.5.1 Convolutional Neural Network

Data Splitting and Random Shuffling
In order to provide the CNN with a sufficient amount of data for training, while still leaving enough data for performance evaluation, the dataset was split into a training set and a test set. 70 percent of the data was allocated for training and 30 percent was allocated for testing. This process was performed using the `train_test_split` function from the `scikit-learn` library. This function also shuffles the dataset randomly, this is advantageous because it breaks up patterns in the dataset and can reduce overfitting.

Data Balancing
Considering the class imbalance that existed in the image dataset due to the fact that vessels spend more time steaming, searching, and being stationary in port than actually fishing. It was crucial to address this imbalance in order to reduce the risk of biased model performance and poor generalizability of the models. The preprocessed 2015 and 2016 datasets contain 23206 images of non-fishing events and contain 1244 fishing events, thus only 5 percent of the dataset contains images of fishing events. To ensure a balanced dataset, a random undersampling method was utilized. The process was performed using the `RandomUnderSampler` function from the `imblearn` library. This function extracts 1244 random images from the non-fishing class.

CNN Architecture
With the vessel trajectory images preprocessed and split into a training dataset and a validation dataset, the CNN model was trained using `TensorFlow Keras`. The input images were reshaped to fit the expected input shape of the CNN, which was (693, 693, 1), representing the height, width, and number of channels (grayscale) for each image.

A CNN architecture should be tailored around the task at hand and the dataset used. Different tasks and datasets require different architectures for optimal performance. The final architecture was found through extensive testing of different layers and hyper-parameters. The final architecture was chosen due to its ability to generalize to the dataset and its computational efficiency. The final CNN model architecture is defined as follows:
CNN Architecture

1. A convolutional layer with 32 filters, a kernel size of (3,3), strides of (3,3), ReLU activation, and an input shape of (693,693,1).
2. A max-pooling layer with a pool size of (2,2).
3. A convolutional layer with 64 filters, a kernel size of (6,6), and ReLU activation.
4. A max-pooling layer with a pool size of (2,2).
5. A convolutional layer with 128 filters, a kernel size of (6,6), and ReLU activation.
6. A dense layer with 512 units and ReLU activation.
7. A Flatten layer to convert the feature maps into a 1D vector.
8. A dropout layer with a rate of 0.5 to prevent overfitting.
9. An output dense layer with 1 unit and sigmoid activation

The model was compiled using the Adam optimizer and a binary cross-entropy loss function. The Adam optimizer was chosen due to its adaptive learning rate and efficient convergence. Binary cross-entropy is a popular loss function in binary classification, making it well-suited as a loss function in this case.

CNN Evaluation
To evaluate the performance of the CNN, the test dataset was utilized to obtain the predicted labels from the model. These predictions were then compared to the fishing labels derived from the ERS data. A variety of validation metrics were used for the evaluation, including accuracy, precision, recall, F1 score, area under the curve (AUC), and receiver operating characteristic curve (ROC). These metrics were used to create bar charts and confusion matrices, allowing for easy interpretation of the results.

The false positive and negative arrays were saved for every model test to analyze and understand the model's predictions and thus improving the performance. The model history, including the accuracy and loss, was saved for every training process. These metrics were used to evaluate the model's training behavior and to identify any potential issues such as overfitting and underfitting.

3.5.2 K-means Clustering

Dataset
The dataset utilized for clustering was derived from the original dataset employed by the CNN. However, it was refined to exclude all entries that were classified as 'fishing' by the CNN. Consequently, this resulting dataset was intended to exclusively include vessel activities categorized as 'steaming', 'searching', or 'stationary'. It is worth noting that this dataset may still contain anomalies due to potential misclassifications by the CNN. The dataset provided to the clustering algorithms was initially the images the were created for the CNN, but due to excessive increase in time complexity when training the clustering algorithms, the approach was changed to only use the sectioned preprocessed AIS data instead.

Model Evaluation
For unsupervised classification problems, there doesn't exist any truth labels to calculate the accuracy of the clustering models' performance. As mentioned in 2.3.2, there are several metrics that can be used to evaluate the performance, but without visual inspection, it is difficult to get a clear picture of the model's performance. One way of visual validation is by examining the images that the algorithm has classified to determine if they appear correctly classified. However, without any contextual information, it can be difficult to determine if the images are correctly classified. An alternative approach is to plot individual vessel trajectories for each day and color-code the segments of the trajectories according to their respective classes. with this approach, a more robust understanding of whether the images have been accurately classified can be achieved.

3.6 Model Training and Evaluation for Unsupervised Approach

In this section, the model training and evaluation methods are outlined. The model evaluation is identical to the unsupervised part of the semi-supervised approach and will therefore not be repeated.

3.6.1 Data Processing

The data preprocessing steps previously carried out were adequate enough to start training a clustering algorithm. The process differs from supervised learning tasks where preprocessing steps such as splitting and shuffling data are essential for satisfactory results. In clustering, the aim is to identify a structure within a set of unlabeled data, so it is not necessary to split the dataset into a training and test set because there are no labels to validate the test set against. Furthermore, random shuffling is not needed either because the order of the data points does not affect the result.

There is one difference in the dataset used in this approach; the AIS data has been segmented with a resolution demand. This was done due to the occasional low quality of the AIS data. At times, the AIS data was missing data for several minutes or hours. This resulted in 30 min segment only containing as little as one data point. This could create a bias in data. In order to potentially improve the performance of the k- means algorithm this was done.

3.6.2 Model Selection

The clustering method used in this approach was the k-means clustering algorithm. The implementation of the method was rather straightforward with few parameters to choose. The parameters to choose are:

Number of clusters: The optimal number of clusters should ideally align with the number of distinct classes of interest, as discussed in 2.5.2. In this context, the activities of interest to be captured through clustering include Fishing, Steaming, Searching, and Stationary activities. If the hypothesis is correct, each of these activities should correspond to a unique cluster. Thus, the ideal outcome of the clustering algorithm would be four discrete clusters.

However, a potential challenge could arise due to data segmentation. It's a possibility that a single data segment might not correspond unambiguously to a unique activity, implying there may be some overlap between activities. This overlap could complicate the clustering process, potentially causing the algorithm to struggle in accurately partitioning the segments into four individual classes.

The Elbow method is a useful tool for determining the optimal number of clusters. This technique involves plotting the inertia score against a range of possible cluster counts. An "elbow" or bend in the plot often reveals the ideal number of clusters.

In this particular scenario, our goal is to ultimately yield four clusters. However, if the Elbow method suggests more than four clusters as optimal, we would need a strategy to reduce the number of clusters after training. One approach is to visually inspect the plotted trajectories and group the clusters based on their similarities. By examining the trajectories, one can determine if certain clusters should be grouped together or separated.

Distance Metric
The default distance metric in the k-means algorithm is the Euclidean distance. Depending on the dataset and the problem, a different distance metric can be used. In this specific case, the Euclidean distance was satisfactory.

Initialization Method
The result of the k-means method heavily relies on the initialization method. There are four popular initialization methods: random, Forgy, MacQueen, and Kaufman. Peña et al. [58] conducted an empirical comparison of these four initialization methods for the k-means algorithm and concluded that the random and Kaufman initialization methods outperformed the other methods. The reason for this was that these methods were more efficient and less dependent on initial clustering and instance order. Therefore the random method was used for this thesis.

4 Results

This chapter presents the results obtained from the ML models trained and tested on the preprocessed AIS and ERS catch data. This study focused on the fishing activities of Norwegian vessels, particularly those involved in fishing NSSH.

4.1 Semi-Supervised Approach

4.1.1 CNN Results

The best-performing CNN model architecture was determined to be the one detailed in Chapter 3.5.1. This model configuration achieved the best results when trained over seven epochs with a batch size 16.

The data used to train and validate this model consisted of vessel trajectories spanning 45 days, derived from data collected in 2015 and 2016. Initially, the dataset comprised 24,450 images, 1,244 of which were labeled as fishing events. After undersampling, the adjusted dataset consisted of 2,488 images.

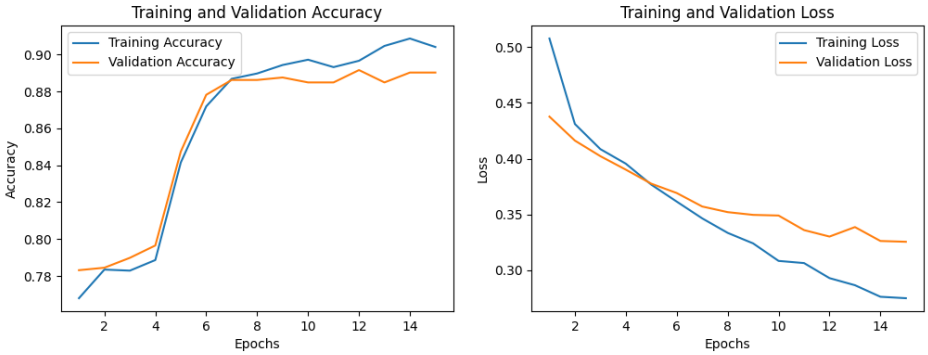
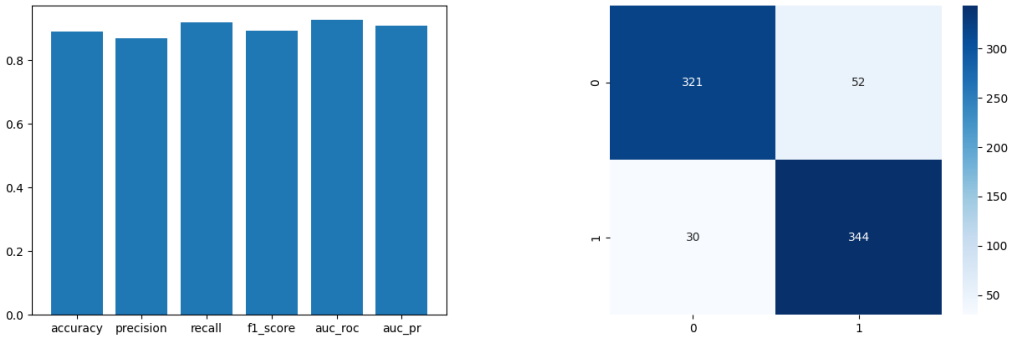


Figure 4.1: Plot of the Training History for the Final CNN Architecture.

Figure 4.1 shows the model's training and validation accuracy and loss throughout a training process with 15 epochs. Figure 4.1 shows signs of overfitting. Hence increasing the number of epochs is redundant. The figure shows that the ideal number of epochs with the given model configuration is between five and seven.



(a) (b)

Figure 4.2: (a) Bar chart of Evaluation Metrics for CNN. (b) Confusion matrix for the trained CNN. Here 0 represents the 'Non-Fishing' class and 1 represents the 'Fishing' class.

The resulting metrics are shown in 4.2a. The resulting scores are:

Metric	**Score**
Accuracy	0.8741633199464525
Precision	0.8465346534653465
Recall	0.9144385026737968
F1 score	0.8791773778920308
AUC ROC	0.9170406159051484

Table 4.1: Evaluation Metrics Results for Final CNN Architecture.

The confusion matrix, as depicted in Figure 4.2b, comprehensively evaluates the model's performance. The model correctly identified 321 instances where both the predicted and actual class were true negatives, signifying instances where the vessel was not engaged in fishing. However, it also misclassified 52 true negatives, indicating a degree of error in the model's predictions.

Conversely, the model demonstrated good accuracy in identifying true positives, which in this case, are instances where the vessel is indeed fishing. Of these, 344 instances were correctly classified, while in 30 instances, the model incorrectly predicted the class as a true negative.

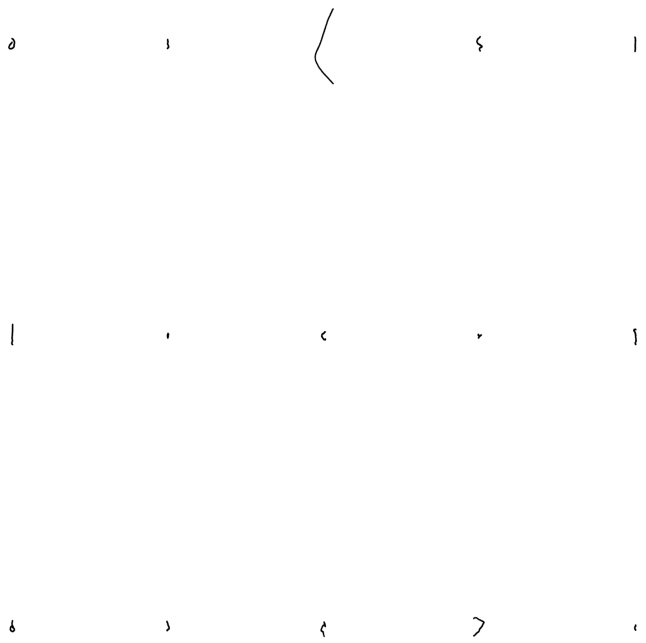


Figure 4.3: Images of Fishing Activity Classified by CNN.

Figure 4.3 displays the images classified as "Fishing" by the CNN.

