Universiteit Leiden

Master Computer Science

Temporal design choices for feature engineering from dynamic networks

Name: Yven Lommen
Student ID: s2040964
Date: 14/08/2023

Specialisation: Data Science

1st supervisor: dr. A.P. Pereira Barata
2nd supervisor: dr. F.W. Takes
Other supervisors: Paul Merkx, MSc
drs. Jasper van Vliet

Master's Thesis in Computer Science

Leiden Institute of Advanced Computer Science
Leiden University
Niels Bohrweg 1
2333 CA Leiden
The Netherlands

Universiteit Leiden

Human Environment and Transport Inspectorate
Ministry of Infrastructure and Water Management

This thesis was made possible by the Human Environment and Transport Inspectorate of the Netherlands. The student participated in an internship which has led to the outcomes of this thesis.

Abstract

In machine learning, feature engineering is an often difficult yet required part of solving any downstream task such as a classification problem. In this thesis we focus on feature engineering from networks whose characteristics may change over time, i.e., dynamic networks. If this change in network characteristics is not accounted for during learning, the final model may underperform throughout its deployment as a direct consequence of the so-called concept drift in time. We examine the impact of temporal design choices for feature engineering from dynamic networks towards the use-case of ship movement networks used within the Human Environment and Transport Inspectorate of the Netherlands (Inspectie Leefomgeving en Transport, ILT). A ship movement network is modelled as a mobility network, a type of network which models ports as locations and observed trips of agents (ships) moving from one port to another. The ILT may use transport data to target non-compliant behaviour, by training a machine learning model on these data. To identify optimal temporal design choices, we make use of an existing feature engineering approach, such that different real-world scenarios are mimicked and evaluated. We measure the impact of the temporal design choices by taking the classification performance of ship type using the Area Under the Receiver Operating Characteristic Curve (ROC AUC). Results indicate that, while concept drift over time negatively impacts learning from network properties, this impact can be mitigated using appropriate design choices such as periodic retraining. Additionally, we find that increasing the size of the data set used for training the classifier does not substantially impact the performance of the classifier when using trajectory-based features. This offers the possibility to reduce the usage of computational resources without making sacrifices with regard to classification performance. The findings of this thesis put emphasis on time awareness when designing machine learning models, serving as an example for evaluating time-aware design choices.

Acknowledgements

First and foremost, I would like to express my sincere gratitude towards Dr. António Pereira Barata, who always gave great feedback and put up with the many questions I had. I have learned a lot through the project and internship which is largely due to him. I also want to thank my second supervisor, Dr. Frank Takes, for the insightful meetings and feedback he provided.

To follow up, I want to take this opportunity to thank the ID-lab team of the ILT, which enabled me to do this project. Special thanks to Paul Merkx and Jasper van Vliet, with whom I had the pleasure to work with the past eight months.

Lastly, since this project closes my past three years of master's education, I would like to thank my partner, family, and friends for always being alongside me. With a special mention to my mom, since I know how much this means to her.

Contents

Chapter 1

Introduction

The Human Environment and Transport Inspectorate (Inspectie Leefomgeving en Transport, ILT) is the supervising authority of the Ministry of Infrastructure and Water Management of the Netherlands (Environment & Inspectorate, 2023). They aim to improve safety and sustainability with regard to transport, infrastructure, environment, and housing. One of their tasks is ensuring that (transportation) organisations are compliant with the laws and regulations which they are responsible for. Given the limited inspection capacity of the ILT, machine learning is used to transform this into a data-driven task.

One of the use cases for machine learning is the risk assessment of ships which want to berth in Dutch harbours. Meaning that they aim to classify ships that are least likely to be compliant with the law in order to further investigate these ships. In this use case, the aim is to maximize performance with regard to classifying non-compliant ships. In order to do so, there needs to be a suitable way of describing ship behaviour which can serve as input for machine learning models.

Networks are often used to model real-world entities and their links to other entities. They are modelled by graphs where nodes in the graph denote an entity in the network and an edge in the graph denotes a link between two entities. In this thesis we make use of mobility networks, which differ from conventional network representations. In a mobility network, rather than entities, there exist agents which move from location to location within the network. A node in such a network denotes a location which can be visited by an agent. An edge in a mobility network represents a trip of an agent moving from one location to another, resulting in a directed network. Furthermore, mobility networks are temporal, meaning that a location visit by an agent is timestamped.

There exist many ways to derive meaningful features from mobility networks and use these features for the aforementioned machine learning tasks. Some methods, such as the one used by de Bruin et al. (2022), make use of centrality measures within mobility networks in order to describe trajectories as feature vectors. Other methods make use of neural networks in order to learn representations of trajectories (Han et al. (2021); X. Li et al. (2018); Yao et al. (2019)). One of the aforementioned methods can be used to represent agents moving through a mobility network. This process results in trajectory-based features, as they describe the trajectory of an agent through the network. As the behaviour of agents changes over time, so do the features which represent them, logically. In addition, overall network characteristics may also change over time, making earlier models that map behaviour through these networks sub-optimal.

This phenomenon is sometimes referred to as concept drift, in which patterns and relations within the data evolve over time (Žliobaitė et al., 2016). Different strategies have been proposed to deal with concept drift (Celik & Vanschoren, 2021). The idea of concept drift raises other questions regarding time and real-world applicability. Especially when dealing with trajectory-based features, because these features may represent periodic movement patterns which may only be captured by feature engineering approaches if a sufficiently long time period is chosen.

Concept drift and other time related aspects of machine learning need to be considered by the ILT when deploying models for real-world applications. Assurance is needed that models remain up to date and are sufficiently reliable for daily practice. This thesis investigates temporal design choices which are related to the applicability of machine learning in the ILT context, with the aim of arriving at better model performance and insights in machine learning practices.

In this thesis we use port calls data, which denote which ships have berthed in which ports, in order to derive a ship movement network, i.e., a mobility network. In such a network, the moving agents are ships and the visited locations are ports. Using this data, a feature engineering approach models ship trajectories through the network. The resulting feature vector represents each ship by its trajectory and is used as input for a ship type classification model to evaluate the meaningfulness of these features. We then mimic real-world design choices, with regard to time, in order to investigate to what degree these design choices impact the performance of the machine learning model. These temporal design choices are all related to the construction of the data set used to train the classifier and the data set used to evaluate the classifier. We focus on two important aspects regarding these data sets. The first aspect is the time period between the deployment of a trained classification model and the classification of an unseen ship. We inspect this in order to examine the durability of a deployed model and thus to detect concept drift. The second aspect which is examined is the size of the data set which is used for training the classifier. With this aspect we aim to investigate whether or not it is beneficial to add more data, both in terms of number of instances (ships) and the size of the considered trajectory of each ship. This is valuable since it could help to improve machine learning performance or reduce computational costs. Both aspects will be explained formally in Chapter 2. The problem which this thesis addresses is formulated as:

How do temporal design choices impact the performance of a machine learning approach based on network features?

To answer the question that the research problem introduces, we use classification performance as a proxy for evaluating the meaningfulness of the network-derived features. We do so because changes in performance can be measured in a concrete manner and changes in the learning ability of the classifier using the presented features are easily identified. A classifier is trained by using a boosted tree learner with a fixed set of hyperparameters. Furthermore, we ensure that, when applicable, train and test data sets are kept the same throughout different design choices. Therefore, we may assume that differences in performance can be accounted for by the representativeness of the features and not by neither (a) the learning algorithm nor (b) the train and test sets. Despite the context of this thesis, which is that of ships, our method can be generalized to other settings that can be represented by mobility networks and similarly modelled features. Using this method we aim to answer the following research questions:

1. *How does the size of the data set used for training a classifier impact classification performance when using trajectory-based features?*
2. *Does classification performance, when using network-derived features, decline over time when using a train-once method?*

This thesis is done in collaboration with the ILT, addressing the specific problems that such organisations may have when implementing similar machine learning solutions. The contributions of this thesis are listed below.

- We empirically show that temporal design choices, with regard to train and test data, impact classification performance when dealing with trajectory-based features.
- We propose a solution which aims to evaluate the impact of the temporal design choices on classification performance.

The remainder of the thesis is structured as follows. We begin by providing a formal problem definition in Chapter 2. After that, we discuss related work in Chapter 3. Following, the methods with which we answer the research questions are described in Chapter 4. The experimental setup is explained in Chapter 5. The results of the conducted experiments are presented in Chapter 6. Our results are elaborated upon in Chapter 7. Finally, we close off by concluding our findings and answering the research questions in Chapter 8.

Chapter 2

Problem Definition

2.1 Mobility networks

In this thesis, we make use of mobility networks. Mobility networks can be described as networks with locations and agents which move through the network from location to location. A mobility network can be modelled as a directed graph $G = (V, E)$, where V denotes the set of nodes and E denotes the set of edges. A node $v \in V$ in this network represents a location, and an edge $(u, v) \in E$ represents an observed trip of an agent from location u to location v. Multiple edges between the same two end points may exist in the graph, denoting different trips between the two endpoints (by different agents or at different times). Using this graph, we can then model the trajectory (or journey) of an agent through the graph as a sequence of visited nodes. The modelled trajectory of an agent can be denoted as $j = (v_1, v_2, .., v_n)$, where v_i denotes the location visited after visiting location v_{i-1}, $1 \leq i \leq n$ and a total of n locations is visited, which is the *length* of the journey. The journey of an agent is constructed by concatenating all of the trips of this agent. Each location $v_i \in j$ contains a timestamp which is used to derive the order of the sequence and is used to derive temporal information.

2.2 Ship classification

In our application, we model a ship movement network as a mobility network where ships are the moving agents. In this ship movement network, ports are equivalent to locations in a mobility network. The set of all ships is denoted by S and the ship movement network is denoted by G. The ship movement network is derived from all of the observed trips from a subset of the ships, S_G. The set of ships is further divided into subsets S_{train} and S_{test} for the training and testing (evaluation) of the classifier. It also holds that S_{train} and S_{test} are mutually exclusive in order to prevent data leakage, the same holds for S_G and S_{test}. However, there can exist an intersection between S_{train} and S_G such that $S_{train} \cap S_G \neq \emptyset$. A journey, j_s, by a ship $s \in S$ represents the sequence of visited ports in the network. The set of all journeys of all ships in S is denoted by the set J_S. We can derive S and J_S from data set D, which contains timestamped information about ships and their visited ports.

The task we focus on using the ship movement network is to classify ships which move through it. In order to do so, a ship, s, needs to be described by a single feature vector describing its journey, j_s, through the network. Thus we need to model the journey with a suitable method. For classification, the goal is to classify a single ship by its ship type. Given this task we can provide a formal problem definition for the classification.

Given a ship movement network G constructed with S_G, a set of ships S_{train}, a set of ships S_{test} and their corresponding sets of trajectories $J_{S_{train}}$ and $J_{S_{test}}$, predict the ship type for all ships $s \in S_{test}$ with a classifier trained on the ships in S_{train} using the features derived from G, $J_{S_{train}}$ and $J_{S_{test}}$.

2.3 Design concepts

Given the problem definition, we can now identify several design concepts which affect the performance of the classification task.

- The *encoding time frame*, which represents the temporal bounds for the journey of a ship. The order of a journey, j_s, of a ship, s, is determined by the timestamps belonging to each port visit. The encoding time frame determines which port visits are included in a journey, thus impacting the resulting feature vector for any ship. More formally, an encoding time frame is represented by its bounds $[t_i, t_k]$. Where both t_i and t_k are timestamps and it always holds that $t_i < t_k$. The size of the encoding time frame is the time that is covered by the bounds of it and is calculated by $t_k - t_i$.
- The *sample time frame*, which represents the time period in which the ships for S_G, S_{train} or S_{test} are sampled. It is used to sample ships that have a port visit that occurred within its bounds. The sample time frame impacts the time period in which unique ships are selected when generating S_G, S_{train}, and S_{test}. A sample time frame for S_{train} consists of the bounds $[t_i, t_k]$ and a sample time frame for S_{test} consists of the bounds $[t_m, t_n]$. Again, all bounds represent timestamps and it holds that $t_k < t_m$. The time period between t_k and t_m is referred to as the *gap* and its size is, logically, calculated by $t_m - t_k$. The constraint $t_k < t_m$ implies that the size of the gap must always be greater than zero. The sample time frame for the ships in S_G has the same constraint; i.e., it must have a gap greater than zero with the sample time frame of S_{test}. The size of the sample time frame can be calculated in the same manner as the size of the encoding time frame.

Chapter 3

Related Work

This thesis aims to investigate the impact of design choices on feature engineering from networks by examining design choices with regard to time and real-world applicability. We firstly talk about such feature engineering approaches in Section 3.1. Secondly, in Section 3.2 we elaborate on concept drift and its relevance to this research.

3.1 Modelling ship behaviour

One straightforward approach to feature engineering from networks is through node embeddings. Node embeddings are continuous feature representations of nodes in a network, which are learned according to specific algorithmic frameworks. A well-established example of this is node2vec (Grover & Leskovec, 2016). A downside of such an approach in our context is that only nodes are represented by features, and not trajectories. Another approach for feature generation is to use graph neural networks (Han et al., 2021). However, deep learning models are ill-equipped with respect to model explainability, a most relevant and often required aspect for deployment.

In the work of de Bruin et al. (2022), a pipeline is created which generates ship trajectory features from a static network, based on both centrality measures and edge attribute distributions. Since this related work satisfies our requirements as per Chapter 2, feature engineering is performed accordingly, with the added implementation of a time-aware component. We describe this addition in Section 4.3. For classification, the target variable is that of ship type and serves as proxy for the assessment of the generated features.

In real-world deployment, a more desired target variable is that of ship (non-)compliance. However, under data availability constraints, this target variable is not an option for the scope of this research. Nevertheless, the results and insights produced in this work remain valuable for at least two reasons. First, we provide a set of design choices which may impact classifier performance: these design choices should be considered when learning other classification tasks. Second, by learning a domain-specific category (in our case: ship type), the notion of *crosslier* (Pereira Barata et al., 2021) may be exploited.

A crosslier is a category-labelled instance of which the feature values and the present label are disharmonious. To put it differently, they are instances which seem to belong to a different category, rather than the one they display, according to a supervised learning classifier. To detect crossliers, a crosslier score is given to each instance: the greater the performance of a classifier, the greater the reliability of the crosslier score. Design choices which impact the performance of the classifier will, therefore, directly impact the assessment of the crosslier score: design choices which negatively impact classifier performance will negatively impact crosslier detection.

3.2 Concept drift

In machine learning, concept drift is the phenomenon under which patterns and relations within the data evolve over time (Žliobaitė et al., 2016). It is often distinguished from data drift, in this thesis we assume that both concepts regard the changes in the distributions of the data over time. Studies have shown that data drift does appear in the global shipping network (GSN). One of these studies is done by Z. Li et al. (2015), who investigated the dynamic changes in centrality of the GSN in the period from 2001-2012. One of their findings is that to a large extent Europe is always at the center of the GSN. They also find that this position is declining which reveals that shifts in centrality distributions are happening. In this study we focus on concept drift with regard to trajectory patterns over time. More specifically, we focus on how different time periods of modelled trajectories impact classification performance. We aim to discover which temporal design choices are most suitable when learning from dynamic network features, in the setting of ship movement networks. We elaborate more on the precise aspects we study in Section 4.3.

Chapter 4

Methodology

In this chapter, we describe the methodology used to answer the research questions. We firstly describe the data in Section 4.1. Secondly, in Section 4.2 the feature engineering approach is explained. Lastly, we elaborate on the temporal design choices in Section 4.3.

4.1 Data

The data set, D, for this research represents calls from ships to ports (i.e. port calls) and stems from EMSA European Maritime Safety Agency (EMSA, 2023). Each data entry contains information about which ship would berth at which port. The data set contains information about approximately 36 000 ships, for a total of circa 6 million port calls, between the years 2014 and 2020, inclusive. The relevant attributes which are used in this study are described in Table 4.1. The table shows two timestamps: one denotes arrival at a port and the other denotes departure from a port. The timestamp for the arrival is used for the determination of the encoding time frame. Both timestamps are used for the calculation of the temporal features described in Section 4.2. In order to make the data suitable for experimentation some necessary preprocessing steps are taken and are described in Appendix A. The distribution of the target class (i.e., ship type) is provided in Figure 4.1.

Attribute	**Description**
IMO Number	Unique identifier of the ship making the port call.
Port Name	Unique identifier of the port onto which the ship wishes to berth or depart from.
Arrival Time	Time at which the ship arrived at the port.
Departure time	Time at which the ship departed from the port.
Ship Type	Ship category; to be used as class label.

Table 4.1: Overview of the attributes in the port calls data.

