Master Thesis Artificial Intelligence

Investigating the Use of Active Learning for Classification of Ship Waste Dumping in the North Sea

Active Learning, Anomaly Detection, Time Series

by
Samuel Meyer

first supervisor
Georg Krempl

second supervisor
Arno Siebes

external supervisors
Jasper van Vliet
Paul Merkx

Submitted to
Utrecht University

on
September 29, 2021

For further questions contact: Samuel Meyer: s.j.meyer2@students.uu.nl, 5648122

Abstract

Detecting occurrences of ships discharging waste into the sea is important to reduce sea pollution, but difficult due to data and resource limitations. The act of inspecting whether a ship has discharged waste is expensive and true occurrences are expected to be rare. This makes it difficult to collect enough labels to use for classification by supervised machine learning. This thesis investigated the use of several active learning approaches (uncertainty sampling, density-weighted sampling, QBC sampling and xPAL sampling) to help increase the rate of learning using fewer training instances to classify looping behavior (a proxy variable for waste discharging). Trajectories were summarized to single instances to allow established active learning methods to select them to be queried. Experiments were performed for different selection/learning pipelines to classify both complete trajectories (post hoc) and partial (initial steps to real time) trajectories. Almost all active learning methods significantly improved learning for complete trajectory classification, on average reaching a macro F1 performance plateau of at least ∼90% within 50 queried instances, compared to ∼78% for random sampling after 100 instances. Different models were trained for different points in elapsed time in the trajectories for partial trajectory classification. Most active learning approaches either matched or outperformed random sampling for partial trajectory classification, depending on the evaluated time point. At the best time point the well performing methods, on average, reached a macro F1 performance plateau of at least ∼60% within 100 queried instances, compared to ∼50% for random sampling after the same amount of instances. These results suggest that active learning methods are a suitable approach to decreasing labelling efforts for the problem of looping detection for both complete and partial trajectories, and possibly for similar problems involving trajectories and/or high class imbalance.

Contents

1 Introduction

The Human Environment and Transport Inspectorate (ILT) is the supervising entity of the Ministry of Human Infrastructure and Water Management. Through inspections they see to it that laws regarding transportation and the environment are upheld. They are looking to improve the allocation of their inspection resources through the use of data science and machine learning. This exploration is done by their Innovation and Data lab (IDlab). While a significant amount of data regarding a variety of problems may be obtainable by the IDlab with relative ease, it is usually very difficult to obtain labels for this data regarding specific illicit activities of interest. This is because these labels can usually only be acquired either through either real world inspections or through inspectors making judgements based on historical data. In addition, for many problem domains the expectation is that violations are relatively rare occurrences. Thus obtaining labels is not only expensive, but only very few positive labels are acquired, making the use of relevant data for machine learning purposes difficult. To help reduce the amount of labels needed to train their machine learning models, this research project served as the first step into investigating the use of active learning for their use cases. While the overall intent was general knowledge generation regarding the use of active learning for high class imbalance situations, this thesis used the IDlab's waste discharge ("zeezwaaien") detection project as a benchmark case. While little is known regarding occurrences of zeezwaaien, it is expected to be an anomaly. This provided an interesting case for machine learning approaches with regards to a high class imbalance.

Zeezwaaien is a Dutch term describing the discharge of waste and/or chemical residues by ships into the sea. This type of waste discharge behavior has been regulated by law in appendix II of The International Convention for the Prevention of Pollution from Ships (MARPOL) agreement[1] in order to limit sea pollution via harmful chemicals. Strict limitations on how and when zeezwaaien is allowed are in place.

While zeezwaaien is a well defined event, it is not well documented. Even though it is known to occur, it is not known exactly when, where, and how often this occurs. The Innovation and Data lab (IDlab) of the ILT has been working on anomaly classification models to detect zeezwaaien. Their first step was in the form of a thesis project by Shpat Cheliku [2]. The project focused on using supervised machine learning to classify zeezwaaien en route for tanker ships. Compared to classifying a complete trajectory, classifying en-route meant having less information to make a prediction the earlier in the trajectory a tanker is. To obtain labels for the zeezwaaien problem, experts looked at visualizations of historical complete tanker trajectories and applied labels using their expertise. However, only tanker trajectories for *looping* trips can be labeled as positive or negative zeezwaaien cases by experts post hoc. A ship makes a loop when it returns to the same port from which it departed within a period of 48 hours

without visiting any other ports. Looping behavior was determined by experts to be behavior where the occurrence of zeezwaaien is likely, and has patterns in visual trajectory representations from which experts can distinguish zeezwaaien labels. Thus, the only cases of zeezwaaien that can be established in the data are looping zeezwaaien cases. Due to a lack of labeled data, the scope of the prior research and this thesis is restricted to the detection of looping behavior, which is essentially a proxy for zeezwaaien in many cases.

Other than detection of zeezwaaien, the problem of detection of anomalous instances in trajectory data is a common problem for the IDlab. This thesis project was an exploration in the use of active learning for this type of problem, with the detection of zeezwaaien as the benchmark case. To maintain cohesion with the research already performed by IDlab on the topic of zeezwaaien detection (described above), this research explores the extension of methods used in their prior and ongoing research with active learning. The purpose of using active learning is to reduce the number of labeled instanced needed in this setting to gain satisfactory model performance for looping detection. The specific attributes of this problem (supervised learning, high class imbalance, trajectory classification) has little to no research with regards to active learning. The following sections will give a more detailed look at the problem description and the research questions for this thesis. Section 2 goes over background knowledge that is useful for the context of the rest of the thesis. Section 3 places this thesis into the broader context of related literature. The description of the data used for this thesis and the applied processing steps are discussed in section 4. The methodology established to summarize (per definition multi-instance) trajectories to allow active learning approaches to query trajectory-wide labels, as well as pipeline designs to make use of this summarization are described in Section 5. Section 6 discusses the performed experiments and the achieved results for Task 1: Complete trajectory classification including some additional analysis, and Task 2: Partial trajectory classification. Section 7 summarizes the results and offers concluding discussion and future work.

1.1 Problem Description

This research is a first exploration into the use of active learning for the IDlab. A variety of the IDlab's problem domains share the problems of expected high class imbalance, trajectory/time-series data, and a lack of labels which are often difficult and/or expensive to obtain. The IDlab's zeezwaaien research project shares all of these attributes while having the largest amount of labeled data available than other comparable projects, making it useful for experimentation and analysis. Thus, thhis thesis expands upon prior research within the zeezwaaien project for the detection of looping cases in the North Sea using supervised learning methods [2]. While the final purpose of this overarching project is to detect waste discharging behavior, there are very few waste discharging labels available. The benefit of the looping

target is that inspectors judge looping behavior to be a good proxy of waste discharging, and the looping label can easily be applied to all trajectories in the available historical dataset. This allowed for the simulation of an oracle labelling any instance selected by an AL approach. To keep the approach from being too specific to the zeezwaaien problem, some additional limitations were considered. This included the assumption of minimal prior labels available and no exact knowledge about the domain distribution other than an assumption of high class imbalance.

Classifying looping behavior given a complete trajectory is trivial, since it is clear a tanker has returned to port within 48 hours as soon as it has returned to port. The zeezwaaien project's aim is to be able to make a prediction while a tanker is en route, meaning only information regarding a tanker's trajectory up to a certain point in time is available. Considering looping is frequently an indicator for zeezwaaien, being able to classify whether a tanker will make a loop is already useful information for inspectors in practice. It would ideally allow for inspectors to approach high risk tankers during or before possible illicit activity, or otherwise wait for a high risk ship at the destination port to conduct an investigation there. While this goes beyond the general scope of the problem as described above, the exploration of active learning regarding en route classification was considered an interesting extension of the methodology designed to approach the more general problem.

Thus, the problem can be described as two-fold. These two sub-problems were divided into two separate tasks, the second one building on the methods and findings of the first. The designed methodologies were designed with regard to maintaining as much coherence with the overarching zeezwaaien project and approaches as possible.

1.2 Research Questions

Given the exploratory nature of this research, a general main research question was posed with a number of sub-questions to address particular aspects of the main question. The second research question addresses the extension into the more specific problem of en route classification.

How can active learning approaches be applied to improve supervised machine learning classifiers of high class imbalance trajectory data with little to no prior information (using looping detection of tankers in the North Sea leaving the port of Rotterdam as an example)?

- How can the active learning pipeline be adjusted to suit trajectory data with trajectory-wide labels?
- Can active learning approaches increase the rate of learning given the described problem scenario?

Can the approaches designed to answer the main research question be extended to improve learning

performance for en route classification of looping behaviors?

- How can the design be adjusted to classify partial trajectories rather than full trajectories?
- Can active learning approaches increase the rate of learning compared to passive learning for en route looping detection?

2 Background

This section will provide a basic understanding of some background topics of relevance to this thesis, as well as provide some relevant literature to serve as examples. A general overview of machine learning is given in Section 2.1, followed by a look into bias in machine learning in Section 2.2 and evaluation in machine learning in Section 2.3. Some examples of relevant decision tree-based classifiers are given in Section 2.4. Since this thesis will explore the application of active learning, a general overview is given in section 2.5, followed by a discussion on some more specific active learning methods in 2.6. Section **??** discusses the application domain of the research in this thesis, going into further detail about the data being used and its implications.

2.1 Machine Learning

Machine learning is the sub-field of artificial intelligence regarding the study and implementation of computer systems that improve with experience and training. The following definition is given by Tom M. Mitchell in the introduction to his book *Machine Learning* [3]:

A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E.

Machine learning methods allow for a program to perform a task without needing to explicitly define how its inputs should result in its outputs. This data-driven approach is particularly useful when defining this relation is either very time-consuming or simply not feasible. An intuitive example of a relevant application is in the field of computer vision. For the task of categorizing the type of object an image depicts (e.g. what type of vehicle is present in this image), it is practically impossible for people to explicitly define rules given such an image in a way that allows proper performance over a range of different categories. A machine learning approach can perform well at this type of task, given enough experience and representative data to learn from. In this case, a collection of images makes up the set of data instances used for training this machine learning model. Each image would then have labels corresponding to the objects they depict. The machine learning model learns the relationship between

the features of these images and this label in order to learn how to categorize them. This problem in particular is an example of *supervised learning*. This specific case is an example of a classification problem, since the model has to classify each image as one of a finite set of possible classes of objects (e.g. bicycle, car, truck etc.). It does this by finding the *decision boundary* between these classes in the feature space of its inputs. Supervised learning models may also predict some numerical value, in which case it is called a regression problem.

Besides supervised learning, the other main branches of machine learning are: unsupervised learning, semi-supervised learning, and reinforcement learning [4]. Unsupervised learning typically attempts to find patterns in data without being given labels. Semi-supervised learning falls between supervised and unsupervised learning. Semi-supervised approaches typically combines a small amount of labeled data together with a large amount of unlabeled data. Reinforcement learning approaches learn from their environment by giving positive or negative rewards based on their behavior in this environment.

The amount of applications for machine learning approaches is vast, but a few recent notable examples of application domains in research include agriculture [5], financial market predictions [6] and social media sentiment analysis [7]. While there are many applications for machine learning, there are some significant limiting factors regarding the success of any machine learning approach. This includes the availability of sufficient data, as well as the quality of the data used and the ability of the specific approach to adequately capture relationships in this data.

2.2 Bias in Machine Learning

In any learning process, there is a danger of bias. For example, given the task of writing a summarizing report regarding some historical conflict, your perception of this event is heavily influenced by the sources you use or have previously encountered about this conflict. Using a book produced by one of the entities that played an active role in the conflict may present different facts and may present these in a different way than if it were produced elsewhere. Additionally your own preconceived notions and unique way thinking may also influence the conclusions you draw from each source. The same is true for machine learning. Any bias present in data or in the way a machine learning algorithm handles this data affects its outcomes. While biases are needed to some extent for learning generalizations [8], they can also lead to models not learning what they are intended to learn.

The following survey discusses a variety of sources of bias in machine learning [9]. Bias regarding data tends to manifest itself through how well represented different groups are within a domain, and in

the specific instance features that are in/excluded. If these groups aren't well represented this results in a misrepresentation of the domain the model is learning about with regards to its task. An example is how machine learning models using historical data of humans performing a task are at risk of inheriting the same biases present in the humans it is based on [10].

A type of bias relevant to active learning is sampling bias. This is bias introduced by non-random sampling of different subgroups within the data. Particularly when not all of the available data is sampled, selective sampling may result in the set of sampled instances not being representative of the underlying data distribution [11]. Active learning uses selective methods to pick samples, so is inherently at risk for sampling bias. Some methods have been developed with this in mind, with Hierarchical sampling being an example [11].

There are different methods developed to reduce bias in data. A simple example is class balancing, where majority classes get *undersampled*, or minority classes get *oversampled* to balance out representation in the data. More advanced recent methods use adversarial approaches for debiasing[12][13][14][15].

2.3 Evaluation in Machine Learning

In order to compare performance between different active learning approaches, or between passive and active learning approaches, there needs to be some way of defining and measuring performance. How high or low performance is defined is dependent on both the application domain and the desired outcome. A common measure of performance is *accuracy*, which compares the proportion of correctly identified instances to all classified instances (1).

$$Accuracy = \frac{TruePositive + TrueNegative}{Total} \tag{1}$$

This generally works well as a performance measure, but if the target class we're trying to classify has far fewer instances in our data than the other classes (class imbalance), we can get a high accuracy while not adequately classifying instances for our target class. The same is true for *misclassification error*, which compares the proportion of all incorrectly classified instances to all classified instances (2).

$$Misclassificationerror = \frac{FalsePositive + FalseNegative}{Total} \tag{2}$$

In a class imbalance situation *precision* would be a better performance measure. Precision looks at the proportion of correctly positively classified cases to all positively classified cases (3). This shows how well the classifier can identify a specific class, but doesn't say anything about performance outside of

this class.

$$Precision = \frac{TruePositive}{TruePositive + FalsePositive} \tag{3}$$

Recall looks at the proportion of positive instances that were correctly identified (4). This is useful in particular when missing instances of the positive class in classification is very undesirable.

$$Recall = \frac{TruePositive}{Positive} \tag{4}$$

To get a balanced measure of precision and recall, an F1 score can be calculated (5). Usually a combination of performance measures is used to get a broad idea of model performance.

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall} \tag{5}$$

Additionally there is the area under the ROC curve measure (AUC). This shows how the ratio of the true positive rate over the false positive rate changes depending on the model classification threshold. This shows how well the model distinguishes between the different classes.

2.4 Decision Tree-Based Classifiers

A decision tree is a mapping of inductive inferences consisting of tests represented as internal nodes linking to sub trees or leaf nodes labeled with a specific class [16]. An example of a decision tree for predicting whether conditions are suitable for outside play is shown by Figure 1. The test in each node is based on a specific feature of an instance of interest. In this example the first node is a test based on whether the weather is sunny, cloudy, or rainy. Each non-leaf node has two or more outgoing edges to other nodes, corresponding to the different number of answers to the test. Each non-root node only has one incoming edge, with the root node having none. After a number of tests, a leaf node is reached. This represents a class of the target variable about which one is trying to make a prediction. In this case this could mean that, given the prior tests, the tree makes the prediction 'yes', the conditions are suitable for outside play. Given a training dataset, the specific tests in the nodes are decided by any of a variety of measures deriving informativeness from how the tests split the dataset (or randomly).

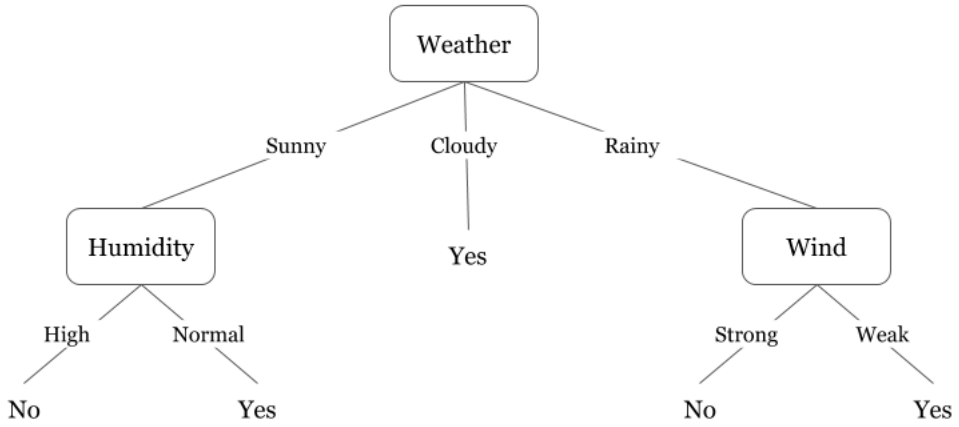

Figure 1: Example of a decision tree from [17] with outside play suitability as the target

2.4.1 Random Forests

Random forests are an ensemble learning method consisting of multiple decision trees. Predictions are made for the same instance of interest by all of the decision trees, each of which casts a vote fore the final prediction. A majority vote of all the decision tree prediction then constitutes the final prediction. A method of producing the different decision trees is via a method called *bagging* (bootstrap aggregating) [18]. Different subsets of the training data are produced (optionally with replacement) selecting instances randomly. Each of these subsets is then used as the training set to produce a decision tree. The process of bagging improves stability for random forests [18] and has demonstrated improved classification performance over the use of single trees [19].

2.4.2 Gradient Boosting Machines

Gradient Boosting Machines create ensembles by producing weak learners iteratively instead of producing them separately [20]. These weak learners are then combined to produce one strong learner. Each of the sequentially produced models is done by sequential error fitting. The instances falsely classified by the ensemble are used to train the next weak learner, which is optimized by minimising loss via gradient descent [20]. One particular implementation of gradient bossintg machines is XGBoost [21]. XGBoost stands for "Extreme Gradient Boosting". This specific implementation uses decision trees as its weak learners, and has been optimized with the intent of being as efficient as possible.

2.5 Active Learning

This section briefly discusses what active learning is, as well as several key approaches. For a machine learning problem, an important aspect is the data used to train the machine learning models. A typical

approach would gather as much labeled data as possible or deemed necessary to train a machine learning model to a point where it performs well. However, it may be desirable to reduce the amount of training instances needed. An example is the case where getting a significant amount of data is not an issue but obtaining labels for this data is expensive either in terms of time and/or resources. In this situation it is desirable to be able to reach a point of high model performance with as few labels as possible. Active learning addresses this problem by selecting the most useful unlabeled data instances, which are then to be labeled as an active part of the training.

Active Learning by Burr Settles (2012) [22] gives an extensive overview of active learning and some of its methods. The following paragraphs give a short version of this overview and relevant literature. The general structure of an active learning approach consists of a data source, a query selection algorithm, an oracle, and a machine learning model. First the query selection algorithm selects an instance from the data source to be queried. This instance is presented to the oracle, which is the entity that can label the given instance. This is usually a human domain expert. The labeled instance is then used to train the machine learning model, and the cycle repeats. Each element in this pipeline can vary depending on the specific problem and chosen active learning method.

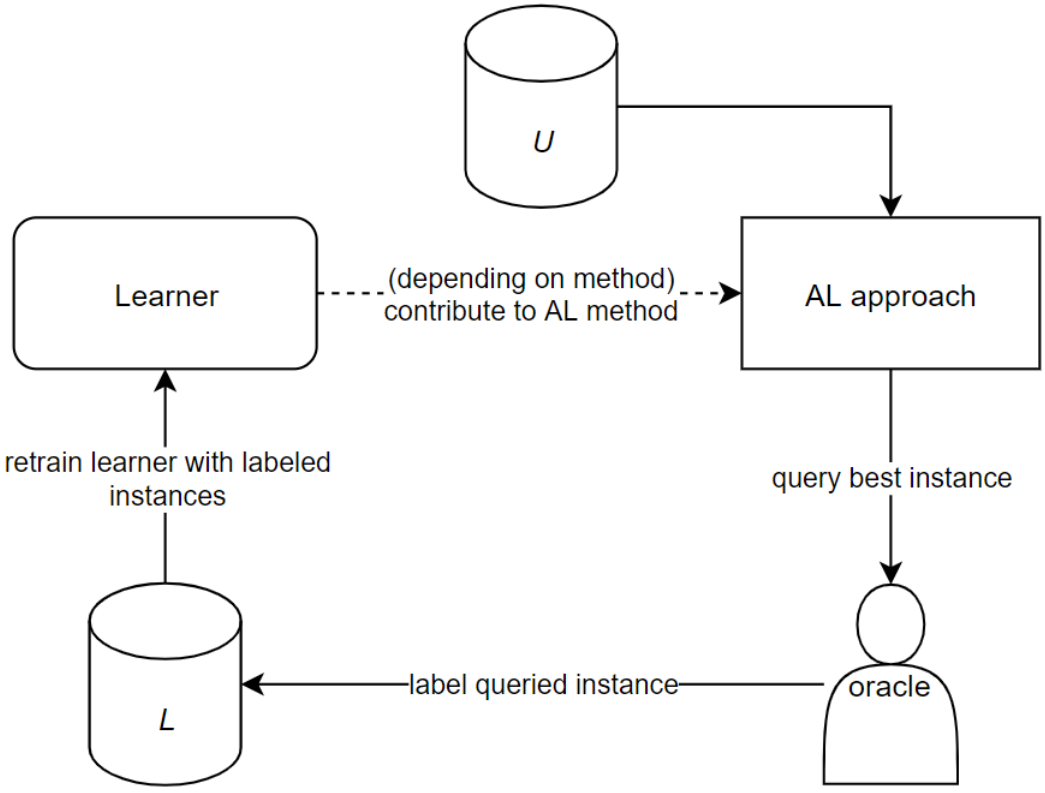

Figure 2: The typical AL pipeline. An AL approach selects the best instance from U to be queried and consequently labeled by an oracle. The labeled instance is then added to the set of all labeled instance L, and the model (learner) is retrained with L as its training set. This is repeated until some pre-specified number of instances has been queried or adequate performance has been reached

2.5.1 Active Learning Categories

Many different query selection methods have been proposed. These are usually particular to different scenarios regarding the form of the data. In the pool-based scenario, all training data is available at once. The following overview will specifically discuss the pool-based scenario for classification problems, since this the most relevant to this thesis. Another problem scenario is stream-based sampling, where unlabeled instances are presented one at a time, and the decision of whether or not to label those instance are made for each of them one at a time. Important to note is that for classification problems, a model learns to find a *decision boundary*. A recent literature survey by Kumar and Gupta (2020)[23] broadly categorizes query selection methods for the pool-based scenario into several categories. This categorization will be used to discuss some of these query selection methods:

- **Informative based methods** look at the informativeness of the individual instances to determine which one to select based on their uncertainty. The idea is that less certain instances are closer to the decision boundary, so they provide more information on the distinction between possible outputs in the feature space. One of the earliest and most common examples of this is *uncertainty sampling*[24]. This method uses the output probability of a probabilistic classifier to determine which instance the model is the most uncertain about. This instance is then selected for labelling and training. This type of method is generally easy to implement and use. However, only sampling close to the decision boundary may lead to sampling bias, and instances close the decision boundary may be similar to each other, so these are less informative overall when sampled together.

- **Representative based methods** look at the structure of the data to determine which ones to sample. The aim is to sample instances that best represent the feature space. A variety of approaches have been proposed. An example is the *density-based* approach, which tries to sample instances in the most densely populated regions of the feature space. An example of this is [25], which summarizes the distance of each instance to its neighbors as a measure of density. The one with the smallest distance is then queried. Due to the more representative instances being used to train the model, representative based methods are less sensitive to sampling bias than informative based methods. However, they generally need more instances to properly approximate the decision boundary.

- **Informative and representative based methods** combine techniques from both previously described methods to determine which instance to query. One possibility is to take individual methods from both of these approaches and combine their "usefulness" measures [26]. However, more novel approaches to creating hybrids between these methods exist. Approaches that include both informative and representative components need to make sure to balance the trade-offs of these approaches appropriately.

- **Other methods** are those that can't be categorized as those mentioned above. These each have their own methodology, strengths and weaknesses. An example is *Expected Gradient Length*[27], which considers the most useful instance to be the one that produces the largest change in the model. After a model has been trained on some labeled data, the expected change in model parameters is calculated for each unlabeled instance. The instance that has the highest expected model change is labeled. This method requires retraining the model for every every instance, for every possible label. Thus, while it performs quite well, it is very computationally expensive.

Some query selection methods require the use of a specific classifier, while others are independent of the classifier and can thus be used in combination with a variety of possible options. This, in combination with how sensitive the query selection method is to the problem context, makes it difficult to determine which method is the most appropriate given a specific problem.

2.5.2 Evaluation and Validation of Active Learning Strategies

Typically in machine learning performance is measured after having fully trained a classifier. However, for active learning it is important to look at the performance in terms of a learning curve. We want to be able to compare the performance of methods at different points of learning so we can see how quickly each method learns from its (passively or actively) selected instances. Thus, performance is measured every time after an instance has been labeled and added to the training pool.

To be able to test whether the models properly generalize, we need to do multiple runs with different instance subsets. This is typically done via k-cross validation, where the instances are divided into k folds. During each round, one fold is left out and used for validation while the rest is used for training. Performance can vary greatly depending on the instances in the pool, so it is important to do multiple repetitions of validation to account for random effects[28]. This is particularly true in the beginning of the learning process, where there is generally a greater variance in performance.

It may also be the case that the goal is not just generalization, but to optimize performance on the training set as well. An example of this is when a system is still learning while deployed, such as in a streaming active learning scenario. In those cases a misclassification comes with a direct cost, so an evaluation of whether an instance is at all predictable before attempting to do so can save resources. A variety of rejection-models are available to estimate which instances to reject [29].