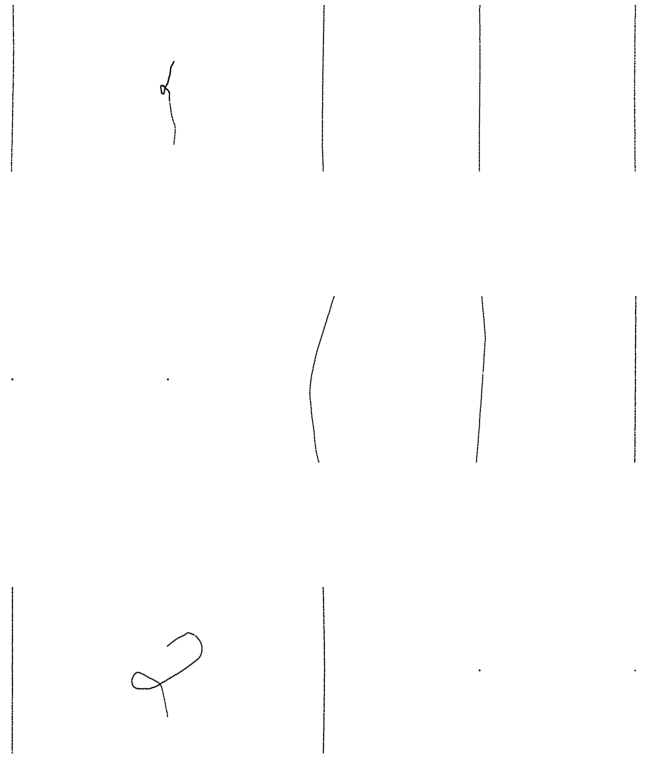


Figure 4.4: Images of Non-Fishing Activity Classified by CNN.

Figure 4.4 displays the images classified as "Not Fishing" by the CNN. The "Not Fishing" class entails all other activities other than fishing, hence in theory, "Steaming," "Stationary," and "Searching".

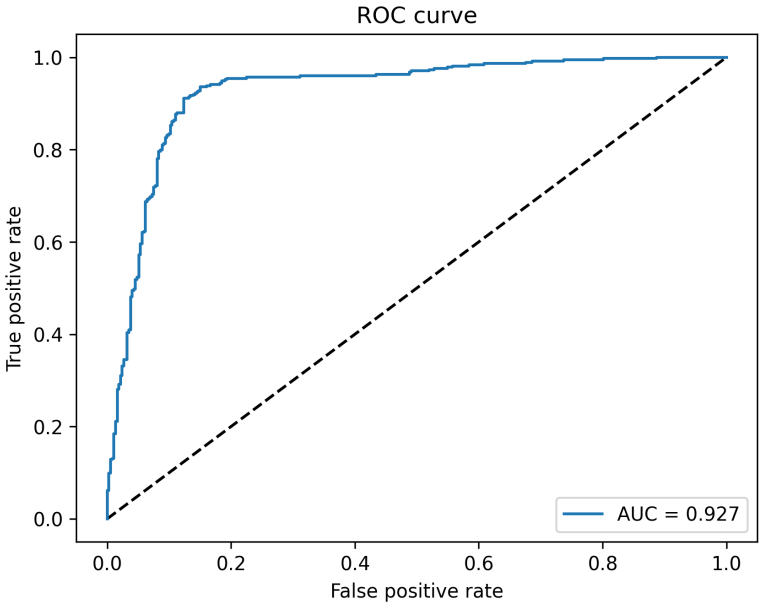


Figure 4.5: Plot of AUC-ROC for Final CNN Architecture.

As seen in figure 4.5 the AUC score was calculated to be 0.927. An AUC score of that size signifies that when presented with a random positive instance and a random negative instance, the model is likely to correctly identify the positive instance over the negative instance, with an approximate probability of 92.7%.

4.1.2 Clustering Results

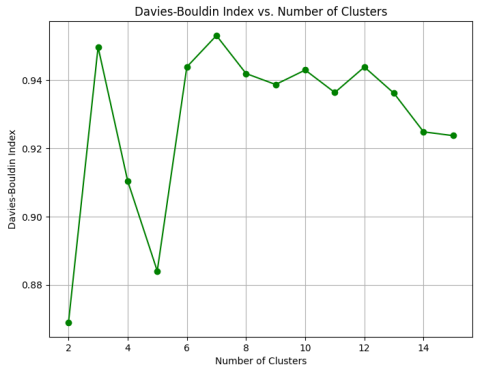


Figure 4.6: DBI Score vs. Number of Clusters for Semi-Supervised Approach

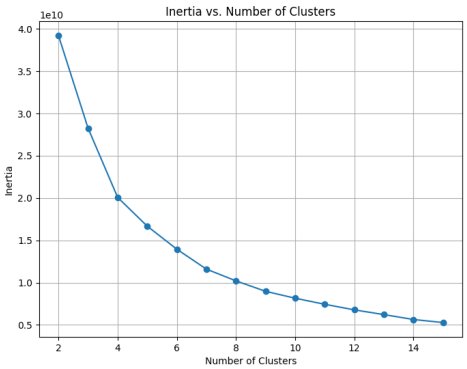


Figure 4.7: Inertia Score vs. Number of Clusters for Semi-Supervised Approach

Figure 4.8: Silhouette Score vs. Number of Clusters for Semi-Supervised Approach

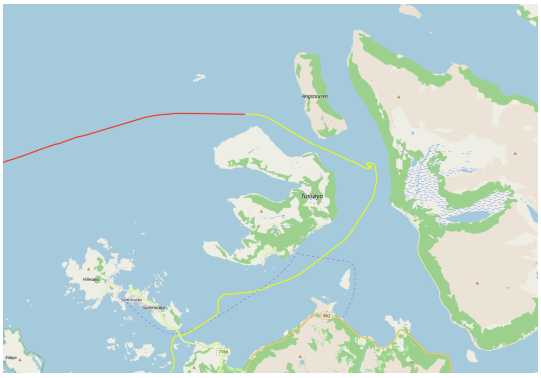
4.1.3 Section-Labeled Trajectories

The figures displayed in the following subsections were created based on the semi-supervised approach, including the classification of the CNN and the k-means clustering algorithm. Yellow trajectories depict times when the vessel is searching for fish, while red and green trajectories depict times when the vessel is steaming or in transit. The purple trajectory depicts times when the vessel is engaged in fishing activity and is trajectories exclusively classified by the CNN. Hence, the remaining colors were classified by the k-means clustering algorithm.

Section-Labeled Trajectory 1

Figure 4.9: CNN and k-means predictions. Yellow trajectories depict times when the vessel is searching for fish, while red and green trajectories depict times when the vessel is steaming. Purple trajectories depict times when the vessel is fishing.

(a) CNN and k-means predictions.

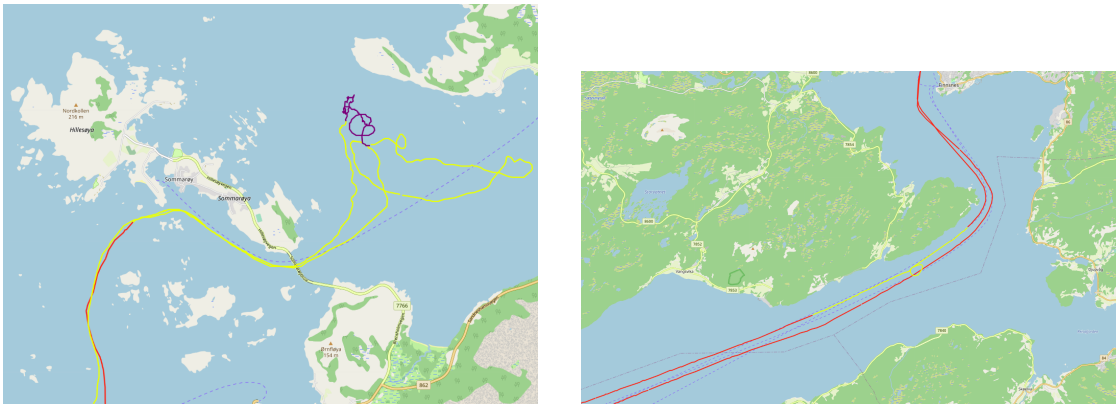


(b) CNN and k-means predictions

Figure 4.10: CNN and k-means predictions for different zoom levels. Yellow trajectories depict times when the vessel is searching for fish, while red and green trajectories depict times when the vessel is steaming. Purple trajectories depict times when the vessel is fishing.

Section-Labeled Trajectory 2

Figure 4.11: CNN and k-means predictions. Yellow trajectories depict times when the vessel is searching for fish, while red and green trajectories depict times when the vessel is steaming. Purple trajectories depict times when the vessel is fishing.



(a) CNN and k-means predictions

(b) CNN and k-means predictions

Figure 4.12: CNN and k-means predictions for different zoom levels. Yellow trajectories depict times when the vessel is searching for fish, while red and green trajectories depict times when the vessel is steaming. Purple trajectories depict times when the vessel is fishing.

4.2 Unsupervised Approach - Trajectory Based (-999 Padding)

The following figures displayed were created solely based on a k-means clustering algorithm. The dataset used as input for the model was sectioned, transformed, normalized, and padded AIS trajectories as covered in 3.4.2. The trajectories are all padded with a value of -999.

4.2.1 Evaluation Metrics

Figures 4.13, 4.14, 4.15 depict the relationship between the number of clusters and the DBI, Inertia, and Silhouette scores for the given dataset, respectively. The x-axis represents the number of clusters, varying from 2 to 15. The y-axis represents the respective scores.

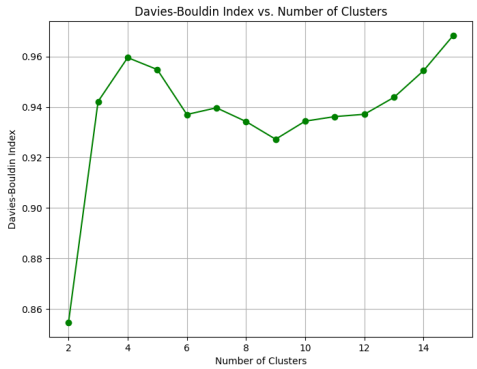


Figure 4.13: DBI Scores vs. Number of Clusters.

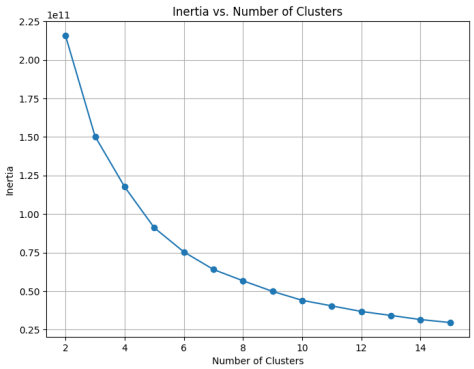


Figure 4.14: Inertia Scores vs. Number of clusters.

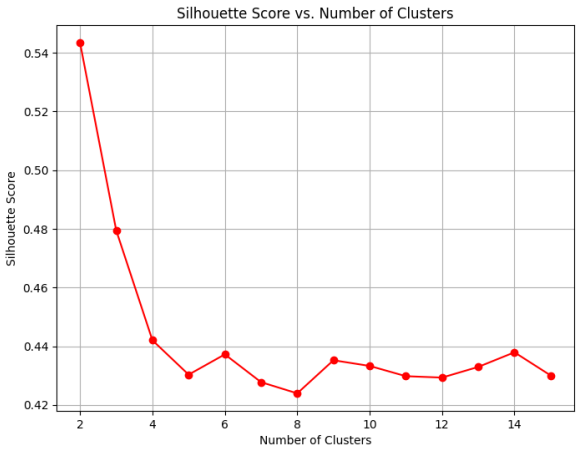


Figure 4.15: Silhouette Scores vs. Number of Clusters.

Figure 4.13 displays an upward trend as clusters increase. Beginning with a DBI score of 0.855 for a single cluster, it increases to 0.958 for fifteen clusters. Figure 4.14 exhibits a consistent decreasing trend from 2 to 15 clusters. The initial inertia at two clusters is approximately 215,815,884,061, reducing to roughly 29,627,462,627 when the number

of clusters reaches fifteen. Figure 4.15 displays a slight downward trend observed as clusters increase from 2 to 15. The highest silhouette score is recorded at 0.543 for two clusters, gradually decreasing to 0.430 as the cluster count grows to 15.

4.2.2 PCA Plot

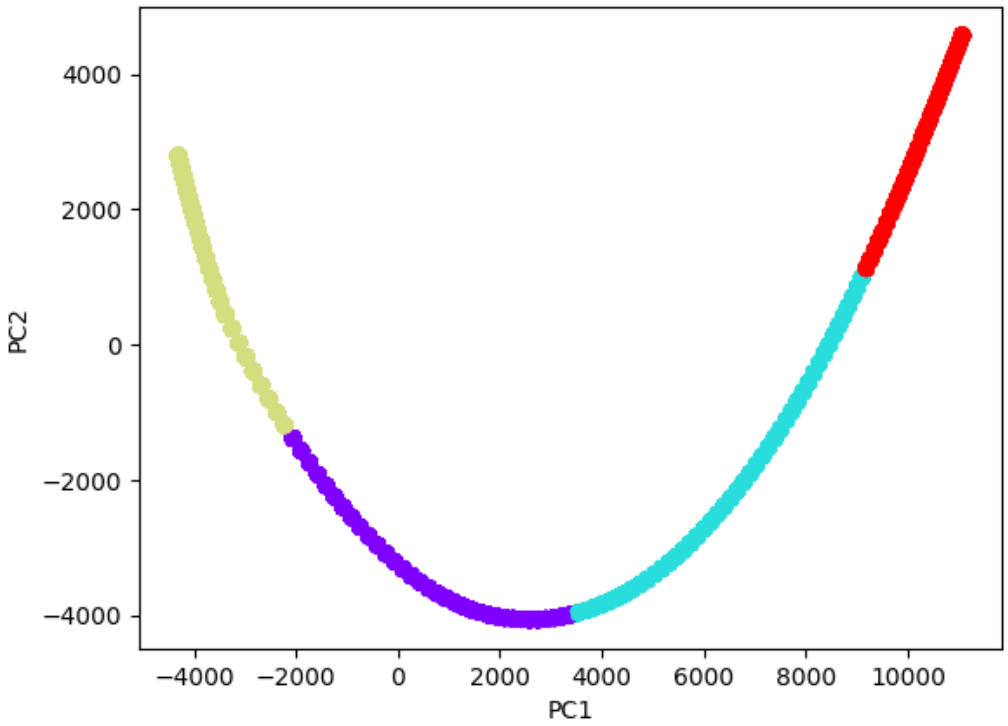


Figure 4.16: PCA plot of the trajectory-based dataset with -999 padding. The points are colored according to the classes from the k-means clustering algorithm.