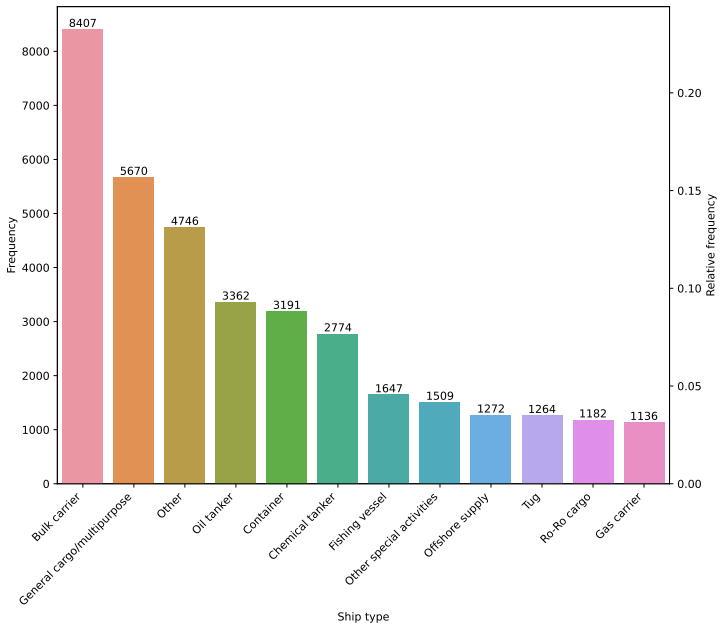


Figure 4.1: Ship type distribution in the port calls data set.

4.2 Feature engineering

The feature engineering approach, upon which the impact of the design choices is tested, stems from de Bruin et al. (2022). This method encodes a journey j_s into a single feature vector. In this section we provide a brief summary of the approach. For the full description and details of the method we refer to the original paper.

At the beginning of the feature engineering process a ship movement network, as described in Chapter 2, is constructed using the subset of ship specific port calls S_G. Here we remark that S_G can be sampled in different manners. This is explicitly stated for each experimental setup in Chapter 5. From this network, twelve centrality measures are calculated for each port in order to obtain its structural importance: in-degree, out-degree, degree, in-strength, out-strength, strength, (weighted) closeness centrality, (weighted) betweenness centrality and (weighted) eigenvector centrality. Using this network, the features for each ship can be calculated.

In their work, de Bruin et al. (2022) make a distinction between two types of features: (1) network features and (2) temporal features. The first type of features relate to the aforementioned centrality measures. The second type of features relate to temporal aspects such as travel time or port stay duration, for example.

Summarily, the values of each centrality measure across the network are stored, as well as the different temporal measurements. These stored measurements are used to create cutoff points with which the encoding and aggregation of journeys may be performed, following a quantile binning strategy. By doing so, each unique ship can, for a specific time frame, be encoded into a single instance for classifier learning. Strictly following the approach provided in the original work generates a total of 503 features, which enable the encoding of ship behaviour across time. The values of the network and temporal features are in range [0;1] due to the aggregation. The remaining features, not representing network or temporal measurements, contain integer values. For the full implementation details, see the code repository (Lommen, 2023).

4.3 Design choices

Our design choices are intrinsically related to the two concepts of *encoding time frame* and *sample time frame*, introduced in Section 2.3. On the one hand, the first concept acts as a tool to answer the first research question regarding the size of the data set used for classifier learning. On the other hand, the second concept is applied to answer the second research question, which investigates the presence of a decline in classification performance over time. Towards answering both research questions, we apply these concepts within three design choices: (1) sliding window; (2) expanding window; and (3) train-test gap. To note, in the following, we describe these choices irrespectively of S_G, which is made explicit in Chapter 5.

The pipeline we use in our method is depicted in Figure 4.2. The second and third step represent the time-aware components, which are at the centre of our study. These two steps are varied across our three design choices and the impact of these variations is measured.

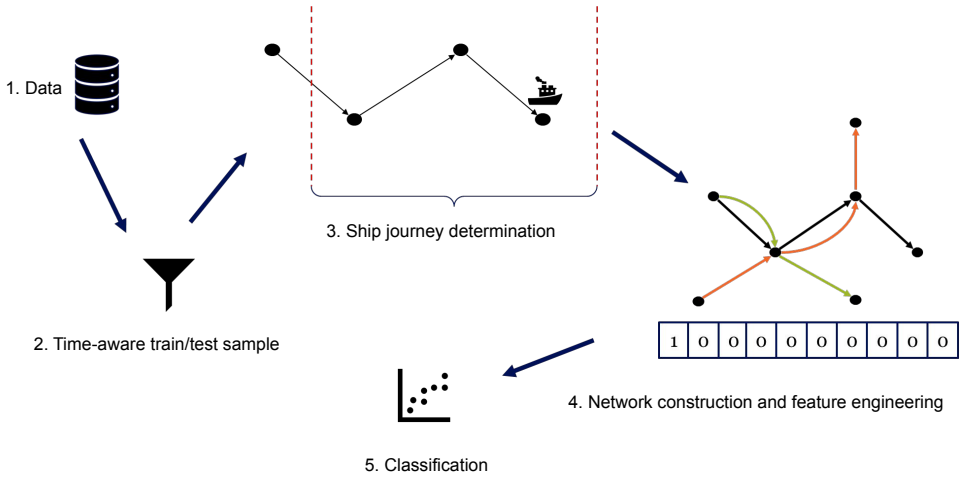


Figure 4.2: Pipeline for feature engineering and classification. The time-aware components are in the second and third step of the pipeline, in which the sample time frame and encoding time frame are implemented.

4.3.1 Sliding window

In the sliding window design, we iteratively shift both the sample time frame and encoding time frame of the train data set and test data set with a fixed time period. It mimics the real-world scenario in which a classifier is updated on a regular interval. A shift of the sample time frame or encoding time frame consisting of the bounds $[t_i, t_k]$ with time period X would result into the bounds $[t_i + X, t_k + X]$. The size of the gap between the sample time frames of the train and test data set is always approximately zero. The sliding window design in illustrated in Figure 4.3.

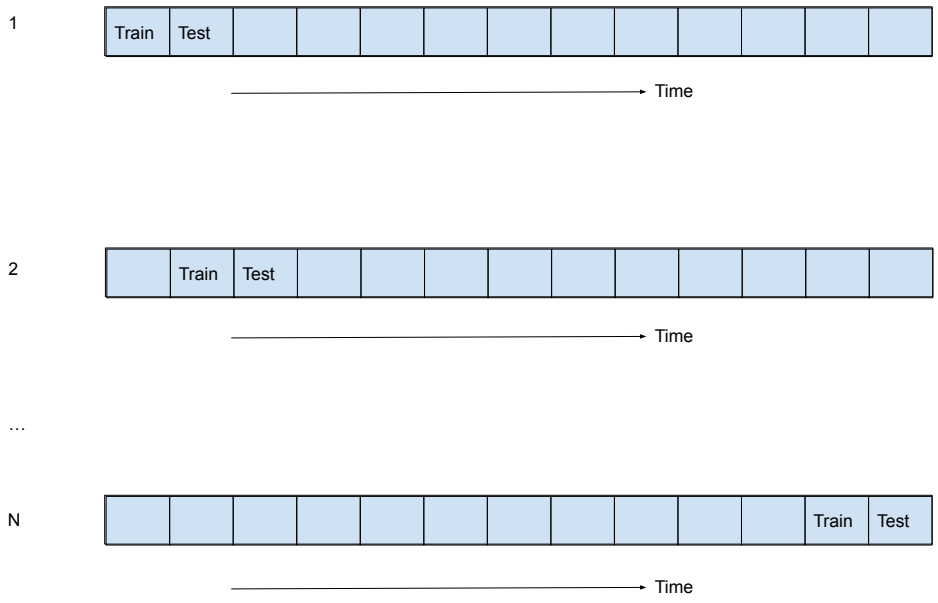


Figure 4.3: Sliding window. The data set is represented as the horizontal bar, where the individual blocks represent a fixed-size time period. The blocks with the text 'train' and 'test' indicate which parts of the data set are considered for the corresponding sample time frames of S_{train} and S_{test}.

4.3.2 Expanding window

In the expanding window design, both the size of the encoding time frame and the size of the sample time frame of the train data set are iteratively increased with a fixed time period. This design investigates different sizes of the train data set relative to the size of the test data set. The size of the train data set is increased in two manners: the number of instances in S_{train} and the length of each journey ($|j_s|$ for $\forall s \in S_{train}$). An increase in the size of the sample time frame or the size of the encoding time frame consisting of the bounds $[t_i, t_k]$ with time period X would result into the bounds $[t_i - X, t_k]$. The sample time frame and encoding time frame of the test data set remain fixed to keep evaluation consistent. The gap between the sample time frames of both the train and test data set is always approximately zero. The expanding window design is illustrated in Figure 4.4.

4.3.3 Train-test gap

In the train-test gap design, we iteratively increase the gap between the sample time frames of the train data set and test data set with a fixed time period. This design mimics the real-world scenario in which a classifier is trained once and is not updated over a longer period of time. In this design, the bounds of the sample time frame of the train data set are shifted backwards at each iteration. The sample time frame of the test data set remains the same to keep evaluation consistent. The sizes of the encoding time frames for both the train and test data set remain fixed. The train-test gap is illustrated in Figure 4.5.

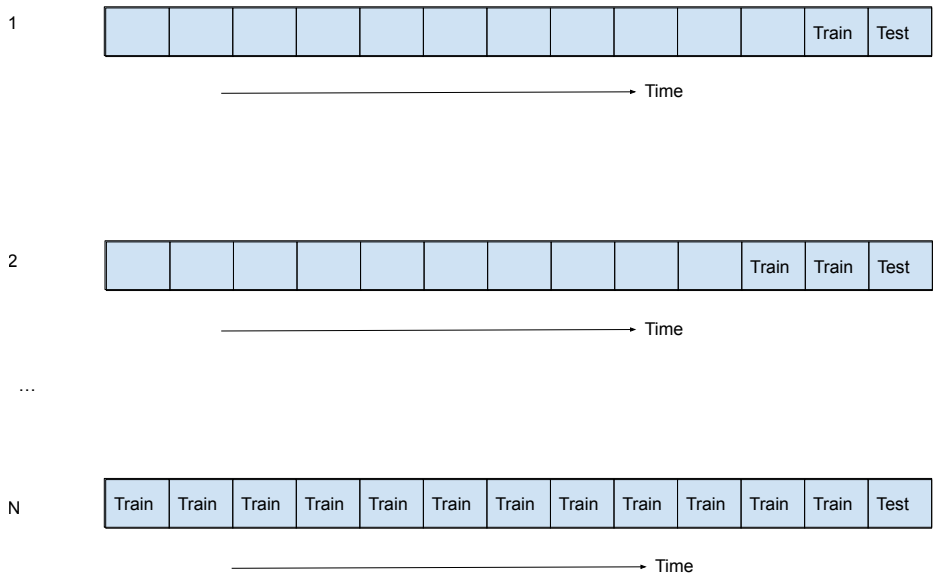


Figure 4.4: Expanding window. The data set is represented as the horizontal bar, where the individual blocks represent a fixed-size time period. The blocks with the text 'train' and 'test' indicate which parts of the data set are considered for the corresponding sample time frames of S_{train} and S_{test}.

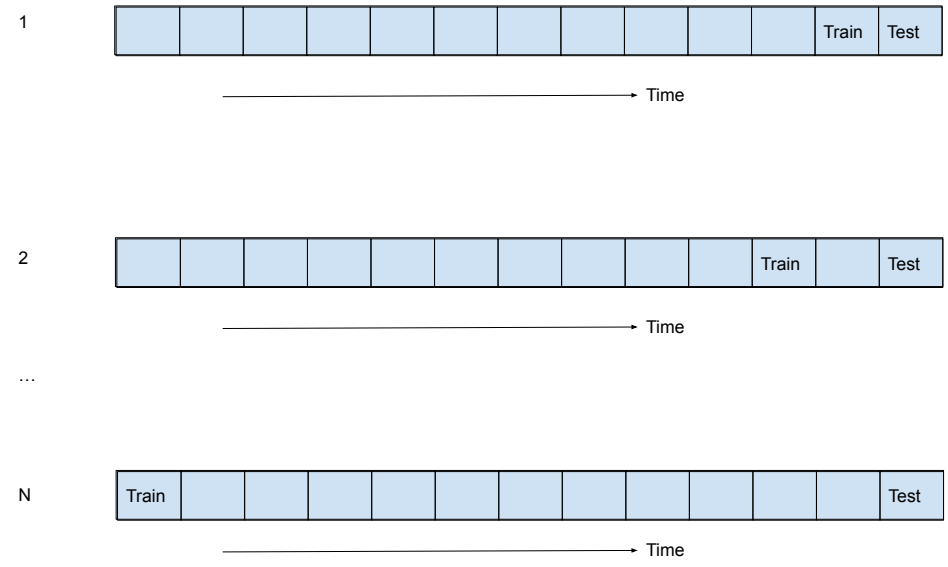


Figure 4.5: Train-test gap. The data set is represented as the horizontal bar, where the individual blocks represent a fixed-size time period. The blocks with the text 'train' and 'test' indicate which parts of the data set are considered for the corresponding sample time frames of S_{train} and S_{test}.

Chapter 5

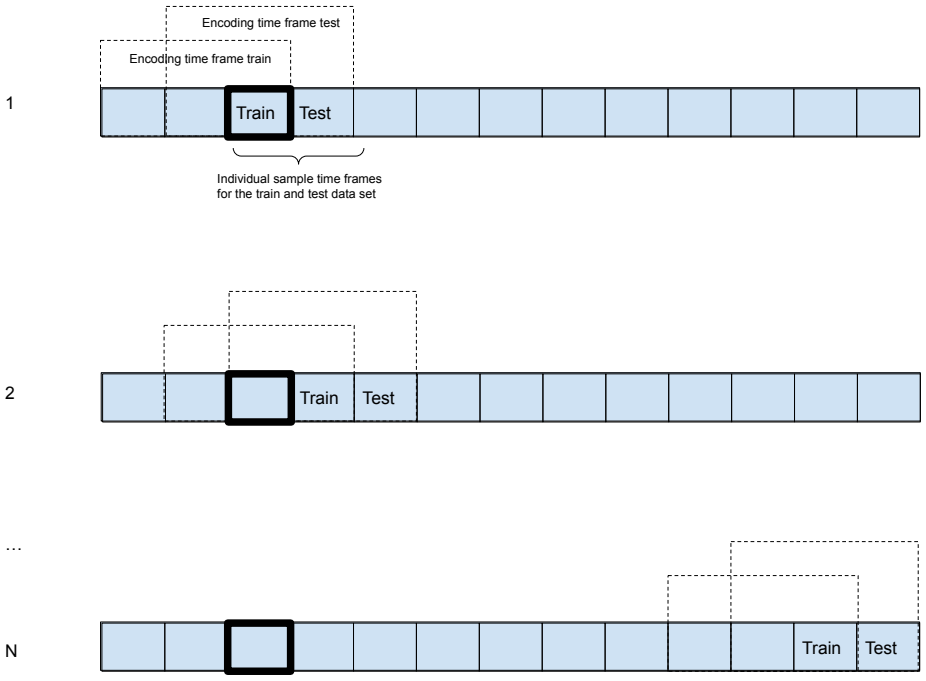
Experiments

In this chapter we describe our experimental setup. Across all experiments, our unit of time is one month. Moreover, for each design choice introduced in Section 4.3, the sampling of S_G may either follow a (1) static, or (2) resampled approach. The first approach assumes a single S_G set across all iterations in time. The second approach assumes a novel S_G set at each iteration in time. The first experimental setup, which implements the sliding window, is described in Section 5.1. The second experimental setup is explained in Section 5.2, which implements the expanding window. The third experimental setup implements the train-test gap design and is elaborated upon in Section 5.3. Fourth and lastly, the method of evaluation is detailed in Section 5.4.

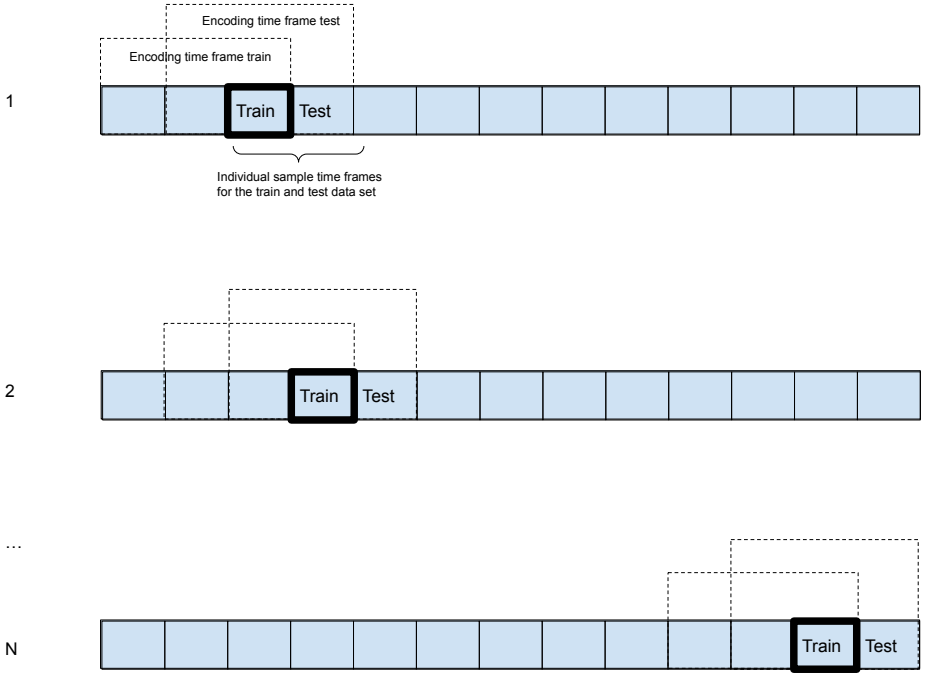
5.1 Sliding window

With respect to the sliding window design choice, both the static and resampled designs are implemented and are run for 40 iterations across time. The results are then aggregated and the mean and standard deviation are recorded. The implementation of the static and resampled network S_G for the sliding window is depicted in Figure 5.1.