Some active learning specific evaluation methods have been proposed and used. These summarize the learning curve into a specific value allowing for easy comparison of method. One example is the area under the learning curve[30]. This typically uses the area under the ROC curve as its base measure for

every point during learning. The area under the resulting curve is then the area under the ROC curve. This measure is dependent on the total number of instances used, so all compared methods need to be trained with the same number of instances for proper comparison. Another possibility is a deficiency score[31] to compare two methods. This calculates the area between the learning curve for an algorithm and its maximum performance line for two different algorithms, and compares these. While this is not dependent on the number of instances used, it is less useful when trying to compare a large variety of active learning techniques.

Validating performance differences between active learning methods introduces the difficulty of not just validating performance from one set of predictions, but doing so over the course of the entire learning process of the compared methods. A typical approach is to take the ALC as a single-value measure for summarizing a method's performance, and then applying the Wilcoxon signed-rank test on results for a collection of different datasets [32]. The Wilcoxon signed-rank test is a non-parametric statistical test that can be used to evaluate whether there is a difference between to paired performance sets. The null hypothesis is that the median of the population difference between these sets is zero, and the alternative hypothesis is that it is not [33]. In the case of active learning, each pair of ALC scores from the compared approaches was obtained by evaluating the performance resulting from applying the different approaches on the same dataset. The main difference from a paired student's t-test is that the Wilcoxon signed-rank test does not assume the data is distributed normally.

2.6 Active Learning Methods

While section 2.5 gave a broader overview of active learning, this section discusses some specific active learning methods of relevance to this research. Considering that a broad goal of this thesis is to investigate the use of active learning methods for this use case, a variety of methods that are representative of different lines of thought in active learning research are considered. The specific methods discussed are uncertainty sampling 2.6.1, query by committee 2.6.2, density weighting2.6.3, expected error reduction 2.6.4, and probabilistic active learning 2.6.5.

2.6.1 Uncertainty Sampling

As described in section 2.5.1, uncertainty sampling is an example of an informative based method. Introduced by [24], this is a widespread method that is efficient and easy to implement. A probabilistic classifier is applied to all available instances, and the most uncertain one is selected to be queried. For a binary classification problem, this means the instance for which the classification of the positive class is closest to 0.5. While generally effective, uncertainty sampling is myopic. Also, being an informative based method, it has the previously described pitfalls of selection bias and redundant sampling. Considering

the high class imbalance in the data for this thesis, the selection bias may be problematic. However, being one the most widely used and easily implementable methods makes it a good baseline for comparison with other active learning methods.

2.6.2 Query By Committee

The query by committee (QBC) method is an informative based method that uses predictions from multiple classifiers (usually of the same type). The different classifiers represent different hypotheses about how the data relates to the labels. Each classifier is a member of the committee, and the amount of disagreement between the members gives the uncertainty for a specific classification. The instance about which the classifiers disagree the most is queried. While first introduced by [34] in 1997, a variety of different version with different underlying models have been created[35]. A machine learning approach that uses a collection of models to make a prediction is called en ensemble approach. If it is already clear that an ensemble method will be used for classification (like in this thesis), the use of QBC is a logical extension. However, it is just as myopic as uncertainty sampling. The same problem with class imbalance may also apply.

Ideally, the different classifiers all represent as different as possible hypotheses[35]. Thus, when applying QBC to extend an ensemble method it may be beneficial to take additional measures to ensure this. This may cause problems when comparing performance between QBC and other active learning methods, since this also changes how the base ensemble forms its classifiers. Because of this it may be necessary to keep the classifier ensemble and the QBC ensemble separate in such a performance comparison scenario. Since this thesis will likely use a random forest and/or XGBoost approach, both of which are ensemble methods, this will need to be considered.

2.6.3 Density Weighting

Density weighting approaches make use of the available data structure to determine instances that are representative of the data. Specifically, they do this by looking at the specific distribution of the data within the feature space. One example of such an approach uses the distance between instances to determine representativeness[25]. In this method, the distance between each instance and all other instances is calculated and added together. The instance that has the shortest total distance is deemed the most representative, and will be queried. This effectively gives instances in dense regions a higher priority. This may not work particularly well in the case of skewed label distributions, since minority classes may be less dense in the data purely through being represented less. This may also be balanced by applying density weighting as a weighting factor for uncertainty sampling. Another method that could also work better in such situations is the ACLStream method[36]. This first clusters instances (using K-means

clustering), then ranks these clusters and their members in order to determine which instance to query. This generally gives a wider view of the feature space than only using density. While the approach was made from data-streams, it should be adaptable for the pool-based scenario.

Density Weighting methods are less myopic since they consider the relationships between instances. Sampling instances over more of the feature space than just the decision boundary reduces the sampling bias that is typical for informative based methods. If relevant clusters for minority classes can be detected this may prove to be useful. However, these clusters need to be related to the label distribution for these methods to work well.

2.6.4 Expected Error Reduction

Expected Error Reduction [37] looks at how the labeling of any instance is expected to change the model's performance. After training the model with a set of labeled instances, performance is then measured based on a measure of choice. For this, a validation set of data needs to be kept aside. Then, an instance x from the unlabeled instances with label y from possible labels Y to the training set and the model is trained again to form a new hypothesis. The performance of this new hypothesis is then measured on the validation set and compared to the prior performance. The resulting loss is then weighted by the classifier's output probability. This process is repeated for all labels y to get the average expected error of adding instance x. This process is repeated for all instances in the pool, and the instance that results in the lowest average expected error is queried.

Since this process requires training and validation of the model for every instance in the pool, for each round of querying, this is a very computationally expensive method. However, due to the almost direct optimization towards a specific performance measure, this method tends to do very well. Focusing on performance measures that are relevant to a class imbalance setting may be promising for the scenario in this thesis. However, the expected performance toward which the model is optimized is dependent on the quality of the validation set.

2.6.5 Probabilistic Active Learning

Probabilistic Active Learning (PAL) is an approach introduced in 2014 by [38]. This can be seen as a combination of informative and representative based methods. Additionally, this method also takes inspiration error reduction .[23]. This approach computes the expected performance change when giving an instance x a label y. To compute this without knowing label y, the *smoothness assumption* is used. This is the assumption that instances close to each other in the feature space should have similar labels. For each instance x a label statistic $ls = (n, \hat{p})$ is computed. This contains the number of labeled in-

stances n in the neighbourhood of x, and the posterior estimate $\hat{p}$ of x's label as the number of positive labels in the neighbourhood over n.

Because the true posterior probability and label are unknown, the label statistics are used as parameters to model their distributions as random variables. The expected values over these variables with regards to a performance measure are then calculated to get a probabilistic performance gain. The specific performance measure for which gain is calculated can be selected depending on the application. The expected performance gain is weighted by the density of x's neighborhood to represent the importance of its neighborhood. The instance with the greatest density-weighted performance gain will be selected for querying.

PAL is an efficient and effective active learning approach with the benefit of being able to select a specific performance measure to optimize, but is dependent on the smoothness assumption. Considering it makes use of labels of instances nearby the scarcity of positive labels in this thesis may make the estimation of expected performance change difficult for many instances.

Further improvements to PAL have been made. Optimized Probabilistic Learning (OPAL) [39] is a non-myopic version of PAL. This takes a given labeling budget and a difference in mislabeling costs for different classes into consideration. A specific version for the multi-class setting, Multi-class Probabilistic Learning [40] extends the approach beyond the binary class to multiple classes. xPAL is a generalized version of probabilistic active learning using a Bayesian approach [41].

2.7 Anomaly Detection

When some pattern of behavior in data does not conform to what is expected under 'normal circumstances', this can be defined as anomalous behavior. The field of study that deals with finding this type of behavior is called anomaly detection[42]. This being a somewhat broad definition, the reason why it would be desirable to find anomalies is very particular to the domain of application. The reason why finding anomalous behavior is desirable is very dependent on the domain of application. It could simply be to find outliers in data in order to remove them before using the data for some other purpose[43]. However, the outliers may also be the particular behavior we would like to detect. Effectively this comes down to classifying a rare class in a highly imbalanced dataset. This imbalance makes bias an issue since it can be difficult to properly represent the desired class during training. This is particularly the case when the amount of available labeled positive cases is slim.

Different machine learning approaches can be applied depending on the available data and the specific

problem. This thesis will build upon a supervised machine learning approach. Supervised machine learning approaches tend to achieve better performance than semi-supervised and unsupervised approaches, but only if there is enough labeled data including the anomaly class to properly train them[44].

3 Related work

When it comes to the problem of detecting looping behavior and/or waste discharging in ships, the only research available is the prior work at the IDlab this thesis will expand upon[2]. However, there is a significant amount of research on problems with similar data, domains, and sub-topics.

The prior work was, in terms of broad topics, a problem of supervised anomaly detection with trajectory data. While this is a very specific combination of topics, research with strong overlap of topics has been performed. An example is [45], which uses semi-supervised learning for anomaly detection in time series of water analysis. A more general method for time series anomaly detection was developed by [46], using unsupervised machine learning. The main reason why supervised techniques are uncommon for anomaly detection problems is because these types of problems usually lack labeled data. For the zeezwaaien detection problem in this thesis, more labels can be acquired with the caveat that this is very expensive. Since the prior research was even more specific due to focusing on time series, an approach that uses supervised machine learning is even rarer.

The main type of data used in this thesis was Automated Identification System (AIS) data. Most ships are required to emit AIS messages at frequent intervals detailing specific information about the ship itself, as well as dynamic data such as positional data and timestamps. A more detailed description will be given in Section 4.1. An overview of machine learning methods that have been used for experimentation with AIS data is given by [47]. Particularly regarding anomaly detection with AIS data, this overview groups problems into three main categories: position, speed, or time anomalies. This differs from the target of this thesis where classification centered around a more abstract anomalous label that is applied to a full trajectory.

There is also the direction of predicting ship trajectories. [48] is an example of this. They first used an unsupervised method to cluster the data instances, then used each cluster to train a neural network for predicting the ship's motion. The resulting models were used as an ensemble to make one prediction. This is likely the most similar research to the prior work, but still differs significantly in the prediction target, which also translates to the specific methods used being quite different.

Adding active learning to the mix makes this research even more specialized. The most similar research topic wise is[49]. This is a semi-supervised time series anomaly detection problem that uses deep reinforcement learning and active learning for real-world time series data. This, for the same reason as described previously uses semi-supervised learning instead of supervised learning.

While this specific combination of topics is very specialized, there is a significant amount of papers that focuses on different combinations of different sub-topics of this thesis. The following section will first give an overview of the use of active learning methods in the maritime domain. This includes an approach that is most similar to the current research. This section is followed by examples of literature regarding the combinations of sub-topics relevant to this thesis.

3.1 Active Learning in the Maritime Domain

The overview of machine learning using AIS data mentioned in the prior section [47] gives a good idea of uses of maritime data. However, none of the approaches in this paper discuss active learning. However, there is a paper that uses AIS data with an active learning approach [50]. In this paper, a Gaussian process model is used to model normal behavior for historical AIS data. A set of different active learning methods were used to select individual AIS updates to use for training. This paper makes the assumption that the historical data is a good example of normal ship trajectories, and generates anomalous trajectories by transforming location data of full historical trajectories. This essentially mimics AIS spoofing, which they are using as a proxy for general anomalous behavior. Anomalies are then detected by comparing deviancy from the combination of normal speed and location at any point during the trajectory.

Since they are using location-based deviancy from normal behavior, the sequential aspect of AIS data is not particularly important to their methods. This differs from this thesis, which used supervised machine learning methods for which behavior in different points of the trip were likely of importance for classification. Because of this, having the active learning approach select individual AIS updates was not considered a suitable approach for our purposes. Additionally, the assumption that all historical data is normal is a stretch, but one they had to make due to a lack of labeled data. Considering research topics, data, and domain, this paper is likely the closest to this thesis.

Beyond the use of AIS data, the only other research concerning active learning in the maritime domain is the following paper with the topic of recognizing ship types from images [51]

3.2 Active Learning and Anomaly Detection

For anomaly detection tasks, bias due to the imbalance in the data is an inherent issue that needs to be addressed. Machine learning models trained on imbalanced data are likely to over-predict majority classes [52]. Debiasing was discussed in section 2.2, but specifically for for class imbalances the use of active learning may reduce the effects of bias in the data. A comparison was done by [53] between random sampling and support vector machine (SVM) based query selection in highly imbalanced data. To measure performance they used the geometric mean, which measures balance on performance between classes. This should be poor if performance on the anomaly class is low. Their results showed that the SVM based query selection method resulted in their model learning much faster compared to when random sampling was used, considering the geometric mean. While this doesn't necessarily mean all possible bias is eliminated, it sets up active learning as a potentially useful method when faced with class imbalances such as in anomaly detection.

As was discussed in section 2.7, anomaly detection approaches tend to be semi-supervised or unsupervised due to the lack of labeled data for the minority class. This is still true for research that also includes active learning. Some examples are research using active learning for deep learning anomaly detection [54] or more general anomaly detection using mixture-model based active learning[55]. A particularly interesting approach first uses unsupervised methods to reduce an anomaly detection problem to a "normal" classification problem, and then uses active learning on the remaining classification task [56].

3.3 Time Series in Active Learning

Considering time series and trajectories are overlapping domains, time series specific research is also of relevance. When using time series data in machine learning, using data at individual time points as data instances loses their relationship to other parts of the same series. Depending on the application, this can mean information important for classification performance is lost. To preserve some of this information to some extent, a variety of methods exist. An example is the *sliding window* approach, which summarizes the last l instances in time by statistical functionals over these l instances for each time point[57]. When using *all* instances prior to a time point, this is called an *expanding window*.

There are few active learning methods made with time series in mind. One of these methods is ACTS [58]. This method uses a nearest neighbor classifier as its basis. It uses shapelet discovery methods[59] to fit patterns onto the data. Probabilistic models over these patterns and the data are then used to calculate uncertainty and utility measures used to select which instance to query.

The paper [50] discussed in section 3.2 queried individual AIS updates out of a pool of historical trajectories. Considering that the relationship between different parts of a trajectory are often important for classification, time-relative information is lost. The updates taken out of the context of their whole trip don't mean as much individually as they would together.

To use time series with active learning methods, there needs to be some way to represent larger parts of trips as a whole for the active learning methods to select from. This is particularly true for this thesis, where a label (e.g. looping vs. non-looping) is applied to a whole trip after querying. It doesn't make much sense to select an individual out of context AIS update, to then label and train with the full trip.

The methodology used for the research that this current thesis builds on used sliding and expanding window approaches to allow individual instances to retain information from a broader view of the trajectory. The same data representations was used for the current thesis to retain cohesion with the prior base approach. These representations were used to allow trip selection through active learning. Complete trajectories (or trajectories up to a certain point) were represented by the last instance in the expanding window. This summarizes the full trip to that point with a set of statistical functions. Taking this point for different trips at the same (partway or final) time point allows for single-instance comparisons of different trajectories. This is an easy to implement summarization, but is likely to lose finer details of the trip. If a small portion, or a combination of smaller parts of the trajectory would be of significant importance to the classification, this may be averaged out by the rest of the trajectory in the statistical functions.

Another approach is to produce a higher-level representation of each trip through clustering. There is a significant amount of research related to the clustering of trajectories. A broad review is given by [60]. Most methods cluster trajectories based on a global comparison of distance or density measures. This has the same problem of averaging out more local parts of a trip of possible importance as summarizing by global statistical functions does. A prominent example of a clustering algorithm is DBSCAN [61]. While this density-based algorithm was not designed to cluster trajectories, it has been extended for, or the basis of, other algorithm to do so. The TRACLUS algorithm first splits trajectories into segments before clustering them using an approach similar to DBSCAN [62]. This allows clustering of sub-trajectories of the full trajectories instead of only full trajectories. Only a small subset of trajectory-clustering research is focused on generating features from these clusters to be used for classification via supervised learning. The TraClass algorithm combines an adjusted version of the TRACLUS algorithm with a region-based clustering algorithm, with the specific focus of generating features to be used for classification [63]. These

features are in the form of trajectory memberships of the different region and segment clusters. The paper introducing this algorithm particularly discusses the use for ship-related trajectory classification. This seems like a very appropriate approach for the problem in this thesis. However, it is a very extensive method and no implementation is available, so implementation of and experimentation with this method may not be feasible given the scope of the project. Implementations of TRACLUS are available (e.g. [64]), so experimentation would this method may be valuable.

3.4 Active Learning in Deployed Model

Informative based active learning methods tend to base their measure of informativeness on the uncertainty about a specific instance. However, is are more than one type of uncertainty. The following distinction between two types of uncertainty can be made. [65]. The first is epistemic uncertainty, which is uncertainty about an instance as a result of a lack of knowledge. This uncertainty is reduced as more instances are used for training. The second is aleatoric uncertainty, which is uncertainty that is inherent to the data being used. Even training a model until its parameters are optimal won't be able to reduce this uncertainty. A method to model both of these types of uncertainty was proposed by [65]. Since epistemic uncertainty can be reduced, this is generally the uncertainty that active learning methods want to use to determine informativeness. Realistically the uncertainty measures used tend to include parts of both.

[66] use the methods to model the two types of uncertainty proposed by [65], and then compares using each of them for their own proposed active learning method. Here, aleatoric uncertainty isn't shown to be helpful in terms of performance when used for sample selection. As reducing the reducible uncertainty as quickly as possible is generally seen as desirable for training, isolating epistemic uncertainty for the purpose of selective sampling in active learning makes sense. Other than [66], another example of an approach to achieve this is [67].

However, there is a possible use of aleatoric uncertainty. Approaches like the one in [66] can be used to determine aleatoric uncertainty for unlabelled instances while training a classifier. If there is an instance for which it can be identified that the model won't be able to make an accurate prediction, an oracle can then be used to classify it instead. This reduces the automation that a classifier provides, but avoids low quality classifications by introducing expertise of an oracle where the classifier will inevitably fall short. Estimations of aleatoric uncertainty can also be used to help estimate epistemic uncertainty as was done in [67].

4 Dataset description

The data used in this research was already preprocessed as a part of the overarching IDlab zeezwaaien detection research project. To maintain coherence with the overarching project, no further changes were made to this dataset. This section will give a description of the data used, the preprocessing steps taken by the IDlab and the feature engineering methods used.

4.1 Data Sources

The data used for this thesis is a combination of Automatic Identification System (AIS) data collected by Made Smart Group and port call data collected by the European Maritime Safety Agency (EMSA). The purpose of using two datasets was to combine informative aspects of both of them to make a single dataset.

As mandated by the International Maritime Organization, ships of 300 gross tonnage or more on international voyages, cargo ships of 500 gross tonnage or more not on international voyages, and all passenger ships regardless of size are required to have an AIS system emitting update messages with high frequency at all times. These updates contain static information (part of which is entered into the system by the crew), and dynamic information collected through a variety of sensors. The most important dynamic features contained in each AIS update are: date and UTC time, latitude, longitude, speed over ground, and and orientation. Examples of static features are a unique ship identifier (IMO numbers), ship type, cargo type, length, and width. The AIS data obtained from Made Smart Group for the zeezwaaien project spans the full year of 2020, for all tanker ships leaving the port of Rotterdam.

Port call data logs information regarding ships departing from or arriving at ports. This includes the ship's IMO number, actual time of arrival (ATA), actual time of departure (ADT), and more ship specific descriptors. The port call data obtained from the EMSA spans all of their logged European port calls over the course of the full year of 2020.

4.2 Preprocessing

The first step of preprocessing was feature selection. Based on prior initial testing done by the IDlab the 6 most useful features were selected from the AIS dataset, which helped reduce the total number of features to be produced by via feature engineering. The selected features that would later be used for feature engineering were *latitude*, *longitude*, *speed over ground*, and *orientation*. Beside these, *IMO*

number and *date and UTC time* were also kept. The important features from the port call dataset were those that would be helpful for separating the AIS updates for tankers into specific trajectories, as well as producing the labels for the looping target.

If a tanker had taken multiple trips in the year 2020, the trajectories from these trips would not be explicitly separated in the AIS dataset. The AIS dataset and port call dataset were merged as to make this separation. Using the *date and UTC time* feature from the AIS dataset as a key, and the *ADT* feature from the port call dataset as a key, the pandas function merge_asof()[68] was used for all instances with matching IMO numbers. This allowed merging based on the AIS update closest to a specific time of departure, as well as all updates within a certain time frame before or after this point. The datasets were merged such that all AIS updates within 48 hours after the corresponding time of departure were grouped into a specific trajectory. This meant that each trajectory was given its own specific *trip_id*. Considering a tanker needs to return to the same port within 48 hours for its behavior to be defined as looping, this time frame in the data covers all cases that can contain looping behavior.

This merging still left differing periods of time at the start of trajectories where tankers were practically stationary at the port of Rotterdam. To make the trajectories more directly comparable, these stationary periods were removed from the trajectories. This was done by setting a virtual "gate" at the exit of the port of Rotterdam. Any updates in trajectories prior to the tankers passing this gate were removed. In addition to the *date and UTC time* feature, an additional time-based feature was added called *time_elapsed_s* (time elapsed in seconds), with the elapsed time starting from the first trajectory point after passing the "gate".

To create the looping target feature, another pass of filtering and merging with the port call dataset was performed. This was done by matching port calls to trajectories to find out whether the tanker returned to the port of Rotterdam by the end of its 48 hour duration. If this was the case, a trajectory was given a positive looping label (label 1). If no such match could be found, it was given a negative looping label (label 0).

The resulting dataset contained 5433 trajectories, of which 88 were positive looping instances. However, due to each trajectory being made up of thousands in individual instances, this dataset was unmanageable given the available resources. To alleviate this, and given the rarity of the value of the positive cases of which the loss was undesirable, a random selection of non-looping cases were removed. The downside of this was that the class imbalance was shifted slightly in the favor of the minority class, making it less extreme. Even still, the dataset now contained a total of 11,785,048 instances, making up

1767 trajectories of which 88 looping trajectories. This gives each trajectory on average ∼6,600 instances. Of these trajectories, with a prevalence for the looping class being ∼5%. While not as extreme as before, this still constitutes as high class imbalance.

4.3 Feature engineering

The features of each individual AIS update in a trajectory only represent a specific point in time. Using these for machine learning would then lose the relationship that point has to others in the trajectory. To generate features that allow the retention of information regarding this relationship, the sliding window and expanding window approaches (as discussed in 3.3) were used to engineer new features. An added benefit is that the calculation of statistical functions over windows for a set of features should be feasible in real time for en route transformation of raw data to the new features. The prior research experimented with windows of different sizes K, with the best results being obtained for a combination of three different sizes [2]. These were window spanning the last 10 minutes, the last 60 minutes, and the expanding window. For each of these time frames a set of eight statistical functions were calculated. These were as follows:

Mean: the average of the K-sized window.

$$mean(x) = \bar{x} = \frac{\sum_{i=1}^{K} x_i}{K} \tag{6}$$

Max: the maximum of the K-sized window.

$$max(x) = \max_i x_i, where\ i \leq K \tag{7}$$

Min: the minimum of the K-sized window.

$$min(x) = \min_i x_i, where\ i \leq K \tag{8}$$

Median: the median of the K-sized window.

$$median(x) = \frac{o_{\lfloor \frac{K}{2} \rfloor} + o_{\lceil \frac{K+1}{2} \rceil}}{2}, where\ o\ ordered\ list\ of\ K numbers \tag{9}$$

Std: the standard deviation of the K-sized window.

$$std(x) = \sqrt{\frac{1}{K-1} \sum_{i=1}^{K} (x_i - \bar{x})^2} \tag{10}$$

Range: the maximum of the K-sized window - the minimum of the K-sized window.

$$range(x) = \max_i x - \min_i x, where\ i \leq K \tag{11}$$

Relative location of the maximum: the relative location of the maximum value of the K-sized window divided by the length of the K-sized window.

$$relative\_location\_max(x) = \frac{argmax(x_i)}{K}, where\ i \leq K \tag{12}$$

Relative location of the minimum: the relative location of the minimum value of the K-sized window divided by the length of the K-sized window.

$$relative\_location\_min(x) = \frac{argmin(x_i)}{K}, where\ i \leq K \tag{13}$$

Applying these functions over the four base AIS features *latitude*, *longitude*, *speed over ground*, and *orientation*, results in 32 new features per window, resulting in a total of 96 new features. Combined with *time_elapsed_s*, and the original AIS features, this gave a total of 101 features.

5 Methodology

This section discusses the experimental methods and design of this thesis. All decisions were made with regard to the specifications of the research problems as well as coherence with prior [2] and concurrent [69] research and methods done by the IDlab for the zeezwaaien detection project.

5.1 The Base Classifier

The prior work in the zeezwaaien project tested performance of a variety of different types of classifiers. Out of those compared, gradient boosting was a consistently among the top performers, with its specific implementation XGBoost [21] also training significantly faster. Based on those results, the base classifier used for this research was the same XGBoost classifier. Only one classifier type as used to allow the focus of performance comparisons to be on the different active learning methods. Considering the restriction of minimal prior initial labels, a parameter tuning set of sufficient size to make a difference seemed out of the question. However, an initialization set of 10 instances that were newly selected each execution (3 positive, 7 negative) was used. The main reason an initialization set was included was due to the active learning library ModAL [70] requiring at least one instance of each class to be supplied for initialization. The initialization set size and distribution was the same for all tested methods, but for QBC this meant

that they had to be divided over multiple individual models, each having the ModAL requirement of at least one instance per class. The size of 10 instances was chosen to allow some initial differences in the initialization sets of the individual committee members of the QBC approach.