4.2.3 Average Padding per Class

Each data point in the dataset contains a different amount of padding depending on the resolution of the transmitted AIS data. The average amount of padding was calculated for each class and is presented in Table 4.2.

Class	Average Padding
0	97.2
1	55.9
2	124.0
3	9.4

Table 4.2: Average Padding per Class.

4.2.4 Class Distribution

Class	Number of Samples
0	2161
1	1595
2	1161
3	6539

Table 4.3: Number of Samples for Each Class in the Dataset.

Table 4.3 provides an overview of the classification distribution. As displayed in the table, Class 3 contains the largest number of data points, while Class 2 has the fewest.

4.2.5 Section-Labeled Trajectories

The sectioned AIS data provided to the clustering algorithm included a resolution demand. The missing data was added when plotting the section-labeled trajectories. The classes present in the figures are fishing, steaming, and searching, which are colored purple, red/green, and yellow, respectively. The sections colored orange depicts sections where the data quality was unsatisfactory and was not a part of the ML dataset.

Section-Labeled Trajectory 1

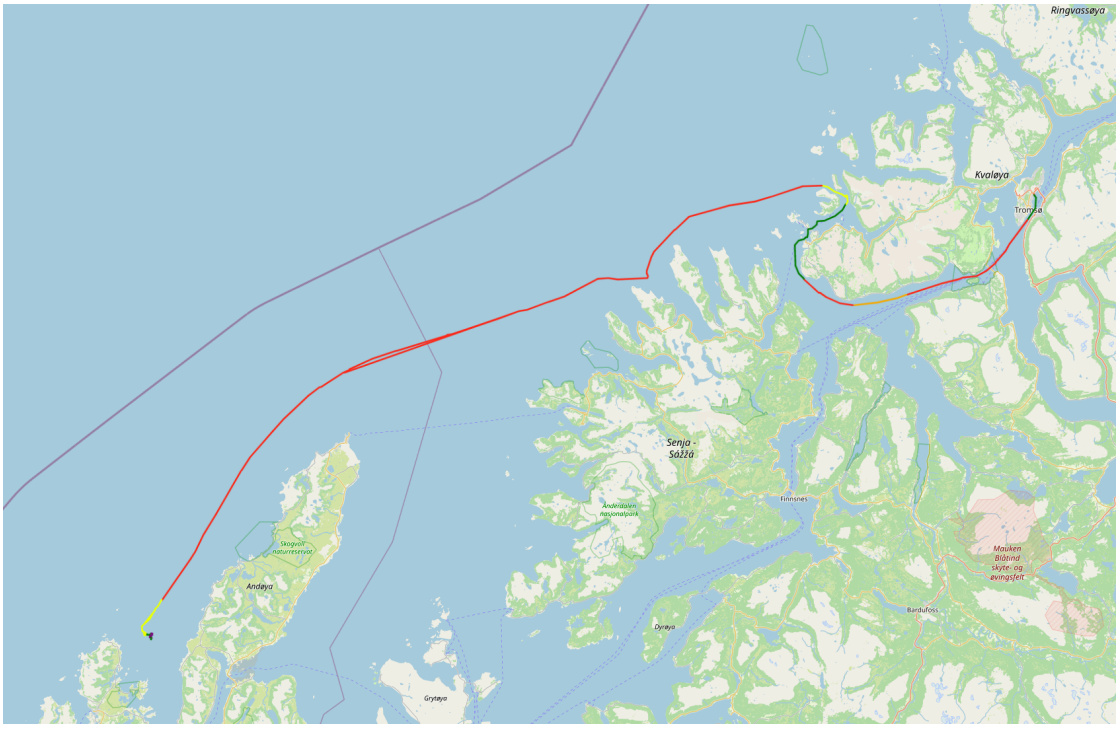
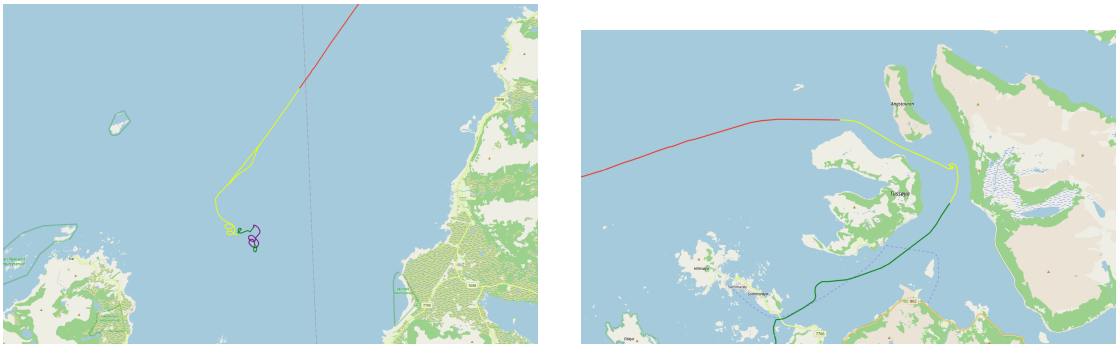


Figure 4.17: k-means predictions with -999 as padding. Yellow trajectories depict times when the vessel is searching for fish, while red/green trajectories depict times when the vessel is steaming. Orange trajectories depict sections where the data quality was below the resolution demand. Purple trajectories depict times when the vessel is fishing.

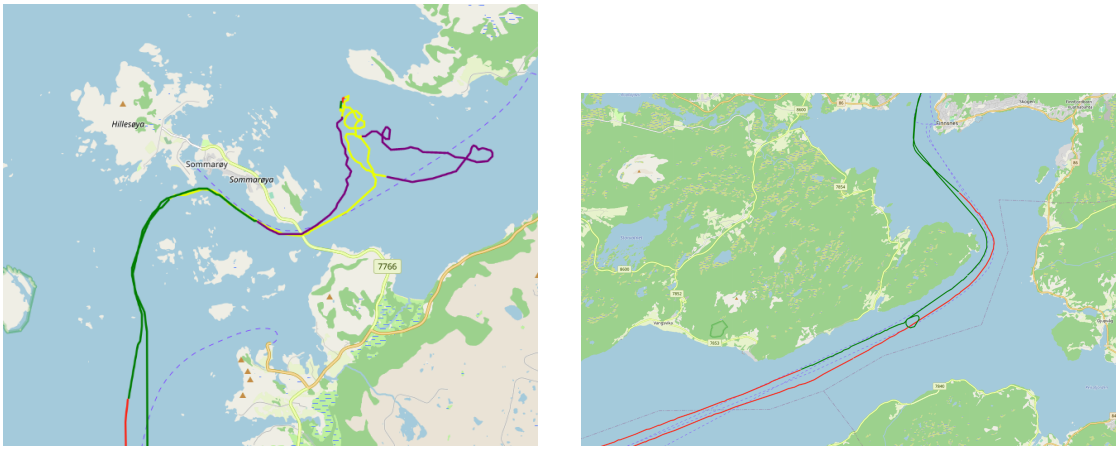


(a) k-means predictions with -999 as padding. (b) k-means predictions with -999 as padding.

Figure 4.18: k-means predictions with -999 as padding. Yellow trajectories depict times when the vessel is searching for fish, while red/green trajectories depict times when the vessel is steaming. Orange trajectories depict sections where the data quality was below the resolution demand. Purple trajectories depict times when the vessel is fishing.

Section-Labeled Trajectory 2

Figure 4.19: k-means predictions with -999 as padding. Yellow trajectories depict times when the vessel is searching for fish, while red/green trajectories depict times when the vessel is steaming. Orange trajectories depict sections where the data quality was below the resolution demand. Purple trajectories depict times when the vessel is fishing.



(a) k-means predictions with -999 as padding.

(b) k-means predictions with -999 as padding.

Figure 4.20: k-means predictions with -999 as padding. Yellow trajectories depict times when the vessel is searching for fish, while red/green trajectories depict times when the vessel is steaming. Orange trajectories depict sections where the data quality was below the resolution demand. Purple trajectories depict times when the vessel is fishing.

4.3 Unsupervised Approach - Trajectory Based (0 Padding)

The following figures displayed were created solely based on a k -means clustering algorithm. The dataset used as input for the model was sectioned, transformed, normalized, and padded AIS trajectories as covered in 3.4.3. The trajectories are all padded with a value of 0.

4.3.1 Evaluation Metrics

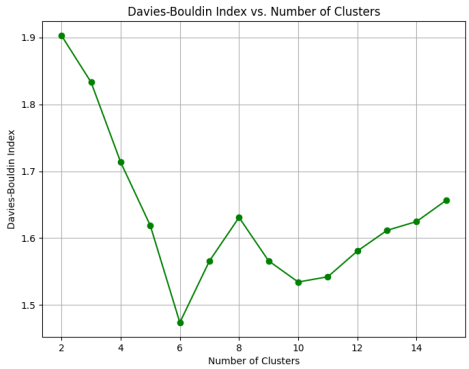


Figure 4.21: DBI Scores vs. Number of Clusters.

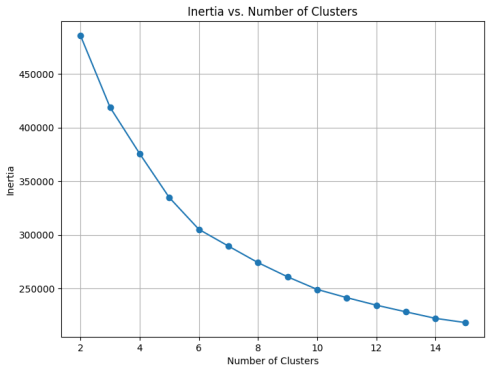


Figure 4.22: Inertia Scores vs. Number of clusters.

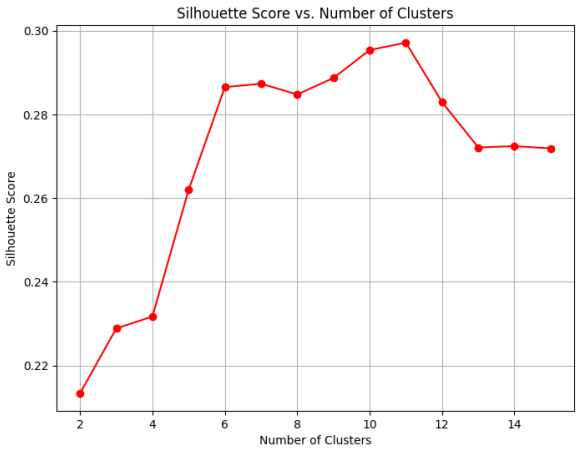


Figure 4.23: Silhouette Scores vs. Number of clusters.

Figure 4.21 exhibit a general downward trend with some fluctuations. The DBI score starts from a relatively high value of 1.903 for two clusters and decreases to 1.657 when the cluster count is fifteen. Figure 4.22 shows a clear and consistent downward trend observed from 2 to 15 clusters. The initial inertia at two clusters is approximately 485,776, which reduces to roughly 218,301 when the number of clusters reaches fifteen. Figure 4.23 displays an upward trend as clusters increase from 2 to 15. The Silhouette score, initially at 0.213 for two clusters, increases to 0.272 for fifteen clusters.

4.3.2 PCA plot

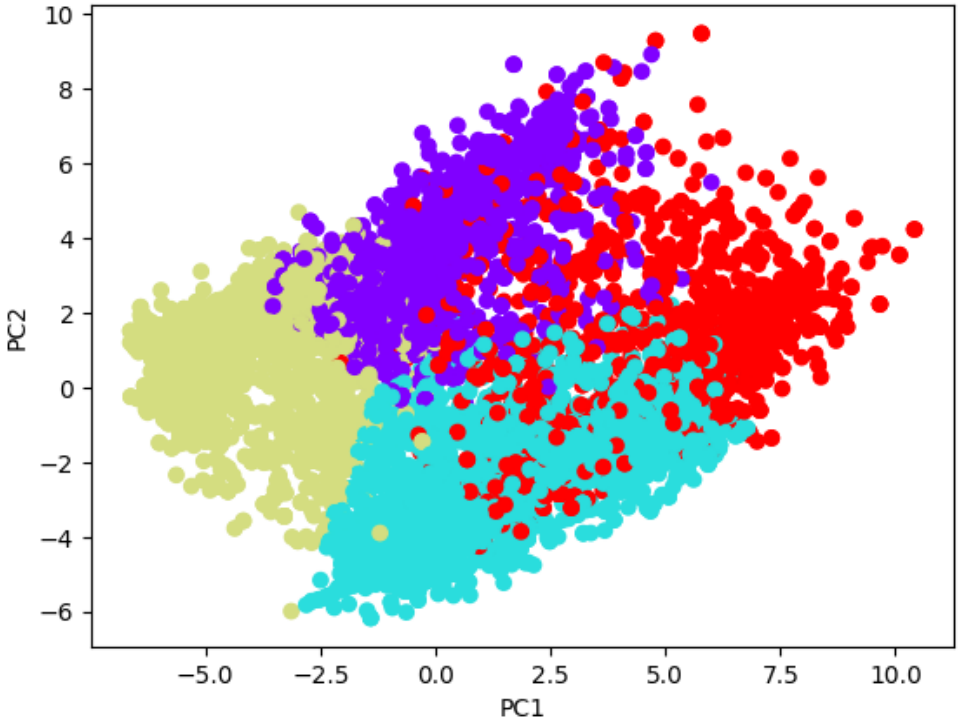


Figure 4.24: PCA plot of the trajectory-based dataset with 0 as padding value. The points are colored according to the classes from the k-means clustering algorithm.