In the static approach, the subset which is used to construct the ship movement network is equal to the train data set of the first iteration and remains the same throughout all iterations. In the resampled approach, the subset used for the construction of the network and for training of the classifier are equal to each other in each iteration. The size of the encoding time frames for the train data set, network construction data set, and the test data set remain fixed on twelve months for both approaches.



(a) Static sliding window



(b) Resampled sliding window

Figure 5.1: Static and resampled sliding window designs for N iterations. The data set is represented as the horizontal bar, where the individual blocks represent a one month period. The block in which the edges are bold indicates the sample time frame of the network subset (S_G). The blocks with the text 'train' and 'test' indicate the sample time frames for the train and test data set (S_{train} and S_{test}). The dashed blocks represent the encoding time frame for either the train or the test data set and are depicted as three months.

5.2 Expanding window

In the expanding window design choice, both the static and resampled designs are implemented and are run for 40 iterations due to data availability constraints. The experiments are repeated 30 times with different initialisation configurations (i.e., the start and end date of each repetition are shifted one month) to ensure reliability of results. The implementation of the static and resampled network S_G for the expanding window is depicted in Figure 5.2.

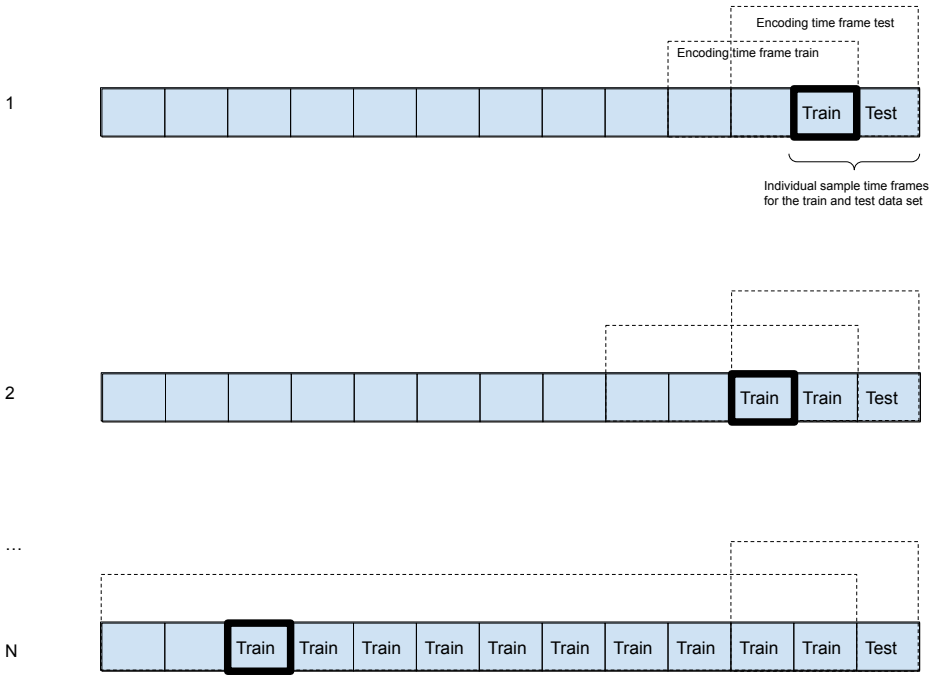
In the static approach, the subset used for the construction of the network shares its sample time frame with the first time period of the sample time frame for the train data set. In the resampled approach, the subset used for the construction of the network and for training of the classifier are equal to each other in each iteration. The test data set remains fixed throughout all of the iterations of a single repetition of the experiment.

In order to examine different sizes of encoding time frames, we make use of four different test data sets that only differ in their size of encoding time frame. The different test data sets have an encoding time frame with a size of 1, 3, 6 and 12 months and are labelled TS_1, TS_3, TS_6 and TS_{12} respectively. The train data set starts with an encoding time frame with a size of twelve months in both designs.

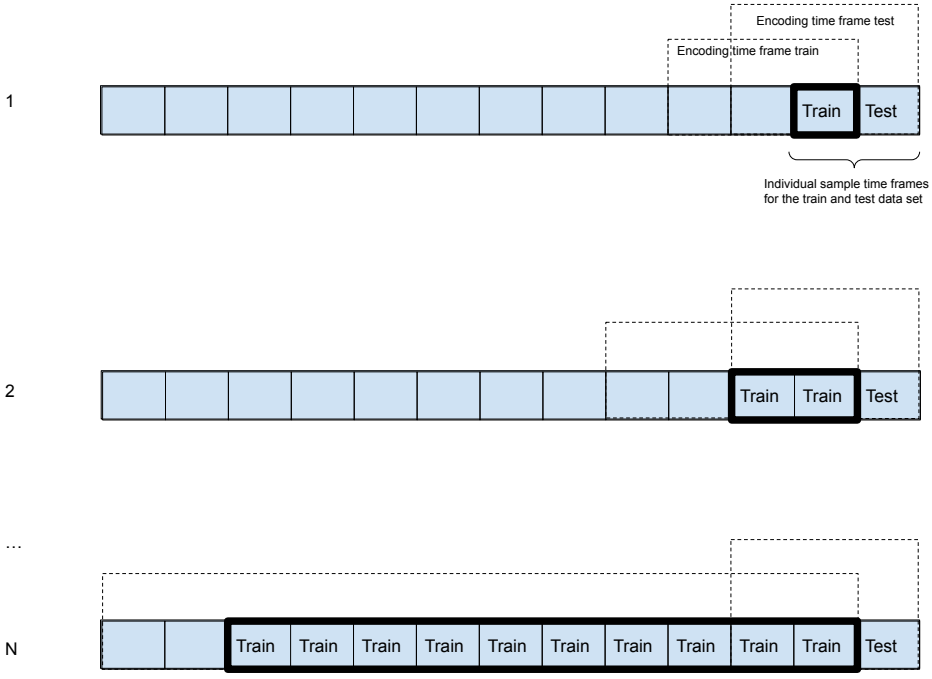
5.3 Train-test gap

For the train-test gap design, the static design is run for 20 iterations and the resampled design is run for 40 iterations, which is the maximum number of iterations under data availability constraints for both designs. Both experiments are repeated 30 times with different initialisation configurations to ensure reliability of results. Both the static and resampled train-test gap designs are graphically displayed in Figure 5.3.

In the static approach, the sample time frame of the network construction data set also has a gap between its sample time frame and that of the train data set; the gap between the sample time frame of the network construction data set and that of the train data set is the same size as the gap between the train data set and the test data set. In the resampled approach, the subset used for the construction of the network and for training of the classifier are equal to each other in each iteration. The size of the encoding time frames for the train data set, network construction data set, and the test data set remain fixed on twelve months for both approaches.

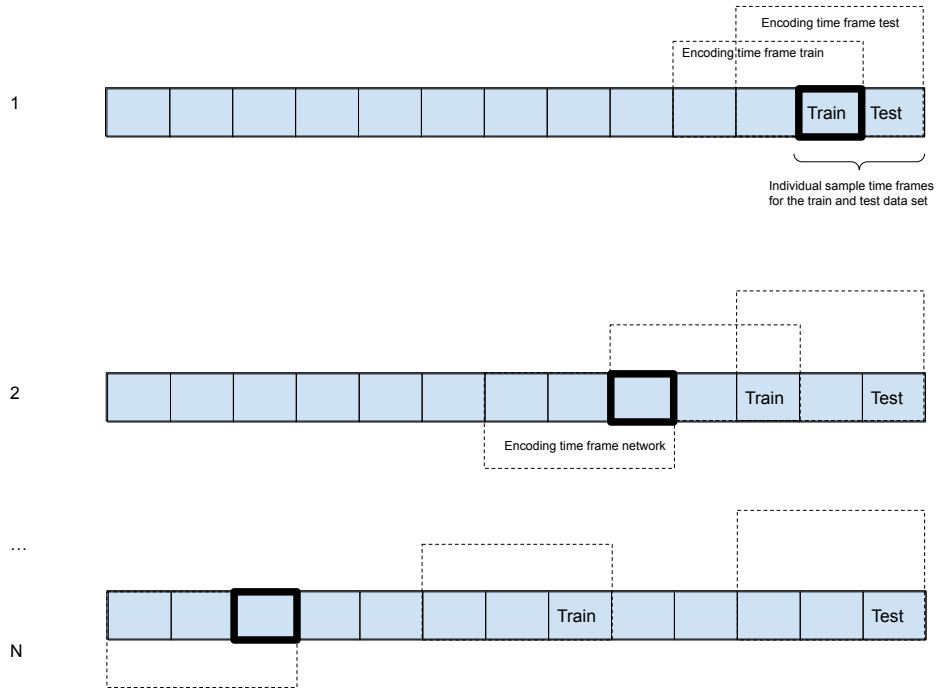


(a) Static expanding window

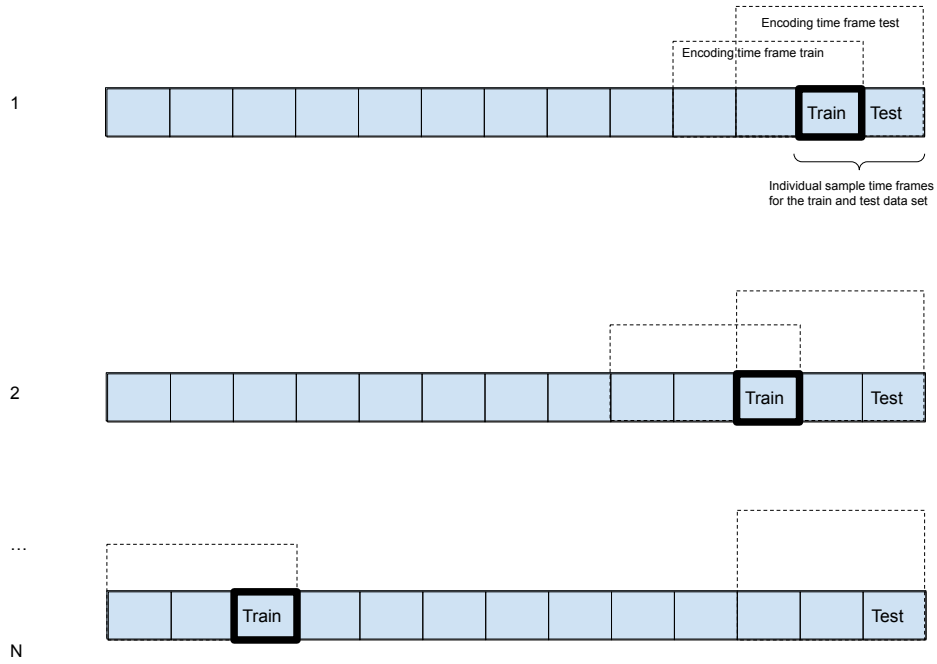


(b) Resampled expanding window

Figure 5.2: Static and resampled expanding window designs for N iterations. The data set is represented as the horizontal bar, where the individual blocks represent a one month period. The block in which the edges are bold indicates the sample time frame of the network subset (S_G). The blocks with the text 'train' and 'test' indicate the sample time frames for the train and test data set (S_{train} and S_{test}). The dashed blocks represent the encoding time frame for either the train or the test data set, for the test data set it remains fixed and is depicted as three months.



(a) Static train-test gap



(b) Resampled train-test gap

Figure 5.3: Static and resampled train-test gap designs for N iterations. The data set is represented as the horizontal bar, where the individual blocks represent a one month period. The block in which the edges are bold indicates the sample time frame of the network subset (S_G). The blocks with the text 'train' and 'test' indicate the sample time frames for the train and test data set (S_{train} and S_{test}). The dashed blocks represent the encoding time frame for either the train data set, network data set, or the test data set and are depicted as three months.

5.4 Evaluation

Two distinct choices are made with respect to the evaluation of our experiments. The choice of (1) learning algorithm - which produces a classifier towards the class label - and (2) performance metric - with which we measure classifier performance - is the same across all of our experiments: XGBoost and the Area Under the Receiver Operating Characteristic Curve (ROC AUC), respectively.

First, XGBoost is chosen due to its attested superiority in classification tasks when using tabular data; it is applied in a One-vs-Rest approach, for each class label. Second, ROC AUC is selected as it is the literature standard for binary classification tasks; a value of 1 reflects a perfect classifier, whilst a value of 0.5 indicates classification at random, and 0 indicates a perfect mislabelling classifier (i.e., always assigns the opposite label).

Throughout all of our experiments, classification performance is first aggregated as the unweighted mean of each class-specific classifier, and then averaged a second time across all different initialisation configurations, when applicable (i.e., expanding window and train-test gap experiments). All code used for the experiments is available through the repository (Lommen, 2023).

Chapter 6

Results

In this chapter we present the results of the experiments described in Chapter 5. We start with the results of the sliding window experiment (Section 6.1). Then, in Section 6.2, the results of the expanding window experiment are shown. Lastly, we display the results of the train-test gap experiment in Section 6.3.

6.1 Sliding window

The results for the sliding window experiment are presented in Table 6.1. The table shows the performance for both the static and resampled sliding window designs. The results show that the resampled sliding window design marginally outperforms the static sliding window design.

In addition to model performance, we are interested in investigating how classifiers are learned across time with respect to feature importance. The feature importance ranking of a classifier provides insights into which features are more important (i.e., which features have a greater weight when assigning a class label) than others for a classifier. When the feature importance rankings between two classifiers are very similar, it suggests that those classifiers will assign similar classification scores to the same instance. Conversely, if the rankings are different, it might suggest that the classification scores have higher variance. Here we note that this change in ranking does not strictly need to reflect, however, in differences in classification performance between the two classifiers. Spearman's rank correlation coefficient was selected to compare the feature importance rankings, since we are interested in whether a monotonic relation exists. The correlation matrices for both static and resampled designs are displayed in Figure 6.1, which shows an overall low positive correlation for both designs (≤ 0.341). The overall correlations for the static design are greater than for the resampled design, with an average of 0.255 ± 0.028 and 0.162 ± 0.023 respectively.

Experimental setup	**ROC AUC**
Sliding window static	0.907 ± 0.008
Sliding window resampled	$\mathbf{0.909 \pm 0.008}$

Table 6.1: Results for the sliding window experiment. Mean ROC AUC values are stated along with their standard deviation (standard deviation).

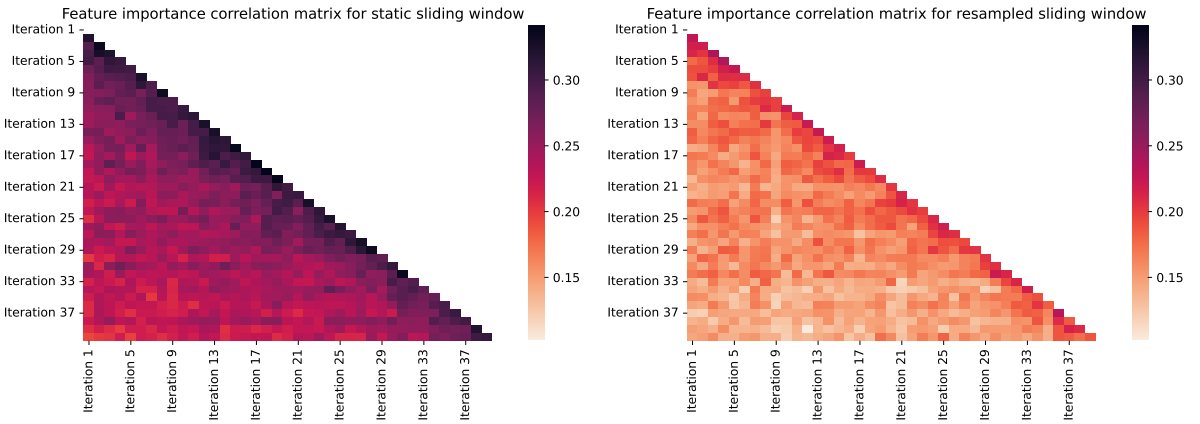


Figure 6.1: Spearman correlation matrices between classifier feature importance rankings of classifiers at different iterations for both sliding window designs. The results are averaged over all classes. The static sliding window matrix has a mean of 0.255 ± 0.028, the resampled sliding window matrix has a mean of 0.162 ± 0.023.