The following sections discuss the selection and set up of the different active learning methods used for this thesis, the steps taken to allow for them to be applied to the trajectory dataset, and the evaluation methods used.

5.2 Active Learning Methods

A variety of methods from the active learning methods described in Section 2.6 was selected. These were selected because they are representative of different categories of active learning approaches discussed in Section2.5.1. One of the goals of this research was to produce results that to some extent inform the use of active learning in a situation where there is little known about the available data and its distribution. Because of this, for most methods the hyperparameters were chosen to be the ones that were set as default or otherwise the most common. The exception to this was QBC, considering there is no standard configuration of members for a committee.

5.2.1 Uncertainty Sampling

To allow for a comparison of the selected active learning methods to passive learning, a random sampling strategy was implemented. The first active learning strategy implemented was uncertainty sampling. Uncertainty sampling was chosen as an example of a basic informative method. It is both simple and relatively common, and is still commonly being used as a basis of comparison when testing newer methods.

5.2.2 QBC Sampling

QBC shares the same reputation as a good testing benchmark method, often surpassing active learning in performance. While still belonging to the class of informative methods, it comes into its own territory with regard informativeness using disagreement between models as an informativeness measure as opposed to uncertainty. Due to there being no common configurations for QBC, a set of tests using different configurations was performed and the best performing one was selected. These tests were performed for the problem of classifying complete trajectories using summarized instances as will be discussed in Section 6.2.1. The results can be foind in Appendix A. This allowed for the evaluation of the performance of QBC with the assumption of a good starting configuration.

5.2.3 Density-weighted Sampling

The specific implementation of density-weighted sampling used in this research used density as a weight for uncertainty sampling. While density-weighted approaches are the typical example of a representative active learning approach, a purely density-weighted approach would be likely to ignore the minority class in a high class imbalance setting, assuming some separation between the classes in the feature space. Using density as a weight for uncertainty sampling allowed for an exploration of the use of a representative method while to some extent addressing the disadvantages introduced by the label distribution for both density-weighted sampling and uncertainty sampling. As uncertainty sampling on its own is also being used for experimentation, The inclusion of this hybrid method gives an indication of how using a representative approach could add to or subtract from performance for the basic uncertainty sampling approach.

5.2.4 xPAL Sampling

xPAL was chosen as an example of the more recent approach of probabilistic active learning. While the method is potentially very powerful, not setting parameters is limiting. For a class imbalance case this was particularly concerning given that the performance measure it optimized towards was misclassification error. However, optimization toward class-balance sensitive measure such as error rate and accuracy are common, making an investigation in their use for active learning an interesting addition for this specific type of problem scenario. Considering XGBoost doesn't produce kernel frequency estimates, these were estimated separately from the classifier. Kernel frequency estimates were produced separately from the classifier using the *pairwise_kernels* method from sci-kit learn [71].

Sampling Method	Parameters
Random	NA
Uncertainty	Informativeness given as proximity to decision boundary (instance with classifier probability closest to 0.5 is best)
QBC	A set of eight random forest models with base parameters. Training sets for QBC learners randomized with replacement to form different hypotheses. Instances are selected through max disagreement sampling
Density-weighted	Density given as summary of cosine similarity between an instance and all other instances.
xPAL	Optimizes towards low misclassification error. Maximal number of hypothetically acquired labels is two. Prior probability of 1 given for all class distributions.

5.3 Summarizing Trajectories and AL Pipeline Design

One of the first challenges regarding the application of active learning for trajectory data stems from the applications of labels to full trajectories. In our looping scenario, a full trajectory has either a positive or negative label for the looping target, but each trajectory consists of thousands of individual instances. Active learning approaches generally select a single instance in the dataset to be queried. To solve this problem, the trajectories were summarized to single instances.

The possibility of generating new trajectory-wide features to represent trajectories as single instances was considered to be out of the scope of this thesis and will likely be part of future research at the IDlab. Instead, advantage was taken of the summarizing characteristics of the features generated through the sliding and expanding window approaches. Particularly, the expanding window features statistically summarize the trajectory for all instances prior to a specific instance, given a time-ordered sequence of trajectory instances. Therefore, the expanding window features of the *last* instance in each trajectory summarizes the corresponding full trajectory. Figure 3 shows how the last instance of each trajectory was used to represent its corresponding full trajectory in a new summarizing dataset.

Apart from keeping the research scope in check, the use of features that were engineered as part of the overarching waste discharge detection project at IDlab was also motivated by the coherence it brings to this overarching project. However, it should be noted that summarizing a trajectory by taking statistical functions over all of its instances causes a loss of information. This includes the order of the instances in the trajectory as well as smaller behaviors that could be indicative of the target behavior but are lost due to the large number of instances per trajectory or time frame that are summarized over.

With the addition of the new dataset of summarized trajectories, there are now two datasets that could potentially take individual roles in the active learning pipeline. Figure 4 shows the different pipelines that were designed to make use of (one or more of) these two datasets. Figure 4a shows how the typical AL pipeline (as seen in Figure 2) was adjusted to query instances from the unlabeled summarized dataset U_s to then label and add all instances of the corresponding full trajectory from the unlabelled full trajectory dataset U_f to the labeled full trajectory dataset L_f. The learner is then trained with the full labeled trajectories. The downside to this approach is that if the AL approach used is dependent on the learner, it will be trained on a different kind of dataset than the one it is querying. This may have caused low performance in early testing.

Figure 4b solves this problem by introducing an additional model called the producer. The producer is a separate model that is exclusively used for querying the best instance from U_s. In this scenario, the

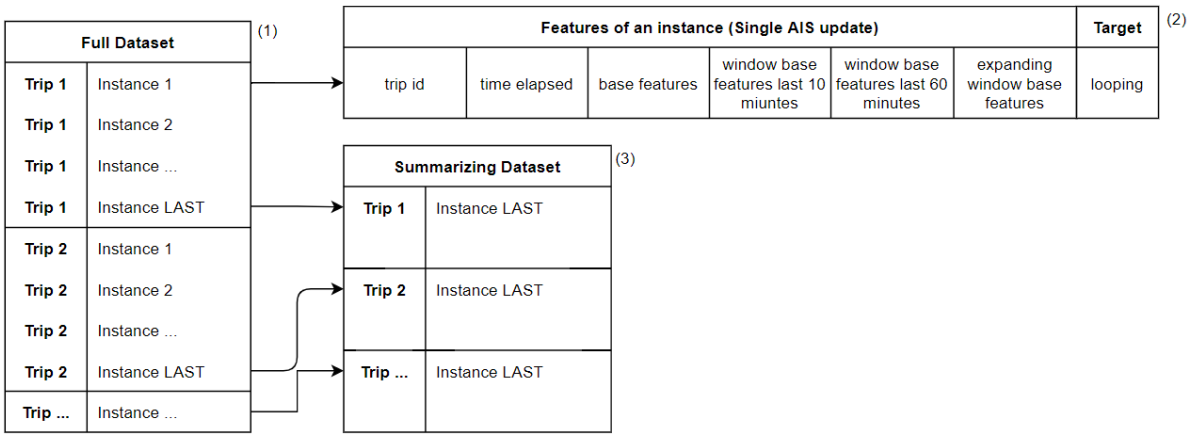

Figure 3: Diagram showing how the last instance of each individual trajectory in the dataset of full trajectories (1) was used to represent the corresponding full trajectory in the new summarized dataset (3). Considering the features (2) of each instance statistically summarize the time points prior to that instance, and in particular the expanding window features of the last instance of a trajectory summarize all prior points in the trajectory, these features for the last trajectory instance statistically summarize the full trajectories.

learner model that is used to make the actual predictions is called the consumer. Every time an instance from U_s is queried and labeled, it is added to the set of labeled summarized instances L_s. The producer is then retrained using L_s as training data. This splitting of tasks between two was inspired by the topic of sample reusability [72]. However, while sample reusability looks to increase performance over a one-classifier model by combining strengths of different model types on the same dataset, in this case the same model type is used for both tasks but on different datasets. The rest of the process is the same as in Figure 4a. An additional benefit of this method is that it is possible to use two different types of models for the consumer or producer if so desired. The concern with this pipeline is that the informativeness of instances selected by a model trained on the summarized set of trajectories may not properly translate to the informativeness of the full trajectories. Additionally, there is the downside of having to train two models, which increases training time. There is also no added benefit to the additional model when the applied AL method does not make use of a model to select a query. For those that do, the pipeline in 4b seemed to perform better than the pipeline in 4a.

Figure 4c is practically identical to the typical AL pipeline shown in Figure2. In this pipeline design, the full trajectories are fully removed from the equation, using only the summarized trajectories for instance selection, training, and classification. This removes some versatility in terms of classification of the learner, since it cannot be expected that the learner performs well outside of classification for the last instances of the trajectory. However, it significantly cuts down on training time by reducing the ∼6000 instances per labeled trajectory to only one instance per labeled trajectory, as well as removing the issue of possible informativeness translation errors from the summarized trajectories to the full trajectories.

Figure 4d Follows the same principle as Figure 4c, but introducing the separate producer/consumer models from Figure 4b. The idea is the same as for the design in Figure 4c, but allowing the flexibility of choosing different, separate models for the producer and consumer.

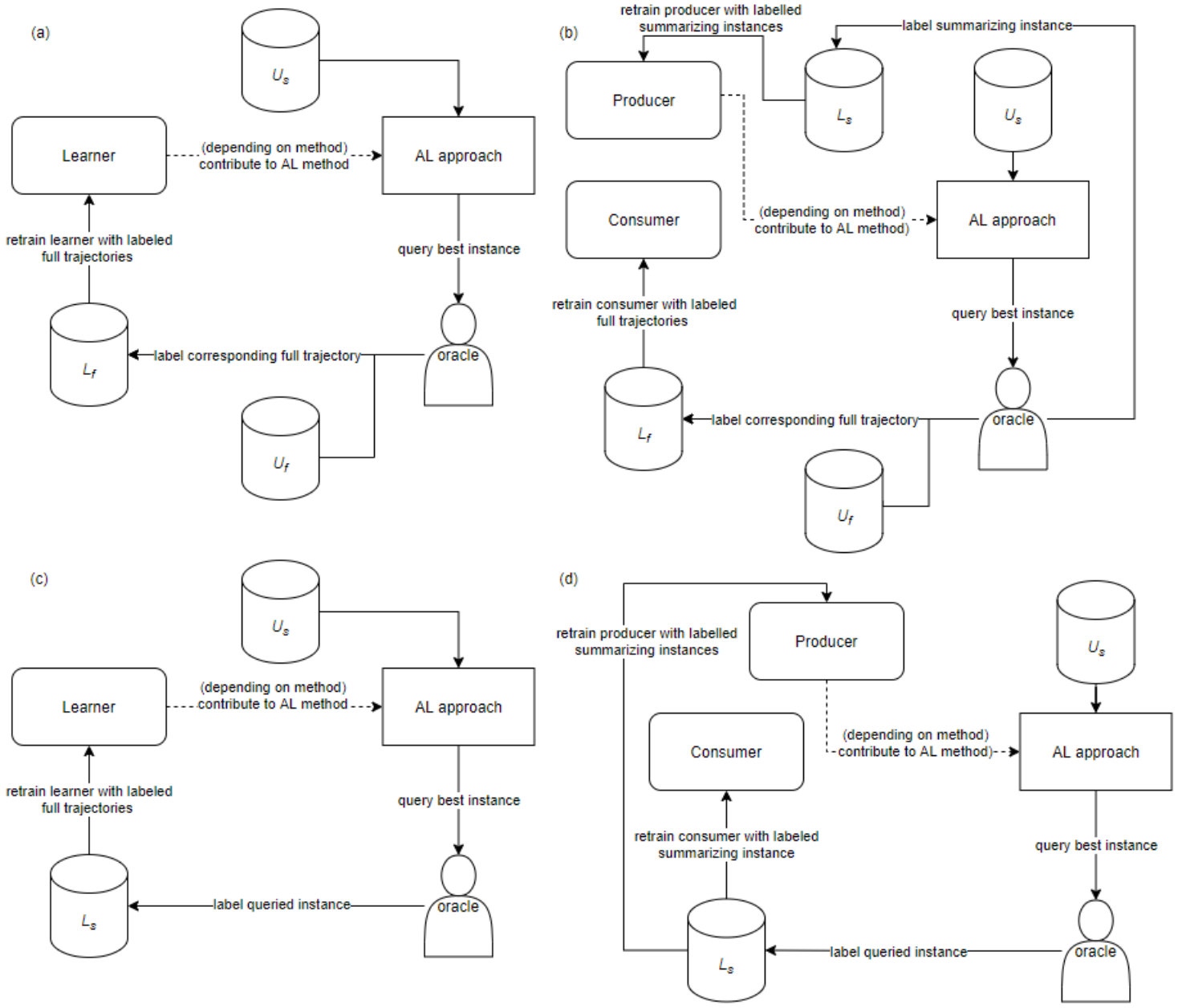

Figure 4: The different possible pipeline designs used in this thesis. Designs a and b use the summarized dataset only for instance selection and the full trajectories represented by the selected instances are used for training the base classifier. Designs c and d train the base classifier directly with the selected summarizing instances. Designs a and c only use the base classifier as the learner for both classifying and instance selection. Designs b and d use a separate classifier (the producer) to select instances to be queried, with the base classifier (consumer) is only used for classification.

When choosing between one of these four designs, there are two main considerations that have to be made:

- Should the predicting classifier be trained with all individual trajectory instances (a and b) or only the summarizing instances (c and d)?
- Should the model used in querying instances be the same as the classifying model (a and c) or separate from the classifying model (b and d)?

The first consideration may depend on a variety of factors such as attributes of the specific datasets and/or method of trajectory summarization used. This thesis experimented with both options to be able to offer a performance comparison given our research scenario. The second consideration is mostly dependent on the specific active learning approach used. The appropriate designs for the active learning methods used in this thesis are as follows:

Random selection does not use a model to select an instance to be queried, so there is no need for there to be an additional model to be used for instance selection. This makes designs **a** or **c** suitable.

Uncertainty sampling, density weighted sampling, and xPAL make use of a single classifier to select an instance to be queried. Considering the presumed performance difference between designs a and b in this case, it makes sense to have a separate producer model when using all instances within trajectories for training. This should not be needed when training on the summarized trajectories, unless using different types of models for the consumer and the producer. Thus, the most suitable designs for these three active learning methods in this case are designs **b** or **c**.

Considering the main classifier in this research was chosen to always be XGBoost with the same parameters, the models that make up the committee of QBC need to be separate from the main classifier. To maintain this separation, the chosen design can only be **b** or **d**.

5.3.1 Summarizing Partial Trajectories

While for complete trajectory classification the last instance of each trajectory was used as the summarizing instance, for partial trajectory classification the instance in each trajectory closest to a specific time point X was used as the summarizing instance. A window of 30 minutes before this time point was used to account for cases where there was a lot of time between consecutive AIS updates. However, an oversight was the fact that part of the instances closest to the specific time point X were beyond time point X. The 30 minute window only extending to the 30 minutes prior to this time point meant that

those instances were not included for testing or training, shrinking the dataset. To allow for the inclusion of shorter trajectories for training, the last instance of trajectories shorter than time point X (and its 30 minute window) were extracted to a separate dataset.

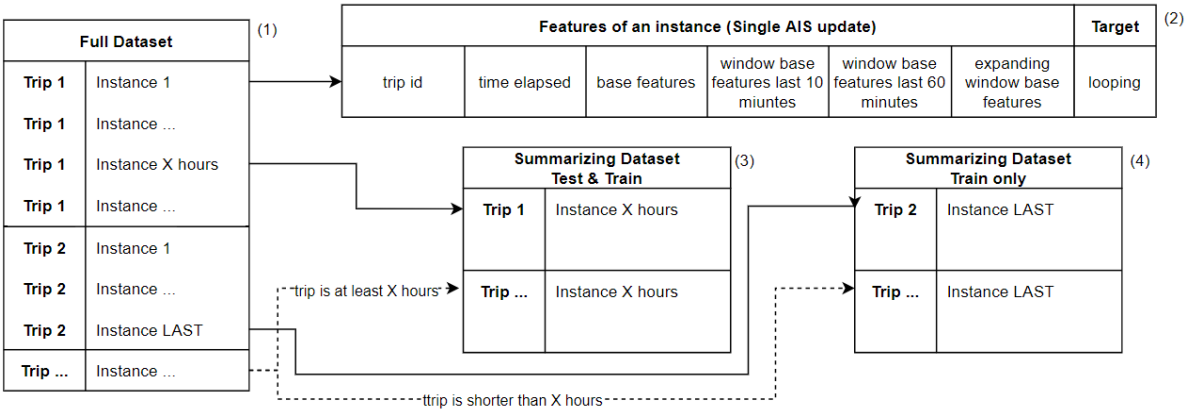

Figure 5: Diagram showing how the instance at a specific hour X of each individual trajectory in the dataset of full trajectories (1) was used to represent the trajectory up to the time point designated by X to create a new summarized dataset for training and testing (3). Considering the features (2) of each instance statistically summarize the time points prior to that instance, and in particular the expanding window features of the last instance of a trajectory summarize all prior points in the trajectory, these features for the last trajectory instance statistically summarize the full trajectories.

5.4 Evaluation and Validation

The main performance measures (described in section 2.3) used to evaluate the performance for the experiments in this thesis were macro precision, macro recall and macro F1. Precision was chosen as an important measure due to the high expenses regarding an inspection in a real world scenario. Low precision would mean a high amount of false positives, which could be seen as a waste of resources. Recall was chosen because of the desire to correctly identify as many of the looping trajectories as possible. A low recall would mean many of these cases would be missed. The cost of missing these is less well defined than a direct cost of resources, but finding these (and eventually zeezwaaien cases in future work) is the main reason for this research. To allow a quick indication of the balance between these two, the F1 score was used.

Considering the target class is the minority class, the performance measures were macro-averaged. Each macro score represents the average of the scores for each individual class. This keeps the majority class from outweighing the minority class. These specific measures were also used in the prior research, allowing for a more direct comparison between results of this research and the prior research. In the rest of the thesis, the macro-averaged measures will generally be referred to without the macro descriptor.

Given that this is an active learning setting however, the performance of a model was evaluated every time a new instance had been queried instead of only after training. Particularly at the start of learning this resulted in a lot of variance in performance, making it important to run many repetitions, reducing this variance as much as possible. 5-fold cross validation was chosen as the method of testing and evaluation, which was repeated as often as possible for each experimental setting. At the start of all testing for a specific method, the same randomization seed was always set. This meant that the different methods were always tested on the same folds generated over all repetitions. The performance measures were averaged over all the performed repetitions for each individual point in learning (e.g. after querying the fourth instance, after querying the fifth, etc.). The dispersion of the performance is shown in interquartile ranges.

As an additional measure, the selected label ratio was used. This shows the averaged ratio of the labels that were selected at each point in learning, with 1 representing only looping cases were selected, and 0 representing only non-looping cases were selected. Considering the major variance in this measure, no dispersion is shown in the corresponding figures. The purpose of this measure was to be an additional source of information that could be related to reasoning behind performance differences.

A more active learning specific measure (discussed in Section 2.5.2) chosen for these experiments was ALC. This allowed for a very quick comparison between learning performance of the different methods, represented via only single values. The main additional benefit of measure was that it could be used for statistical significance testing. The ALC values for the different methods were used to perform Wilcoxon signed-rank tests. However, an adaptation had to be made to the method described in Section 2.5.2. While typically the paired values in two compared sets correspond to performance given a specific dataset, this research is specifically oriented toward performance on the available AIS-based dataset. It is more of an exploration of performance for different active learning methods given this dataset (and its specific characteristics e.g. class imbalance and trajectories), rather than a general performance comparison of the methods overall. To adjust the typical method in which the Wilcoxon signed-rank test is used, pairs of ALC scores for compared methods correspond to performance for specific validation runs over the dataset. Because the tests for comparing different active learning approaches all started with the same seed, the train and test sets for each round of evaluation were the same for each method at the same round of evaluation. Thus, their performances could be paired for these different test sets, which all were identical subsets of the dataset of interest.

The prior research experimented with smoothing predictions by taking the majority vote of a set of predictions within a certain timeframe. Because this did not seem to have any added benefit, this

smoothing is excluded from this research. Thus, any prediction for the complete trajectories is made by classifying its last instance. For partial trajectory classification, the instance at a time point of interest is classified.

For partial trajectory classification, evaluating performance at a specific time point means that trajectories that end before this time point should not be part of the evaluation set. This was the reason the instances were split as shown in Figure 5. However, the shorter trips could still contribute to the classifier's learning. To keep these separated while still using the shorter trips and doing k-fold validation, first all instances at the specific time point were divided into the k-folds. Each time a new set of folds was used for training, the shorter trajectories were added to the training set. This allowed for the inclusion of shorter trajectories in training, while still only evaluating performance for trajectories at the specific time point.

6 Experiments and Results

The following sections discuss the performed experiments and their results. These are first discussed for Task 1, for which the methodologies discussed in section 5 were applied to the problem of classifying complete trajectories. Some additional analysis was done to get a better idea of which trajectories were selected often and why, as well as some analysis regarding important features. Afterwards, the experiments and results for Task 2 are discussed. These built on the previously mentioned methodologies and findings for Task 1 to tackle the problem of partial trajectory classification.

6.1 Task 1: Complete Trajectory Classification

The experimentation for classifying complete trajectories as looping or non-looping was done in two parts. These two parts reflect the two possible pipeline considerations regarding whether to train the predicting classifier on all trajectory instances or only the summarizing instances (discussed in Section 5.3). For quick reference, the defining features of the different designs shown in Figure 4 are re-iterated in Table 1.

	One model (learner)	Separate models for classification and selecting instances (consumer and producer)
Train on all instances in selected trajectories	a	b
Train only on selected trajectory summarizing instances	c	d

Table 1: Defining characteristics of the different pipeline designs

For the first part, pipelines a and b are used depending on the sampling method used. For the second part, pipelines c and d are used depending on the sampling method used.

6.1.1 Training on all instances in selected trajectories

Using all instances in each selected trajectory for training meant that every time a trajectory from the pool of unlabeled trajectories was queried and labeled, ∼6000 instances were added to the training set. This rapidly growing number of instances that had to be used to retrain the XGBoost model every time a new trajectory was queried strongly limited the amount of testing that could be done. This effect was increased by the combination with the instance information-value operations that had to be applied to the millions of instances in the pool. The parameters for the rounds of testing were set to keep the amount of time required to obtain results reasonable. They are given below by Table 2.

Number of folds	5
Number of repetitions	3
Labeling budget	80 instances

Table 2: The experimentation parameters for training with all instances in trajectories

With these parameters, only a total of 15 runs of training and validation were done. This is far fewer than would be considered desirable, but collecting results using these settings were already on the border of unreasonable. Running the experiments for random sampling took∼seven hours, ∼11 hours for uncertainty sampling. Since the training durations for xPAL and even more so for QBC were expected to take significantly longer than for the others, these were excluded in this part of experimentation. Table 3 shows the pipeline designs used for each method.

Sampling Method	Pipeline Design
Random Sampling	a
Uncertainty Sampling	b
Density-Weighted Sampling	b

Table 3: The pipeline design used with each sampling method for training with all instances in trajectories

The averaged results and their interquartile ranges over all 15 runs for random sampling, uncertainty sampling, and density sampling are shown in Figure 6.

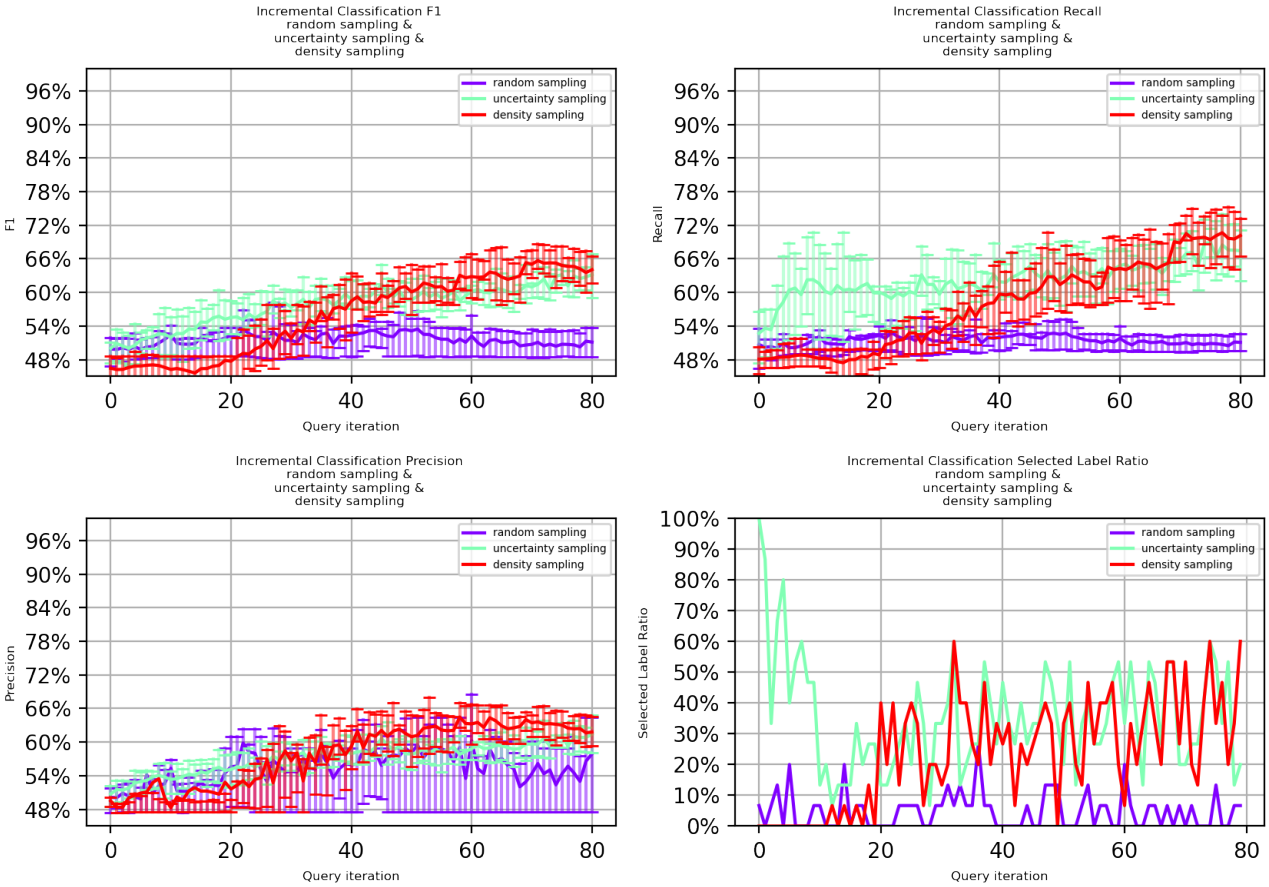

Figure 6: (Macro) F1, Recall, and Precision for training on all instances in selected trajectories, as well as the percentage of selected positive labels.