4.3.3 Average Padding per Class

Each data point in the dataset contains a different amount of padding depending on the resolution of the transmitted AIS data. The average amount of padding was calculated for each class and is presented in Table 4.4.

Class	Average Padding
0	104.7
1	157.4
2	113.1
3	46.42

Table 4.4: Average Padding per Class.

4.3.4 Class Distribution

Table 4.5 provides an overview of the classification distribution. As displayed in the Class 1 contains the largest number of data points, while Class 3 has the fewest.

Class	Number of Samples
0	1964
1	4293
2	3438
3	1725

Table 4.5: Number of Samples for Each Class in the Dataset.

Section-Labeled Trajectory 1

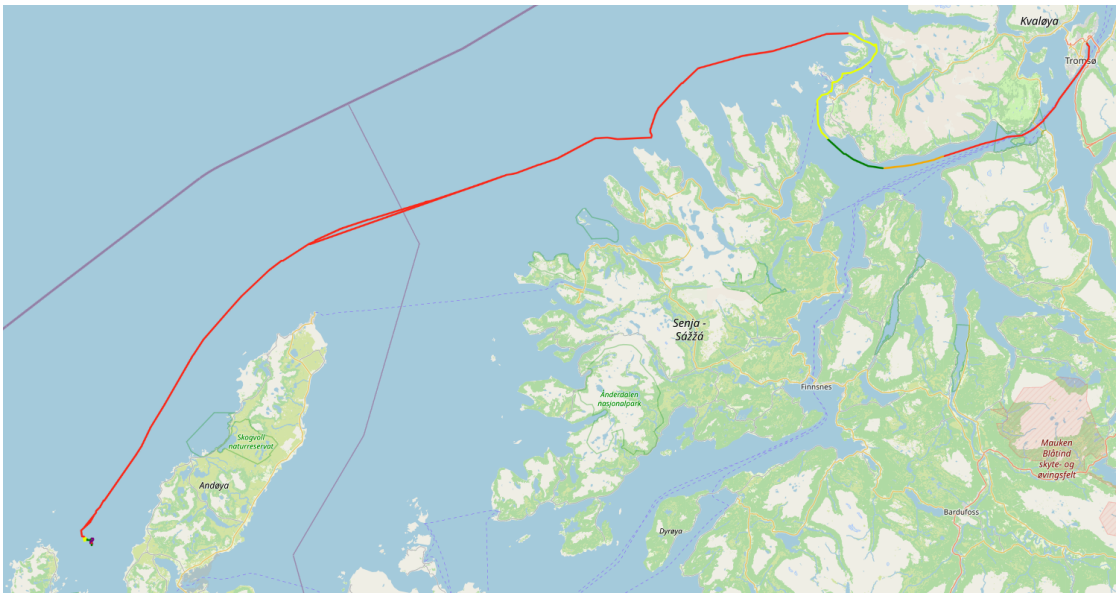
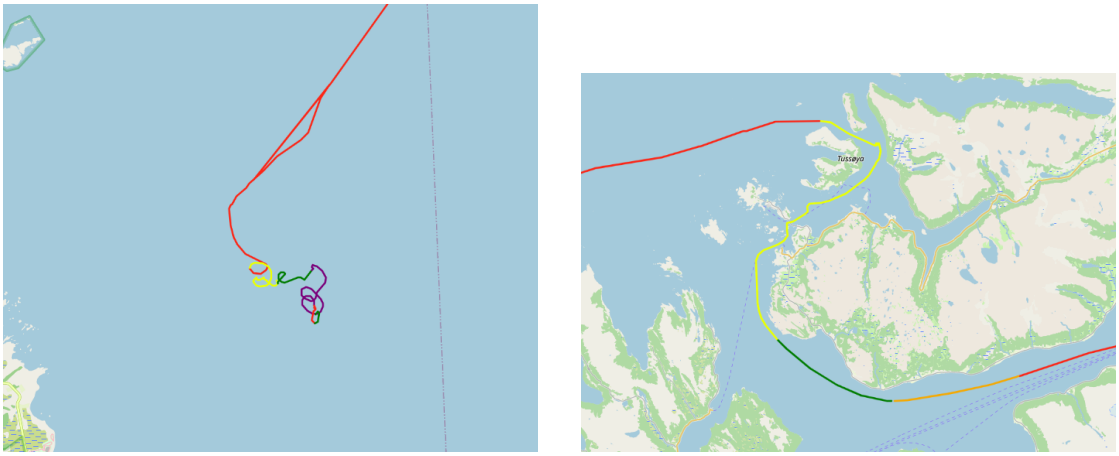


Figure 4.25: k-means predictions with 0 as padding value. Yellow trajectories depict times when the vessel is searching for fish, while red/green trajectories depict times when the vessel is steaming. Orange trajectories depict sections where the data quality was below the resolution demand. Purple trajectories depict times when the vessel is fishing.



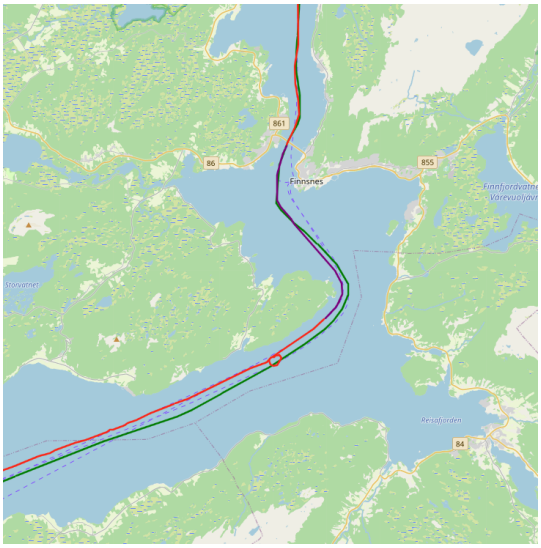
(a) k-means predictions with 0 as padding value.

(b) k-means predictions with 0 as padding value.

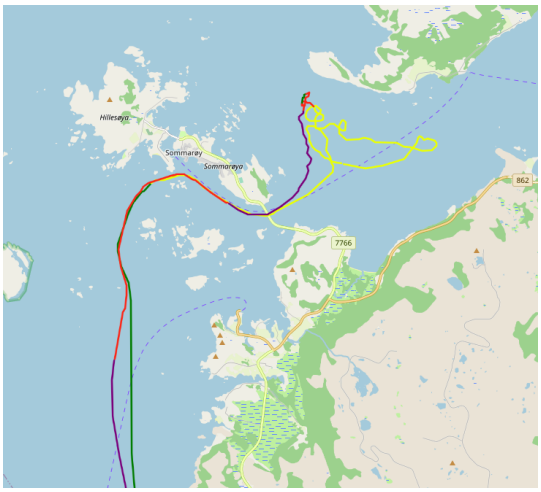
Figure 4.26: k-means predictions with 0 as padding value for different zoom levels. Yellow trajectories depict times when the vessel is searching for fish, while red/green trajectories depict times when the vessel is steaming. Orange trajectories depict sections where the data quality was below the resolution demand. Purple trajectories depict times when the vessel is fishing.

Section-Labeled Trajectory 2

Figure 4.27: k-means predictions with 0 as padding value. Yellow trajectories depict times when the vessel is searching for fish, while red/green trajectories depict times when the vessel is steaming. Orange trajectories depict sections where the data quality was below the resolution demand. Purple trajectories depict times when the vessel is fishing.



(a) k-means predictions with 0 as padding value.



(b) k-means predictions with 0 as padding value.

Figure 4.28: k-means predictions with 0 as padding value for different zoom levels. Yellow trajectories depict times when the vessel is searching for fish, while red/green trajectories depict times when the vessel is steaming. Orange trajectories depict sections where the data quality was below the resolution demand. Purple trajectories depict times when the vessel is fishing

4.4 Unsupervised Approach (Feature Based)

4.4.1 Evaluation Metrics

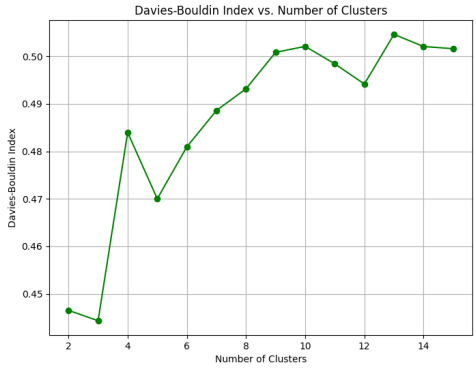


Figure 4.29: DBI Scores vs. Number of clusters.

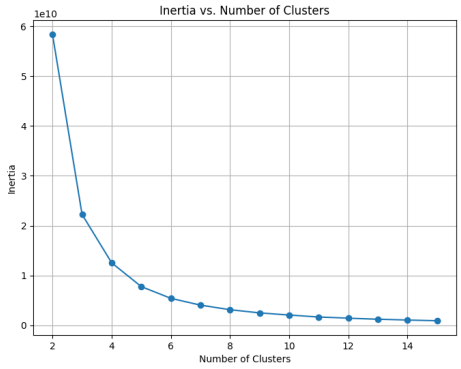


Figure 4.30: Inertia Scores vs. Number of clusters.

Figure 4.31: Silhouette Scores vs. Number of clusters.

Figure 4.29 demonstrate a general upward trend. Starting from a low DBI score of 0.446 for two clusters, it ends at about 0.502 for thirteen clusters. Figure 4.30 exhibits a consistent decrease in the Inertia score as the number of clusters increases. The score reduces dramatically from approximately 58,365,863,208 for two clusters to around 1,071,879,267 for thirteen clusters. Figure 4.31 shows a slight drop in the Silhouette score as the number of clusters increases, starting from a high of about 0.816 for two clusters and decreasing to around 0.661 for fifteen clusters.

4.4.2 PCA plot

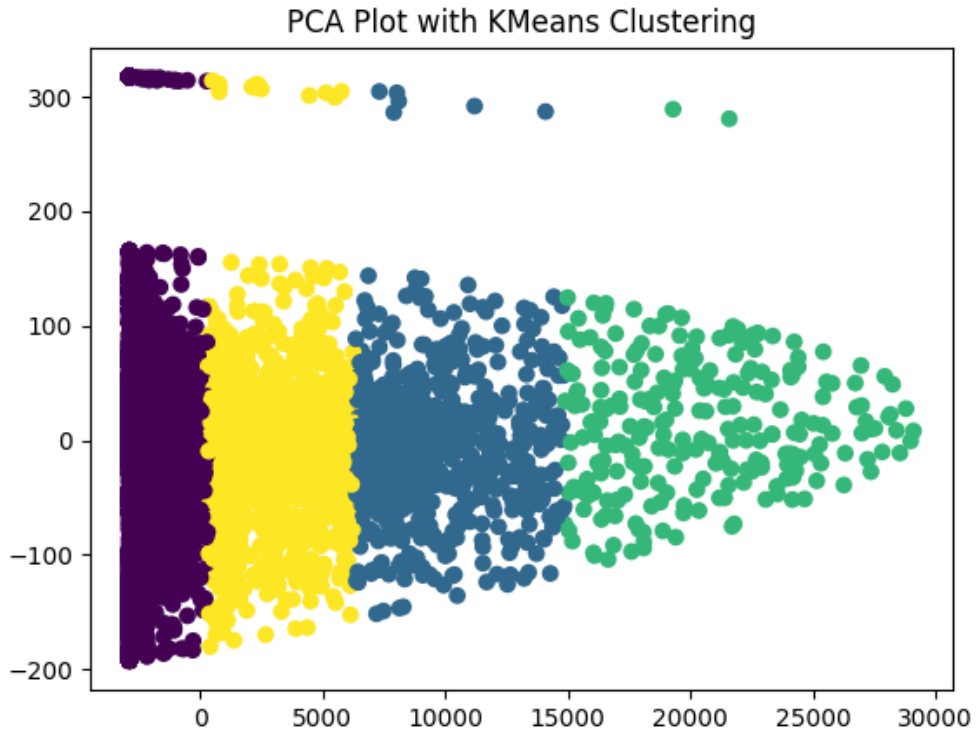


Figure 4.32: PCA plot of the feature-based dataset. The points are colored according to the classes from the k-means clustering algorithm.

4.4.3 Class Distribution

Class	Number of Samples
0	5684
1	709
2	248
3	996

Table 4.6: Number of Samples for Each Class in the Dataset

Table 4.6 provides an overview of the classification distribution. As displayed in the Class 0 contains the largest number of data points, while Class 2 has the fewest.