Other than feature importance ranking correlations, we are interested in sample score ranking correlations between the static and resampled sliding window designs. The sample score ranking correlation between the two designs gives insight into how the evaluation of instances is impacted by the different designs. Table 6.2 displays the Spearman correlation between sample scores assigned by (1) a classifier resulting from the static sliding window design and (2) a classifier resulting from the resampled sliding window design. The results are aggregated over all iterations. The table shows an overall moderately strong positive correlation (≥ 0.550) between sample scores. In Appendix B two similar tables are given: one table represents the subset of positive samples (i.e., the relevant ship type), the average correlation is moderately strong positive (≥ 0.652); the other table represents the subset of negative samples, the average correlation is moderately strong positive (≥ 0.516).

Ship type	**Spearman correlation**
Bulk carrier	0.895 ± 0.013
General cargo/multipurpose	0.751 ± 0.036
Oil tanker	0.680 ± 0.037
Container	0.727 ± 0.039
Chemical tanker	0.678 ± 0.040
Fishing vessel	0.812 ± 0.036
Other special activities	0.707 ± 0.035
Tug	0.689 ± 0.042
Offshore supply	0.710 ± 0.038
Ro-Ro cargo	0.627 ± 0.055
Gas carrier	0.550 ± 0.060

Table 6.2: Spearman correlation between sample scores assigned by classifiers which stem from the static and resampled sliding window designs for each ship type. Correlations are averaged over all 40 iterations and are displayed with their standard deviation.

6.2 Expanding window

The results for the expanding window experiment are presented in Figure 6.2 for both the static and resampled designs. The results show in both experimental designs that increasing the size of the train data set (i.e., increasing both the sample and encoding time frames) does not have a marked impact with regard to performance. Yet, it can be observed that increasing the size of the encoding time frame of the test data set does impact performance: the greater the encoding time frame, the greater the performance. The performance differences between the static and resampled designs are marginal. In Figures C.1 and C.2 the individual results for each test encoding size are presented for both expanding window designs.

In addition to model performance, we are interested in investigating how classifiers are learned across iterations (i.e., different sizes of encoding and sample time frames) with respect to feature importance. We again use Spearman's rank correlation coefficient. The correlation matrices for both the static and resampled designs are displayed in Figure 6.3.

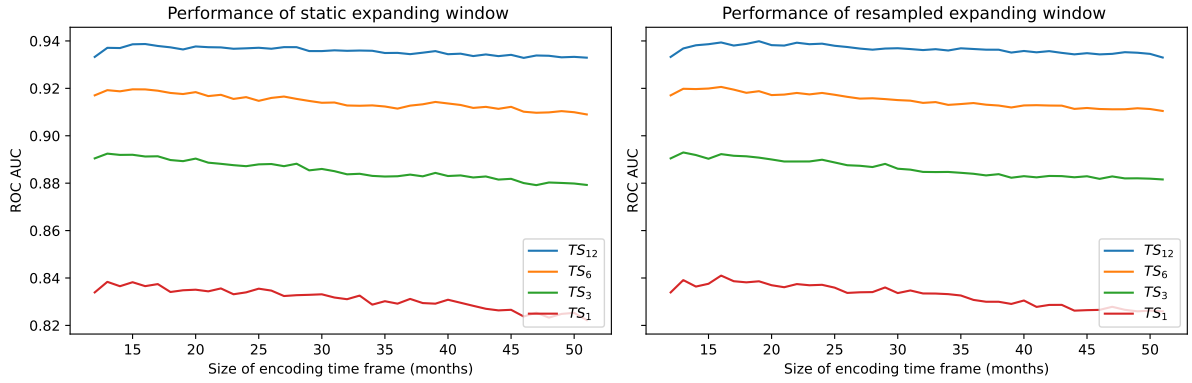


Figure 6.2: Results for both the static and resampled designs of the expanding window. The horizontal axes denote the size of the encoding time frame of the train data set (S_{train}) in months. The vertical axes denote the mean ROC AUC value over all classes. The results are averaged over all 30 repetitions.

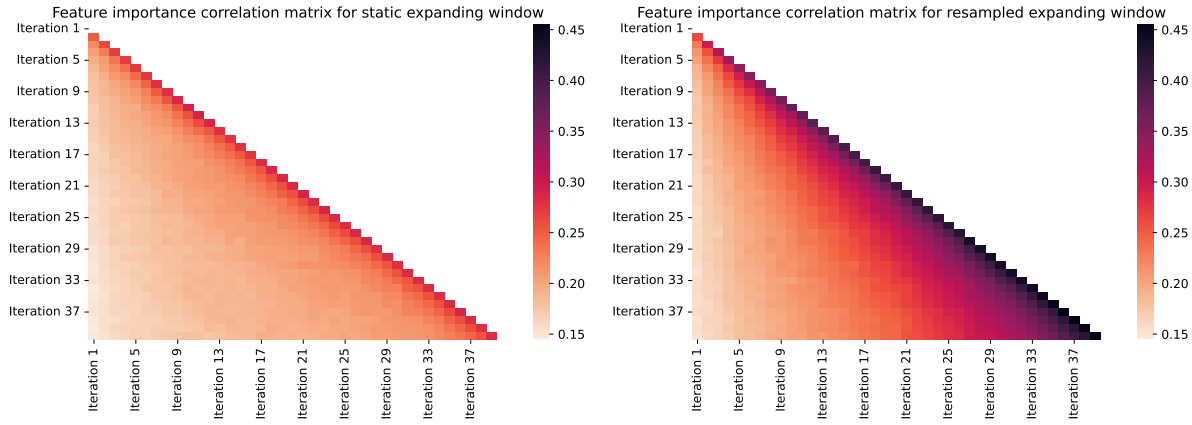


Figure 6.3: Spearman correlation matrices between feature importance rankings of classifiers at different iterations for both expanding window designs. The results are averaged over all classes, where each class is averaged over all 30 repetitions of the experiments. The static expanding window matrix has a mean of 0.202 ± 0.031, the resampled expanding window matrix has a mean of 0.259 ± 0.070.

The average correlation of the resampled expanding window design matrix (0.259 ± 0.070) is greater than the average correlation of the static expanding window design matrix (0.202 ± 0.031). For the resampled expanding window design, the correlation values between later iterations are greater. This can not be observed in the matrix for the static expanding window design.

Alongside feature importance ranking correlation matrices, similar matrices are constructed for the ranking correlation between sample scores assigned by classifiers at different iterations. Furthermore, we record the sample score ranking correlations between classifiers stemming from the static and resampled designs. For all correlations, we use the classified instances of test data set TS_{12}. The sample score ranking correlations are presented in Table 6.3.

The table displays the average of the correlation matrix for each individual ship type in the second and third columns, the matrices are presented in Figures C.3, C.4, C.5 and C.6. These columns show an overall moderately strong positive correlation (≥ 0.596) for both designs. The resampled design shows a greater correlation than the static design for every ship type. The correlation regarding the difference between the static and resampled designs is overall moderately strong positive (≥ 0.624). Tables C.1 and C.2 present the correlations for the subset of positive and negative samples across iterations respectively, both show an overall moderately strong positive correlation (≥ 0.642 and ≥ 0.562).

Ship type	**Static**	**Resampled**	**Static vs resampled**
Bulk carrier	0.853 ± 0.017	0.869 ± 0.023	0.858 ± 0.007
General cargo/multipurpose	0.815 ± 0.017	0.831 ± 0.024	0.820 ± 0.010
Oil tanker	0.695 ± 0.031	0.728 ± 0.044	0.703 ± 0.011
Container	0.742 ± 0.022	0.769 ± 0.035	0.749 ± 0.008
Chemical tanker	0.705 ± 0.026	0.736 ± 0.038	0.713 ± 0.013
Fishing vessel	0.788 ± 0.016	0.801 ± 0.028	0.803 ± 0.011
Other special activities	0.726 ± 0.025	0.743 ± 0.037	0.744 ± 0.009
Tug	0.689 ± 0.047	0.710 ± 0.058	0.723 ± 0.010
Offshore supply	0.705 ± 0.024	0.717 ± 0.036	0.723 ± 0.018
Ro-Ro cargo	0.647 ± 0.032	0.676 ± 0.046	0.665 ± 0.014
Gas carrier	0.596 ± 0.041	0.660 ± 0.063	0.624 ± 0.021

Table 6.3: Spearman correlations of sample scores for the expanding window designs. The second and third column denote the correlations between sample scores assigned by classifiers in different iterations for both the static and resampled designs. Correlations are the average of the resulting correlation matrix (which considers all iterations). The fourth column denotes the sample score correlation between the static and resampled designs, where each correlation is the average of all iterations. All results are aggregated over all 30 repetitions of the experiment and are recorded with their standard deviation.

6.3 Train-test gap

The results for the train-test gap experiment are shown in Figure 6.4 for both the static and resampled designs. The results show that performance decreases as the size of the gap increases. The performance difference between the static and resampled designs is not noteworthy and thus the performance can be considered identical. The decline in performance over time is a most probable indicator of concept drift. This result is on par with the literature consensus. In Figure D.1 the individual results for both the static and resampled designs are plotted which include the standard deviations.

The feature importance ranking correlations between classifiers in different iterations are presented in Figure Figure 6.5 for both train-test gap designs. We are again interested in investigating how classifiers are learned across iterations (i.e., with different gap sizes). The average correlations of both the static and resampled design matrices can be considered low positive and are nearly identical (0.167 ± 0.017 and 0.167 ± 0.022 respectively).

Other correlation matrices are constructed representing the sample score ranking correlations between classifiers which are constructed at different iterations. Furthermore, we record the sample score ranking correlations between classifiers stemming from the static and resampled designs. The resulting correlations are presented in Table 6.4. It displays the average of the correlation matrix for each individual ship type in the second and third columns, the matrices are presented in Figures D.2, D.3, D.4 and D.5. These columns show an overall moderately strong positive correlation (≥ 0.508) for both train-test gap designs. For the difference between the static and resampled designs, the table shows an overall moderately strong positive correlation (≥ 0.578). Tables D.1 and D.2 present the correlations for the subset of positive and negative samples across iterations respectively, both show an overall moderately strong positive correlation (≥ 0.497 and ≥ 0.471 respectively).

Figure 6.4: Results for the train-test gap experiment. The horizontal axis denotes the size of the gap between the sample time frames of the train data set (S_{train}) and the test data set (S_{test}). The vertical axis denotes the mean ROC AUC value over all classes. The results are averaged over all 30 repetitions.

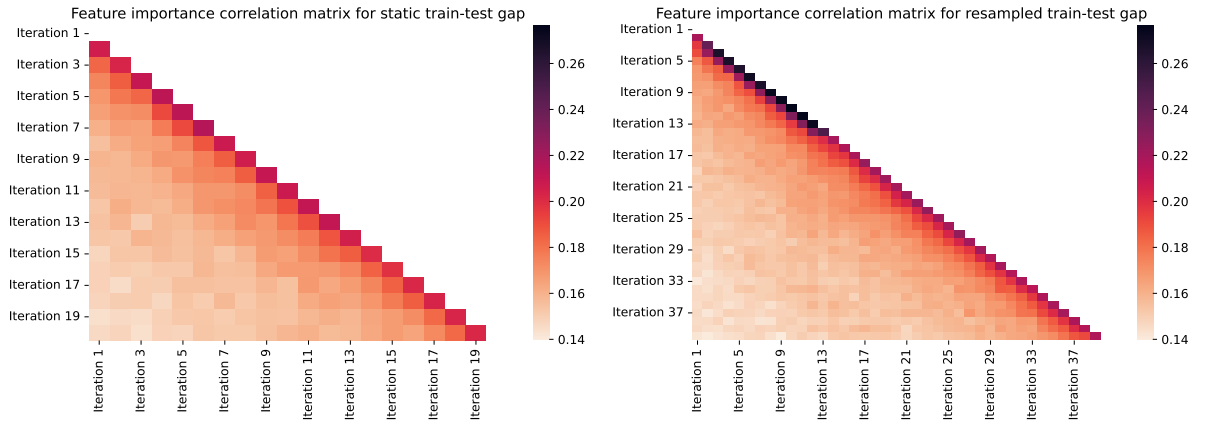


Figure 6.5: Spearman correlation matrices between feature importance rankings of classifiers at different iterations for both train-test gap designs. The results are averaged over all classes, where each class is averaged over all 30 repetitions of the experiments. The static train-test gap matrix has a mean of 0.167 ± 0.017, the resampled train-test gap matrix has a mean of 0.167 ± 0.022.

Ship type	**Static**	**Resampled**	**Static vs resampled**
Bulk carrier	0.816 ± 0.010	0.811 ± 0.011	0.837 ± 0.009
General cargo/multipurpose	0.812 ± 0.011	0.810 ± 0.013	0.836 ± 0.010
Oil tanker	0.641 ± 0.020	0.633 ± 0.024	0.687 ± 0.012
Container	0.740 ± 0.016	0.738 ± 0.019	0.769 ± 0.012
Chemical tanker	0.673 ± 0.018	0.668 ± 0.020	0.713 ± 0.011
Fishing vessel	0.789 ± 0.009	0.793 ± 0.012	0.816 ± 0.011
Other special activities	0.653 ± 0.014	0.653 ± 0.016	0.702 ± 0.012
Tug	0.612 ± 0.020	0.607 ± 0.020	0.688 ± 0.015
Offshore supply	0.678 ± 0.016	0.673 ± 0.017	0.726 ± 0.017
Ro-Ro cargo	0.656 ± 0.020	0.660 ± 0.022	0.695 ± 0.014
Gas carrier	0.516 ± 0.029	0.508 ± 0.035	0.578 ± 0.017

Table 6.4: Spearman correlations of sample scores for the train-test gap design. The second and third column denote the correlations between sample scores assigned by classifiers in different iterations for both the static and resampled designs. Correlations are the average of the resulting correlation matrix (which considers all iterations). The fourth column denotes the sample score correlation between the static and resampled designs, where each correlation is the average of the first 20 iterations (maximum of the static design). All results are aggregated over all 30 repetitions of the experiment and are recorded with their standard deviation.

Chapter 7

Discussion

In this chapter, we elaborate on our findings and their implications. We firstly discuss the technical aspects of our results in Section 7.1. Afterwards, in Section 7.2, we elaborate on the practical implementations of our results which are of relevance to the application domain. We close off this chapter with the limitations of our work (Section 7.3).

7.1 Technical assessment

In this section, we elaborate on the results of each experiment (i.e. design choice) in a structured manner. We discuss the following three aspects for each design choice: the differences in performance; the Spearman correlation matrices with regard to feature importance; the sample score correlations. We inspect the results of the sliding window experiment, expanding window experiment, and train-test gap experiment in Sections 7.1.1, 7.1.2, and 7.1.3 respectively.

7.1.1 Sliding window

In the sliding window experiment, the resampled design shows a marginally better performance than the static design. Meaning that updating the characteristics of the ship movement network does not greatly impact classification performance.

We observe a greater difference between the two designs when we study the feature importance ranking correlation matrices. The matrices show that the overall correlations are greater for the static design than for the resampled design, although the averages of both matrices are considered low positive. Indicating that classifiers in different iterations have a more similar feature importance ranking between each other in the static design than in the resampled design. This can be explained by examining the two factors that impact the feature vector of a ship: (1) the contents of its journey j and (2) the network characteristics of the ship movement network G. The instances in S_{train} contain, for each iteration in the static design, a constant factor (the non-changing G), which does not exist in the resampled design. Ensuring that feature vectors are more similar across iterations in the static design than in the resampled design. The greater similarity between feature vectors, in turn, accounts for more similar feature importance rankings.

Taking into account that there is no noteworthy difference in classification performance when G changes, it could imply that the representativeness of features (i.e., how well ship types are represented by a feature vector) mainly comes from the contents of a journey j (i.e., the visited ports in the ship movement network) regardless of networks characteristics. In other words, network characteristics are necessary for encoding ship journeys, but updating the network adds little information needed for classification of ship type.

The sample score ranking correlations between the static and resampled designs are moderately strong positive for all ship types. Here, the difference in network characteristics is the source of the dissimilarity of features (for the same instances) produced by the static and sliding window designs. To put it differently, the variance between sample scores of both designs is mainly due to the different network characteristics that are considered (other contents in G). However, the existing variance between sample scores is not reflected in performance, since both designs perform approximately identically.

The results of this experiment indicate that if (ship type related) movement through the ship movement network (i.e., contents of ship journeys) evolves over time, the representativeness of feature vectors of ships is also impacted over time. This could be an indication of concept drift, which partly answers the second research question.