While only 15 runs total were performed for each active learning method, the dispersion of the performances is quite low for all three methods. This may be a result of the thousands of individual instances used for training leading to consistent behavior. Uncertainty sampling starts sampling looping trajectories with a high frequency, which corresponds to a quick rise in recall compared to the other two methods. Density-weighted sampling's recall even dips slightly while sampling only negative instances in the beginning. After the first 40 instances, both uncertainty sampling and density-weighted sampling reach very similar performance for all measures, showing greater performance compared to random sampling. However, performance after 80 instances for both of these methods only reaches around 65%.

6.2 Training on selected trajectory summarizing instances

When using the trajectory summarizing instances not only for selecting trajectories, but also for training, the speed at which experiments could be run increased dramatically. This allowed for the test parameters to be as shown in table 4.

Number of folds	5
Number of repetitions	30
Labeling budget	100 instances

Table 4: The experimentation parameters for training with trajectory summarizing instances

With these settings, a total of 150 runs of training and evaluation were performed for each tested method. As well as getting a smoother picture of performance for these methods, it also allowed testing with the QBC and xPAL approaches. The corresponding pipeline designs are shown in table 5.

Sampling Method	Pipeline Design
Random Sampling	c
Uncertainty Sampling	c
QBC Sampling	d
Density-Weighted Sampling	c
xPAL	c

Table 5: The pipeline design used with each sampling method for training with trajectory summarizing instances

The averaged results and their interquartile ranges over all test runs for random sampling, uncertainty sampling, QBC, density sampling, and xPAL sampling are shown in Figure 7. Results for corresponding significance tests can be found in Appendix B, and will be referenced when relevant.

To gain additional analytical insight, pre-testing was performed comparing random sampling using the dataset as described in Section 4 with random sampling when the majority class in the dataset was undersampled. The non-looping/looping ratio in the base dataset was 20:1, while in the undersampled dataset non-looping cases were removed at random to achieve a ratio of 4:1. The purpose of this experiment was to investigate the effect of increasing the rate at which instances of the minority class were sampled without the further bias or influence through the application of a guided selection strategy. The results can be seen in Figure 41 in Appendix C. The results suggest that a higher sampling rate of the minority looping trajectories results in a significant increase in the learning rate for macro recall and a slight increase in the learning rate for precision initially, but slightly lower learning rate for precision later on.

Looking at the performances for the active learning methods, we can see that all except xPAL clearly outperformed the random sampling approach. Considering that early testing with parameter changes had little effect on its performance (as discussed in Section 5.2) a possible explanation for the relatively low performance of xPAL in this case is that it is optimizing towards lowest expected misclassification error. Given a highly imbalanced dataset, a low misclassification score can easily be attained by always predicting the majority class. This may have skewed the selection strategy to favor non-looping cases over looping cases. As random selection also mostly selects the majority class due to the class imbal-

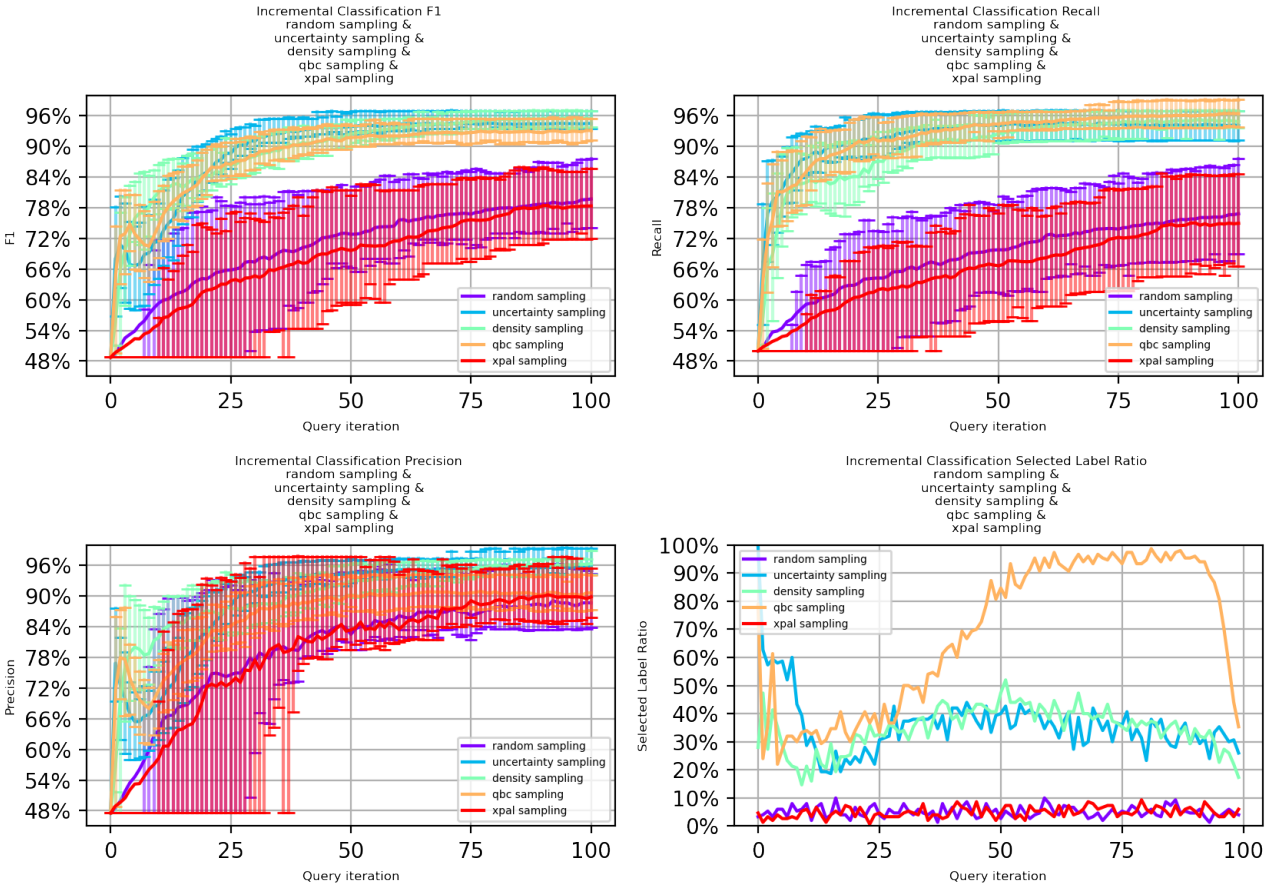

Figure 7: (Macro) F1, Recall, and Precision for training on selected trajectory summarizing instances, as well as the percentage of selected positive labels.

ance, this could lead to similar performance in precision. If this xPAL implementaion has enough of a preference toward the majority class that samples the minority class even less than random sampling does, this could explain that its recall performance is slightly worse.

Uncertainty sampling, QBC sampling, and density-weighted sampling all performed very similarly. For both uncertainty sampling and QBC sampling a dip in F1 performance can be seen as the start of training, as a result of this same initial dip in their precision scores. These dips may be explained by the switch from training with the initialization set to using active learning methods. As discussed in Section 5.1, all models were first initialized with 10 instances that were randomly selected, but with a ratio of 10:3 for the non-looping/looping ratio. As discussed above for the undersampled random sampling experiment, this shifts performance towards the majority class and thus to precision. Once uncertainty and QBC sampling are introduced, they start sampling looping cases with high frequency as can be seen from their selected label ratios. This could then result in this drop in precision and rapid increase in recall until they both stop sampling the looping cases so aggressively. Interestingly, the QBC strategy returns to the aggressive selection of looping cases. At this point the model had likely been trained with

enough instances for its ability to correctly classify the majority instances to not be affected as strongly. This does cause it to have a slightly lower precision and slightly higher recall after these 100 instances of training. Significance testing finds the performance difference between uncertainty sampling and QBC sampling to be insignificant.

Density-weighted sampling shows no dip, and instead shows slightly slower learning for recall. Given that it is likely that instances from the majority class will be in a more densely populated area of the feature space, density-weighted sampling didn't aggressively sample the minority class as much as uncertainty sampling and QBC sampling did. While ths resulted in slower recall learning, it mitigated the original drop in precision score, resulting in density-weighted sampling taking a short lead in F1 score until the other two catch up.

From the prior observation that recall increases and precision decreases when simply selecting randomly, it could be expected that for uncertainty sampling, query by committee sampling, and density-weighed sampling, the precision should generally be lower than it was for random sampling judging from the selected label ratios. However, all three of these outperform random sampling w.r.t. precision. This could be due to the increase in the rate at which precision improves for the minority looping class being greater than the decrease in the rate at which precision improves for the majority class. It could also be due to more specific qualities of the selected majority non-looping instances being more informative than any randomly selected ones, as one would hope when using an active learning approach.

Comparing performance between all sampling methods when using all instances in trajectories for training compared to using only the summarizing instances, it is clear that, for classification of complete trajectories, the latter method is much more effective. These findings were used for designing the approach to Task 2 for Section 6.3. However, first a further investigation was performed into the summarized-instance models and the trajectories they selected.

6.2.1 Additional Analysis

As an additional step to this research, it was of interest to not only test the performance differences resulting from using active learning methods, but to do some further analysis regarding important features and interesting trajectories. This was only performed for training using instances summarizing complete trajectories.

For each sampling method, a table shows Gini importance over the course of training in Appendix D. The training period was divided into nine bins, showing average importance within each bin. The following are some interesting observations regarding these feature importances:

- Random sampling and xPAL sampling result in extremely similar feature importances in the resulting models.
- The feature *rolling_std_speed_60min* (standard deviation of speed in the last 60 minutes) starts with high importance for all methods, but loses importance for all but random sampling and xPAL. Given both of these sampling methods mostly sample the non-looping trajectories, this feature could be exclusively important for identifying non-looping trajectories. An example of how this could happen is if, when speed varies little over the course of the last hour, this is a strong indicator of a non-looping tanker that is still underway, while strong variations in speed could be common to both looping and non-looping trajectories.
- The reverse case can be made for the feature *expanding_range_lon* (range of longitude over the complete trajectory). This feature is only mildly important for random and xPAL sampling, while gaining high importance for all other methods. That likely makes it a useful predictor for the looping trajectories. One possible explanation is that over the course of their trajectories, the looping cases will stay relatively close to the port of Rotterdam and have low longitudinal ranges, while non-looping tankers will have further destinations (especially for international destinations) and thus have to traverse farther longitudinally.
- Similarly, an important feature for all methods is *rolling_rel_max_lat_10min* (relative maximum latitude over the last 10 minutes). Latitude being important for the last 10 minutes of the trajectory makes sense given that if the trajectory were to be looping, it should be close in latitude to the port of Rotterdam. If it is non-looping, the odds of being close to specifically this latitude in the last 10 minutes of its trajectory are very slim. Interestingly, the relative minimum latitude in the last 10 minutes is not important for any method. Given that the window only covers the last minutes, only one of the two would be needed to represent latitudinal proximity to the port of Rotterdam.
- The only method that results in importance for orientation-based features is uncertainty sampling. This grows more toward the end of learning, specifically for features *rolling_max_orientation_10min* and *rolling_median_orientation_60min*. This change is interesting because performance for uncertainty sampling plateaus halfway through the active learning process. The increased importance on orientation seems to neither harm or improve model performance. One could imagine orientation to be of some use, considering it seems likely that a looping tanker would be oriented toward the port of Rotterdam near the end of its trajectory, but there may be too much variation due to other factors such as angles of approach.

To get an overview of which trajectories were often selected by the different sampling methods, as well as during what point in training, heat maps were plotted over the learning process for the frequency selection of the trajectories. Because of the large amount of trajectories, minimum sampling thresholds at each time point were used to leave out the less important trajectories. Different active learning methods required different thresholds to produce heat maps that allowed a good overview for a qualitative inspection. Random sampling was excluded since any frequently selected instance is only frequently selected through random chance. The different trajectories are shown over the x-axis, where green trip id's represent looping trajectories and the red trip id's represent non-looping trajectories. Selection frequencies were binned over the y-axis for every 10 consecutive queries. The following are the produced heat maps:

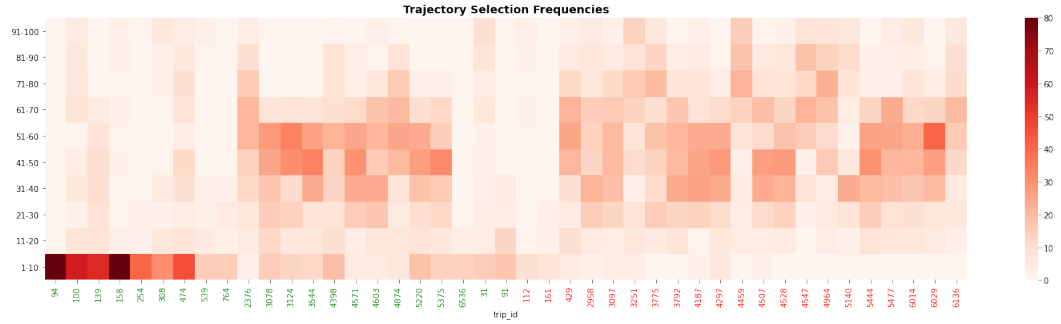

Figure 8: Uncertainty sampling selection frequency heat map. Minimum sampling threshold = 6

For uncertainty sampling there is a clear preference of a select few looping instances at the start of learning, with a more divided preference towards the middle of learning while it is reaching its performance plateau, with no specific preference toward the end.

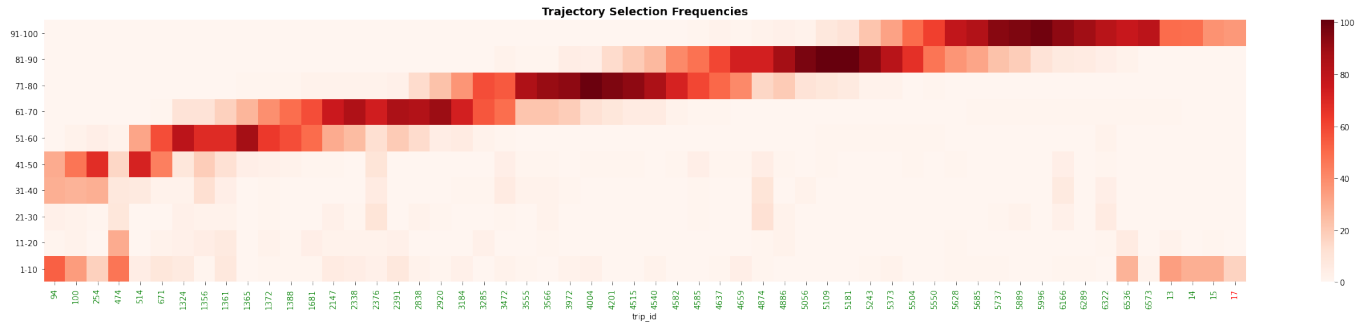

Figure 9: QBC sampling selection frequency heat map. Minimum sampling threshold = 8

QBC shows preference for a select few positive instance at the start of learning, followed by a short period of no preference for either class, followed by extreme preference toward a large set of specific looping trajectories over the rest of learning, in a consistent order.

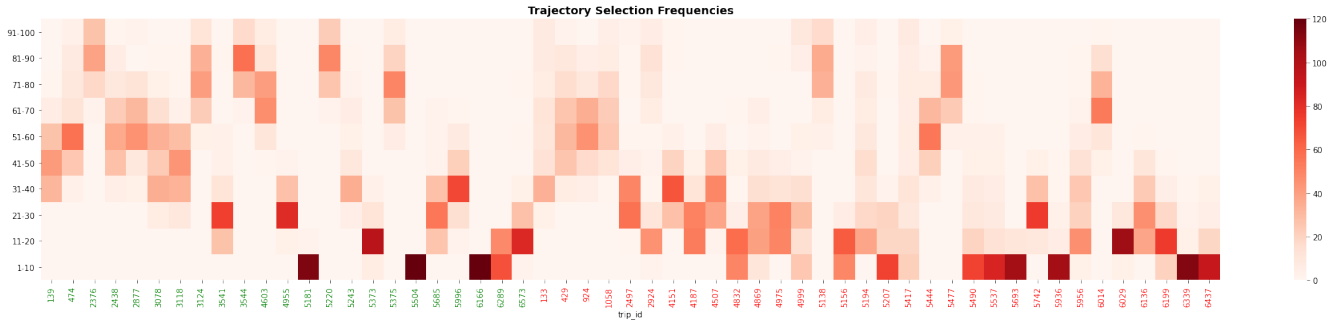

Figure 10: Density-weighted sampling selection frequency heat map. Minimum sampling threshold = 14

Density-weighted sampling shows strong preference toward both specific looping and non-looping trajectories in the start, after which preference for instances for both classes weakens over the course of training.

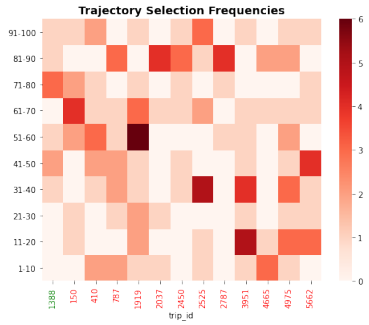

Figure 11: xPAL sampling selection frequency heat map. Minimum sampling threshold = 3

xPAL sampling shows preference toward almost exclusively non-looping instances, in a practically random distribution.

In terms of general preference toward looping or non-looping cases, the selected trajectories closely correspond to the ratio as was shown in Figure 7. After this an exploration followed into the specific instances that were frequently selected by the active learning approaches. First a few examples will be show for trajectories frequently selected at the start of learning, and then a few trajectories frequently selected near the end of learning. A larger selection of frequently selected trajectories can be found in Appendix E. The color gradations in the figures indicate the speed in knots (nautical miles per hour) of the tanker at that point in the trajectory:

The following figures show a few examples of looping trajectories that were frequently selected at the start of learning.

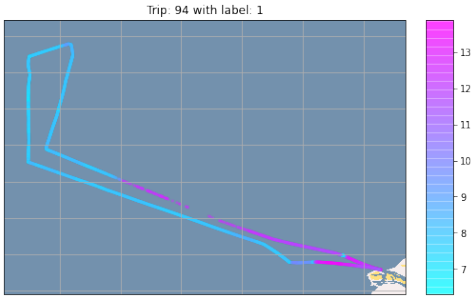

Figure 12: Looping trajectory frequently selected near the start of learning (uncertainty sampling and QBC sampling)

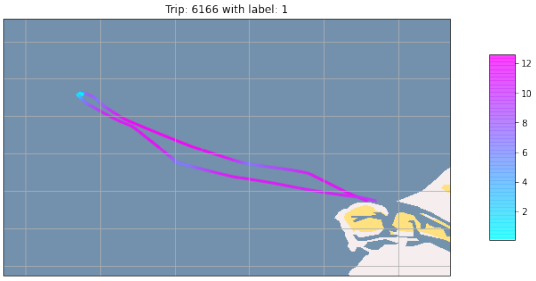

Figure 13: Looping trajectory frequently selected near the start of learning (density-weighted sampling)

The purpose of active learning approaches is to try and find the decision boundary between classes as quickly as possible. Sampling instances near the decision boundary would mean that the selected trajectories should be similar in terms of features. Both Figures 12 and 13 show clear looping trajectories. However, trip 94 is moving slowly throughout most of the trajectory, and trip 6166 shows a point that is likely stationary. Experts consider trajectories such as 6166 to be anchoring. This is behavior observed when a tanker is anchored outside of port overnight to avoid having to pay fees. Since looping trajectories are usually fairly short, the longer duration of both of these looping trajectories adds similarity to typical non-looping trajectories.

The following figures show a few examples of non-looping instances that were frequently selected near the start of learning.

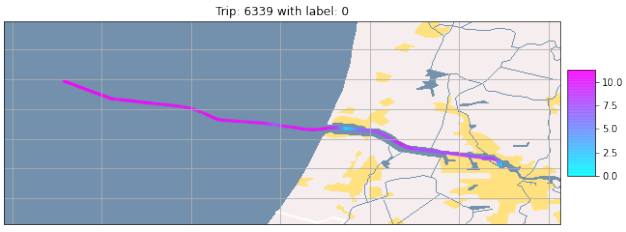

Figure 14: Looping trajectory frequently selected near the end of learning (density-weighted sampling)

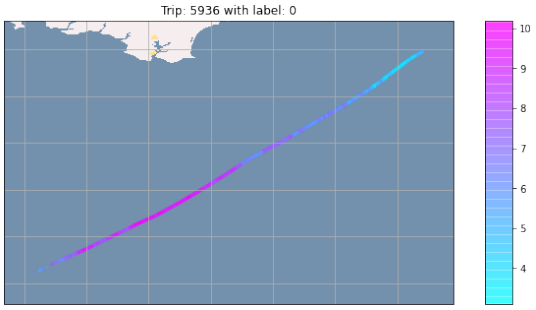

Figure 15: Looping trajectory frequently selected near the end of learning (density-weighted sampling)

Figure 14 shows a non-looping trajectory that is leaving the port of Amsterdam instead of the port of Rotterdam. This is likely due to an error in preprocessing. The fact that the entire trajectory takes place in a very small longitudinal space is similar to a looping trajectory in that respect. Figure 15 shows a trajectory that takes place in its entirety just south of the United Kingdom, near Plymouth. Considering this trajectory doesn't start from the Port of Rotterdam either, this is also an erroneous trajectory. For such an example similarities to typical looping trajectories are less clear, other than perhaps a period of

slower movement that one might see for a looping tanker that is at its turning point.

The following figures show a few examples of looping instances that were frequently selected at near the end of learning:

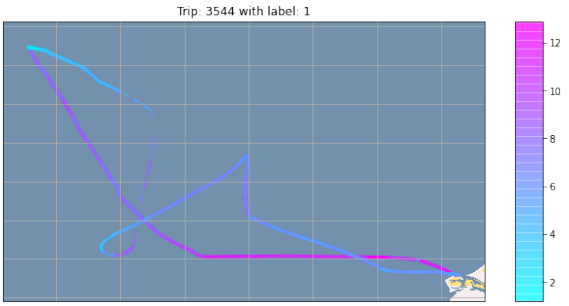

Figure 16: Non-looping trajectory frequently selected near the end of learning (density-weighted sampling)

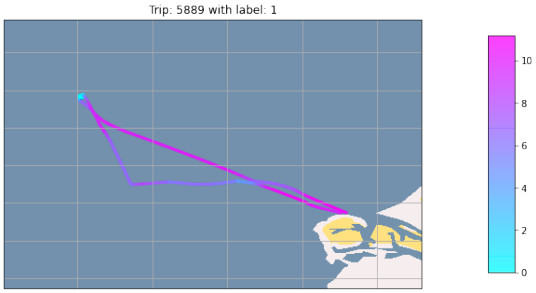

Figure 17: Non-looping trajectory frequently selected near the end of learning (QBC sampling)

We can see for Figures 16 and 17 even more looping trajectories that are clearly of longer durations than a non-stop looping trip would be. For looping trajectories seemingly not much has changed for frequently selected trajectories. Considering it is the minority class these instances are likely still quite informative toward the end of learning.

The following figures show a few examples of non-looping instances that were frequently selected near the end of learning.