Section-Labeled Trajectory 1

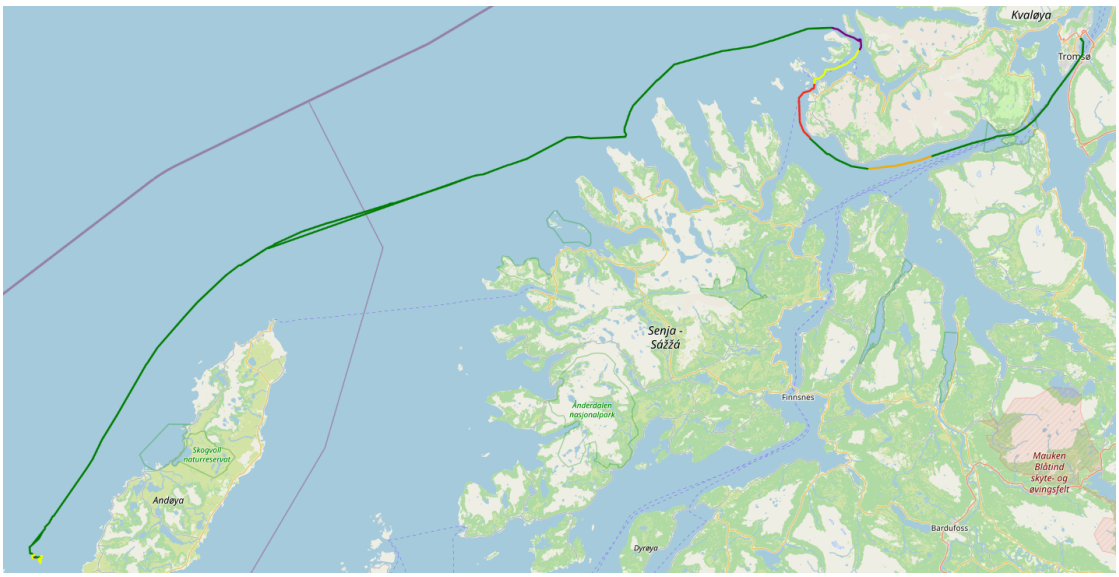
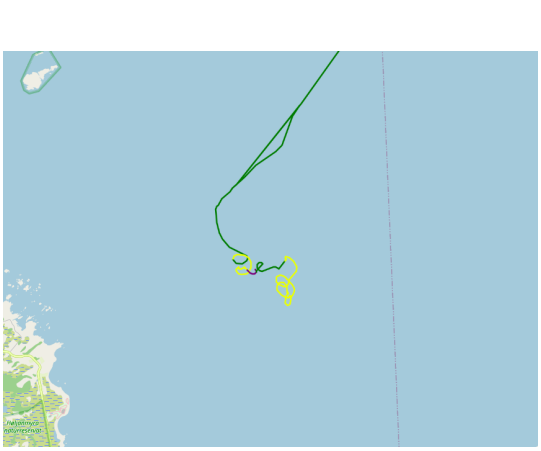


Figure 4.33: k-means predictions for feature-based dataset. Yellow/purple trajectories depict times when the vessel is searching/fishing; this is unclear. Red/green trajectories depict times when the vessel is steaming. Orange trajectories depict times when the data quality was below the resolution demand.



(a) k-means predictions for feature-based dataset.

(b) k-means predictions for feature-based dataset.

Figure 4.34: k-means predictions for feature-based dataset for different zoom levels. k-means predictions for feature-based dataset. Yellow/purple trajectories depict times when the vessel is searching/fishing; this is unclear. Red/green trajectories depict times when the vessel is steaming. Orange trajectories depict times when the data quality was below the resolution demand.

Section-Labeled Trajectory 2

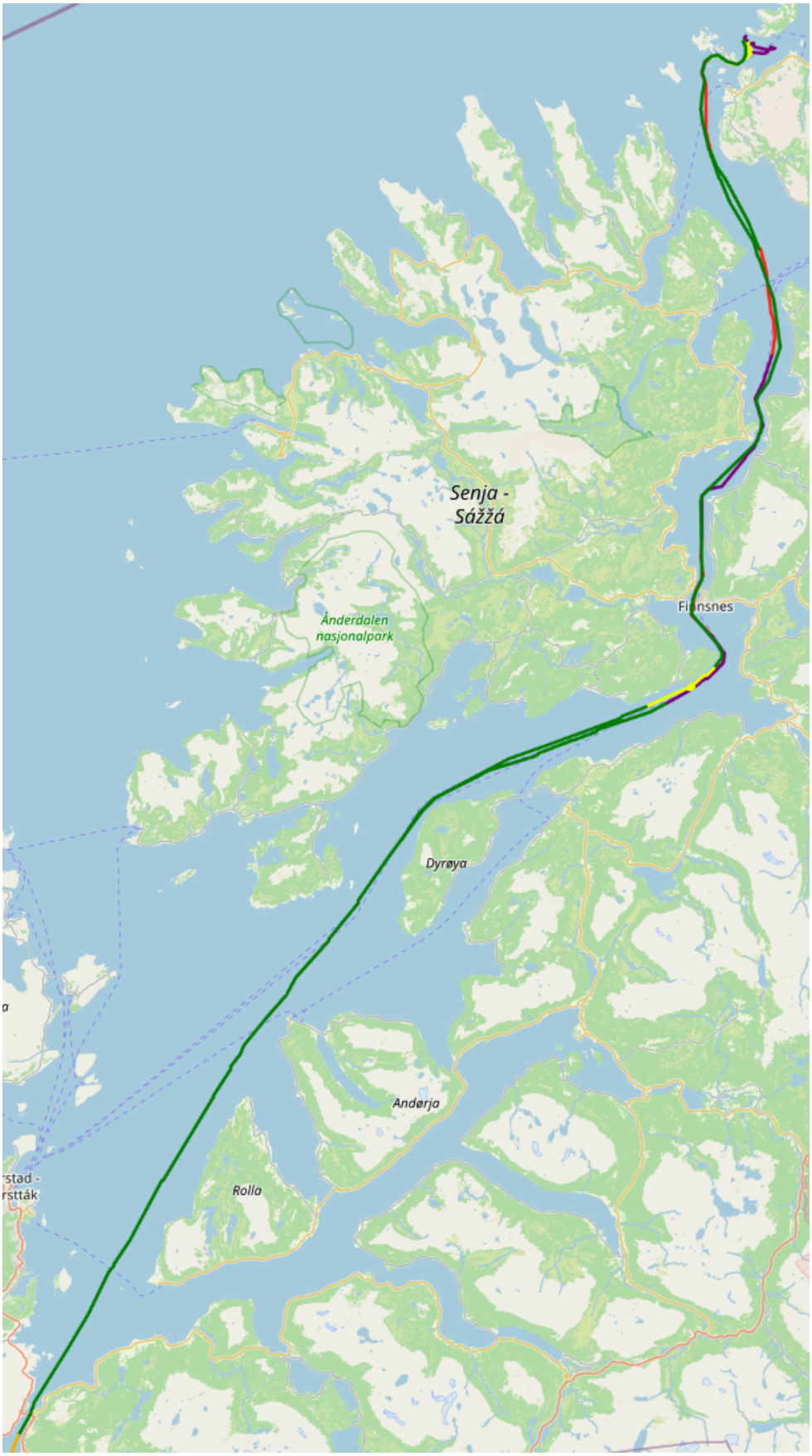
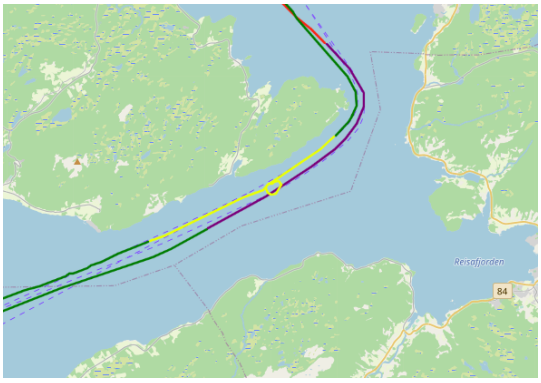
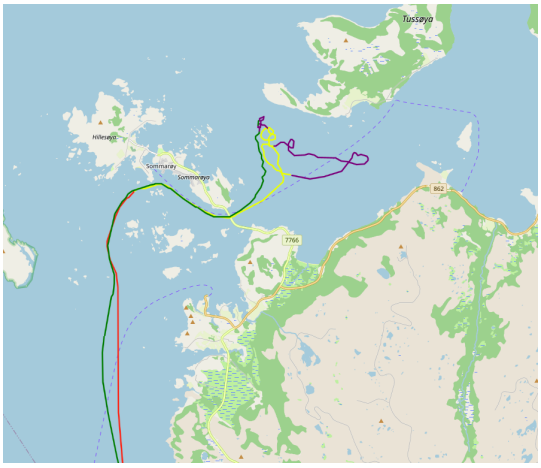


Figure 4.35: k-means predictions for feature-based dataset. Yellow/purple trajectories depict times when the vessel is searching/fishing; this is unclear. Red/green trajectories depict times when the vessel is steaming. Orange trajectories depict times when the data quality was below the resolution demand.



(a) k-means predictions with feature-based dataset.



(b) k-means predictions with feature-based dataset.

Figure 4.36: k-means predictions for feature-based dataset for different zoom levels. k-means predictions for feature-based dataset. Yellow/purple trajectories depict times when the vessel is searching/fishing; this is unclear. Red/green trajectories depict times when the vessel is steaming. Orange trajectories depict times when the data quality was below the resolution demand.

4.5 ERS Catch Labels

The plots featured in this section are labeled according to the ERS catch report. Therefore, whenever the vessel trajectories appear as 'Fishing', the vessel was engaged in fishing activities. The section-labeled trajectories depicted in this section serve as a validation tool to compare with the plots presented in 4.1 and 4.2.

4.5.1 Section-Labeled Trajectories

The following figures illustrate a single day's trajectory of a fishing vessel. The green trajectory signifies a period where the vessel is not engaged in any fishing activity. In contrast, a purple trajectory denotes times the vessel is actively engaged in fishing the NSSH. The particular trajectories depicted were selected due to their interesting path throughout the day. Evidently, the vessel did not take a direct route to its final fishing grounds; instead, it appears to have deviated from a direct path, presumably in search of fish. Following this diversion, the vessel reaches a location where it successfully locates fish and initiates fishing activities.

Section-Labeled Trajectory 1

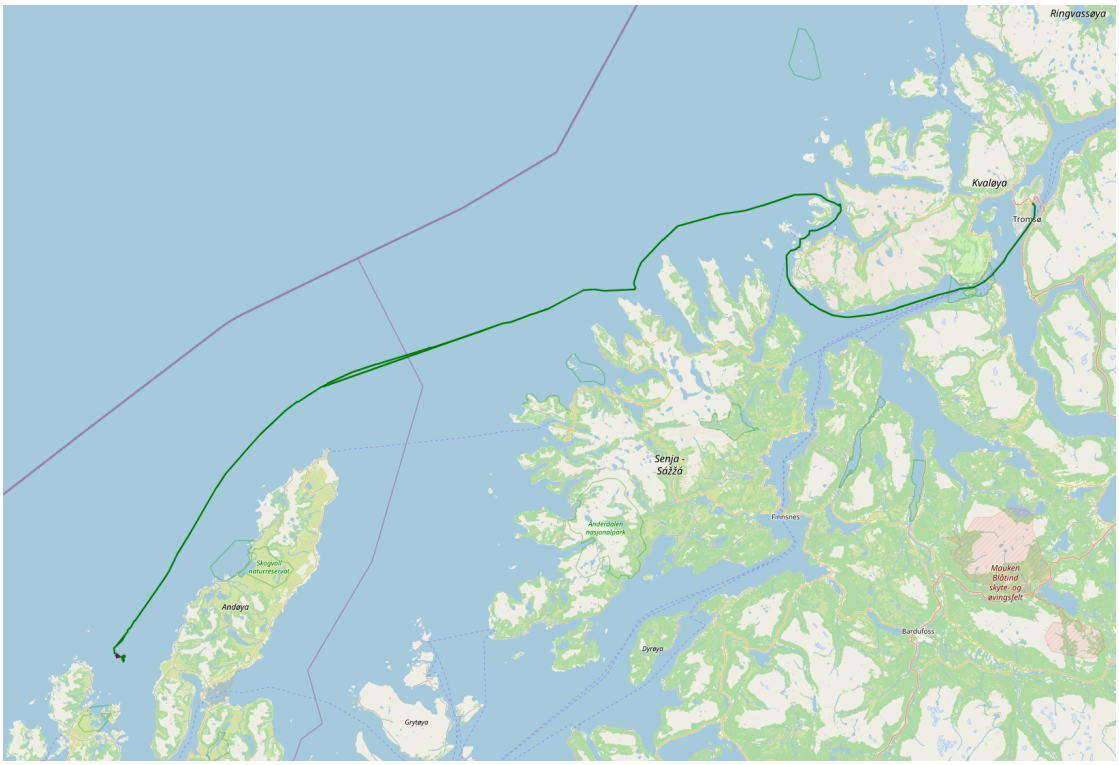
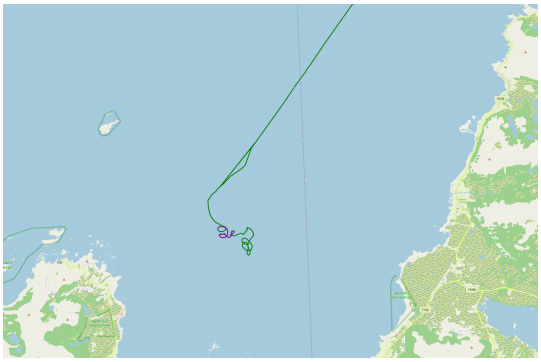
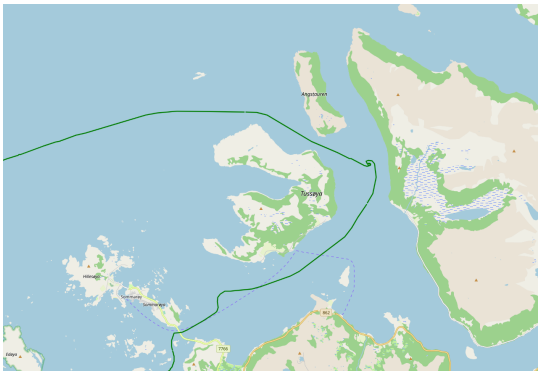


Figure 4.37: Fish labels derived from ERS catch data. Green trajectories depict times when the vessel is not fishing. Purple trajectories depict times when the vessel is fishing.