7.1.2 Expanding window

In the expanding window experiment, we again observe a negligible difference between the performance of the static and resampled designs. Again suggesting that the changing network characteristics have little impact on performance (and thus feature representativeness). Furthermore, increasing the size of the train data set (i.e., increasing the sample and encoding time frames), has in our scenario no positive impact on classification performance. We do, however, see a great difference in performance between the different encoding time frame sizes of the test data sets. This could be explained by the nature of our feature vectors, which represent ship journeys over time. If ship movement through the ship movement network changes over time, ship movement from earlier periods becomes non-representative of present-day ship movement (i.e., the data becomes outdated). This results in little to no added value to feature representativeness by increasing either the sample time frame or encoding time frame for the train data set. However, when we increase the size of the encoding time frame of the test data set, the percentage of shared time periods between the encoding time frames of the train and test data set becomes greater. This ensures that the added data in the test data set has greater added value than the added data in the train data set.

The feature importance ranking correlation matrices reveal an average low positive correlation between feature importance rankings across iterations for both designs. However, the matrix for the resampled design shows that correlations between later iterations are greater than among other iterations. This behaviour is logical for the resampled design, since both (1) the journeys in the train data set ($J_{S_{train}}$) and (2) the network construction data set (S_G) increase in size with an approximately constant rate (i.e., not every month contains an equal amount of data). This means that there is a decline in the percentage of data added after each iteration. This ensures that the train data sets (which are constructed with $J_{S_{train}}$ and S_G) of different iterations become more similar over time, resulting in more similarity between features. Consequently, the increased similarity between features accounts for more similar feature importance rankings.

In the static design, both the encoding time frame and the sample time frame of S_G do not increase in size but rather shift. This adds variance between the feature vectors of different iterations and explains why the behaviour we observe in the resampled design correlation matrix is not visible in the static design correlation matrix. This is in line with our suggestion that feature similarity across iterations is impacted by the changing network characteristics of G, however, changes in G do not greatly impact classification performance.

The sample score correlations between iterations for each ship type can be considered moderately high for both designs, with the lowest and highest value being 0.596 and 0.869 respectively. The resampled design shows a greater correlation than the static design for every ship type. This is again explained by the higher similarity between features in different iterations in the resampled design than in the static design. The variance between sample scores stems from the following three factors: (1) the additional ships in S_{train}, due to the increased sample time frame; (2) the difference in the contents of the journeys in the train data set ($J_{S_{train}}$), due to the increased encoding time frame and sample time frame of S_{train}; and (3) the change in network characteristics of G between iterations, due to extension (in the resampled design) or shift (in the static design) of both the sample and encoding time frame of S_G. However, the added variance across iterations by these three factors has little impact on feature representativeness, as is reflected by the marginal differences in performance across iterations. The correlation between sample score rankings of the static and resampled designs is overall moderately strong positive. Such as with the sliding window experiment, the variance between sample score rankings of the static and resampled designs is not reflected in performance differences.

The results of the expanding window experiment relate to the first research question, regarding the impact of the size of the train data set on performance. The results indicate that increasing the train data set does not necessarily ensure a (positive) impact on classification performance. However, the nature of our features (i.e., trajectory-based) plays an important role when assessing this aspect. We must consider that ship behaviour may change over time and that past data may or may not be of relevance when constructing such features.

7.1.3 Train-test gap

In the train-test gap experiment, we can observe a declining classification performance as the size of the gap increases, which is a probable indication of concept drift. The performance difference between the static and resampled designs is again negligible. We suspect that evolving movement behaviour through the ship movement network is the biggest contributor to the performance decline. Since other results show that training on older network characteristics does not greatly impact performance. As the gap increases, so too do the movement differences between time periods and, in turn, feature representations. It must be noted that the decline in performance is not very great, but is noticeable nonetheless.

The correlations between feature importance rankings displayed in the correlation matrices for both designs are overall low positive, meaning that different iterations have very different feature importance rankings. This is logical, considering that in every iteration both the ships in S_{train} and the ships in S_G are resampled for both designs; resulting in little similarity between features of different iterations.

The sample score correlations across iterations are overall moderately strong positive, with the lowest and highest value being 0.508 and 0.816 respectively. As stated earlier, performance declines as the gap increases, however, this decline in performance is not very great; which helps to explain as to why sample score correlations remain moderately high. The sample score ranking correlation between the static and resampled designs is overall moderately strong positive. The negligible difference in performance does not reflect this existing variance, which is in consensus with both results of the sliding window and expanding window experiments. Since we find that differences in network characteristics do not impact performance in any of our experiments, there is strong evidence in favour of the hypothesis that the representativeness of features mostly comes from the contents of ship journeys rather than the contents of G.

This experiment helps to answer the second research question, regarding the presence of concept drift. Results indicate that in our scenario, although not drastically, concept drift most likely impacts classification performance over time. This is probably due to the maritime movement in general, which is prone to change over time.

7.2 Relevance to the domain

In this section, we elaborate on the implications of our findings for domain-related applications. We also comment on the relevance of time awareness when feature engineering similar features or validating models.

The results of the experiments show that the considered time-aware design concepts (encoding time frame and sample time frame) can impact model performance. In our case, results indicate that concept drift is most likely affecting classification performance. In a deployment scenario, these effects could, for example, be mitigated with proper strategies such as periodic retraining. When retraining, it should be considered that changing the ship movement network does not greatly impact performance, thus implying that a choice must be made regarding reusing, replacing or updating the ship movement network. Additionally, results show that increasing the size of the train data set (i.e., adding more ships or more data per ship) has an insignificant impact on classification performance. As a result, the size of the train data set can be reduced to an optimum; implying a reduction in computing cost without sacrificing classification performance. This has beneficial consequences for training time and resources. Regardless of what the case-specific impact is on model performance, it is recommended to have a suitable strategy to mitigate negative effects or exploit the newly obtained knowledge.

For all design choices, overall feature importance ranking correlations are low. Meaning that predictions regarding model behaviour, when training a new classifier, are not well grounded. In other words, when training a new classifier, regardless of using an updated ship movement network, we can not confidentially estimate which features are important to the new classifier and thus the intrinsic behaviour (i.e., assigning classification scores) of the classifier may change drastically. In our case, individual samples are still ranked similarly across iterations (i.e., in the expanding window and train-test gap experiments) and between the static and resampled designs (i.e., in the sliding window experiment). Moreover, model performance is almost not impacted by the intrinsic changes in behaviour. However, this change in intrinsic classifier behaviour may result in implications on (possible) fairness constraints. The possible effect on intrinsic behaviour should, therefore, be considered when designing and validating machine learning models.

Looking past the case-specific consequences of our results, the experimental setup and results of this research provide an example of how time-aware design concepts can be researched. Adjustments or case-specific alternations can (and should) be made when copying the approach used in this study. We show that machine learning models should be constructed and validated in a time-aware manner when relevant, by looking both at classifier performance and intrinsic classifier behaviour.

7.3 Limitations

As with all research, the findings of our study are limited by multiple aspects which we elaborate upon in this section. We start by discussing the limitations of the results by not isolating the sample time frame and encoding time frame. Then, we describe the limitations regarding feature engineering, feature importance and performance metric. Lastly, we comment on the generalisability of our findings.

In this thesis, we focus on the temporal aspect when constructing trajectory-based features. However, we do not explicitly compare the importance of the sample time frame and encoding time frame. The sample time frame increases the train data set by extending the number of instances (ships). The encoding time frame extends the number of considered trips (i.e., increasing the size of j) for a ship in S_{train}, thus increasing the amount of data per ship. We do not isolate either the encoding time frame nor the sample time frame. Therefore, we can not make any remarks about which of these aspects has more impact on classification performance.

Another aspect that limits the findings of this thesis is that we only examine one feature engineering method. The scope of this thesis does not allow for implementing and examining multiple feature engineering methods. We remark that this limitation also holds for the machine learning algorithm. Furthermore, we analyse feature importance only in one manner. This limits the information about intrinsic changes that we can obtain by comparing the feature importance of different classifiers. The performance of the classification is also measured using only one metric. There exist multiple metrics which could produce distinct changes across experimental setups.

The final limitation of this thesis is regarding the generalisability of our findings. This is mainly due to the fact that only one (type of) data set is used for the experiments. We can not state that the results of this thesis transfer to other problem settings which deal with trajectory-based features or mobility networks, such as trucks or aircrafts.

Chapter 8

Conclusion

In machine learning, feature engineering is a prevalent requirement for a plethora of problems, one of which being that of classification. Within the context of the inspections domain, classification problems often translate to targeting real-world non-compliance of laws and regulations. Therefore, investigating the adequacy of feature engineering practices is of paramount importance to the ILT.

Motivated by the use case of the ILT and the phenomenon of concept drift which is typical of real-world applications, in this thesis, we investigated different temporal design choices with respect to the feature engineering processes used to model ship behaviour. We were interested in assessing how these different choices impacted the downstream task of classification, rather than considering the classification itself as the object of study. Specifically, we considered the temporal feature engineering processes under a ship movement network framework.

We found that increasing the size of the data set (i.e, the number of instances (ships) and the amount of data per ship) used for training the classifier does not have a great impact on classification performance in our case. However, increasing the encoding time frame (impacting the amount of data per ship) for instances in the test data set proved to have a positive impact on performance. We speculate that ship type related behaviour (i.e., ship trajectories through the ship movement network) gradually changes over time, thus incorporating time periods in the train data set which are not incorporated in the test data set has no added value for classification performance.

Moreover, results showed that classification performance declines over time when a classifier is deployed using a train-once method (i.e., no retraining). Classification performance declines gradually as the time period between the construction of the train data set and test data set (used for evaluation) increases. We estimate that feature representativeness (i.e., how well ship types are represented by features) mostly comes from the contents of a ship trajectory (i.e., visited ports). Whereas the ship movement network characteristics are a necessity for encoding ship journeys but updating them has little impact on performance. The decline in performance over time implies that the representativeness of features for their corresponding ship type declines, which is an indicator that ship movement behaviour through the ship movement network changes gradually over time.

All in all, our results indicate that the applied temporal design choices can have an impact on both classification performance and the intrinsic behaviour of classifiers. Using the context of maritime movement, we show how these design choices can be implemented and evaluated. Despite the limitations in our experimental design (see Chapter 7.3), our findings enable the ILT to make better informed decisions with respect to model design, evaluation, and, ultimately, deployment. Even though we can not necessarily state how these design choices will affect similar use cases, we can state that they should be investigated.

The results of this thesis provide an opportunity for further research. The generalisability of our findings to other mobility networks could be examined, investigating how these temporal design choices impact other forms of transportation or mobility. Alternatively, it could be examined how different feature engineering methods and machine learning algorithms are affected by these temporal design choices. Lastly, we have identified that these temporal aspects can impact performance. However, we only make suggestions, based on the results of our experiments, on how to tackle the negative effects. Other research could investigate which strategies are suitable for mitigating potential negative effects, this would be especially relevant for trajectory-based features.

Bibliography

Celik, B., & Vanschoren, J. (2021). Adaptation strategies for automated machine learning on evolving data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *43*(9), 3067–3078. https://doi.org/10.1109/tpami.2021.3062900

de Bruin, G. J., Barata, A. P., van den Herik, H. J., Takes, F. W., & Veenman, C. J. (2022). Fair automated assessment of noncompliance in cargo ship networks. *EPJ Data Science*, *11*(1), 1–19. https://doi.org/10.1140/epjds/s13688-022-00326-w

EMSA. (2023). *Emsa/thetis-eu* [Accessed: 01-01-2023]. https://portal.emsa.europa.eu/web/thetis-eu

Environment, H., & Inspectorate, T. (2023). *About the ilt* [Accessed: 15-05-2023]. https://english.ilent.nl/about-the-ilt

Grover, A., & Leskovec, J. (2016). Node2vec: Scalable feature learning for networks. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 855–864. https://doi.org/10.1145/2939672.2939754

Han, P., Wang, J., Yao, D., Shang, S., & Zhang, X. (2021). A graph-based approach for trajectory similarity computation in spatial networks. *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 556–564. https://doi.org/10.1145/3447548.3467337

Li, X., Zhao, K., Cong, G., Jensen, C. S., & Wei, W. (2018). Deep representation learning for trajectory similarity computation. *2018 IEEE 34th International Conference on Data Engineering (ICDE)*, 617–628. https://doi.org/10.1109/ICDE.2018.00062

Li, Z., Xu, M., & Shi, Y. (2015). Centrality in global shipping network basing on worldwide shipping areas. *GeoJournal*, *80*(1), 47–60. https://doi.org/10.1007/S10708-014-9524-3

Lommen, Y. (2023). *Thesis yven lommen* [Accessed: 30-06-2023]. https://github.com/yvenlommen/Thesis-Yven-Lommen

Pereira Barata, A., Takes, F., Herik, H., & Veenman, C. (2021). The eXPose approach to crosslier detection. *2020 25th International Conference on Pattern Recognition (ICPR)*, 2312–2319. https://doi.org/10.1109/ICPR48806.2021.9412644

Yao, D., Cong, G., Zhang, C., & Bi, J. (2019). Computing trajectory similarity in linear time: A generic seed-guided neural metric learning approach. *2019 IEEE 35th International Conference on Data Engineering (ICDE)*, 1358–1369. https://doi.org/10.1109/ICDE.2019.00123

Žliobaitė, I., Pechenizkiy, M., & Gama, J. (2016). An overview of concept drift applications. In *Big data analysis: New algorithms for a new society* (pp. 91–114). https://doi.org/10.1007/978-3-319-26989-4_4

Appendix A

Data preprocessing

In order to make the port calls data set suitable for our needs, some preprocessing steps are taken. Starting with a flaw in the data that needs a suitable solution. As stated in Section 4.2, one of the features makes use of the travel times of trips from a ship. The set of all ports in the ship movement network (G) is denoted by the set of all nodes V, a port is denoted by $v \in V$. In order to derive the travel time of a ship from port v_i to port v_{i+1}, the Departure Time attribute at v_i is used and the Arrival Time attribute at v_{i+1} is used. However, in some cases a *'negative trip'* occurs, meaning a trip that has a negative travel time. This could occur when the recorded arrival time at v_{i+1} is earlier than the departure time at v_i. This can obviously never occur in the real world and we must assume that a mistake is made at the construction of this data set. Nevertheless, this problem needs a solution and so it it measured how many ships possess at least one negative trip. Approximately 10 000 out of the roughly 36 000 ships contain at least one negative trip. Removing all of these ships would mean a great loss of data and thus the percentage of negative trips for each ship is examined. Meaning the number of trips that is observed to have a negative duration divided by the number of total trips of a ship. By doing this for every ship with at least one negative trip we are able to determine the elbow value of the negative trip proportion. The elbow value is in this scenario the *greatest* value for the percentage of negative trips of a ship for which the added proportion to the total number of ships containing one negative trip is greater than the added proportion to percentage of negative trips of a ship. The elbow value is depicted in Figure A.1. Using the elbow value ($\sim$9,84%), we keep ships that have less than 9,84% of their trips with a negative duration in the data set. Ships that do not meet this criteria are discarded from the data set. From the figure it can be observed that in total 93% from the ships are kept that at least had one negative trip. Overall 715 ships are discarded which means a loss of $\sim$2% of all the ships in our data set.

The second preprocessing step that is needed is due to sparsity of data in a particular period within the data. The data set contains data from the time period 2014 - 2020, meaning that we have seven years worth of data at our disposal. In our experiments we make use of time periods of one month (Chapter 5). With a data set of seven years, this means that we have 84 time periods available. However, two months specifically (November and December 2019) do not contain enough data to be used. It is therefore decided that these two months are neglected.

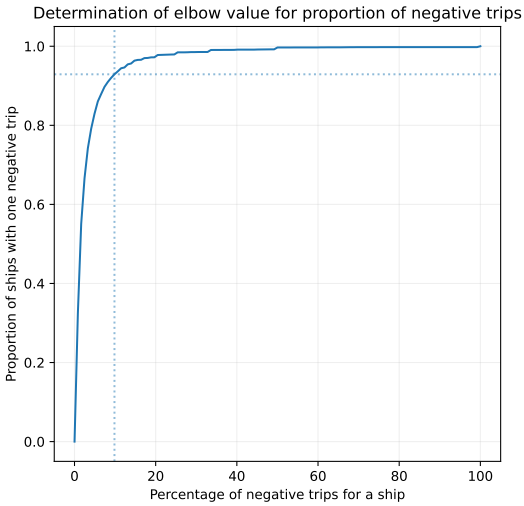


Figure A.1: Elbow value determination for the threshold for the percentage of negative trips a ship may contain. The vertical dashed line represents the value for the elbow value (∼9,84) on the horizontal axis and the horizontal dashed dashed line represents the elbow value (∼0.93) on the vertical axis.

The last preprocessing step that is taken is the determination of the number of classes for the classification. Recall that the classification target is the ship type, for example a ship type can be 'Container'. Since some ship types occur so little in the data set we again make use of the elbow value in order to determine the classes which are used for classification. In this scenario, we prefer the ship types with most occurrences and thus have the greatest added value to the total proportion of ships. Using the elbow value we settle on the eleven most occurring classes to use for classification. These classes are (from most occurring to least occurring): Bulk carrier, General cargo/multipurpose, Oil tanker, Container, Chemical tanker, Fishing vessel, Other special activities, Tug, Offshore supply, Ro-Ro cargo and Gas carrier. All other classes in the data set are relabelled to 'Other' and are still used for training.