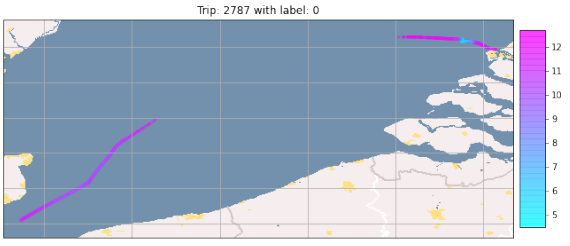

Figure 18: Non-looping trajectory frequently selected near the start of learning (density-weighted sampling)

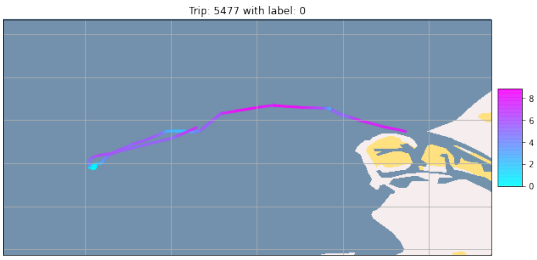

Figure 19: Non-looping trajectory frequently selected near the start of learning (density-weighted sampling)

Figure 18 shows a trajectory for which a large part is missing. This shows a flaw in the interpolation option used when extracting the original data. Similarity to looping trajectories may stem from the fact that, due to the missing middle part, a relatively larger part of the trajectory instances is close to the port of Rotterdam. This would affect the features generated by the sliding and expanding window approaches. Figure 19 shows a trajectory that likely anchored and is making its way back to the port of Rotterdam. That means it is likely that this would have performed a loop, but it was anchored outside of the port for so long that the return trip was cut short by the 48 hour limit.

From the examples seen, it generally seems that most of the looping trajectories that were selected show similarities with typical non-looping trajectories, meaning that sampling is likely able to find the decision boundary fairly effectively. This was also true for the non-looping trajectories, but especially the last trajectories only share similarities to typical looping trajectories due to data processing errors.

6.3 Task 2: Partial Trajectory Classification

The experimental approach to partial trajectory classification was both guided by the prior research [2] as well as informed by the experimentation and results for Task 1. To evaluate performance for partial trajectories, a set of time points at specific full hours after departure within the maximum time period of 48 hours were selected. These time points were: 10h, 16h, 20h, 25h, 30h, 35h, and 40h. Based on expert opinion, it is very unlikely that any anomalous behavior will occur in the first ten hours of a trajectory. The purpose of en route classification is to eventually intervene before an act such as zeezwaaien can occur, or at the least have the ability to mobilize inspectors to the port of arrival. Because of this, any classification beyond 40 hours into the trajectory is likely too late.

Training a classifier using all instances within trajectories should allow for the classifier to make predictions for different time points of trajectories. However, the additional overhead of re-training classifiers with all these instances throughout the active learning process, as well as the relatively low performance of the resulting classifiers, the choice was made to use pipeline designs c and d for training only on summarizing instances. New datasets for each different time point were made in accordance to what could be seen in Figure 5. With different datasets for different time points, a different classifier was trained for each specific time point following the same methodology and settings as for Task 1. For quick reference these are shown in Tables 6 and 7

Number of folds	5
Number of repetitions	30
Labeling budget	100 instances

Table 6: The experimentation parameters for training with partial trajectory summarizing instances

Following this methodology for partial trajectory classification, in practice, the resulting models would be applied consecutively to an en route ship as soon as it reaches the corresponding time point. Given enough time points, the time span between classifications should be short enough that the delay before a new classification (compared to consistent real time classification) should still leave enough room for a timely response.

Sampling Method	Pipeline Design
Random Sampling	c
Uncertainty Sampling	c
QBC Sampling	d
Density-Weighted Sampling	c
xPAL	c

Table 7: The pipeline design used with each sampling method for training with partial trajectory summarizing instances

To give some additional context to the results, Table 8 shows the label distributions at each time point, including complete trajectories for reference. The upper two columns show the label distributions for the different datasets produced for only training (pre-time point only) and both testing and training (at or max. 30 minutes before time point). The bottom two columns show how these would be divided on average between the train and test folds for k-fold validation. Figures 20 - 24 show the performance for the different sampling methods at the different time points. Results for corresponding significance tests can be found in Appendix B, and will be referenced when relevant.

		Comp.	10h	16h	20h	25h	30h	35h	40h
Train	1	NA	4	16	26	39	60	70	78
only	0	NA	149	146	146	265	303	358	433
Train	1	88	44	36	35	29	15	10	7
& Test	0	1679	944	845	857	785	777	723	674
Avg. in	1	70	35	45	54	62	72	78	84
train folds	0	1343	755	822	832	893	925	936	972
Avg. in	1	18	9	7	7	6	3	2	1
test fold	0	336	189	169	171	157	155	145	135

Table 8: Label distributions for the complete trajectory and the different partial trajectory datasets, and an average (rounded) label distribution for each train and test set produced for 5-fold cross validation. Note that the partial trajectory datasets don't include all available instances due to an oversight explained at the end of Section 5.3

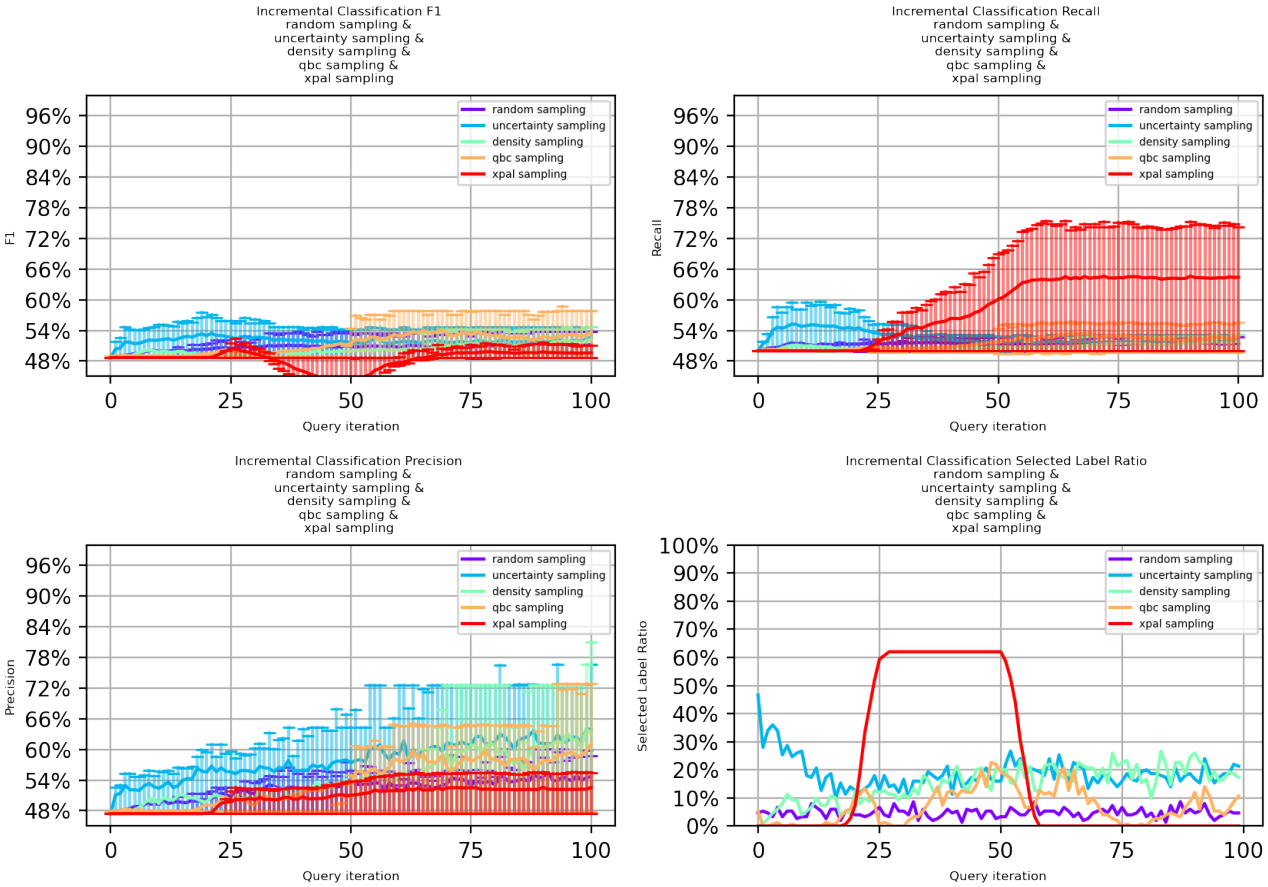

Figure 20: Partial Trajectories at 10 hours. (Macro) F1, recall, and precision for training selected trajectory summarizing instances, as well as the percentage of selected positive labels.

For most methods the performance at 10 hours was as expected. While uncertainty sampling initially selected more looping trajectories than the other methods, thus giving an initial small increase in recall and precision for the looping class, performance is generally quite low compared to learning with complete trajectories. Considering this is very early in the trajectory, the differences between looping and non-looping trajectories are likely to be very small. The outlying active learning method in this case is xPAL. After ∼20 instances that were seemingly purely non-looping, there is a consistent switch to sampling positive instances about 60% of the time on average until the ∼60th instance. This results in a significant performance increase for recall, as well as a smaller one for precision, throughout this period. This may have been a result of the presumed small differences between the trajectories of the different classes up to this point in time. xPAL uses kernel density estimation as a part of its expected performance impact calculations for label selection. Even if the initialization set is skewed toward the majority class, combined with the issue of optimization toward misclassification error, if the trajectories in the different classes are similar this increases the odds of selecting a trajectory from the minority class. In turn, adding labels for the minority class also increases its representation in the kernel density estimate, thus making minority class classification seem more valuable. Significance testing at this time point finds no

significant difference between the performances of random sampling and density-weighted sampling, as well as between random sampling and QBC sampling.

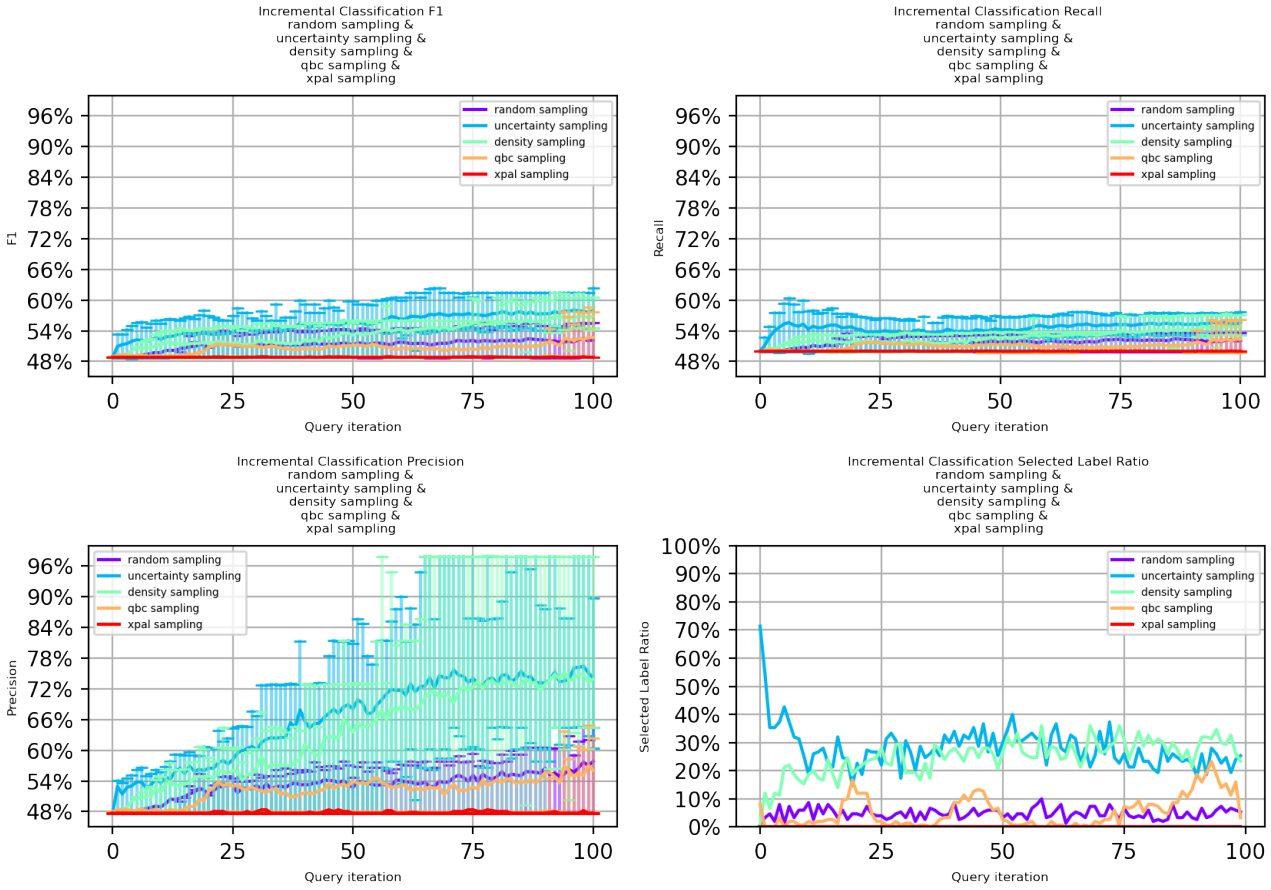

Figure 21: Partial Trajectories at 16 hours. (Macro) F1, recall, and precision for training selected trajectory summarizing instances, as well as the percentage of selected positive labels.

After 16 hours a point was reached where the trajectories in the different classes likely weren't similar enough for xPAL to presume a high performance increase for labelling an instance that ends up being a looping trajectory. No real performance increase for xPAL is seen across the rest of the tested time points. Significant improvements were seen in precision, particularly for uncertainty sampling and density-weighted sampling. Again this is accompanied by higher sampling rates for the looping class. The dispersion of performance is very varied however. QBC sampling had some spiked periods of sampling looping instances, but did not outperform random sampling. It may have been the case that at this time point there weren't enough properly informative positive instances available for training at this point to allow the different committee members to form a good set of different hypotheses.

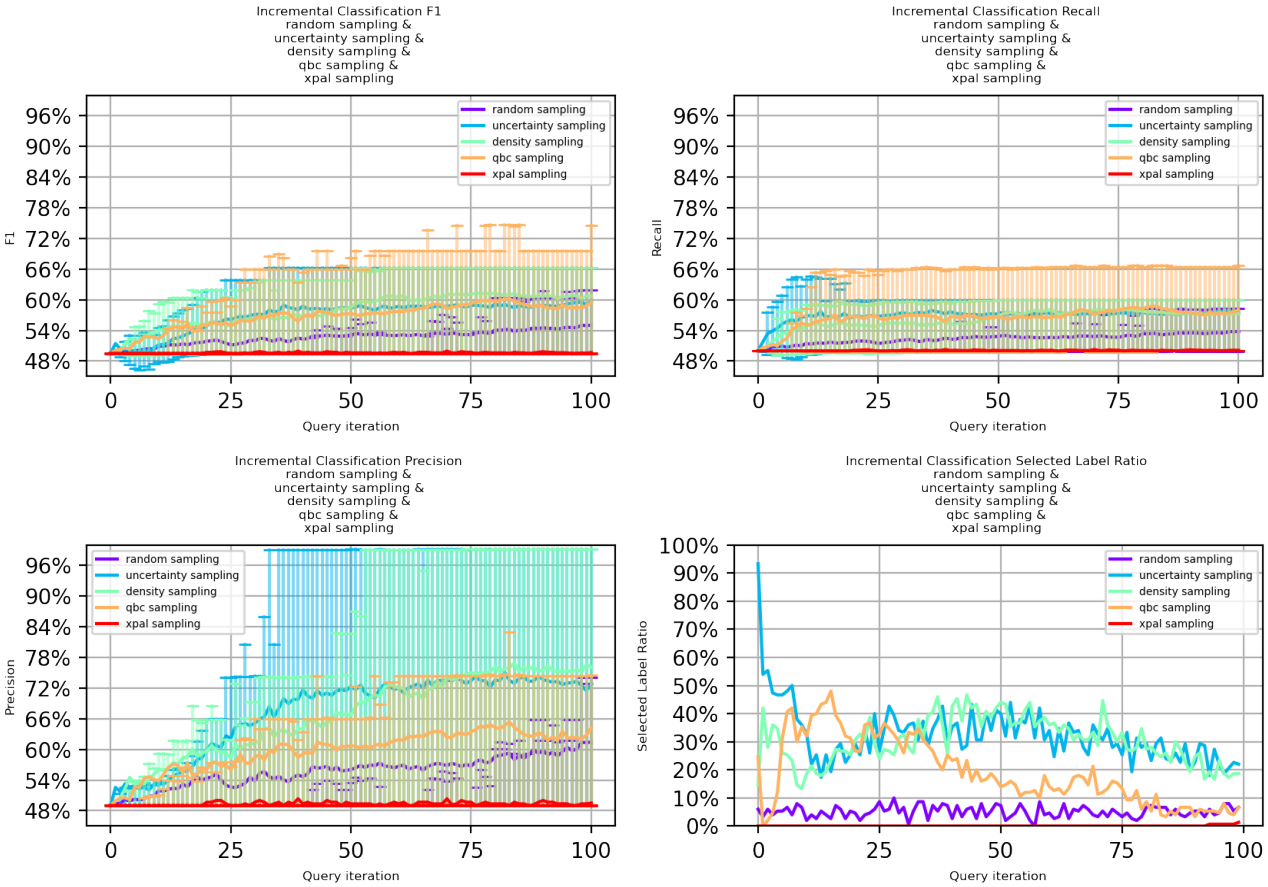

Figure 22: Partial Trajectories at 30 hours. (Macro) F1, recall, and precision for training selected trajectory summarizing instances, as well as the percentage of selected positive labels.

Over the course of the time points 20 hours, 25 hours (Appendix C), Figures **??**, 43) there was a consistent increases in learning rates for all methods in precision and consequently F1 scores, reaching a peak at 30 hours (Figure 22). Uncertainty sampling and density-weighted sampling increased their lead over the other methods in terms of precision with statistically insignificant differences both at 20, hour,s 30 ours, and beyond. QBC sampling's learning rate gradually overtook random sampling with regard to both recall and precision, with its recall and F1 score being very similar to those of uncertainty sampling and density-weighted sampling. At 30 hours the difference between performance for QBC and both uncertainty sampling and density-weighted sampling was not statistically significant. At this time point here was also a notable increase in the dispersion of performances for all methods. This is likely due to the increasingly small number of looping trajectories in the test set. As can be seen in Table 8, on average only 3 looping trajectories were in the test set for each round of evaluation. This was due to trajectories at time points ending before the current time point being used exclusively for training, as described in Section 5.3.1. Because of this, when evaluating at later time points, more of the looping trajectories have ended before this time point and were exclusively used for training. With only 3 looping trajectories in the test set on average any one or more or less correct classifications of the looping trajectories resulted

in large changes in performance.

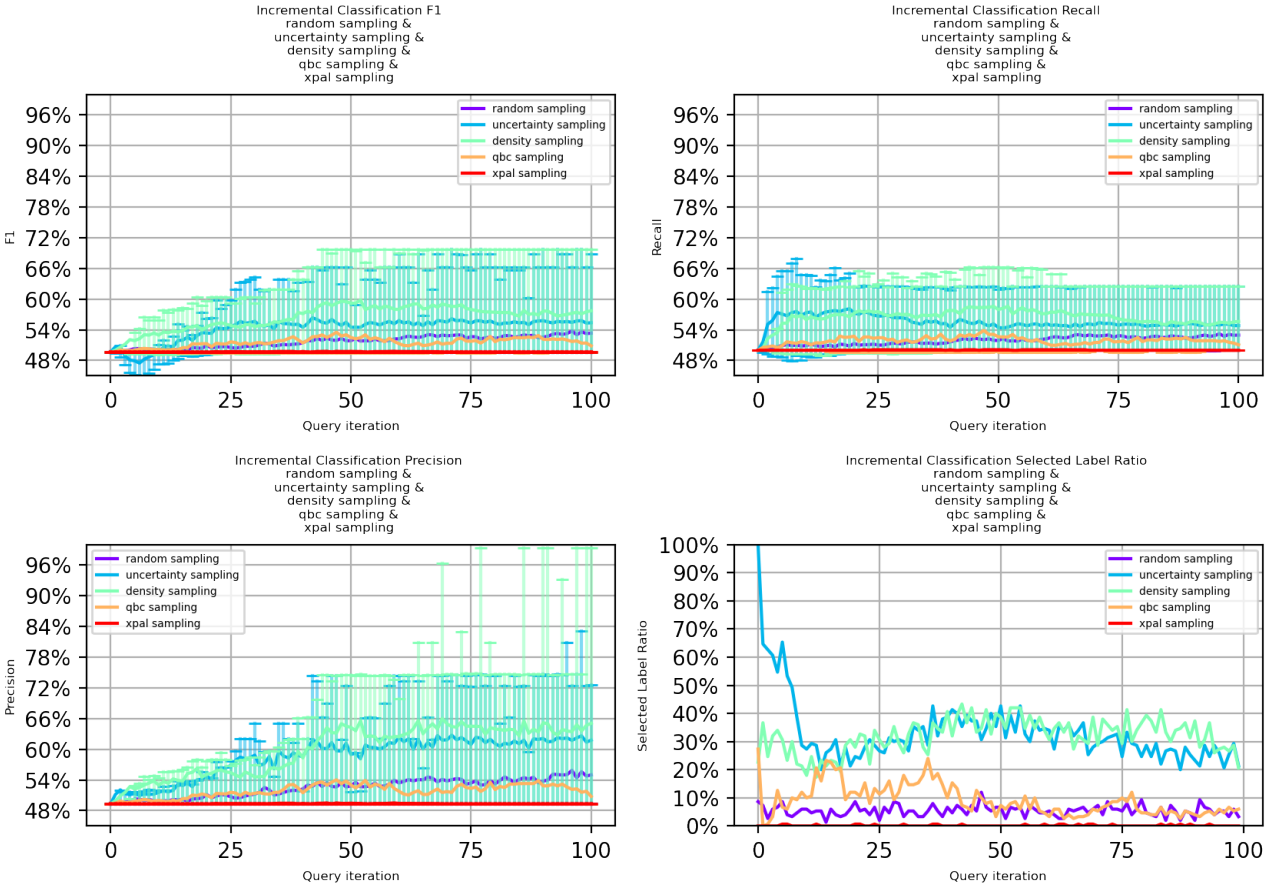

Figure 23: Partial Trajectories at 35 hours. (Macro) F1, recall, and precision for training selected trajectory summarizing instances, as well as the percentage of selected positive labels.

After 35 hours there is a general decrease in performance, with the exception of density-weighted sampling showing no decrease for recall. With the number of looping trajectories in the training set increasing, but the number of looping trajectories at the evaluated point in time decreasing (2 per evaluation round at this time point), there may have been an over-representation of the looping trajectories at this point resulting in an increased number of occurrence of false positives where non-looping trajectories were falsely classified as looping trajectories. From this time point on QBC showed no statistically significant performance different with random sampling.

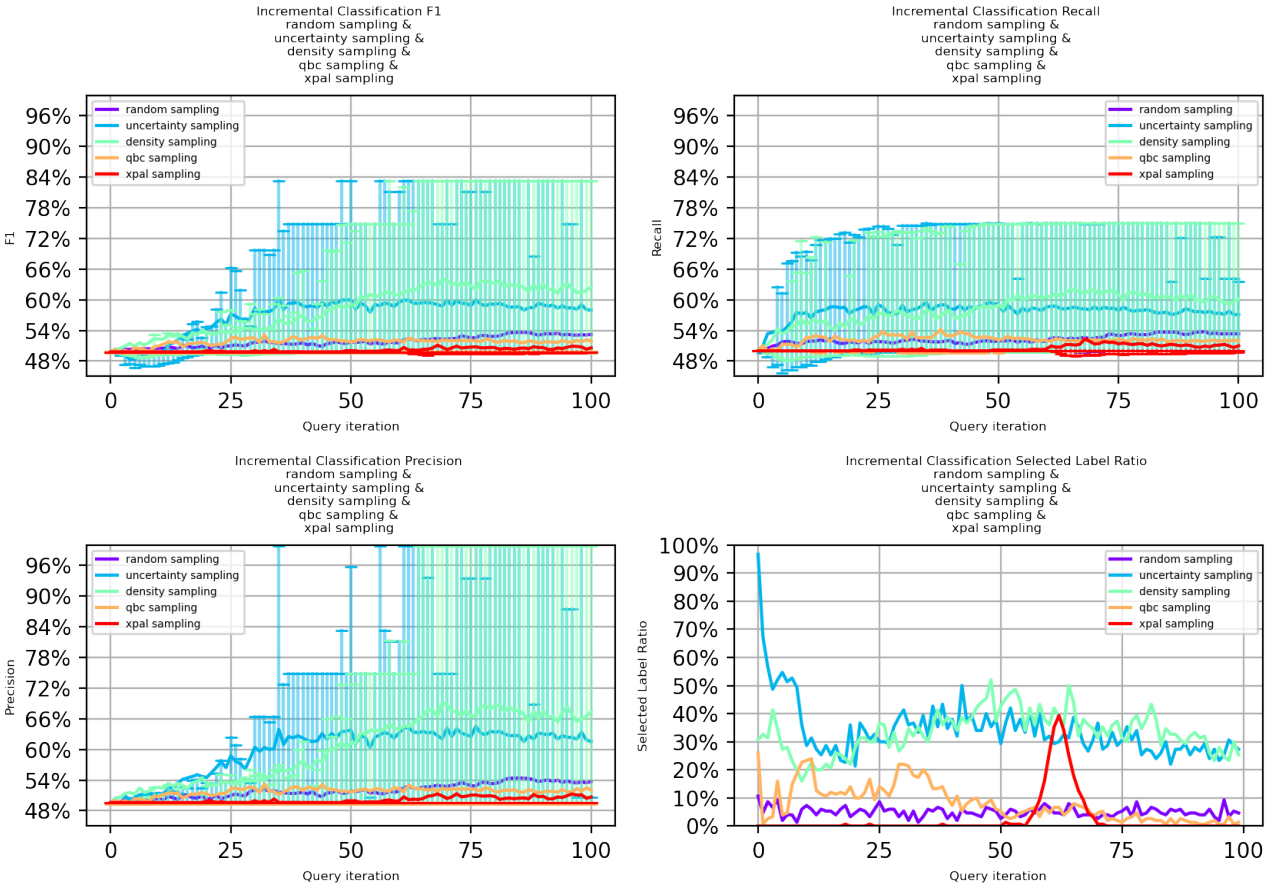

Figure 24: Partial Trajectories at 40 hours. (Macro) F1, recall, and precision for training selected trajectory summarizing instances, as well as the percentage of selected positive labels.