(a) Fish Labels Derived from ERS Catch Data.



(b) Fish Labels Derived from ERS Catch Data.

Figure 4.38: Fish labels derived from ERS catch data. Green trajectories depict times when the vessel is not fishing. Purple trajectories depict times when the vessel is fishing.

Section-Labeled Trajectory 2

Figure 4.39: Fish labels derived from ERS catch data. Green trajectories depict times when the vessel is not fishing. Purple trajectories depict times when the vessel is fishing.

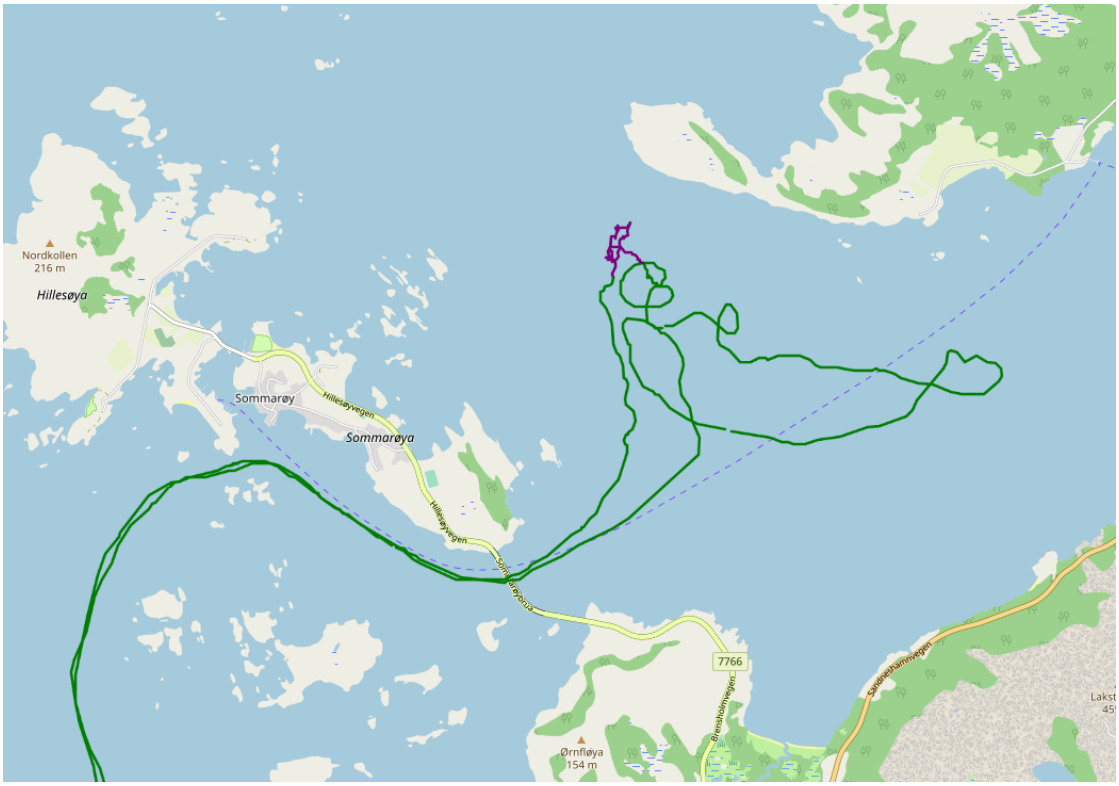


Figure 4.40: Fish labels derived from ERS catch data. Green trajectories depict times when the vessel is not fishing. Purple trajectories depict times when the vessel is fishing.

5 Discussions

5.1 Semi-Supervised Approach

5.1.1 CNN

The performance of the CNN was evaluated using various architectures, configurations, and datasets. Initially, tests were conducted using non-rotated images and inaccurate labelling of fishing events. It was assumed that all entries in the ERS catch report corresponded to fishing, but further exploration revealed this assumption to be incorrect. The initial test results showed a performance accuracy of 79 percent.

Upon analyzing the clustering results, it was discovered that the direction of the plotted trajectories significantly affected the clusters. This insight was obtained through visual inspection of the classified images. Subsequently, the CNN dataset was refined by addressing these two uncertainties. As a result, the performance of the CNN improved by approximately 10 percent after fixing these mistakes.

The training time of the CNN was greatly influenced by factors such as the image array size (image resolution), the number of layers in the CNN, the kernel size, the stride, and the number of epochs. The image resolution was incrementally reduced through visual inspection of the images until it reached a point where further reduction would compromise the information content. Experimentation was conducted to determine the impact of the number of layers, strides, and kernels on model performance. Through an iterative process, it was observed that reducing the stride size to (3,3) and utilizing kernel sizes of (3,3) and (6,6) produced satisfactory outcomes. Attempting smaller sizes led to a substantial increase in training time without yielding significant performance improvements. Hence, the chosen configurations balanced computational efficiency and desirable results.

A bias was discovered in the dataset, as there was a disproportionate number of white pixels compared to the black pixels representing the trajectory in the images. Typically, an image's percentage of white pixels was around 99 percent. To obtain meaningful results, pixel normalization was mandatory. Although normalization yielded meaningful results, it is essential to note that the bias persists and may impact the outcome. To mitigate the bias, one potential approach is to increase the trajectories' thickness while still ensuring that the trajectory information remains intact.

Initially, the model displayed apparent symptoms of overfitting. Experiments with the dropout layer, number of epochs, and learning rate were conducted to solve the problem.

The final dataset and CNN architecture achieved an acceptable performance with a final accuracy of 0.874 and an AUC ROC score of 0.93. These metrics indicate that the model has demonstrated a good ability to classify the data accurately. The CNN was capable of accurately classifying a high proportion of correct positive classes while maintaining a low number of false positives.

5.1.2 Trajectory Based Clustering for Semi-Supervised Approach

The initial clustering tests were conducted using the non-fishing image arrays obtained from the CNN. However, due to the exceedingly long training time required by the k-means algorithm, this approach was deemed impractical to pursue further. As a result, an alternative approach was adopted, wherein the non-fishing events from the CNN were clustered using the segmented AIS data instead of the image arrays. This modification significantly improved training times, with the entire dataset now being clustered in under one minute compared to the previously encountered training times of multiple days.

The trajectory-based clustering results of the non-fishing dataset display performance similar to that of the trajectory-based unsupervised approach. The Silhouette, DBI, and Inertia score for 3 clusters were 0.462, 0.911, and 1.761×10^{10}, respectively. The Silhouette score indicates that the data points within the clusters are relatively closer to each other than to other clusters. Ideally, this value should have been closer to 1. The DBI score is relatively high; this value can indicate that the clusters in the model need to be better separated. Finally, while Inertia should ideally be as low as possible, this value strongly depends on the scale of the dataset and the number of data points, making its analysis difficult. Nevertheless, the Inertia plot depicted in Figure 4.7 reveals convergence after four clusters, suggesting this as an appropriate cluster count. However, given that the aim is to classify three activities from the CNNs negative class, an ideal cluster number would be three.

5.2 Overall Performance

As seen visually in the section-labeled trajectories in Chapter 4.1.3 and 4.1.3, the semi-supervised model demonstrates good performance in classifying the different activities. Comparing Figure 4.10a and 4.38a, it is clear that the model accurately predicts the time when the vessel is, in fact, fishing, but over-classifies a bit by classifying a large portion of the trajectory as fishing than what it was. This accuracy is also seen when comparing Figure 4.12a and 4.40. Furthermore, the model classifies the period where the vessel is on a detour and likely searching for fish as "Searching" but also overclassifies a bit here as well, including more of the trajectory than needed, as seen in Figure 4.38b. Overall the model displays good performance.

The semi-supervised model demonstrates good performance in classifying different vessel activities, as displayed from the section-labeled trajectories visualized in Chapter 4.1.3 and 4.1.3. Comparing Figure 4.10a and 4.38a proves the model's ability to identify "Fishing" periods accurately. However, the model overclassifies, labeling a more significant part of the trajectory as fishing than it should. This pattern of overclassification also is also displayed in the model's classification of "Searching" periods, where it identifies more trajectory as searching than needed, as seen in Figure 4.38b. Despite these tendencies, the model's overall performance is prominent.

5.2.1 Strengths and Weaknesses

The primary strength of the approach lies in its ability to accurately classify fishing events, thereby creating a more robust model and easing the clustering process. Another strength is the model's ability to handle poor-quality AIS data. Given that AIS data

can be noisy, incomplete, and unreliable, plotting the trajectories mitigates some of these problems, notably eliminating the need for padding. However, the semi-supervised approach also has its weaknesses. The dependence on the quality of ERS data for training is a significant limitation. During the model training phase, it was discovered that not all entries in the ERS catch report corresponded to actual fishing events, which required an adjustment in the approach. Inaccurate reporting by vessels can introduce significant bias to the model. Furthermore, the presence of bias in the dataset, shown by a disproportionate number of white pixels compared to black pixels, presents a challenge. While pixel normalization helped yield meaningful results, the persistent bias could still impact the outcomes, indicating an area for improvement in data preprocessing.

For the clustering phase, the high DBI score indicates that the clusters are not well separated. This suggests that the model might struggle to distinguish different non-fishing events effectively. As later revealed in the Unsupervised approach, the clustering method used in this semi-supervised approach is mainly dependent on the padding value of the coordinate arrays. Unfortunately, this dependency diminishes the model's robustness, marking an area that requires further improvement.

5.2.2 Conclusion

The semi-supervised approach demonstrates promise in addressing the complex task of classifying vessel activities. Despite certain challenges, it offers a plausible solution by tackling the poor quality of AIS data, achieving high classification accuracy. However, the quality of ERS data, the separation of clusters, the padding reliance, and the persistent bias in the dataset represent important areas for improvement.

5.3 Unsupervised Approach - Trajectory Based Clustering

5.3.1 -999 Padding

The k-means algorithm with trajectory-based clustering exhibited mixed results with -999 padding.

On the positive side, the section-labeled trajectories demonstrated the model's capacity to classify the "Fishing", "Searching", and "Steaming" activities. As the depicted trajectories lacked stationary points, it posed challenges in assessing the model's performance in detecting the "Stationary" class. The model appeared to accurately classify "Fishing" periods close to the actual fishing times as depicted in Figures 4.17 and 4.18. Despite slight deviations, this level of precision might be sufficient for its intended application, namely, providing feedback to the FishGuider model.

The course traced by the vessel on the way to the fishing destination in Figure 4.17 suggested a phase of searching for fish, supporting the model's accurate classification. Also, the "Searching" period identified before the fishing activity makes sense, as vessels are likely searching for fish before fishing. On the whole, the model seemed proficient in classifying "Steaming" periods as seen in Figures 4.19 and 4.20.

Conversely, the performance metrics and the PCA plot narrate a different story—the clustering quality peaks at four clusters according to both the metrics and the segmented trajectories. The four clusters' Silhouette, DBI, and Inertia scores were 0.43, 0.95, and 7.5×10^{12}, respectively. The Silhouette score of 0.43 does not mark exceptional cluster

compactness or separation. A value closer to 1 would be preferred. The DBI score of 0.95 is relatively high, indicating potential cluster proximity or size issues. Inertia, dependent on the variables and the total data, is difficult to interpret but is generally preferred to be low.

The PCA plot in Figure 4.16 reveals a tight clustering along a convex curve. This distinctive formation suggests an underlying structure in the multidimensional data. The classes, indicated by colours, change along the x-axis. This might be caused by the linear transformation technique of PCA, aiming to maximize data variance and the use of the -999 padding value.

Interestingly, the average quantity of padding per segmented AIS data, as shown in Table 4.2, might be the primary trigger for the k-means algorithm. Despite the convincing performance on the segmented trajectories, this padding effect may suggest a correlation between the resolution of AIS data and the activity or speed of the vessel. However, the padding amount shouldn't be the primary trigger for classification. The distribution of samples across classes appears to be well balanced, considering the real-world likelihood of a vessel spending more time in one activity, such as steaming, than others. As shown in Table 4.3, one class significantly outnumbers the others with 5234 samples. Meanwhile, the remaining classes exhibit a relatively balanced distribution with around 1500 samples each.