Appendix B

Additional tables sliding window experiment

Ship type	**Spearman correlation**
Bulk carrier	0.652 ± 0.051
General cargo/multipurpose	0.877 ± 0.022
Oil tanker	0.688 ± 0.068
Container	0.742 ± 0.054
Chemical tanker	0.764 ± 0.050
Fishing vessel	0.864 ± 0.060
Other special activities	0.698 ± 0.099
Tug	0.718 ± 0.076
Offshore supply	0.853 ± 0.061
Ro-Ro cargo	0.818 ± 0.064
Gas carrier	0.756 ± 0.080

Table B.1: Spearman correlation between sample scores assigned by classifiers which stem from the static and resampled sliding window designs for each ship type, only instances that have a positive class label (i.e., the relevant ship type) are considered. Correlations are averaged over all 40 iterations and are displayed with their standard deviation.

Ship type	**Spearman correlation**
Bulk carrier	0.843 ± 0.020
General cargo/multipurpose	0.633 ± 0.048
Oil tanker	0.630 ± 0.046
Container	0.622 ± 0.057
Chemical tanker	0.621 ± 0.046
Fishing vessel	0.792 ± 0.041
Other special activities	0.688 ± 0.037
Tug	0.676 ± 0.044
Offshore supply	0.683 ± 0.042
Ro-Ro cargo	0.585 ± 0.060
Gas carrier	0.516 ± 0.068

Table B.2: Spearman correlation between sample scores assigned by classifiers which stem from the static and resampled sliding window designs for each ship type, only instances that have a negative class label (i.e., not the relevant ship type) are considered. Correlations are averaged over all 40 iterations and are displayed with their standard deviation.

Appendix C

Additional figures expanding window experiment

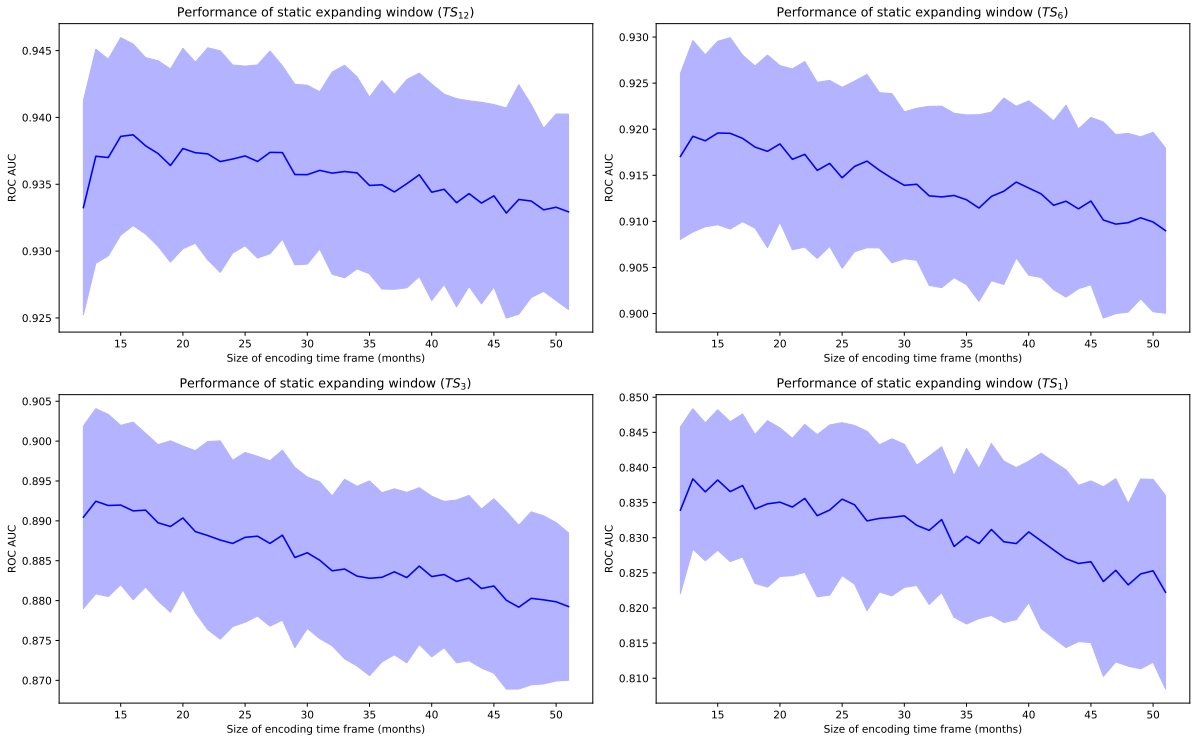


Figure C.1: Plots for the static expanding window design for each encoding time frame size of the test data set. The horizontal axes denote the size of the encoding time frame of the train data set in months. The vertical axes denote the mean ROC AUC value over all classes. Results are averaged over all 30 repetitions and are plotted with their standard deviation.

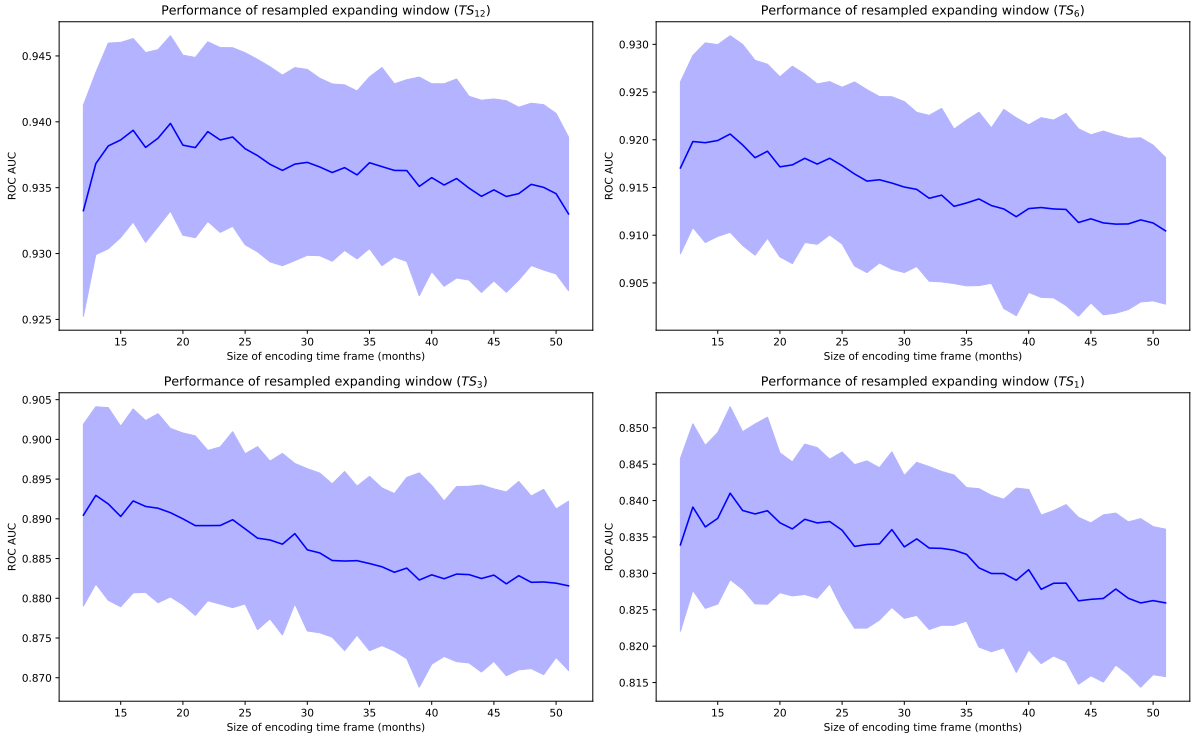
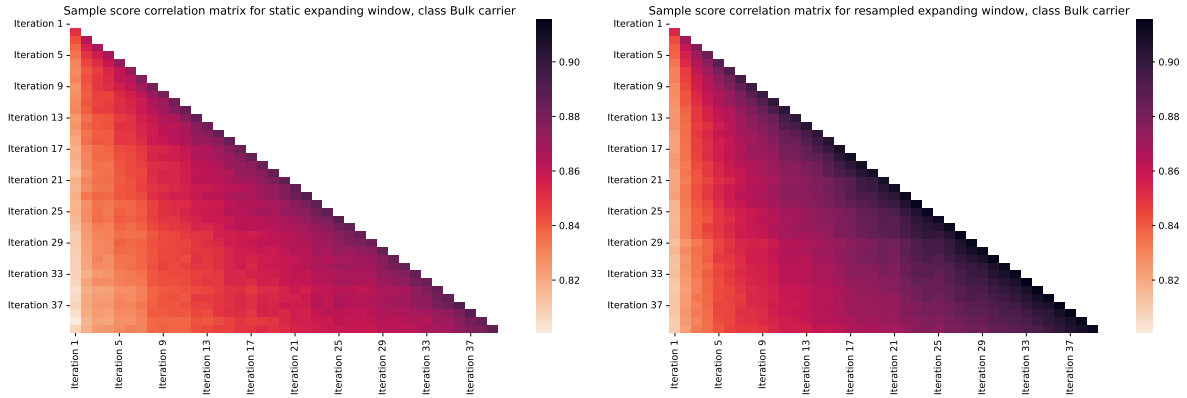


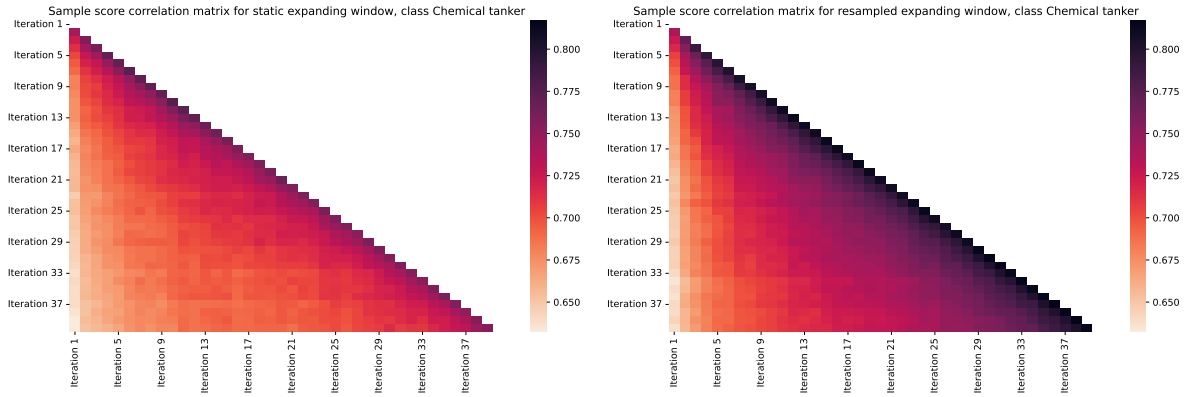
Figure C.2: Plots for the resampled expanding window design for each encoding time frame size of the test data set. The horizontal axes denote the size of the encoding time frame of the train data set in months. The vertical axes denote the mean ROC AUC value over all classes. Results are averaged over all 30 repetitions and are plotted with their standard deviation.

Ship type	**Static expanding window** (r)	**Resampled expanding window** (r)
Bulk carrier	0.715 ± 0.038	0.758 ± 0.055
General cargo/multipurpose	0.821 ± 0.017	0.851 ± 0.022
Oil tanker	0.757 ± 0.032	0.791 ± 0.041
Container	0.755 ± 0.031	0.774 ± 0.038
Chemical tanker	0.754 ± 0.026	0.767 ± 0.034
Fishing vessel	0.781 ± 0.036	0.807 ± 0.043
Other special activities	0.642 ± 0.041	0.670 ± 0.047
Tug	0.642 ± 0.064	0.661 ± 0.077
Offshore supply	0.824 ± 0.029	0.836 ± 0.034
Ro-Ro cargo	0.766 ± 0.034	0.787 ± 0.041
Gas carrier	0.719 ± 0.039	0.756 ± 0.048

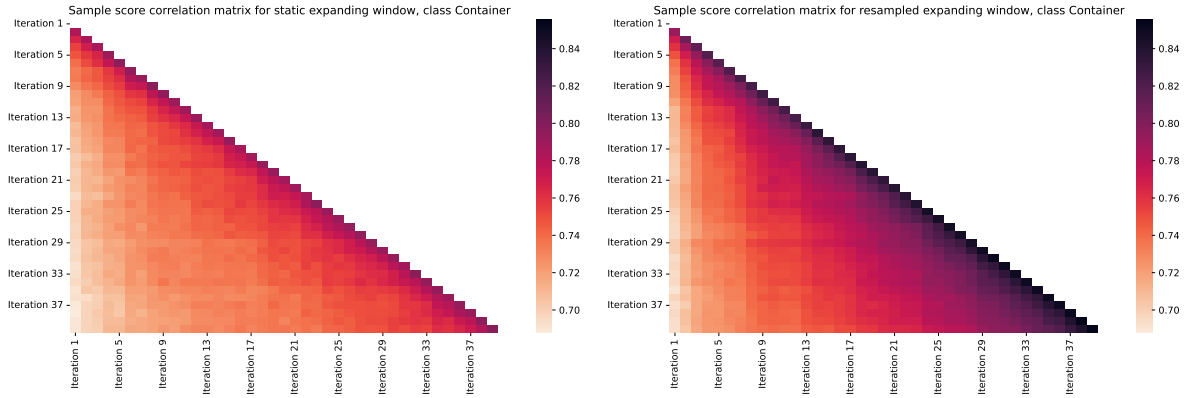
Table C.1: Spearman correlation between sample scores assigned by classifiers in different iterations of the expanding window experiments, only instances that have a positive class label are considered. Correlations are the average of the resulting correlation matrix (which considers all iterations) and are aggregated over all 30 repetitions of the experiment.



(a) Bulk carrier

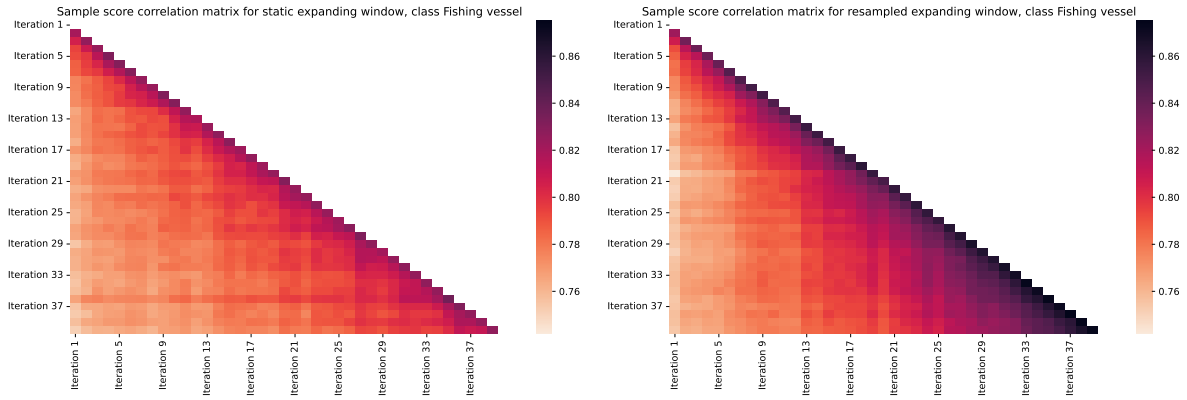


(b) Chemical tanker

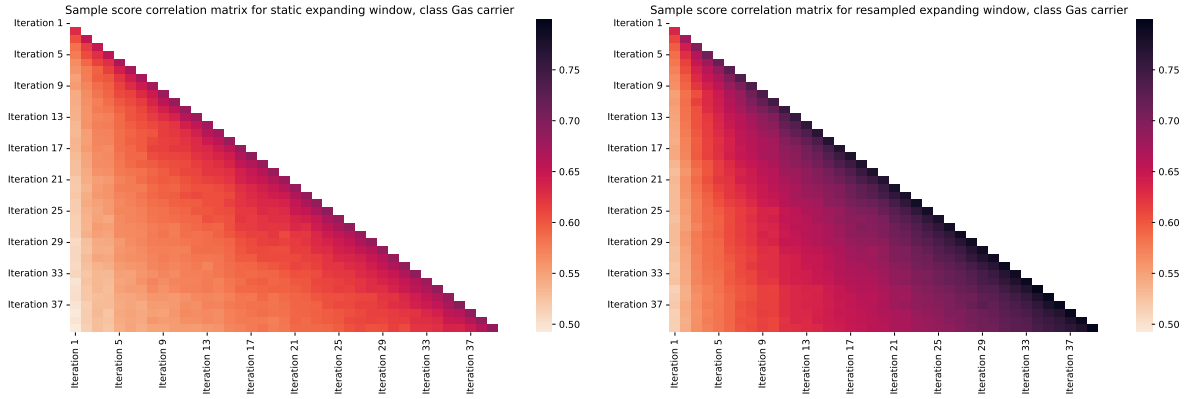


(c) Container

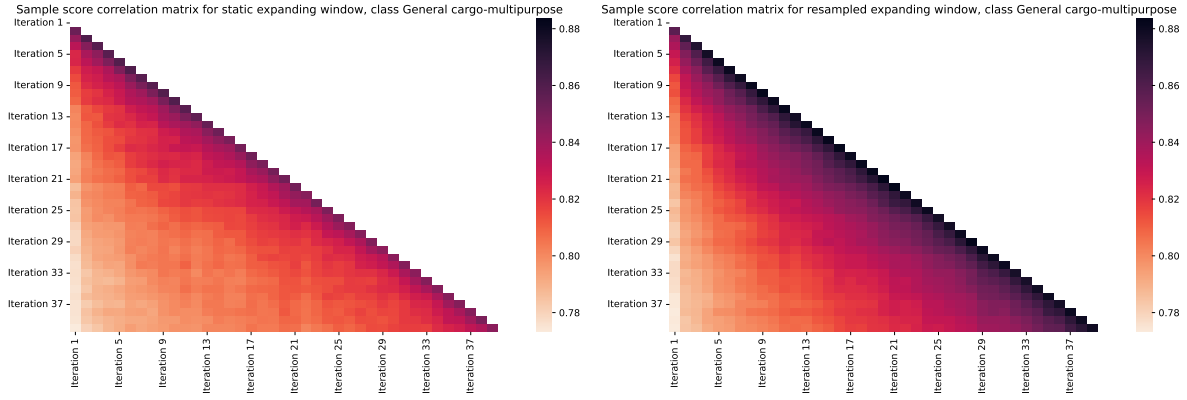
Figure C.3: Correlation matrices (per ship type) which represent the Spearman correlation between sample scores assigned by classifiers at different iterations for both expanding window designs. The results are averaged over 30 repetitions of the experiment.