After 40 hours, with only a single looping trajectory in the evaluation set, the dispersion of performance for recall was higher than at any of the former time points. However, a small average increase in performance is seen for uncertainty sampling and density-weighted sampling over performance at the 35 hour time point. Only one looping trajectory on average was in the test set for each round of evaluation. Considering both uncertainty sampling and density-weighted sampling had a greater tendency to sample looping trajectories than the other methods, they likely had a greater chance at correctly classifying that one instance, resulting in the looping-half of the macro scores to be significantly boosted. For all other methods performance stayed roughly the same.

Similar to the results for classifying complete trajectories using summarizing instances, the active learning approaches consistently matched or outperformed random sampling. However, for none of the time points the performance matched that for complete trajectory classification.

As a final overview, Table 25 gives a performance overview in terms of average ALC scores for both complete and partial summarized trajectory classification. While this doesn't capture how the learning rate changed exactly over the course of training, the values do generally correspond to the impressions given by the performance figures.

	Comp.	10h	16h	20h	25h	30h	35h	40h
random	19.47	14.31	14.41	14.48	14.56	14.63	14.48	14.59
uncertainty	26.50	14.67	15.28	15.46	15.72	16.00	15.58	16.23
density	25.87	14.16	14.72	15.39	15.33	15.76	15.86	16.42
qbc	26.55	14.37	14.21	14.00	15.16	15.88	14.49	14.58
xpal	18.71	16.43	13.90	NaN	13.90	13.92	13.91	14.06

Figure 25: Average ALC scores for the different active learning methods over the summarized trajectory learning scenarios. Since each time point used different training and test sets, scores should only be compared between the active learning methods at each separate time point.

7 Conclusions

This thesis consisted of an exploration into the use of active learning methods for the detection of zeezwaaien. For the purpose of having enough labeled data available for experimentation and evaluation, the feature *looping behavior* was used as a proxy for zeezwaaien. The approach taken was chosen to be moderately non-specific to allow some insight into how the performance on this dataset could translate to the more general problem of active learning for high class imbalance trajectory data given little prior information. This section will first go over the posed research questions and their answers in the light of the research performed for this thesis. This is followed by a more detailed discussion of specific aspects of the findings in Section 7.1. Finally, some suggestions are made for future work in Section 7.2.

The first research question:

- *How can active learning approaches be applied to improve supervised machine learning classifiers of high class imbalance trajectory data with little to no prior information (using looping detection of tankers in the North Sea leaving the port of Rotterdam as an example)?*

By applying a variety of active learning approaches in general forms to the problem of classifying looping behavior for these trajectories, this research was able to gain some insight into how these different types of approaches perform as out-of-the-box solutions to this type of problem (Task 1). This question was answered in two parts corresponding to the following posed sub-questions:

The first sub-question:

- How can the active learning pipeline be adjusted to suit trajectory data with trajectory-wide labels?

The key to applying active learning to trajectory data with trajectory-wide labels was the summarization of these trajectories to single instances. This allowed for the application of the traditional pool-based active learning pipeline on the new dataset of summarized instances, with the exception of the use of a separate classifier for sample selection for QBC sampling. For complete trajectory classification, a summarization based on expanding and sliding window features resulted in satisfactory results. Concurrent research at the IDlab using the same dataset [69] indicates that high performance can be achieved in a non-active learning setting when training using all instances in trajectories. However, the active learning experiments with pipeline adjustments to allow training with all instances in trajectories showed much slower learning for complete trajectory classification.

The second sub-question:

- Can active learning approaches increase the rate of learning given the described problem scenario?

All but one of the tested active learning approaches significantly increased the learning rate over random sampling, resulting in better performance w.r.t. (macro) scores for F1, precision, and recall. No statistically significant difference in performance was seen for performance between uncertainty sampling and QBC sampling. Given the additional pre-testing and training time that were involved in QBC sampling, this suggests that in this type of problem scenario the more simple uncertainty sampling would likely preferable. Uncertainty sampling and density-weighted sampling plateau at approximately the same levels of performance. The learning curves suggest uncertainty sampling would be a better choice for the detection of anomalies early in learning due to a faster initial increase in recall, while density-weighted sampling balances this with cost of misclassification through a more steady rise in precision. The xPAL implementation optimizing towards reduction of misclassification error showed similar learning to random sampling, with performance for recall being slightly worse.

The second research question:

- *Can the approaches designed to answer the main research question be extended to improve learning performance for en route classification of looping behaviors?*

The findings for the first research question motivated the use of summarized trajectory instance training as the main approach to answering the second research question. The same active learning approaches in their general forms were used, now for classifying looping behavior for partial trajectories (Task 2). This question was also answered in two parts corresponding to the following posed sub-questions:

The first sub-question:

- How can the design be adjusted to classify partial trajectories rather than complete trajectories?

The same pipeline designs that were used for summarized complete trajectory classification could be used for partial trajectory classification, but now summarizing trajectories up to specific points in time into the trajectory. Different models were trained for different pre-specified points of elapsed time into the trajectories. For each time point the active learning approaches were applied using only instances summarizing trajectories up to that specific time point for training and evaluation. Summarizing instances for complete trajectories of trips with shorter durations than the specified time point were only used for training.

The second sub-question

- Can active learning approaches increase the rate of learning compared to passive learning for en route looping detection?

Depending on the evaluated point in time, all active learning approaches other than xPAL showed either matching or increased learning rates compared to random sampling. To apply this to en route classification in a real-world scenario, the different models for different time points would be used as a tanker

reaches the specific time point. For the earliest time point into the trajectory these methods showed no real improvement over random sampling. With later time points performance increased across the board, with uncertainty sampling and density-weighted sampling showing faster increasing precision than QBC sampling. Towards the end of the observed time points overall performance decreased, likely due to almost no looping trajectories being present at these later time points to be used for training or evaluation.

To summarize the above: Summarizing trajectories into single instances to use for selecting trajectories and then training using the selected summarizing instances was an effective approach for using active learning to classify trajectories given the problem scenario. When applied, active learning showed improvements over the rate of learning in nearly all situations. Particularly uncertainty sampling and density-weighted sampling never seemed at risk of underperforming compared to random sampling. Considering the performance for the looping classification using minimal guided selection of parameters (excluding QBC), active learning may be promising for other (trajectory) class imbalance scenarios where there is little to no prior information available. These statements also hold for en route classification, however there may be better approaches than multiple time-specific models, particularly when the distribution of the labels is also imbalanced in relation to the specific time point.

7.1 Further Discussions

This section adds some additional discussion regarding the methodologies and experiments in this thesis.

- For a situation in which not much is known beforehand, the use of an approach with many different hyperparameters is generally counter-intuitive. This was a problem for both QBC sampling and xPAL. For QBC sampling the pre-experimentation to find the most suitable configuration of committee members is both not possible if no labeled data is available beforehand, and it disqualified its performance as a look into general QBC performance for class imbalance trajectory classification. xPAL allows for the measure toward which it is optimized to be changed [41], and with at least the expectation of high class imbalance it would make sense to do so. However, this was not part of the implementation used in this thesis, and changing it would have taken a considerable amount of time that was not available during this thesis project. Its addition to this thesis is the demonstration of how strong the effect of an active learning method that is not suited or set up properly for the problem to which it is applied.

- The partial trajectory classification approach of training for and classifying at specific time points did allow for en route classification, but performance was significantly lower than was achieved in concurrent non-active learning research at the IDlab using the same dataset. For complete trajectories, training on summarizing instances resulted in competing performances within a 100

instances compared to concurrent research using all instances of all trajectories for training (hold-out set excluded) [69]. This difference is likely explained by multiple factors. This includes the fact that classifying looping earlier in trajectories is overall a more difficult problem than for complete trajectories, so more instances are likely needed to get better performance. More specific to the approach however, there is a lot of variation between trajectories at the same time point. At the end of complete trajectories, all looping trajectories will be exhibiting behavior very specific to entering the port of Rotterdam (captured particularly in the sliding window features for the last 10 and 60 minutes). However, after X hours, different looping trajectories will be at completely different parts of their looping trajectories depending on the size of the loop and its speed. With more time for experimentation it may have been worth trying to train a single classifier with instances from summarized trajectories at multiple different time points.

- Particularly for the non-looping class, the additional analysis in Section 6.2.1 seemed to have a tendency to select non-looping instances for which there were data-related issues. While not the purpose of active learning, this may be a useful feature for data cleaning. Particularly in a pool-based scenario, like a scenario where visualizations of complete trajectories are shown to inspectors for labelling, effective active learning would then both increase the rate of learning while aiding with data cleaning at the same time.

7.2 Future work

This section discusses some possible directions of future work based on the results from this thesis, as well use-cases for the IDlab.

- Following the discussion about how trajectories with the same label at different points in time may still exhibit lots of variation in behavior, it may be useful to look to other methods of feature generation of summarization. "Milestone" type features that relate to specific behaviors of the traversing object that either have or haven't occurred may be a very informative (e.g. for looping classification, tanker rotated 180 degrees in the trajectory so far, tanker has left coastal area during any time point in the trajectory so far, etc.). These would have to be crafted using domain knowledge, being heavily dependent on the knowledge of inspectors. Other notable methods of feature generation for trajectories include perennial object mining [73] and trajectory clustering [63].

- The pool-based scenario in this thesis is of use specifically w.r.t. zeezwaaien detection for the future purpose of reducing complete trajectory labeling. En route classification of zeezwaaien in a deployed learning situation would likely be more akin to a stream-based scenario. In this case, the decision has to be made to inspect a tanker or not while it is en route. This begs the question

at which point during the trajectory to decide whether to inspect or not. A possible direction of research for this is aleatoric uncertainty [67]. Estimating aleatoric uncertainty given an en route tanker would give a measure of how high the irreducible uncertainty is for classifying this tanker at this time point. With high aleatoric uncertainty, applying a label to the tanker would result in very little classifying power gained, since its uncertainty is highly irreducible. Experimentation would then involve finding the appropriate threshold of aleatoric uncertainty before considering whether inspecting an instance will be informative for the learner.

- For a deployed learning situation, active learning could be used as a tool to help inspectors at the ILT to decide when to perform inspections. However, there are more factors at play that go into making such decisions. Active learning serves to streamline the exploration aspect of learning, with the aim of faster learning for the classifier. At the same time it is desirable to exploit what we have learned to identify illegal activities. Considering that the successful active learning approaches significantly increased sampling of our target class, these approaches were beneficial with regard to both of these factors. Since these approaches generally try to sample close the decision boundary between classes, an increase in sampling of the minority class would be expected. A third important factor is ethical concerns. Measures need to be taken to be able to address concerns of ethical equity. Ways of addressing this include a randomization factor for inspections and explainable methodologies. In order to estimate the optimal use of active learning for scenario's closer to the en route use-cases, experiments including these different factors and interplay between them would have to be performed.

References

[1] Ministerie van Binnenlandse Zaken en Koninkrijksrelaties. *Internationaal Verdrag Ter Voorkoming van Verontreiniging Door Schepen, 1973, Zoals Gewijzigd Door Het Protocol van 1978 Daarbij*. URL: https://wetten.overheid.nl/BWBV0003241/2021-01-01#Verdrag_2_VerdragtekstII_3 (visited on 05/12/2021).

[2] S. Cheliku. *APPLYING AI-BASED ANOMALY DETECTION TECHNIQUES TO IDENTIFY WASTE DISCHARGERS AT THE NORTH SEA*. 2020. URL: http://localhost/handle/1874/399716 (visited on 04/09/2021).

[3] Tom M. Mitchell. *Machine Learning*. McGraw-Hill Series in Computer Science. New York: McGraw-Hill, 1997. 414 pp. ISBN: 978-0-07-042807-2.

[4] The MIT Press. *Introduction to Machine Learning, Fourth Edition — The MIT Press*. URL: https://mitpress.mit.edu/books/introduction-machine-learning-fourth-edition (visited on 04/10/2021).

[5] Konstantinos G. Liakos, Patrizia Busato, Dimitrios Moshou, et al. "Machine Learning in Agriculture: A Review". In: *Sensors* 18.8 (8 Aug. 2018), p. 2674. DOI: 10.3390/s18082674. URL: https://www.mdpi.com/1424-8220/18/8/2674 (visited on 04/10/2021).

[6] Praveen Kumar Donepudi. "AI and Machine Learning in Retail Pharmacy: Systematic Review of Related Literature". In: *ABC Journal of Advanced Research* 7.2 (2 Nov. 15, 2018), pp. 109–112. ISSN: 2312-203X. DOI: 10.18034/abcjar.v7i2.514. URL: https://i-proclaim.my/journals/index.php/abcjar/article/view/514 (visited on 04/10/2021).

[7] Zulfadzli Drus and Haliyana Khalid. "Sentiment Analysis in Social Media and Its Application: Systematic Literature Review". In: *Procedia Computer Science*. The Fifth Information Systems International Conference, 23-24 July 2019, Surabaya, Indonesia 161 (Jan. 1, 2019), pp. 707–714. ISSN: 1877-0509. DOI: 10.1016/j.procs.2019.11.174. URL: https://www.sciencedirect.com/science/article/pii/S187705091931885X (visited on 04/10/2021).

[8] Tom M. Mitchell. *The Need for Biases in Learning Generalizations*. Carnegie Mellon University, 1980, p. 3.

[9] Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, et al. *A Survey on Bias and Fairness in Machine Learning*. Sept. 17, 2019. arXiv: 1908.09635 [cs]. URL: http://arxiv.org/abs/1908.09635 (visited on 04/10/2021).

[10] Daniel Fuchs. "The Dangers of Human-Like Bias in Machine-Learning Algorithms". In: *Missouri S&T's Peer to Peer* 2.1 (May 4, 2018). URL: https://scholarsmine.mst.edu/peer2peer/vol2/iss1/1.

[11] Sanjoy Dasgupta and Daniel Hsu. "Hierarchical Sampling for Active Learning". In: *Proceedings of the 25th International Conference on Machine Learning - ICML '08*. The 25th International Conference. Helsinki, Finland: ACM Press, 2008, pp. 208–215. ISBN: 978-1-60558-205-4. DOI: 10.1145/1390156.1390183. URL: http://portal.acm.org/citation.cfm?doid=1390156.1390183 (visited on 04/10/2021).

[12] Mario Arduini, Lorenzo Noci, Federico Pirovano, et al. *Adversarial Learning for Debiasing Knowledge Graph Embeddings*. Comment: MLG 2020 at the ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD) 2020. Feb. 17, 2021. arXiv: 2006.16309 [cs, stat]. URL: http://arxiv.org/abs/2006.16309 (visited on 04/12/2021).

[13] Luke Darlow, Stanisław Jastrzebski, and Amos Storkey. *Latent Adversarial Debiasing: Mitigating Collider Bias in Deep Neural Networks*. Comment: 10 pages, 4 figures, submitted to AISTATS 2021. Nov. 19, 2020. arXiv: 2011.11486 [cs]. URL: http://arxiv.org/abs/2011.11486 (visited on 04/12/2021).

[14] Brian Hu Zhang, Blake Lemoine, and Margaret Mitchell. "Mitigating Unwanted Biases with Adversarial Learning". In: *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*. AIES '18. New York, NY, USA: Association for Computing Machinery, Dec. 27, 2018, pp. 335–340. ISBN: 978-1-4503-6012-8. DOI: 10.1145/3278721.3278779. URL: https://doi.org/10.1145/3278721.3278779 (visited on 04/12/2021).

[15] Christian Reimers, Paul Bodesheim, Jakob Runge, et al. *Towards Learning an Unbiased Classifier from Biased Data via Conditional Adversarial Debiasing*. Mar. 10, 2021. arXiv: 2103.06179 [cs]. URL: http://arxiv.org/abs/2103.06179 (visited on 04/12/2021).

[16] Vili Podgorelec, Peter Kokol, Bruno Stiglic, et al. "Decision Trees: An Overview and Their Use in Medicine". In: *Journal of Medical Systems* 26 (2002), pp. 445–463.

[17] *Decision Tree Tutorials & Notes — Machine Learning*. HackerEarth. URL: https://www.hackerearth.com/practice/machine-learning/machine-learning-algorithms/ml-decision-tree/tutorial/ (visited on 09/25/2021).

[18] Leo Breiman. "Bagging Predictors". In: *Machine Learning* 24.2 (Aug. 1996), pp. 123–140. ISSN: 0885-6125, 1573-0565. DOI: 10.1007/BF00058655. URL: http://link.springer.com/10.1007/BF00058655 (visited on 09/25/2021).

[19] Leo Breiman. "Random Forests". In: *UC Berkeley TR567* (1999).

[20] Alexey Natekin and Alois Knoll. "Gradient Boosting Machines, a Tutorial". In: *Frontiers in neurorobotics* 7 (2013), p. 21.

[21] Tianqi Chen and Carlos Guestrin. "XGBoost: A Scalable Tree Boosting System". In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '16. New York, NY, USA: ACM, 2016, pp. 785–794. ISBN: 978-1-4503-4232-2. DOI: 10.1145/2939672.2939785. URL: http://doi.acm.org/10.1145/2939672.2939785.

[22] Burr Settles. "Active Learning". In: *Synthesis Lectures on Artificial Intelligence and Machine Learning* 6.1 (June 30, 2012). Book that gives extensive overview of active learning as a whole, a variety of methods, and more, pp. 1–114. ISSN: 1939-4608, 1939-4616. DOI: 10.2200/S00429ED1V01Y201207AIM018. URL: http://www.morganclaypool.com/doi/abs/10.2200/S00429ED1V01Y201207AIM018 (visited on 03/02/2021).

[23] Punit Kumar and Atul Gupta. "Active Learning Query Strategies for Classification, Regression, and Clustering: A Survey". In: *Journal of Computer Science and Technology* 35.4 (July 1, 2020). Comparison of a range of active learning methods, as well as discussion on how to properly validate, pp. 913–945. ISSN: 1860-4749. DOI: 10.1007/s11390-020-9487-4. URL: https://doi.org/10.1007/s11390-020-9487-4 (visited on 12/16/2020).

[24] David D. Lewis and William A. Gale. "A Sequential Algorithm for Training Text Classifiers". In: *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR '94. Berlin, Heidelberg: Springer-Verlag, Aug. 1, 1994, pp. 3–12. ISBN: 978-0-387-19889-7.

[25] Y. Wu, I. Kozintsev, J. Bouguet, et al. "Sampling Strategies for Active Learning in Personal Photo Retrieval". In: *2006 IEEE International Conference on Multimedia and Expo*. 2006 IEEE International Conference on Multimedia and Expo. July 2006, pp. 529–532. DOI: 10.1109/ICME.2006.262442.

[26] Burr Settles and Mark Craven. "An Analysis of Active Learning Strategies for Sequence Labeling Tasks". In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. EMNLP '08. USA: Association for Computational Linguistics, Oct. 25, 2008, pp. 1070–1079.

[27] Burr Settles, Mark Craven, and Soumya Ray. "Multiple-Instance Active Learning". In: *Proceedings of the 20th International Conference on Neural Information Processing Systems*. NIPS'07. Red Hook, NY, USA: Curran Associates Inc., Dec. 3, 2007, pp. 1289–1296. ISBN: 978-1-60560-352-0.

[28] Daniel Kottke, Adrian Calma, Denis Huseljic, et al. *Challenges of Reliable, Realistic and Comparable Active Learning Evaluation*. General overview of how active learning methods should be validated. Proceedings of the Workshop and Tutorial on Interactive Adaptive Learning. Sept. 13, 2017. URL: http://localhost/handle/1874/359528 (visited on 12/16/2020).

[29] Corinna Cortes, Giulia DeSalvo, and Mehryar Mohri. "Learning with Rejection". In: *International Conference on Algorithmic Learning Theory*. Springer. 2016, pp. 67–82.

[30] Tobias Reitmaier and Bernhard Sick. "Let Us Know Your Decision: Pool-Based Active Training of a Generative Classifier with the Selection Strategy 4DS". In: *Information Sciences*. Mobile and Internet Services in Ubiquitous and Pervasive Computing Environments 230 (May 1, 2013), pp. 106–131. ISSN: 0020-0255. DOI: 10.1016/j.ins.2012.11.015. URL: https://www.sciencedirect.com/science/article/pii/S0020025512007682 (visited on 04/12/2021).

[31] Erelcan Yanık and Tevfik Metin Sezgin. "Active Learning for Sketch Recognition". In: *Computers & Graphics* 52 (Nov. 2015), pp. 93–105. ISSN: 00978493. DOI: 10.1016/j.cag.2015.07.023. URL: https://linkinghub.elsevier.com/retrieve/pii/S0097849315001260 (visited on 04/12/2021).

[32] Yukun Chen, Subramani Mani, and Hua Xu. "Applying Active Learning to Assertion Classification of Concepts in Clinical Text". In: *Journal of Biomedical Informatics* 45.2 (Apr. 2012), pp. 265–272. ISSN: 1532-0464. DOI: 10.1016/j.jbi.2011.11.003. pmid: 22127105. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3306548/ (visited on 09/24/2021).

[33] Andrew P. King and Robert J. Eckersley. "Chapter 6 - Inferential Statistics III: Nonparametric Hypothesis Testing". In: *Statistics for Biomedical Engineers and Scientists*. Ed. by Andrew P. King and Robert J. Eckersley. Academic Press, Jan. 1, 2019, pp. 119–145. ISBN: 978-0-08-102939-8. DOI: 10.1016/B978-0-08-102939-8.00015-3. URL: https://www.sciencedirect.com/science/article/pii/B9780081029398000153 (visited on 09/24/2021).

[34] Ray Liere. "Active Learning with Committees for Text Categorization." In: (1997), pp. 591–596. URL: http://www.aaai.org/Library/AAAI/1997/aaai97-092.php.

[35] Prem Melville and Raymond J. Mooney. "Diverse Ensembles for Active Learning". In: *Proceedings of the Twenty-First International Conference on Machine Learning*. ICML '04. New York, NY, USA: Association for Computing Machinery, July 4, 2004, p. 74. ISBN: 978-1-58113-838-2. DOI: 10.1145/1015330.1015385. URL: https://doi.org/10.1145/1015330.1015385 (visited on 04/30/2021).

[36] Dino Ienco, Albert Bifet, Indrė Žliobaitė, et al. "Clustering Based Active Learning for Evolving Data Streams". In: *Discovery Science*. Ed. by Johannes Fürnkranz, Eyke Hüllermeier, and Tomoyuki Higuchi. Red. by David Hutchison, Takeo Kanade, Josef Kittler, et al. Vol. 8140. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 79–93. ISBN: 978-3-642-40896-0 978-3-642-40897-7. DOI: 10.1007/978-3-642-40897-7_6. URL: http://link.springer.com/10.1007/978-3-642-40897-7_6 (visited on 05/02/2021).

[37] Nicholas Roy and Andrew McCallum. "Toward Optimal Active Learning through Sampling Estimation of Error Reduction". In: *Proceedings of the Eighteenth International Conference on Machine Learning*. ICML '01. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., June 28, 2001, pp. 441–448. ISBN: 978-1-55860-778-1.

[38] Georg Krempl, Daniel Kottke, and Myra Spiliopoulou. "Probabilistic Active Learning: Towards Combining Versatility, Optimality and Efficiency". In: *Discovery Science*. Ed. by Sašo Džeroski, Panče Panov, Dragi Kocev, et al. Lecture Notes in Computer Science. Cham: Springer International Publishing, 2014, pp. 168–179. ISBN: 978-3-319-11812-3. DOI: 10.1007/978-3-319-11812-3_15.

[39] Georg Krempl, Daniel Kottke, and Vincent Lemaire. "Optimised Probabilistic Active Learning (OPAL)". In: *Machine Language* 100.2-3 (Sept. 1, 2015), pp. 449–476. ISSN: 0885-6125. DOI: 10.1007/s10994-015-5504-1. URL: https://doi.org/10.1007/s10994-015-5504-1 (visited on 04/28/2021).

[40] Daniel Kottke, Georg Krempl, Dominik Lang, et al. "Multi-Class Probabilistic Active Learning". In: *ECAI 2016* (2016), pp. 586–594. DOI: 10.3233/978-1-61499-672-9-586. URL: https://ebooks.iospress.nl/doi/10.3233/978-1-61499-672-9-586 (visited on 04/28/2021).

[41] Daniel Kottke, Marek Herde, Christoph Sandrock, et al. *Toward Optimal Probabilistic Active Learning Using a Bayesian Approach*. Comment: 11 pages, 8 figures, appendix. June 2, 2020. arXiv: 2006.01732 [cs, stat]. URL: http://arxiv.org/abs/2006.01732 (visited on 09/25/2021).