5.3.2 0 Padding

The k-means algorithm with trajectory-based clustering demonstrated a less favourable performance with 0 padding than with -999 padding. The results using 0 padding revealed similar and dissimilar characteristics compared to those of -999 padding.

Initially, the section-labeled trajectories displayed in Chapter 4.3.4 and Chapter 4.3.4 portray a diverse performance. The trajectory in Chapter 4.3.4 exhibits comparable performance to the -999 padding, with the model classifying the segmented trajectories similarly. However, there is an issue with the classification of inshore steaming as searching, which is sub-optimal. Conversely, the section-labeled trajectories depicted in chapter 4.3.4 demonstrate the model's inadequate performance, as it appears to misclassify evident steaming events as fishing events, as indicated by the straight purple trajectory segments.

The PCA plot for the model, illustrated in Figure 4.24, shows the data points spread out in a structured yet overlapping manner. This distribution may reflect the effects of using 0 padding. This visual overlap aligns with the unfavourable metrics displayed in Figures 4.21, 4.22, and 4.23. The Silhouette, DBI, and Inertia scores for four clusters stand at 0.26, 1.71, and 334816.9, respectively. These metrics suggest potential issues with cluster separability, compactness, and intra-cluster distance, indicating a less optimal clustering solution.

Similar to the results with -999 padding, the average padding per class seems to be ordered by class as depicted in Table 4.4. This pattern could imply that padding continues to play a vital role in triggering the classification process, which is not the ideal circumstance for a robust model. The distribution of classes appears to be more evenly distributed, with two classes containing roughly double the number of samples compared to the other two classes.

5.3.3 Strengths and Weaknesses

The k-means algorithm with trajectory-based clustering presents several strengths and weaknesses when applied to vessel behaviour classification with both -999 and 0 padding.

Among its strengths, the model demonstrated worthy performance in classifying certain vessel activities such as "Fishing," "Searching," and "Steaming", particularly with -999 padding. The model displayed promising results in labeling "Fishing". The labels aligned relatively well with the actual times of the activities. Moreover, the PCA plot suggests that the model can detect underlying patterns in the multidimensional data.

However, this approach also exhibits certain weaknesses. While the classification for some activities was successful, the model's performance for the "Stationary" class was uncertain due to the lack of stationary points in the trajectories. The Silhouette, DBI, and Inertia scores for both padding methods indicated sub-optimal cluster compactness, separation, and intra-cluster distance, suggesting potential issues with cluster quality. Furthermore, the model seemed to rely heavily on padding for the classification process, which is undesirable for a robust model. Misclassifications were also observed, such as labeling clear steaming events as fishing, especially with 0 padding.

5.3.4 Conclusion

In summary, the unsupervised trajectory-based clustering approach using the k-means algorithm displays promising results in classifying certain vessel activities. While it displays potential, it also reveals areas for improvement, particularly in classification robustness and reliability and reducing dependence on padding for the classification process.

5.4 Unsupervised Approach - Feature Based Clustering

Feature-based clustering offers the advantage of mitigating the padding issue by encapsulating the 2D array of coordinates into a series of relevant features.

Upon visual inspection, the section-labeled trajectories show a degree of accuracy in classifying sectioned trajectories, particularly as evident in Figure 4.33. However, this assumption is challenged by the somewhat arbitrary labels observed in Figure 4.35. The model seems to be proficient in classifying "Steaming" events, while its classification of "Fishing" and "Searching" activities appear somewhat arbitrary. Distinguishing between the latter two classes remains difficult. Similar to the trajectory-based models, the lack of stationary points in the model makes the assessment challenging. Further examination of additional figures reaffirms the classification's relative inaccuracy. However, the model seems to factor in the vessel's navigational angle and traveled distance into its classification – a logical approach given the provided features. Still, a clear separation between behaviors is lacking.

The PCA plot depicted in Figure 4.32 supports this observation. The data points appear to be classified along the x-axis, spread across the plot, with seemingly arbitrary classification boundaries. Adjusting these boundaries might yield improved results, potentially allowing for an optimal behavioral index distinction.

As for class distribution, a significant imbalance was observed in Table 4.6. Class 0 is

significantly larger, with 5684 samples, compared to the other classes, which average around 600 samples each.

5.4.1 Strengths and Weaknesses

The feature-based clustering approach demonstrates several strengths. Firstly, it resolves the padding problem inherent in the trajectory-based models by collapsing the 2D array of coordinates into a series of relevant features. This can make the model more efficient and resistant to noise or missing data. Furthermore, the model exhibits competence in classifying "Steaming" events, likely the most frequent class of activities in the dataset.

However, the model also exhibits some weaknesses. The classifications of "Fishing" and "Searching" activities appear somewhat arbitrary and less reliable than the "Steaming" classification. The inability to distinguish between these two classes may limit the model's usefulness for the purpose. Furthermore, the lack of stationary points for evaluation adds a layer of uncertainty to the model's overall performance.

The PCA plot indicates that the classification boundaries are somewhat arbitrary and might benefit from further optimization. While the model seems to account for the navigational angle and distance traveled, the unclear separation between behaviors indicates a need for better feature engineering or choice of algorithm.

Lastly, the class imbalance issue, despite its justification by the higher occurrence of "Steaming" events, might still affect the model's performance. Though justified, it can be wise to consider this imbalance while clustering and interpreting the model's performance.

5.4.2 Conclusion

In conclusion, the feature-based clustering approach displayed an unreliable ability to classify vessel activities. Its demonstrates strength and robustness to padding-related issues, which could make it a valuable tool for further development. The model's current shortcomings call for additional research and fine-tuning. This might include optimizing the classification boundaries, refining feature engineering strategies, and considering different algorithms to improve performance.

5.5 Post-Processing of Model Output

The output generated by the models discussed is raw data, which requires further processing to yield valuable data for the FishGuider model. The FishGuider model requires input in the form of positive and negative fishing events. Positive fishing events are instances where a vessel has successfully found fish, indicating the presence of fish in the area. Conversely, negative fishing events represent situations where a vessel has actively searched for fish but failed to locate any, indicating an absence of fish in the area.

In its current state, the ML model does not readily provide this specific output, making it imperative to process the model's output accordingly. A plausible approach to translating the model's output into positive and negative fishing events would be to treat all instances of the model's 'Fishing' event classification as positive fishing events. However, the identification of negative fishing events presents a more significant challenge. For

instance, when the model classifies 'searching' preceding 'fishing,' this 'searching' event should not be misinterpreted as a negative fishing event. Consequently, additional constraints may need to be imposed, such as requiring the model to demonstrate a sequence of 'steaming' to 'searching' to 'steaming' again over a specified duration to classify the occurrence as a negative fishing event.

6 Conclusion and Further Work

6.1 Final Conclusion

This study aimed to investigate whether ML models could be used to classify the behaviors of individual fishing vessels based on AIS data and ERS catch data. The classification of vessel behavior aimes to provide feedback to the FishGuider model that provides information about herring distribution.

Two main approaches were employed, refined, and evaluated in this thesis: a semi-supervised approach and an unsupervised approach.

In the case of the semi-supervised approach, the CNN demonstrated prominent performance in classifying fishing activities from non-fishing activities. This classification was achieved on images of plotted and sectioned vessel trajectories. The clustering component of this approach demonstrated mixed performance. While internal evaluation metrics suggested relatively poor performance, the visual assessment of the section-labeled trajectories indicated a more promising outcome. Despite certain challenges, this approach offers a potential solution to addressing the quality issues associated with AIS data, and it has achieved a high classification accuracy. However, the quality of ERS data, the separation of clusters, the padding reliance, and the persistent bias in the dataset represent important areas for improvement.

For the unsupervised approach, two main methods were tested utilizing the k-means algorithm: trajectory-based clustering and feature-based clustering. Each method exhibited different performance characteristics. Based on the visual assessment of the section-labeled trajectories, the trajectory-based method seemed to outperform the feature-based method. However, it revealed that the trigger in the trajectory-based model was likely the amount of padding in each array. For internal evaluation metrics, the feature-based method showed superior performance, displaying acceptable cluster compactness and separation values compared to the trajectory-based method.

In conclusion, both the semi-supervised and unsupervised approaches show potential for classifying the behavior of fishing vessels but also underscore the need for further refinement and research. The best-performing model is the semi-supervised model; it has some crucial reliance issues that must be handled for the model to be useful. The unsupervised feature-based clustering method shows the highest potential for an accurate and effective model, but the classification boundary must be adjusted to fit the different vessel activities. The methods used in this study may provide a foundation for future work, focusing on improving the model's robustness, reliability, and performance. Lastly, appropriate post-processing is required for the raw data generated by these models to be useful for the FishGuider model.

6.2 Further Work

- **Refining feature engineering strategies:** The feature choice and feature selection can significantly impact the model's performance. Hence, experimenting with new features and feature selection strategies could potentially improve the model's performance.
- **Optimize classification boundary feature based clustering:** The PCA plot depicted in Figure 4.32 suggests that the data classification occurs along a certain continuum. It appears that the classification boundaries are arbitrary. Therefore, it could be beneficial to experiment with adjustments to these boundaries, aligning them more effectively with the specific activities of the vessels.
- **End-to-End Trip Plotting:** A potentially beneficial modification could involve plotting the vessel trajectories for complete journeys (starting when a vessel leaves the port and ending upon its return) instead of daily plotting, as conducted in this thesis. This adjustment might provide a more precise and more comprehensive picture for analysis.
- **Post-processing model output:** An important area for future work lies in processing the raw data output from the ML models into distinct positive and negative fishing events. This process involves creating a new methodology for accurately identifying negative fishing events, likely requiring the implementation of specific constraints based on action sequences.

Bibliography

[1] ChatGPT. URL https://chat.openai.com.

[2] Norwegian Fisheri managment. URL https://www.regjeringen.no/globalassets/upload/fkd/brosjyrer-og-veiledninger/folder.pdf.

[3] Cian Kelly. Assimilation of real-time measurements with an individual-based model for estimation of geographical distribution and abundance of Norwegian herring. URL https://ntnuopen.ntnu.no/ntnu-xmlui/handle/11250/3066402.

[4] Petter Holm and Katrine Tveiterås. Fiskeriforvaltning i bevegelse. URL https://torskeprogrammet.no/wp-content/uploads/sites/16/2014/03/PH-Fiskeriforvaltning-i-bevegelse.pdf.

[5] Peter Tyedmers. Fisheries and Energy Use. In *Encyclopedia of Energy*, pages 683–693. Elsevier. ISBN 978-0-12-176480-7. doi: 10.1016/B0-12-176480-X/00204-7. URL https://linkinghub.elsevier.com/retrieve/pii/B012176480X002047.

[6] David E. Busch. *Monitoring Ecosystems*. Island Press. ISBN 978-1-59726-264-4. URL https://books.google.no/books?id=QkcNorxksaIC.

[7] Andrew W. Jones, Katie A. Burchard, Anna M. Mercer, John J. Hoey, Michael D. Morin, Giovanni L. Gianesin, Jacob A. Wilson, Calvin R. Alexander, Brooke A. Lowman, Debra G. Duarte, David Goethel, James Ford, James Ruhle, Rodman Sykes, and Troy Sawyer. Learning From the Study Fleet: Maintenance of a Large-Scale Reference Fleet for Northeast U.S. Fisheries. 9. ISSN 2296-7745. URL https://www.frontiersin.org/articles/10.3389/fmars.2022.869560.

[8] Amir Yaghoubi Shahir, Tilemachos Charalampous, Mohammad A. Tayebi, Uwe Glasser, and Hans Wehn. TripTracker: Unsupervised Learning of Fishing Vessel Routine Activity Patterns. In *2021 IEEE International Conference on Big Data (Big Data)*, pages 1928–1939. IEEE. ISBN 978-1-66543-902-2. doi: 10.1109/BigData52589.2021.9671492. URL https://ieeexplore.ieee.org/document/9671492/.

[9] Sintef - anvendt forskning, teknologi og innovasjon, . URL https://www.sintef.no/.

[10] SINMOD - SINTEF, . URL https://www.sintef.no/sintef-ocean/satsinger/sinmod/.

[11] Decision support for fishing vessels based on marine ecosystem models and fishery data. URL https://www.sintef.no/en/projects/2019/decision-support-for-fishing-vessels-based-on-marine-ecosystem-models-and-fishery-data/.

[12] de Boerder Kristina Souza, Erico N., Stan Matwin, and Boris Worm. Improving Fishing Pattern Detection from Satellite AIS Using Data Mining and Machine Learning. 11(7):e0158248. ISSN 1932-6203. doi: 10.1371/journal.pone.0158248.

URL https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0158248.

[13] A. L. Samuel. Some studies in machine learning using the game of checkers. 44 (1.2):206–226. ISSN 0018-8646. doi: 10.1147/rd.441.0206. URL https://ieeexplore.ieee.org/abstract/document/5389202.

[14] Ayush Pant. Introduction to Machine Learning for Beginners. URL <https://towardsdatascience.com/introduction-to-machine-learning-for-beginnerseed6024fdb08>:.

[15] Yoshua Bengio, Ian Goodfellow, and Aron Courville. *Deep Learning*. URL https://www.deeplearningbook.org/.

[16] Jorge S. Marques. Machine Learning Slides.

[17] Applications of Machine Learning - Javatpoint. URL https://www.javatpoint.com/applications-of-machine-learning.

[18] Top 10 Machine Learning Applications and Examples in 2023. URL https://www.simplilearn.com/tutorials/machine-learning-tutorial/machine-learning-applications.