(a) Fishing vessel

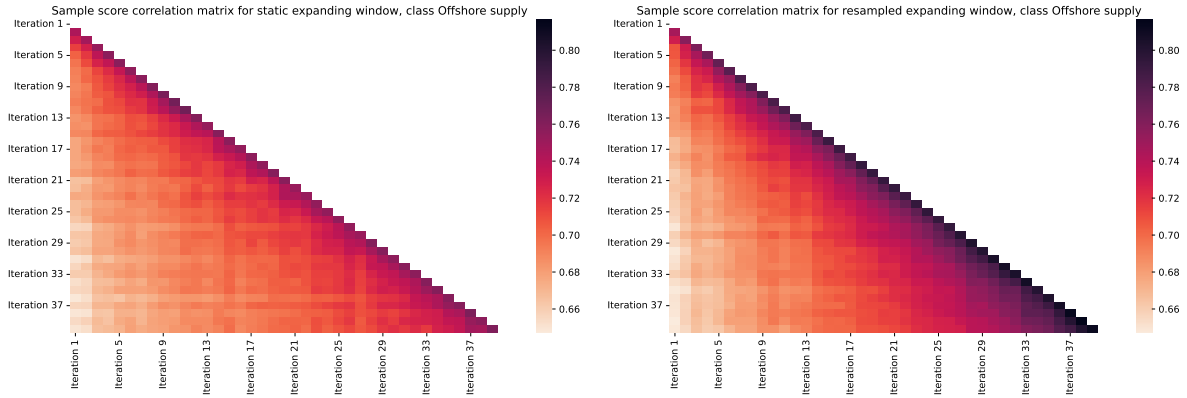


(b) Gas carrier

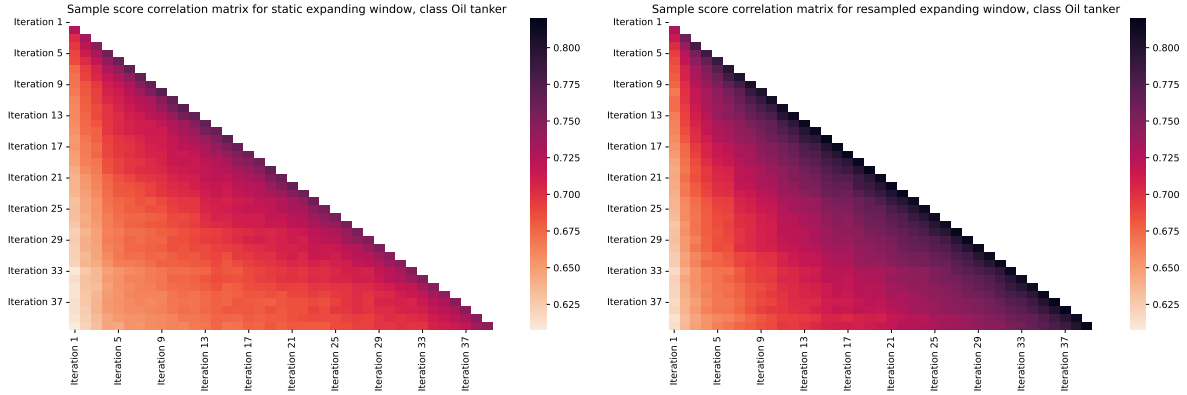


(c) General cargo-multipurpose

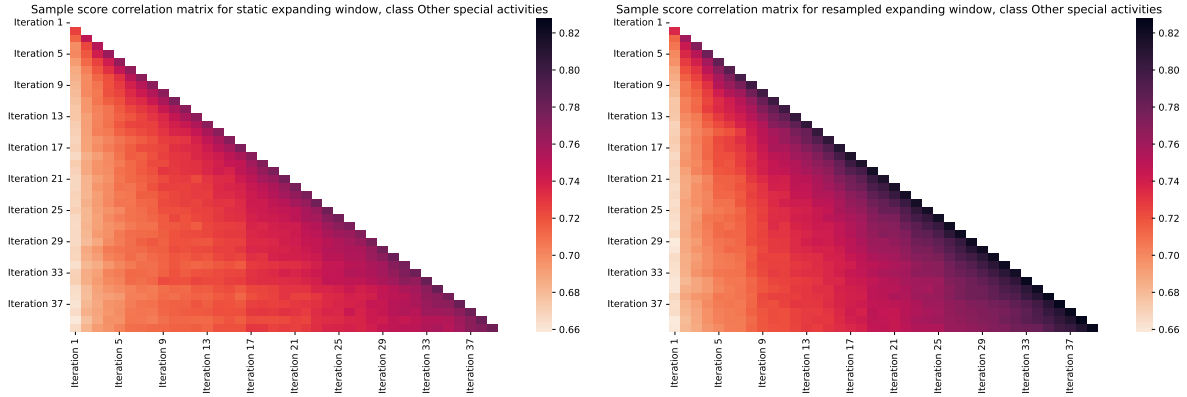
Figure C.4: Correlation matrices (per ship type) which represent the Spearman correlation between sample scores assigned by classifiers at different iterations for both expanding window designs. The results are averaged over 30 repetitions of the experiment.



(a) Offshore supply

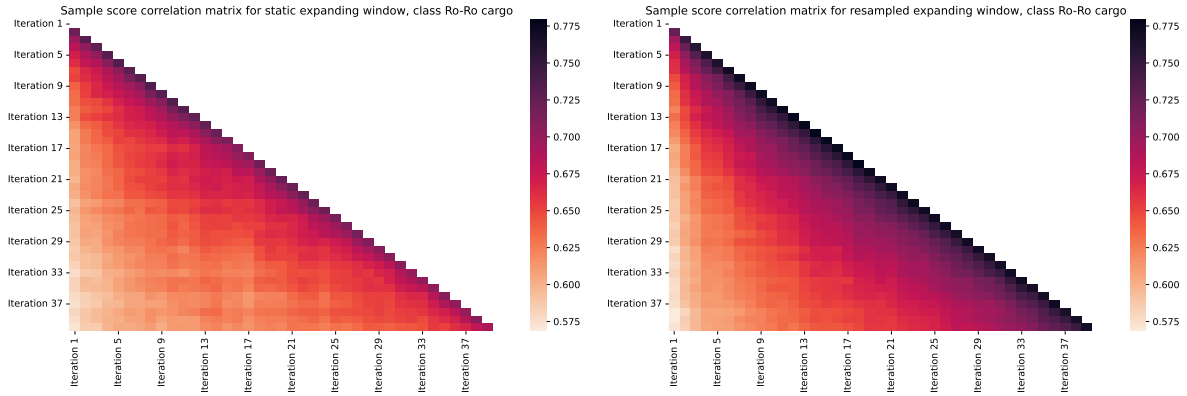


(b) Oil tanker

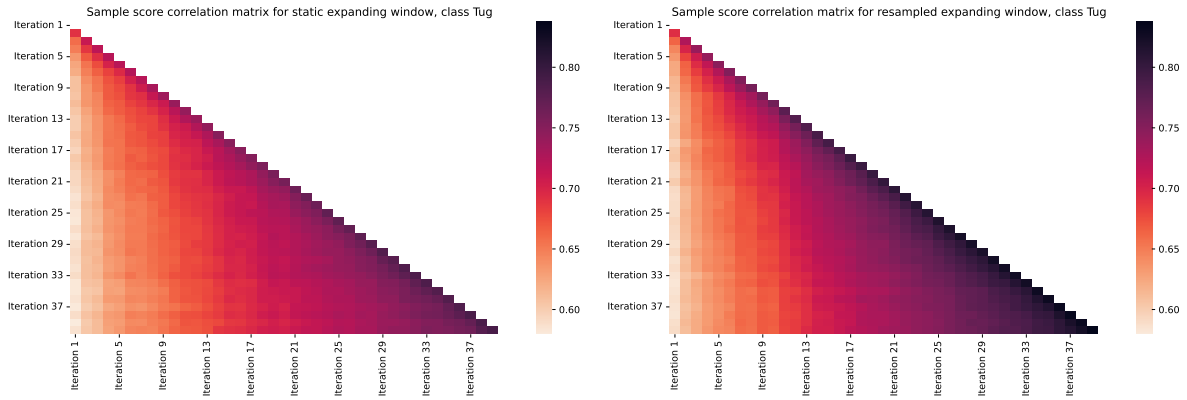


(c) Other special activities

Figure C.5: Correlation matrices (per ship type) which represent the Spearman correlation between sample scores assigned by classifiers at different iterations for both expanding window designs. The results are averaged over 30 repetitions of the experiment.



(a) Ro-Ro cargo



(b) Tug

Figure C.6: Correlation matrices (per ship type) which represent the Spearman correlation between sample scores assigned by classifiers at different iterations for both expanding window designs. The results are averaged over 30 repetitions of the experiment.

Ship type	**Static expanding window** (r)	**Resampled expanding window** (r)
Bulk carrier	0.797 ± 0.024	0.819 ± 0.031
General cargo/multipurpose	0.627 ± 0.033	0.655 ± 0.049
Oil tanker	0.647 ± 0.036	0.684 ± 0.051
Container	0.631 ± 0.032	0.670 ± 0.050
Chemical tanker	0.633 ± 0.032	0.671 ± 0.048
Fishing vessel	0.773 ± 0.018	0.786 ± 0.030
Other special activities	0.705 ± 0.027	0.722 ± 0.039
Tug	0.677 ± 0.049	0.700 ± 0.060
Offshore supply	0.674 ± 0.026	0.687 ± 0.040
Ro-Ro cargo	0.599 ± 0.036	0.632 ± 0.052
Gas carrier	0.562 ± 0.044	0.631 ± 0.068

Table C.2: Spearman correlation between sample scores assigned by classifiers in different iterations of the expanding window experiments, only instances that have a negative class label are considered. Correlations are the average of the resulting correlation matrix (which considers all iterations) and are aggregated over all 30 repetitions of the experiment.

Appendix D

Additional figures train-test gap experiment

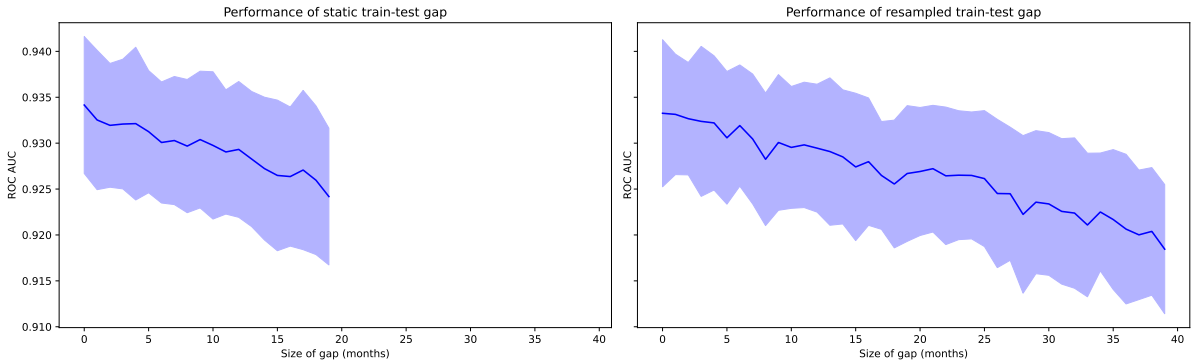
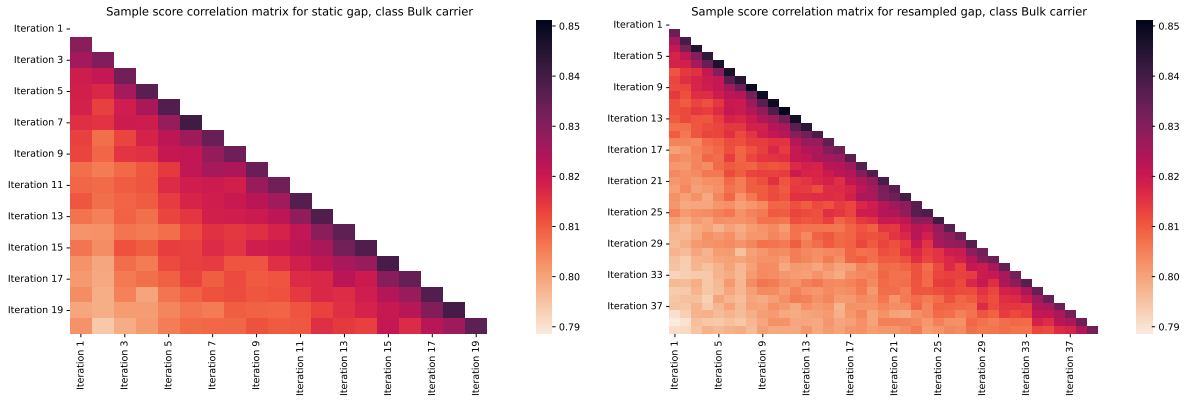
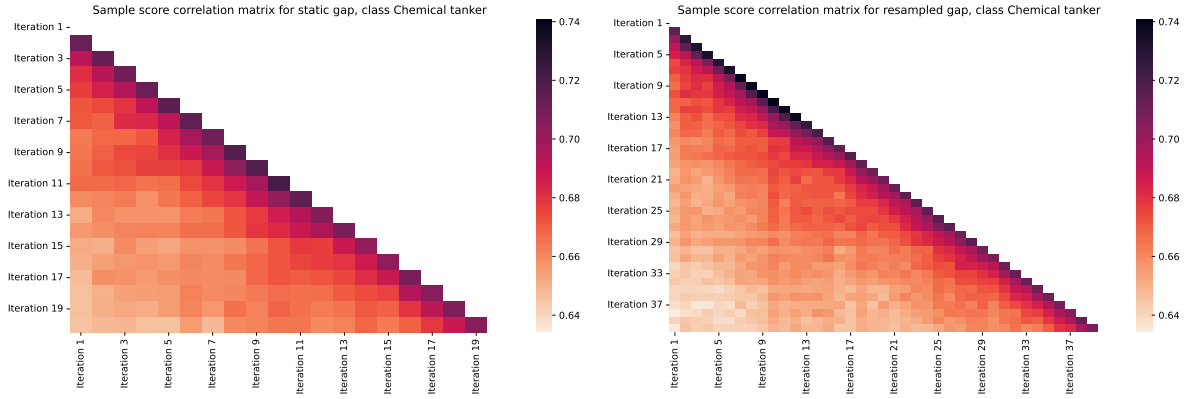


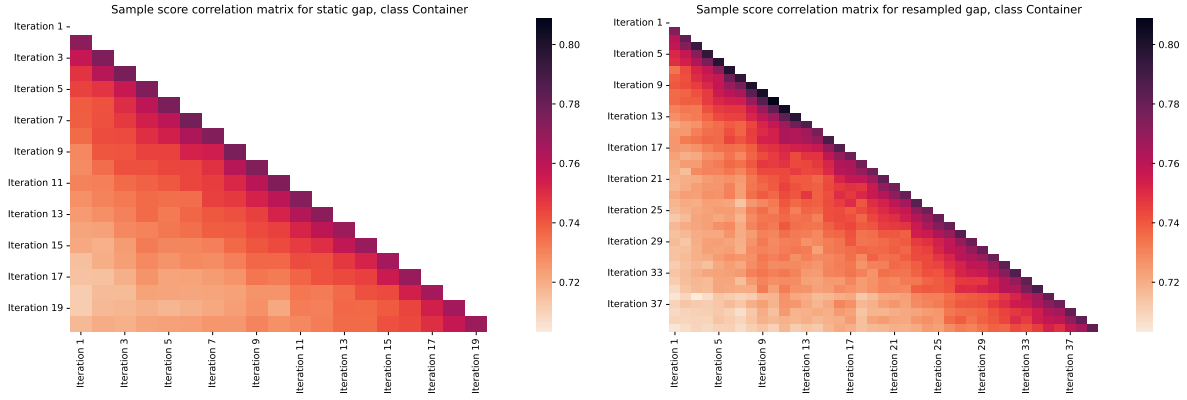
Figure D.1: Plots for both the static and resampled train-test gap design experiments. The horizontal axes denote the size of the gap between the sample time frame of the train data set (S_{train}) and test data set (S_{test}) in months. The vertical axes denote the mean ROC AUC value over all classes. Results are averaged over all 30 repetitions and are plotted with their standard deviation.