[42] Varun Chandola, Arindam Banerjee, and Vipin Kumar. "Anomaly Detection: A Survey". In: *ACM Computing Surveys* 41.3 (July 2009), pp. 1–58. ISSN: 0360-0300, 1557-7341. DOI: 10.1145/1541880.1541882. URL: https://dl.acm.org/doi/10.1145/1541880.1541882 (visited on 04/12/2021).

[43] Markus Goldstein and Seiichi Uchida. "A Comparative Evaluation of Unsupervised Anomaly Detection Algorithms for Multivariate Data". In: *PLOS ONE* 11.4 (Apr. 19, 2016), e0152173. ISSN: 1932-6203. DOI: 10.1371/journal.pone.0152173. URL: https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0152173 (visited on 04/12/2021).

[44] Salima Omar, Universiti Teknologi Malaysia, Asri Ngadi, et al. *Machine Learning Techniques for Anomaly Detection: An Overview*.

[45] Vincent Vercruyssen, Wannes Meert, Gust Verbruggen, et al. "Semi-Supervised Anomaly Detection with an Application to Water Analytics". In: *2018 IEEE INTERNATIONAL CONFERENCE ON DATA MINING (ICDM)*. Vol. 2018-November. IEEE, pp. 527–536. ISBN: 978-1-5386-9158-8. DOI: 10.1109/ICDM.2018.00068. URL: https://ieeexplore.ieee.org/xpl/conhome/8591042/proceeding, (visited on 04/13/2021).

[46] Hansheng Ren, Bixiong Xu, Yujing Wang, et al. "Time-Series Anomaly Detection Service at Microsoft". In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (July 25, 2019). Comment: KDD 2019, pp. 3009–3017. DOI: 10.1145/3292500.3330680. arXiv: 1906.03821. URL: http://arxiv.org/abs/1906.03821 (visited on 04/13/2021).

[47] Enmei Tu, Guanghao Zhang, Lily Rachmawati, et al. "Exploiting AIS Data for Intelligent Maritime Navigation: A Comprehensive Survey From Data to Methodology". In: *IEEE Transactions on Intelligent Transportation Systems* 19.5 (May 2018), pp. 1559–1582. ISSN: 1524-9050, 1558-0016. DOI: 10.1109/TITS.2017.2724551. URL: https://ieeexplore.ieee.org/document/8025635/ (visited on 03/06/2021).

[48] Enmei Tu, Guanghao Zhang, Shangbo Mao, et al. *Modeling Historical AIS Data For Vessel Path Prediction: A Comprehensive Treatment*. Comment: 11 pages. Mar. 30, 2020. arXiv: 2001.01592 [cs]. URL: http://arxiv.org/abs/2001.01592 (visited on 04/13/2021).

[49] Tong Wu and Jorge Ortiz. *RLAD: Time Series Anomaly Detection through Reinforcement Learning and Active Learning*. Mar. 31, 2021. arXiv: 2104.00543 [cs]. URL: http://arxiv.org/abs/2104.00543 (visited on 04/12/2021).

[50] Kira Kowalska and L. Peel. *Maritime Anomaly Detection Using Gaussian Process Active Learning*. Jan. 1, 2012, p. 1171. 1164 pp. ISBN: 978-1-4673-0417-7.

[51] Sizhe Huang, Huosheng Xu, and Xuezhi Xia. "Active Deep Belief Networks for Ship Recognition Based on BvSB". In: *Optik* 127.24 (Dec. 1, 2016), pp. 11688–11697. ISSN: 0030-4026. DOI: 10.1016/j.ijleo.2016.09.089. URL: https://www.sciencedirect.com/science/article/pii/S0030402616311081 (visited on 04/13/2021).

[52] Josh Attenberg. *6 Class Imbalance and Active Learning*.

[53] Seyda Ertekin, Jian Huang, and C. Lee Giles. "Active Learning for Class Imbalance Problem". In: *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR '07. New York, NY, USA: Association for Computing Machinery, July 23, 2007, pp. 823–824. ISBN: 978-1-59593-597-7. DOI: 10.1145/1277741.1277927. URL: https://doi.org/10.1145/1277741.1277927 (visited on 03/03/2021).

[54] Tiago Pimentel, Marianne Monteiro, Adriano Veloso, et al. *Deep Active Learning for Anomaly Detection*. Comment: Accepted for publication at IJCNN 2020. Apr. 7, 2020. arXiv: 1805.09411 [cs, stat]. URL: http://arxiv.org/abs/1805.09411 (visited on 04/13/2021).

[55] Dan Pelleg and Andrew W Moore. "Active Learning for Anomaly and Rare-Category Detection". In: (), p. 8.

[56] N. Goernitz, M. Kloft, K. Rieck, et al. "Toward Supervised Anomaly Detection". In: *Journal of Artificial Intelligence Research* 46 (Feb. 20, 2013), pp. 235–262. ISSN: 1076-9757. DOI: 10.1613/jair.3623. URL: https://jair.org/index.php/jair/article/view/10802 (visited on 03/06/2021).

[57] Nesreen K. Ahmed, Amir F. Atiya, Neamat El Gayar, et al. "An Empirical Comparison of Machine Learning Models for Time Series Forecasting". In: *Econometric Reviews* 29.5-6 (Aug. 30, 2010), pp. 594–621. ISSN: 0747-4938. DOI: 10.1080/07474938.2010.481556. URL: https://doi.org/10.1080/07474938.2010.481556 (visited on 04/13/2021).

[58] Fengchao Peng, Qiong Luo, and Lionel M. Ni. "ACTS: An Active Learning Method for Time Series Classification". In: *2017 IEEE 33rd International Conference on Data Engineering (ICDE)*. 2017 IEEE 33rd International Conference on Data Engineering (ICDE). San Diego, CA, USA: IEEE, Apr. 2017, pp. 175–178. ISBN: 978-1-5090-6543-1. DOI: 10.1109/ICDE.2017.68. URL: http://ieeexplore.ieee.org/document/7929964/ (visited on 03/06/2021).

[59] Lexiang Ye and Eamonn Keogh. "Time Series Shapelets: A New Primitive for Data Mining". In: *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '09. New York, NY, USA: Association for Computing Machinery, June 28, 2009, pp. 947–956. ISBN: 978-1-60558-495-9. DOI: 10.1145/1557019.1557122. URL: https://doi.org/10.1145/1557019.1557122 (visited on 04/13/2021).

[60] Guan Yuan, Penghui Sun, Jie Zhao, et al. "A Review of Moving Object Trajectory Clustering Algorithms". In: *Artificial Intelligence Review* 47.1 (Jan. 1, 2017), pp. 123–144. ISSN: 0269-2821. DOI: 10.1007/s10462-016-9477-7. URL: https://doi.org/10.1007/s10462-016-9477-7 (visited on 05/05/2021).

[61] Martin Ester, Hans-Peter Kriegel, Jörg Sander, et al. "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise". In: *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*. KDD'96. Portland, Oregon: AAAI Press, Aug. 2, 1996, pp. 226–231.

[62] Jae-Gil Lee, Jiawei Han, and Kyu-Young Whang. "Trajectory Clustering: A Partition-and-Group Framework". In: *Proceedings of the 2007 ACM SIGMOD International Conference on Management of Data*. SIGMOD '07. New York, NY, USA: Association for Computing Machinery, June 11, 2007, pp. 593–604. ISBN: 978-1-59593-686-8. DOI: 10.1145/1247480.1247546. URL: https://doi.org/10.1145/1247480.1247546 (visited on 05/09/2021).

[63] Jae-Gil Lee, Jiawei Han, Xiaolei Li, et al. "*TraClass*: Trajectory Classification Using Hierarchical Region-Based and Trajectory-Based Clustering". In: *Proceedings of the VLDB Endowment* 1.1 (Aug. 2008), pp. 1081–1094. ISSN: 2150-8097. DOI: 10.14778/1453856.1453972. URL: https://dl.acm.org/doi/10.14778/1453856.1453972 (visited on 05/05/2021).

[64] apolcyn. *Apolcyn/Traclus_impl*. Apr. 1, 2021. URL: https://github.com/apolcyn/traclus_impl (visited on 05/09/2021).

[65] Robin Senge, Stefan Bösner, Krzysztof Dembczyński, et al. "Reliable Classification: Learning Classifiers That Distinguish Aleatoric and Epistemic Uncertainty". In: *Information Sciences* 255 (Jan. 10, 2014), pp. 16–29. ISSN: 0020-0255. DOI: 10.1016/j.ins.2013.07.030. URL: https://www.sciencedirect.com/science/article/pii/S0020025513005410 (visited on 04/13/2021).

[66] Vu-Linh Nguyen, Sébastien Destercke, and Eyke Hüllermeier. *Epistemic Uncertainty Sampling*. Comment: Draft version of a paper to be published in the proceedings of DS 2019, 22nd International Conference on Discovery Science, Split, Croatia, 2019. Aug. 31, 2019. arXiv: 1909.00218 [cs, stat]. URL: http://arxiv.org/abs/1909.00218 (visited on 04/09/2021).

[67] Moksh Jain, Salem Lahlou, Hadi Nekoei, et al. *DEUP: Direct Epistemic Uncertainty Prediction*. Feb. 16, 2021. arXiv: 2102.08501 [cs, stat]. URL: http://arxiv.org/abs/2102.08501 (visited on 04/13/2021).

[68] The pandas development team. *Pandas-Dev/Pandas: Pandas*. Version latest. Zenodo, Feb. 2020. DOI: 10.5281/zenodo.3509134. URL: https://doi.org/10.5281/zenodo.3509134.

[69] Shpat Cheliku, Heysem Kaya, Paul Merkx, et al. "TitleTBD". unpublished. 2021.

[70] Tivadar Danka and Peter Horvath. "modAL: A Modular Active Learning Framework for Python". In: (). available on arXiv at https://arxiv.org/abs/1805.00979. URL: https://github.com/modAL-python/modAL.

[71] F. Pedregosa, G. Varoquaux, A. Gramfort, et al. "Scikit-Learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[72] Katrin Tomanek and Katherina Morik. "Inspecting Sample Reusability for Active Learning". In: *Active Learning and Experimental Design Workshop In Conjunction with AISTATS 2010*. Active Learning and Experimental Design Workshop In Conjunction with AISTATS 2010. JMLR Workshop and Conference Proceedings, Apr. 21, 2011, pp. 169–181. URL: https://proceedings.mlr.press/v16/tomanek11a.html (visited on 09/29/2021).

[73] Zaigham Siddiqui and Myra Spiliopoulou. "Classification Rule Mining for a Stream of Perennial Objects". In: July 19, 2011, pp. 281–296. ISBN: 978-3-642-22545-1. DOI: 10.1007/978-3-642-22546-8_22.

Appendix A QBC Tuning

The following figures were the result of testing different configurations of committees for QBC. Model types tested were: Linear logistic regression (denotes as classifier 1), random forest (classifier 2), and XGBoost (classifier 3). To limit the amount of test-configurations, the amount of different types of models in a configuration was always balanced, and the committee could have a maximum of 8 members. This maximum was only exceeded for the configuration using three of each model type.

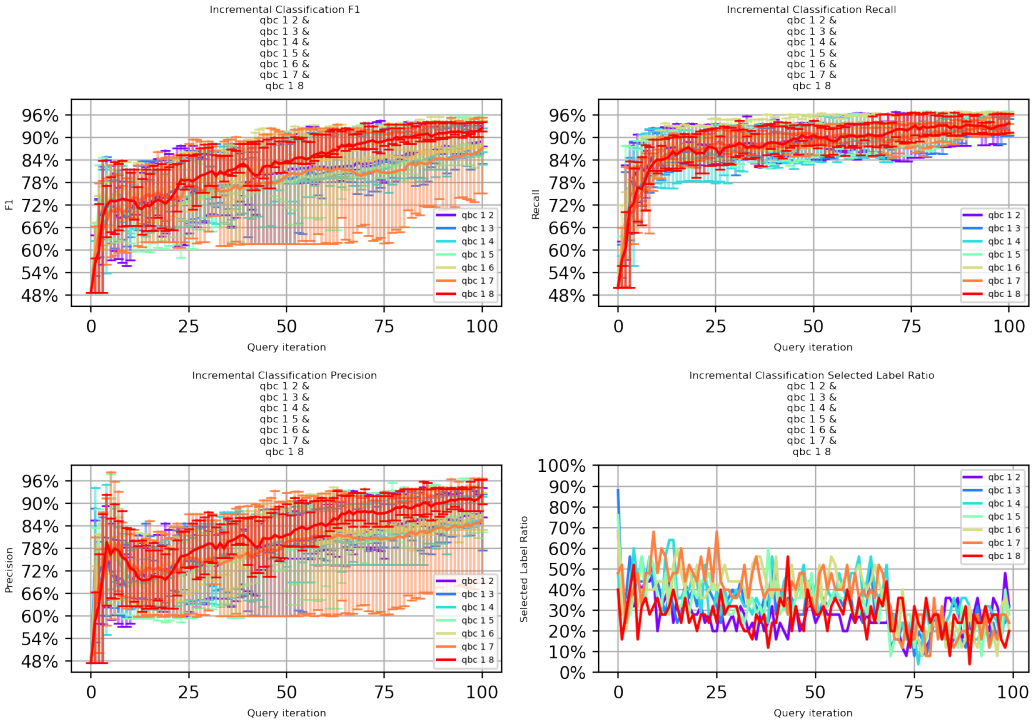

Figure 26: (Macro) F1, recall, and precision for complete summarizing instance learning of QBC configurations only using linear logistic regression models.

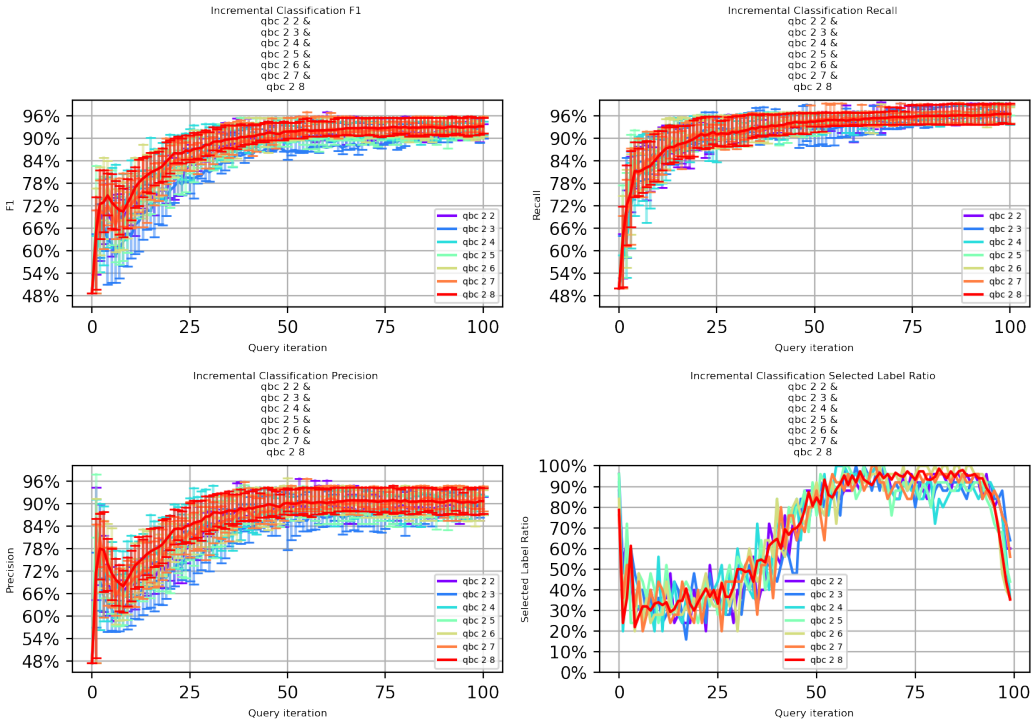

Figure 27: (Macro) F1, recall, and precision for complete summarizing instance learning of QBC configurations only using random forest models.

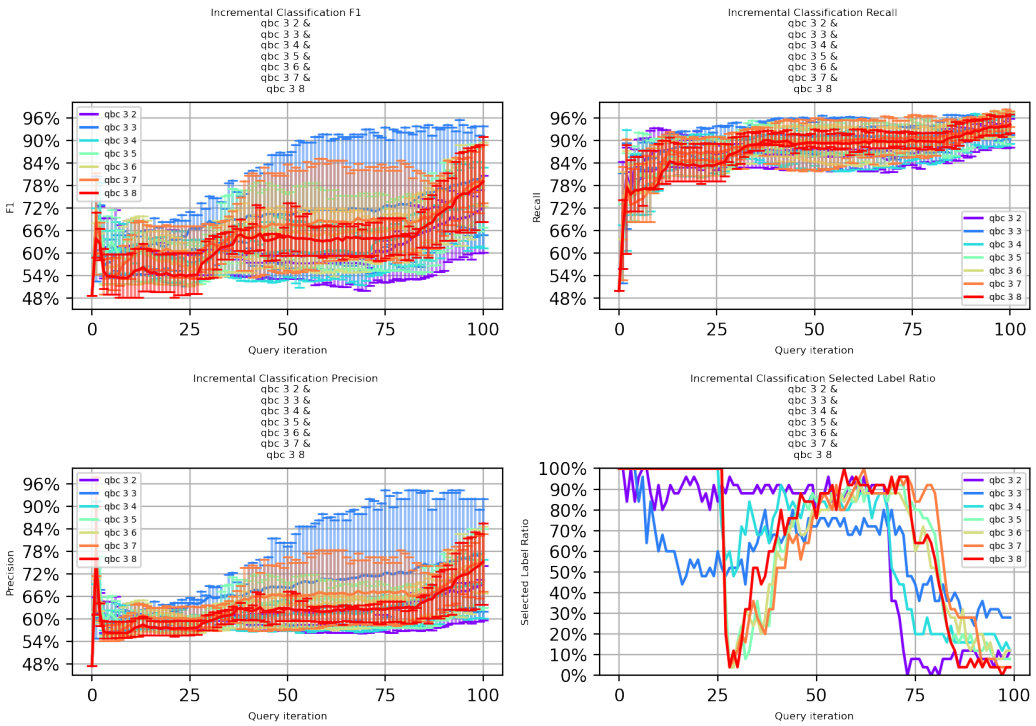

Figure 28: (Macro) F1, recall, and precision for complete summarizing instance learning of QBC configurations only using XGBoost models.

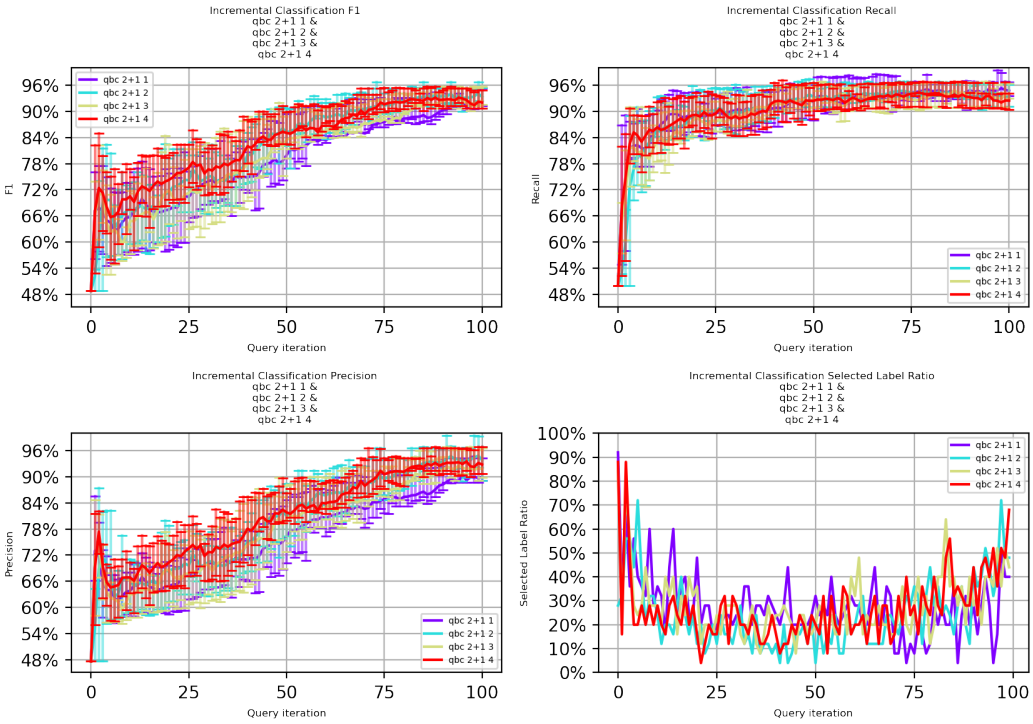

Figure 29: (Macro) F1, recall, and precision for complete summarizing instance learning of QBC configurations using combinations of random forest and linear logistic regression models.

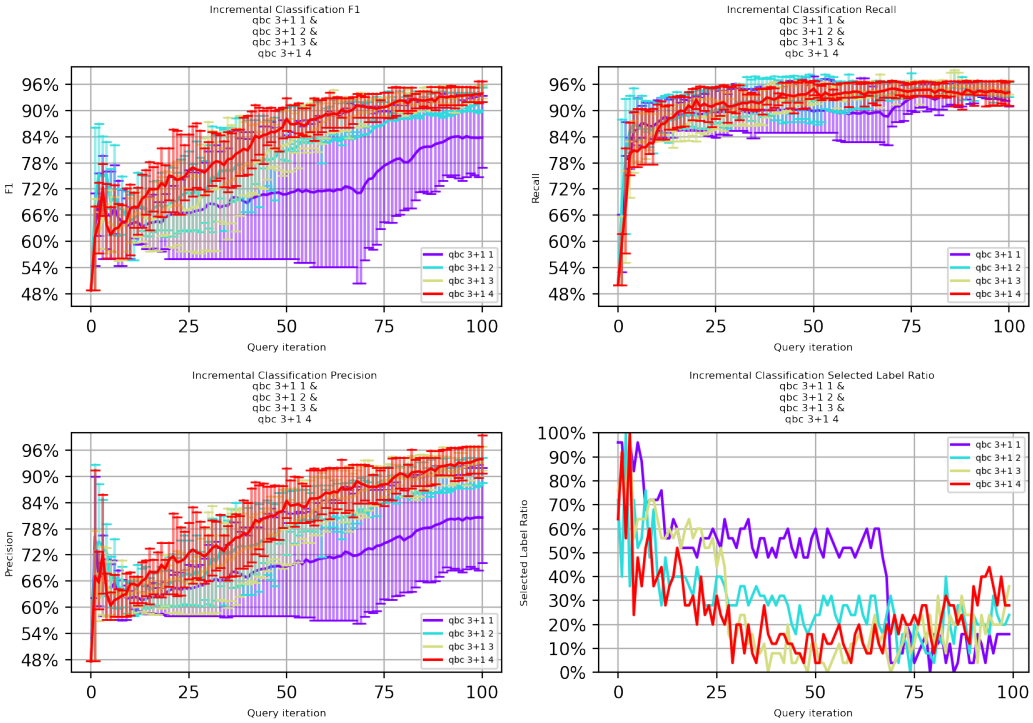

Figure 30: (Macro) F1, recall, and precision for complete summarizing instance learning of QBC configurations using combinations of XGBoost and logistic linear regression models.

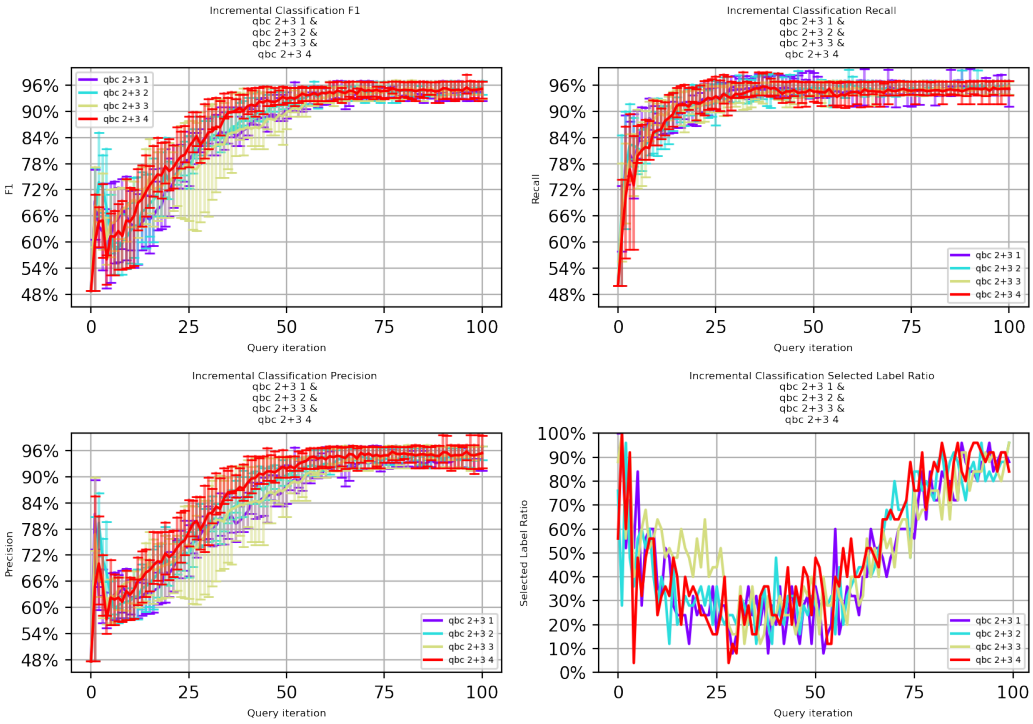

Figure 31: (Macro) F1, recall, and precision for complete summarizing instance learning of QBC configurations using combinations of random forest and XGBoost models.