[19] Anirudha Ghosh, A. Sufian, Farhana Sultana, Amlan Chakrabarti, and Debashis De. Fundamental Concepts of Convolutional Neural Network. pages 519–567. ISBN 978-3-030-32643-2. doi: 10.1007/978-3-030-32644-9_36. URL https://www.researchgate.net/profile/A-Sufian/publication/337401161_Fundamental_Concepts_of_Convolutional_Neural_Network/links/5ed612a7299bf1c67d3292e6/Fundamental-Concepts-of-Convolutional-Neural-Network.pdf.

[20] Rikiya Yamashita, Mizuho Nishio, Richard Kinh Gian Do, and Kaori Togashi. Convolutional neural networks: An overview and application in radiology. 9(4):611–629. ISSN 1869-4101. doi: 10.1007/s13244-018-0639-9. URL https://insightsimaging.springeropen.com/articles/10.1007/s13244-018-0639-9.

[21] A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way — Saturn Cloud Blog. URL https://saturncloud.io/blog/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way/.

[22] Introduction to Deep Learning: What Are Convolutional Neural Networks Video, . URL https://www.mathworks.com/videos/introduction-to-deep-learning-what-are-convolutional-neural-networks--1489512765771.html.

[23] Max Pooling Explained — Papers With Code. URL https://paperswithcode.com/method/max-pooling.

[24] Chigozie Nwankpa, Winifred Ijomah, Anthony Gachagan, and Stephen Marshall. Activation Functions: Comparison of trends in Practice and Research for Deep Learning. URL http://arxiv.org/abs/1811.03378.

[25] Diego Unzueta. Convolutional Layers vs Fully Connected Layers. URL https://towardsdatascience.com/convolutional-layers-vs-fully-connected-layers-364f05ab460b.

[26] Vishal Yathish. Loss Functions and Their Use In Neural Networks. URL https://towardsdatascience.com/loss-functions-and-their-use-in-neural-networks-a470e703f1e9.

[27] Geoffrey E. Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R. Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. URL http://arxiv.org/abs/1207.0580.

[28] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. URL https://www.cs.toronto.edu/~rsalakhu/papers/srivastava14a.pdf.

[29] (1) Introduction to Dropout to regularize Deep Neural Network — LinkedIn, . URL https://www.linkedin.com/pulse/introduction-dropout-regularize-deep-neural-network-saurav-singla/.

[30] Connor Shorten and Taghi M. Khoshgoftaar. A survey on Image Data Augmentation for Deep Learning. 6(1):60. ISSN 2196-1115. doi: 10.1186/s40537-019-0197-0. URL https://journalofbigdata.springeropen.com/articles/10.1186/s40537-019-0197-0.

[31] Jiawei Han, Micheline Kamber, and Jian Pei. *The-Morgan-Kaufmann-Series-in-Data-Management-Systems-Jiawei-Han-Micheline-Kamber-Jian-Pei-Data-Mining.-Concepts-and-Techniques-3rd-Edition-Morgan-Kaufmann-2011.Pdf*. 3 edition. URL http://myweb.sabanciuniv.edu/rdehkharghani/files/2016/02/The-Morgan-Kaufmann-Series-in-Data-Management-Systems-Jiawei-Han-Micheline-Kamber-Jian-Pei-Data-Mining.-Concepts-and-Techniques-3rd-Edition-Morgan-Kaufmann-2011.pdf.

[32] Alper Taner, Yeşim Benal Öztekin, and Hüseyin Duran. Performance Analysis of Deep Learning CNN Models for Variety Classification in Hazelnut. 13(12):6527. ISSN 2071-1050. doi: 10.3390/su13126527. URL https://www.mdpi.com/2071-1050/13/12/6527.

[33] Sarang Narkhede. Understanding AUC - ROC Curve. URL https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5.

[34] Charu Aggarwal and Chandan Reddy. *Data Clustering, Algorithms and Applications*. URL https://web.archive.org/web/20160410055656id_/http://haralick.org/ML/data_clustering.pdf#page=27.

[35] T. Soni Madhulatha. An Overview on Clustering Methods. URL http://arxiv.org/abs/1205.1117.

[36] Mahamed G.H. Omran, Andries P. Engelbrecht, and Ayed Salman. An overview of clustering methods. 11(6):583–605. ISSN 15714128, 1088467X. doi: 10.3233/IDA-2007-11602. URL https://www.medra.org/servlet/aliasResolver?alias=iospress&doi=10.3233/IDA-2007-11602.

[37] Julio-Omar Palacio-Niño and Fernando Berzal. Evaluation Metrics for Unsupervised Learning Algorithms. URL http://arxiv.org/abs/1905.05667.

[38] Ashutosh Bhardwaj. Silhouette Coefficient : Validating clustering techniques. URL https://towardsdatascience.com/silhouette-coefficient-validating-clustering-techniques-e976bb81d10c.

[39] Junwei Xiao, Jianfeng Lu, and Xiangyu Li. Davies Bouldin Index based hierarchical initialization K-means. 21(6):1327–1338. ISSN 1088467X. doi: 10.3233/IDA-163129. URL https://search.ebscohost.com/login.aspx?direct=true&db=a9h&AN=126400516&site=ehost-live.

[40] Michael J. Brusco and Douglas Steinley. A Comparison of Heuristic Procedures for Minimum Within-Cluster Sums of Squares Partitioning. 72(4):583–600. ISSN 0033-3123, 1860-0980. doi: 10.1007/s11336-007-9013-4. URL http://link.springer.com/10.1007/s11336-007-9013-4.

[41] Principal Component Analysis (PCA) - NTNU Universitetsbiblioteket, . URL https://bibsys-almaprimo.hosted.exlibrisgroup.com/primo-explore/openurl?sid=google&auinit=T&aulast=Kurita&atitle=Principal%20component%20analysis%20(PCA)&id=doi:10.1007%2F978-3-030-03243-2_649-1&vid=NTNU_UB&institution=NTNU_UB&url_ctx_val=&url_ctx_fmt=null&isSerivcesPage=true.

[42] Principal component analysis, . URL https://en.wikipedia.org/w/index.php?title=Principal_component_analysis&oldid=1154463284.

[43] Dong Yang, Lingxiao Wu, Shuaian Wang, Haiying Jia, and Kevin X. Li. How big data enriches maritime research – a critical review of Automatic Identification System (AIS) data applications. 39(6):755–773. ISSN 0144-1647. doi: 10.1080/01441647.2019.1649315. URL https://doi.org/10.1080/01441647.2019.1649315.

[44] Ais norge, . URL https://www.kystverket.no/navigasjonstjenester/ais/ais-artikkelside/.

[45] AIS (Automatic Identification System) Overview, . URL https://shipping.nato.int/nsc/operations/news/2021/ais-automatic-identification-system-overview.aspx.

[46] Pablo Kaluza, Andrea Kölzsch, Michael T. Gastner, and Bernd Blasius. The complex network of global cargo ship movements. 7(48):1093–1103. ISSN 1742-5689. doi: 10.1098/rsif.2009.0495. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2880080/.

[47] Marc West, Tristan Cooper, and Bernard Kachoyan. AIS Analysis in Support of Counter-Piracy Operations. 2:110. doi: 10.1080/18366503.2010.10815665.

[48] Morten Winther, Jesper H. Christensen, Marlene S. Plejdrup, Erik S. Ravn, Ómar F. Eriksson, and Hans Otto Kristensen. Emission inventories for ships in the arctic based on satellite sampled AIS data. 91:1–14. ISSN 1352-2310. doi: 10.1016/j.atmosenv.2014.03.006. URL https://www.sciencedirect.com/science/article/pii/S1352231014001678.

[49] Shilavadra Bhattacharjee. What is Automatic Identification System (AIS)- Types And Working (FAQs). URL https://www.marineinsight.com/marine-navigation/automatic-identification-system-ais-integrating-and-identifying-marine-communication-channels/.

[50] Athanassios Goudossis and Sokratis K. Katsikas. Towards a secure automatic identification system (AIS). 24(2):410–423. ISSN 0948-4280, 1437-8213. doi: 10.1007/s00773-018-0561-3. URL http://link.springer.com/10.1007/s00773-018-0561-3.

[51] Fishing Factors. URL https://www.fishranger.com.au/fishing_factors.

[52] Electronic Reporting Systems. URL https://www.fiskeridir.no/English/Fisheries/Electronic-Reporting-Systems.

[53] Åpne data: elektronisk rapportering (ers). URL https://www.fiskeridir.no/Tall-og-analyse/AApne-data/elektronisk-rapportering-ers.

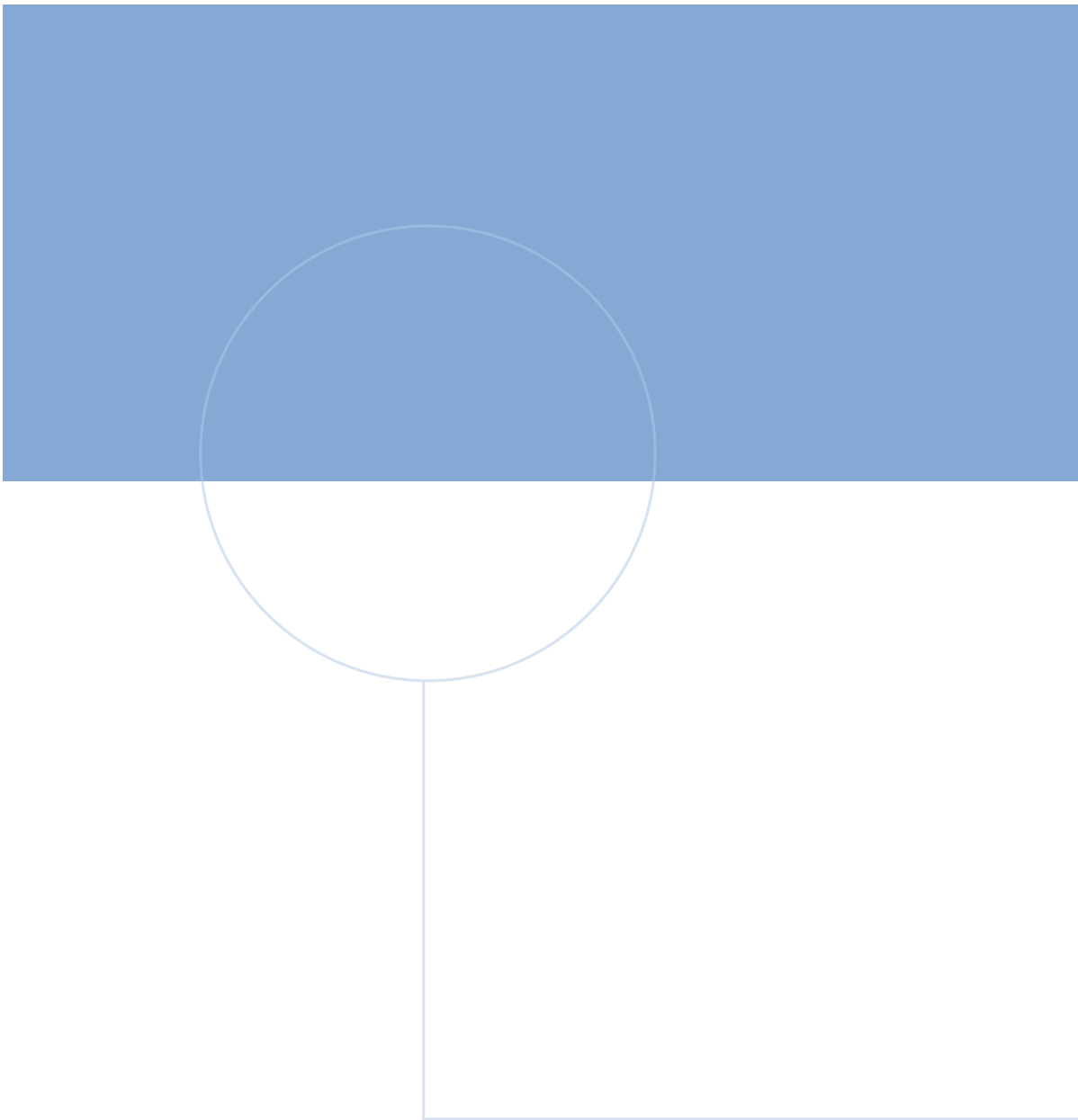
[54] G. Huse, S. Railsback, and A. Feronö. Modelling changes in migration pattern of herring: Collective behaviour and numerical domination. 60(3):571–582. ISSN 1095-8649. doi: 10.1111/j.1095-8649.2002.tb01685.x. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1095-8649.2002.tb01685.x.

[55] Ian H McQuinn. Metapopulations and the Atlantic herring. URL https://link.springer.com/article/10.1023/A:1018491828875.

[56] I Huse. Tilt angle distribution and swimming speed of overwintering Norwegian spring spawning herring. 53(5):863–873. ISSN 10543139. doi: 10.1006/jmsc.1996.9999. URL https://academic.oup.com/icesjms/article-lookup/doi/10.1006/jmsc.1996.9999.

[57] Ad Corten. The role of "conservatism" in herring migrations. URL https://link.springer.com/article/10.1023/A:1021347630813.

[58] J. M Peña, J. A Lozano, and P Larrañaga. An empirical comparison of four initialization methods for the K-Means algorithm. 20(10):1027–1040. ISSN 0167-8655. doi: 10.1016/S0167-8655(99)00069-0. URL https://www.sciencedirect.com/science/article/pii/S0167865599000690.



NTNU

Norwegian University of
Science and Technology