(a) Bulk carrier

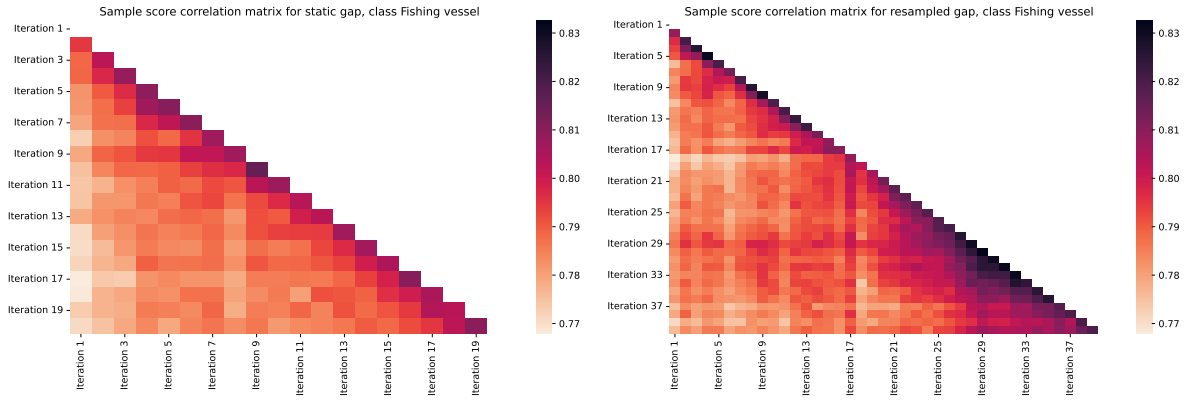


(b) Chemical tanker

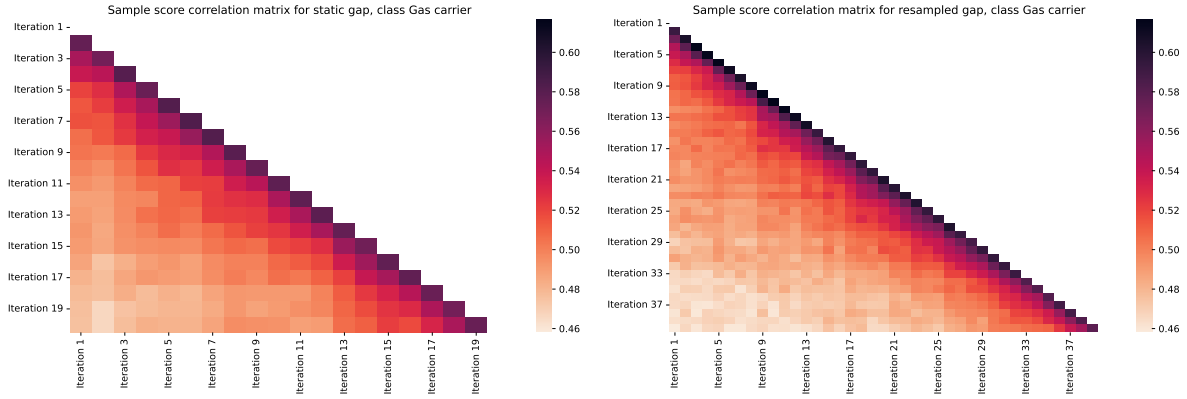


(c) Container

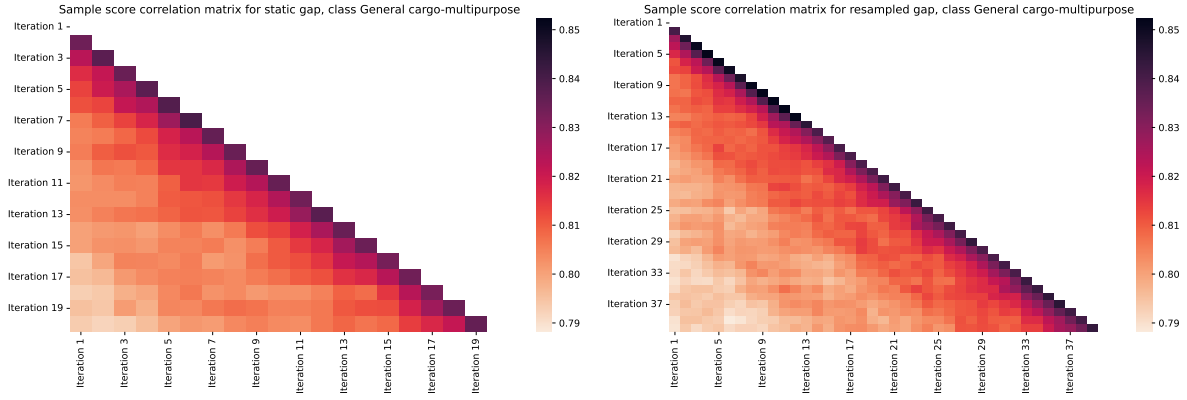
Figure D.2: Correlation matrices (per ship type) which represent the Spearman correlation between sample scores assigned by classifiers at different iterations for both train-test gap designs. The results are averaged over 30 repetitions of the experiment.



(a) Fishing vessel

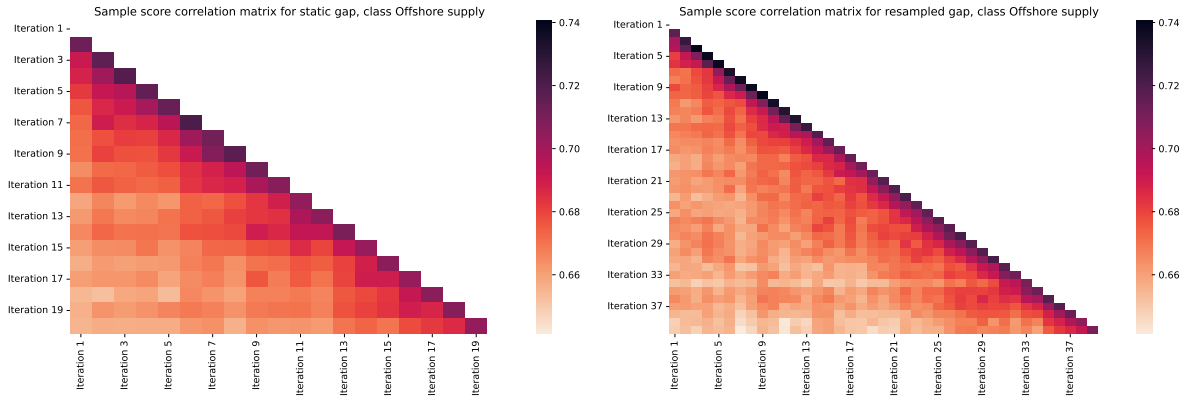


(b) Gas carrier

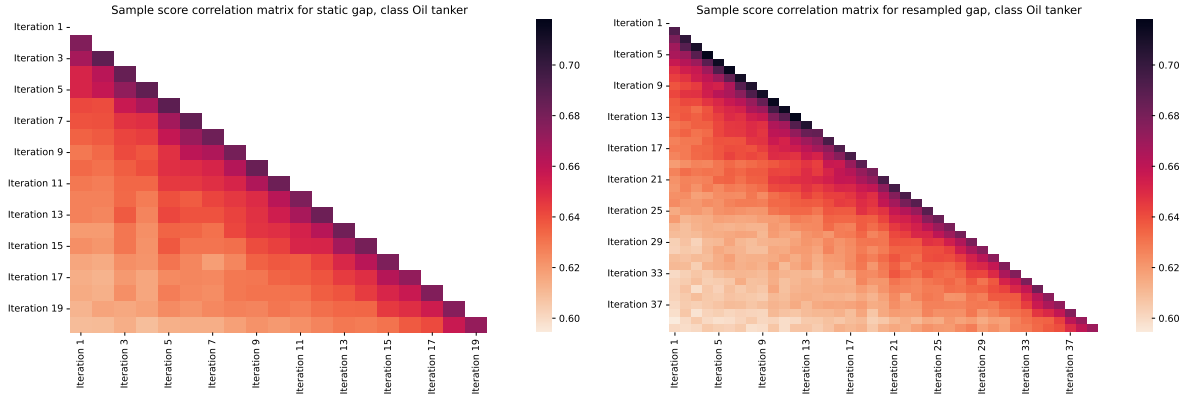


(c) General cargo-multipurpose

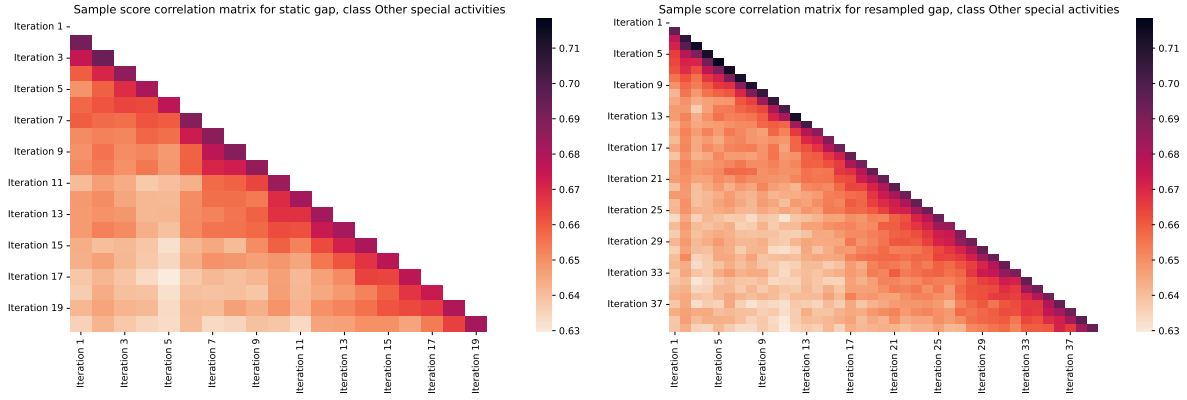
Figure D.3: Correlation matrices (per ship type) which represent the Spearman correlation between sample scores assigned by classifiers at different iterations for both train-test gap designs. The results are averaged over 30 repetitions of the experiment.



(a) Offshore supply

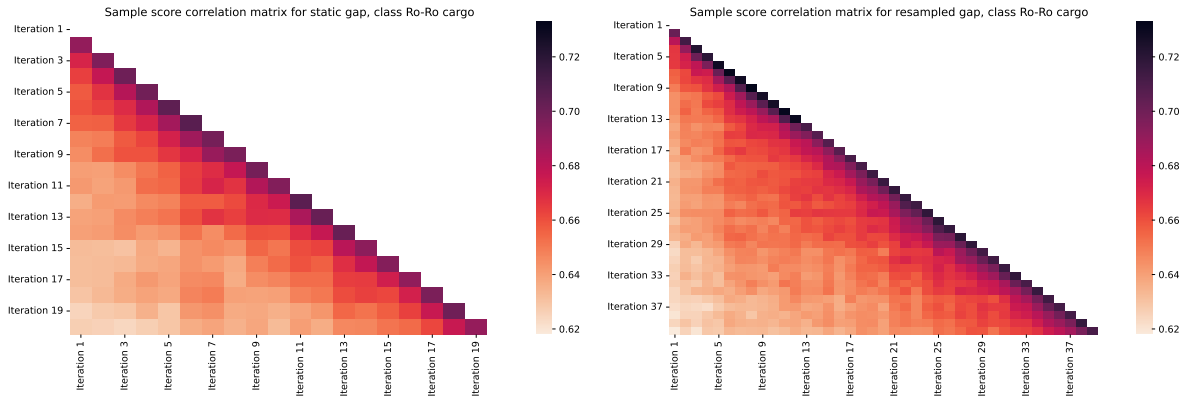


(b) Oil tanker

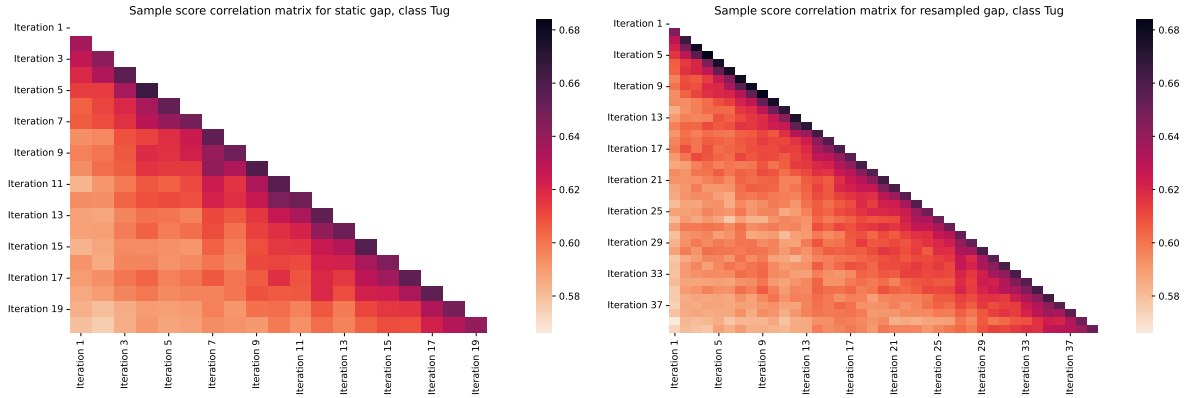


(c) Other special activities

Figure D.4: Correlation matrices (per ship type) which represent the Spearman correlation between sample scores assigned by classifiers at different iterations for both train-test gap designs. The results are averaged over 30 repetitions of the experiment.



(a) Ro-Ro cargo



(b) Tug

Figure D.5: Correlation matrices (per ship type) which represent the Spearman correlation between sample scores assigned by classifiers at different iterations for both train-test gap designs. The results are averaged over 30 repetitions of the experiment.

Ship type	**Static train-test gap** (r)	**Resampled train-test gap** (r)
Bulk carrier	0.621 ± 0.023	0.609 ± 0.027
General cargo/multipurpose	0.837 ± 0.010	0.832 ± 0.013
Oil tanker	0.680 ± 0.024	0.671 ± 0.030
Container	0.709 ± 0.032	0.703 ± 0.034
Chemical tanker	0.766 ± 0.016	0.763 ± 0.018
Fishing vessel	0.760 ± 0.026	0.729 ± 0.042
Other special activities	0.595 ± 0.038	0.559 ± 0.045
Tug	0.551 ± 0.049	0.497 ± 0.059
Offshore supply	0.814 ± 0.022	0.792 ± 0.026
Ro-Ro cargo	0.741 ± 0.032	0.723 ± 0.038
Gas carrier	0.658 ± 0.039	0.641 ± 0.043

Table D.1: Spearman correlation between sample scores assigned by classifiers in different iterations of the train-test gap experiments, only instances that have a positive class label are considered. Correlations are the average of the resulting correlation matrix (which considers all iterations) and are aggregated over all 30 repetitions of the experiment.

Ship type	**Static train-test gap (r)**	**Resampled train-test gap (r)**
Bulk carrier	0.747 ± 0.013	0.741 ± 0.015
General cargo/multipurpose	0.624 ± 0.022	0.623 ± 0.025
Oil tanker	0.590 ± 0.022	0.582 ± 0.027
Container	0.629 ± 0.022	0.627 ± 0.027
Chemical tanker	0.595 ± 0.022	0.590 ± 0.024
Fishing vessel	0.773 ± 0.010	0.779 ± 0.013
Other special activities	0.627 ± 0.015	0.628 ± 0.017
Tug	0.598 ± 0.020	0.594 ± 0.020
Offshore supply	0.644 ± 0.017	0.638 ± 0.019
Ro-Ro cargo	0.610 ± 0.023	0.614 ± 0.025
Gas carrier	0.478 ± 0.032	0.471 ± 0.037

Table D.2: Spearman correlation between sample scores assigned by classifiers in different iterations of the train-test gap experiments, only instances that have a negative class label are considered. Correlations are the average of the resulting correlation matrix (which considers all iterations) and are aggregated over all 30 repetitions of the experiment.