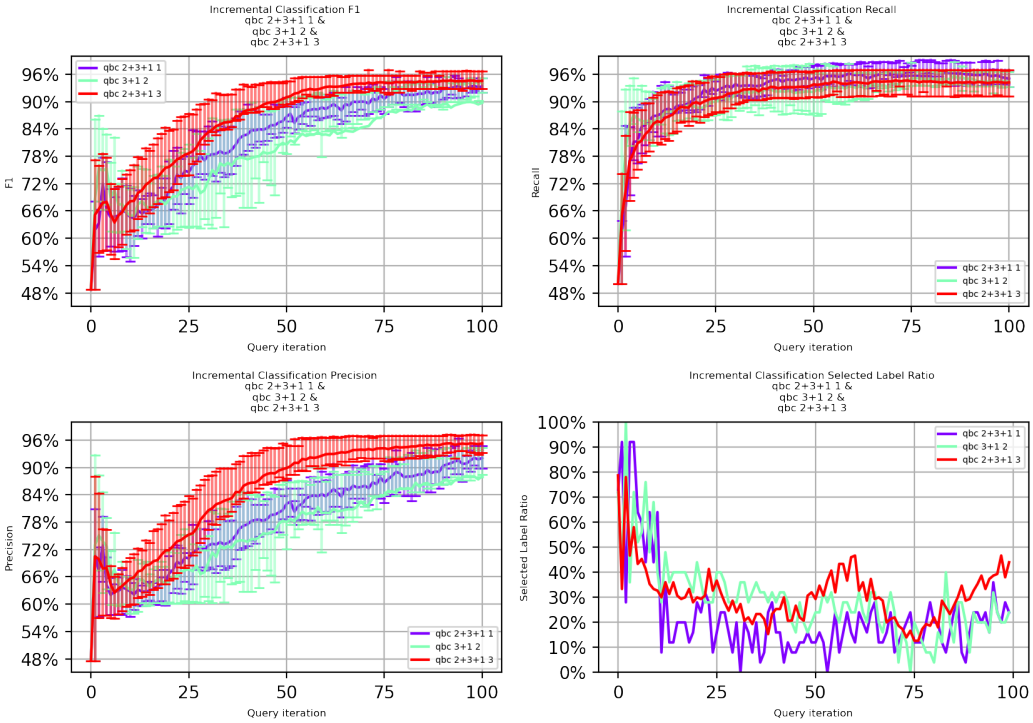

Figure 32: (Macro) F1, recall, and precision for complete summarizing instance learning of QBC configurations using combinations of random forest, XGBoost, and linear logistic regression models.

Appendix B Wilcoxon Signed-Rank Significance Tests

The following are the results for applying the Wilcoxon signed-rank test as described in Section 5.4 to test whether the performance between the observed performance differences between the active learning methods for the summarized instance training tests were significant. Red values mean the p-value was below 0.05, thus the difference between the performances of the methods wasn't statistically significant.

	random	uncertainty	density	qbc	xpal
random	NA	2.44E-26	2.35E-26	2.35E-26	3.15E-02
uncertainty	2.44E-26	NA	1.32E-08	9.53E-01	2.30E-26
density	2.35E-26	1.32E-08	NA	8.96E-10	2.93E-26
qbc	2.35E-26	9.53E-01	8.96E-10	NA	2.35E-26
xpal	3.15E-02	2.30E-26	2.93E-26	2.35E-26	NA

Figure 33: Wilcoxon-signed rank test results for the performances of different active learning methods for complete trajectory classification

	random	uncertainty	density	qbc	xpal
random	NA	3.00E-06	1.80E-01	1.91E-01	7.85E-15
uncertainty	3.00E-06	NA	1.28E-12	3.07E-04	1.50E-12
density	1.80E-01	1.28E-12	NA	1.66E-03	2.60E-14
qbc	1.91E-01	3.07E-04	1.66E-03	NA	7.94E-15
xpal	7.85E-15	1.50E-12	2.60E-14	7.94E-15	NA

Figure 34: Wilcoxon-signed rank test results for the performances of different active learning methods for partial trajectory classification at the 10 hour time point.

	random	uncertainty	density	qbc	xpal
random	NA	1.11E-14	3.08E-06	2.99E-03	7.50E-19
uncertainty	1.11E-14	NA	4.67E-09	1.09E-21	3.58E-26
density	3.08E-06	4.67E-09	NA	2.85E-13	3.80E-26
qbc	2.99E-03	1.09E-21	2.85E-13	NA	3.53E-14
xpal	7.50E-19	3.58E-26	3.80E-26	3.53E-14	NA

Figure 35: Wilcoxon-signed rank test results for the performances of different active learning methods for partial trajectory classification at the 16 hour time point.

	random	uncertainty	density	qbc	xpal
random	NA	1.77E-15	1.25E-14	1.13E-02	2.53E-13
uncertainty	1.77E-15	NA	7.73E-01	3.48E-08	6.51E-24
density	1.25E-14	7.73E-01	NA	9.85E-08	7.95E-25
qbc	1.13E-02	3.48E-08	9.85E-08	NA	1.45E-16
xpal	2.53E-13	6.51E-24	7.95E-25	1.45E-16	NA

Figure 36: Wilcoxon-signed rank test results for the performances of different active learning methods for partial trajectory classification at the 20 hour time point.

	random	uncertainty	density	qbc	xpal
random	NA	9.63E-14	6.60E-12	1.83E-05	1.13E-17
uncertainty	9.63E-14	NA	2.61E-03	1.02E-04	4.55E-26
density	6.60E-12	2.61E-03	NA	2.77E-02	3.72E-26
qbc	1.83E-05	1.02E-04	2.77E-02	NA	1.85E-23
xpal	1.13E-17	4.55E-26	3.72E-26	1.85E-23	NA

Figure 37: Wilcoxon-signed rank test results for the performances of different active learning methods for partial trajectory classification at the 25 hour time point.

	random	uncertainty	density	qbc	xpal
random	NA	9.04E-09	4.95E-10	7.12E-07	1.65E-11
uncertainty	9.04E-09	NA	4.42E-01	4.87E-01	3.00E-21
density	4.95E-10	4.42E-01	NA	8.30E-01	2.33E-21
qbc	7.12E-07	4.87E-01	8.30E-01	NA	4.23E-22
xpal	1.65E-11	3.00E-21	2.33E-21	4.23E-22	NA

Figure 38: Wilcoxon-signed rank test results for the performances of different active learning methods for partial trajectory classification at the 30 hour time point.

	random	uncertainty	density	qbc	xpal
random	NA	3.45E-08	1.43E-13	5.64E-01	2.77E-05
uncertainty	3.45E-08	NA	5.14E-02	2.69E-08	1.38E-19
density	1.43E-13	5.14E-02	NA	4.94E-12	7.88E-24
qbc	5.64E-01	2.69E-08	4.94E-12	NA	3.56E-04
xpal	2.77E-05	1.38E-19	7.88E-24	3.56E-04	NA

Figure 39: Wilcoxon-signed rank test results for the performances of different active learning methods for partial trajectory classification at the 35 hour time point.

	random	uncertainty	density	qbc	xpal
random	NA	1.81E-08	4.95E-10	1.29E-01	1.20E-02
uncertainty	1.81E-08	NA	6.55E-01	7.17E-11	1.11E-14
density	4.95E-10	6.55E-01	NA	9.08E-11	1.45E-16
qbc	1.29E-01	7.17E-11	9.08E-11	NA	5.41E-01
xpal	1.20E-02	1.11E-14	1.45E-16	5.41E-01	NA

Figure 40: Wilcoxon-signed rank test results for the performances of different active learning methods for partial trajectory classification at the 40 hour time point.

Appendix C Additional Performance Graphs

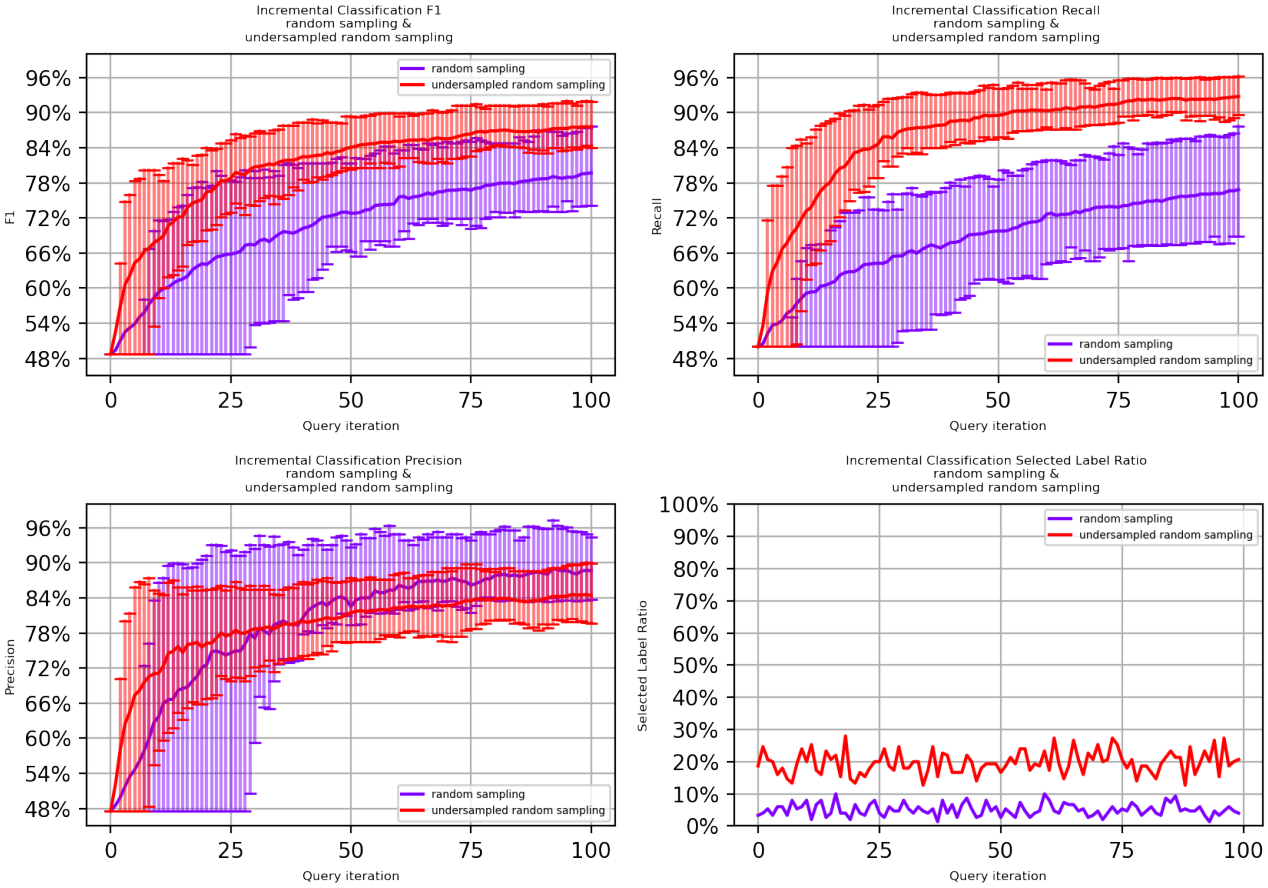

Figure 41: (Macro) F1, recall, and precision for training on undersampled and not undersampled randomly selected trajectory summarizing instances, as well as the percentage of selected positive labels.

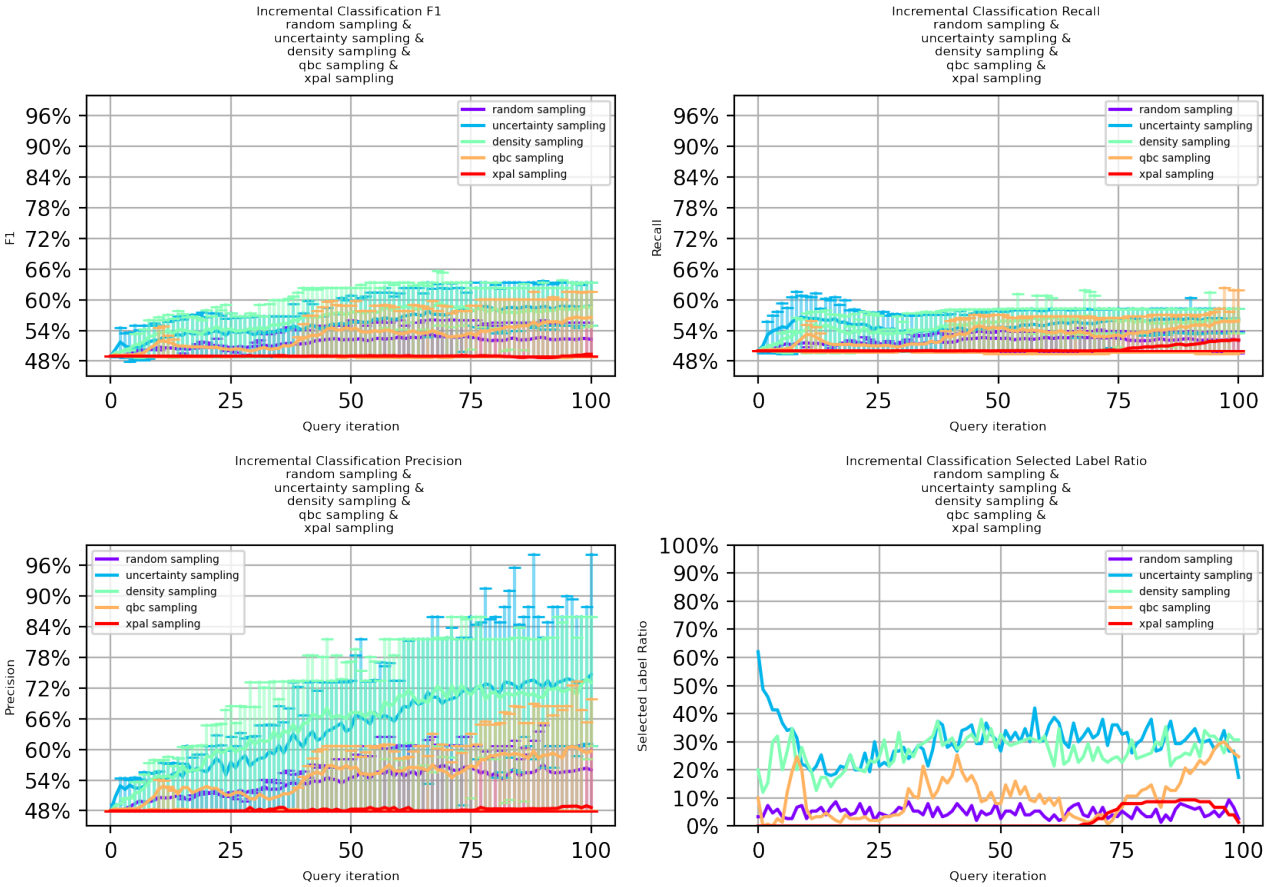

Figure 42: Partial Trajectories at 20 hours. (Macro) F1, recall, and precision for training selected trajectory summarizing instances, as well as the percentage of selected positive labels.

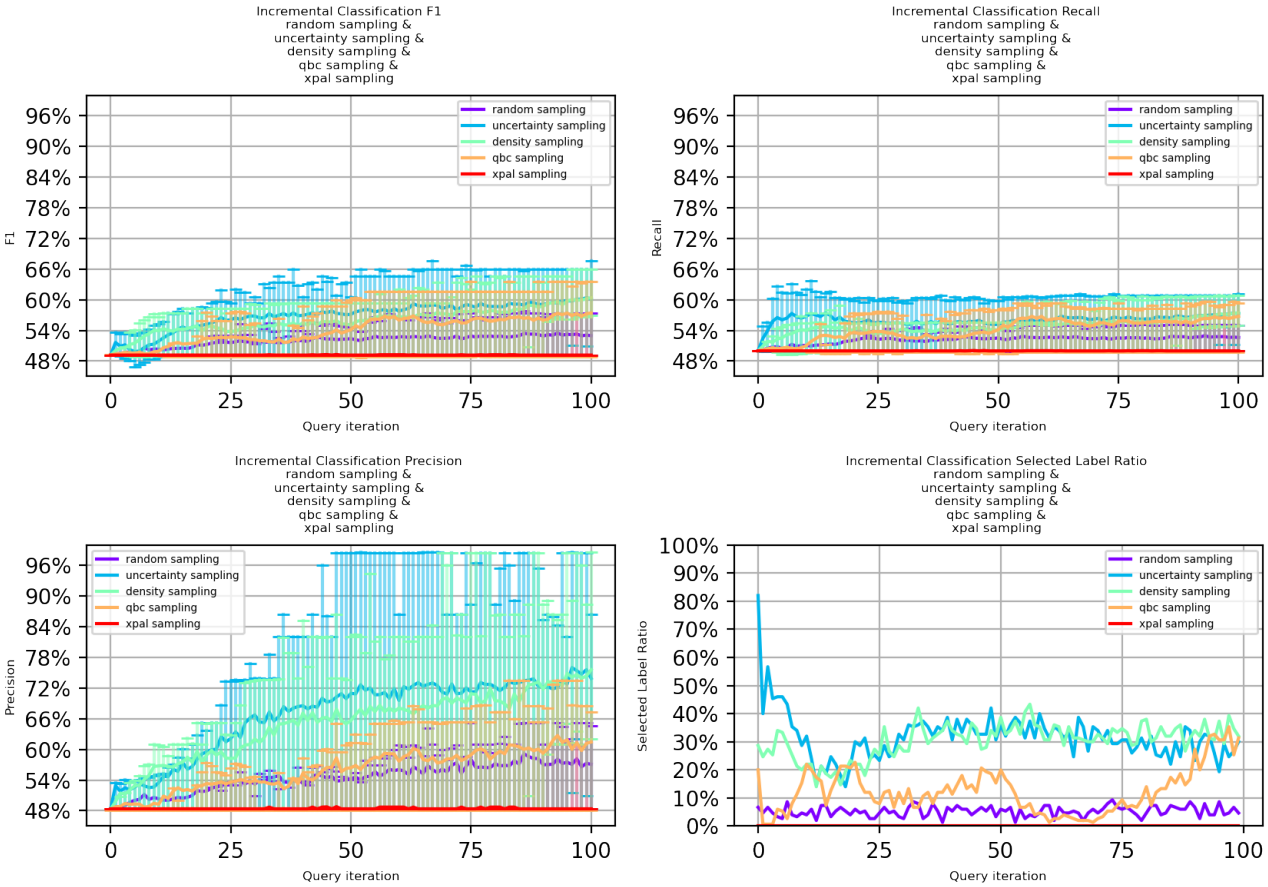

Figure 43: Partial Trajectories at 25 hours. (Macro) F1, recall, and precision for training selected trajectory summarizing instances, as well as the percentage of selected positive labels.

Appendix D Feature Importances of Summarized Complete Trajectory Classifiers

The following tables show the feature importances as a measure of gain for the classifiers resulting from the different sampling methods for complete trajectory classification with summarized instance learning. The importances were binned and averaged, with each column number showing the last included instance in each bin.

	10	21	32	43	54	65	76	87	98
Latitude	0.001152	0.000446	0.007633	0.013396	0.017533	0.018113	0.019103	0.021762	0.021390
Longitude	0.009207	0.006190	0.003679	0.005327	0.009422	0.014405	0.014229	0.018399	0.014405
SOG	0.009420	0.001704	0.003382	0.002498	0.010837	0.009538	0.019794	0.024560	0.017552
COG	0.012429	0.006797	0.007472	0.013790	0.014996	0.025208	0.023154	0.019926	0.023823
time_elapsed_s	0.082822	0.056782	0.058669	0.049343	0.044319	0.041920	0.042197	0.042447	0.038408
expanding_mean_lat_full	0.004966	0.000419	0.001044	0.001886	0.003694	0.001606	0.002117	0.002804	0.000891
expanding_mean_lon_full	0.002152	0.000878	0.001020	0.000535	0.002058	0.000125	0.002736	0.002978	0.001739
expanding_mean_speed_full	0.004423	0.001406	0.001407	0.006751	0.007718	0.009426	0.012044	0.007723	0.012923
expanding_mean_orientation_full	0.007122	0.000673	0.001376	0.001791	0.003006	0.008582	0.008847	0.006380	0.003670
expanding_max_lat_full	0.010293	0.003295	0.004959	0.005484	0.003313	0.000209	0.000040	0.000247	0.000976
expanding_max_lon_full	0.003118	0.002628	0.004644	0.003169	0.001723	0.006427	0.005488	0.007543	0.007311
expanding_max_speed_full	0.012056	0.006433	0.004956	0.002858	0.004887	0.001929	0.001308	0.000243	0.000675
expanding_max_orientation_full	0.003770	0.000758	0.000375	0.000102	0.000578	0.000138	0.000956	0.001358	0.001126
expanding_min_lat_full	0.002676	0.002249	0.001882	0.001459	0.000858	0.000934	0.000891	0.002407	0.001210
expanding_min_lon_full	0.008993	0.003600	0.002793	0.002858	0.002818	0.000684	0.002732	0.003334	0.002129
expanding_min_speed_full	0.001710	0.000000	0.000403	0.000447	0.000749	0.000000	0.000296	0.000000	0.001114
expanding_min_orientation_full	0.002002	0.000547	0.003315	0.004687	0.003598	0.003298	0.002554	0.002312	0.002213
expanding_std_lat_full	0.058142	0.063035	0.042818	0.049156	0.050138	0.047315	0.043582	0.043999	0.045376
expanding_std_lon_full	0.035506	0.033941	0.021001	0.017337	0.019940	0.027303	0.039482	0.042646	0.052721
expanding_std_speed_full	0.004379	0.002978	0.001420	0.000287	0.002443	0.002130	0.003103	0.001969	0.002536
expanding_std_orientation_full	0.034554	0.022713	0.017628	0.016775	0.019118	0.019599	0.026186	0.026637	0.024704
expanding_median_lat_full	0.000377	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000164
expanding_median_lon_full	0.000000	0.000000	0.001039	0.001066	0.000969	0.000056	0.000000	0.000026	0.000000
expanding_median_speed_full	0.008388	0.005648	0.005288	0.006122	0.004527	0.006592	0.003796	0.002591	0.002328
expanding_median_orientation_full	0.002319	0.000292	0.001141	0.002429	0.001558	0.001041	0.000577	0.000820	0.000535
expanding_range_lat_full	0.008111	0.012978	0.019191	0.020012	0.016457	0.014955	0.014486	0.016228	0.016646
expanding_range_lon_full	0.022393	0.020688	0.027478	0.029204	0.030402	0.024984	0.029243	0.033204	0.028725
expanding_range_speed_full	0.003287	0.000388	0.001550	0.001015	0.000849	0.002058	0.002112	0.000677	0.000800
expanding_range_orientation_full	0.003289	0.000950	0.000801	0.000093	0.000000	0.000000	0.000000	0.000024	0.000063
expanding_rel_max_lat_full	0.001362	0.000377	0.000000	0.000179	0.001313	0.001422	0.001832	0.000607	0.002418
expanding_rel_max_lon_full	0.007247	0.005650	0.004042	0.004196	0.002166	0.002112	0.001595	0.000985	0.000204
expanding_rel_max_speed_full	0.021473	0.011905	0.006337	0.004780	0.007369	0.004414	0.003781	0.003305	0.005785
expanding_rel_max_orientation_full	0.004643	0.001276	0.000516	0.000442	0.000329	0.000948	0.002132	0.001073	0.001172
expanding_rel_min_lat_full	0.005072	0.003030	0.001674	0.001899	0.004406	0.001905	0.001341	0.002983	0.000791
expanding_rel_min_lon_full	0.000349	0.000778	0.000858	0.000842	0.000051	0.000000	0.000266	0.000229	0.003034
expanding_rel_min_speed_full	0.000938	0.000209	0.000257	0.000667	0.000124	0.000035	0.000348	0.000162	0.000571
expanding_rel_min_orientation_full	0.000701	0.000870	0.000646	0.000582	0.001908	0.001067	0.001566	0.000504	0.001112
rolling_mean_lat_10min	0.000000	0.000317	0.000000	0.000000	0.000000	0.000000	0.000000	0.000082	0.000000
rolling_mean_lon_10min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000256	0.000297	0.000000	0.000000
rolling_mean_speed_10min	0.008138	0.001683	0.000641	0.001948	0.001086	0.008023	0.005793	0.005999	0.006505
rolling_mean_orientation_10min	0.018442	0.004619	0.001974	0.002967	0.003827	0.006915	0.005969	0.013890	0.016311
rolling_max_lat_10min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000459	0.000583
rolling_max_lon_10min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
rolling_max_speed_10min	0.010969	0.002932	0.001467	0.000320	0.000245	0.000983	0.001411	0.002324	0.000003
rolling_max_orientation_10min	0.012411	0.008395	0.007597	0.008584	0.006480	0.011363	0.007814	0.008128	0.008739
rolling_min_lat_10min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
rolling_min_lon_10min	0.000000	0.000000	0.000000	0.000162	0.000012	0.000355	0.000373	0.000374	0.000420
rolling_min_speed_10min	0.000000	0.000000	0.000184	0.000000	0.000000	0.001399	0.001907	0.001972	0.004407
rolling_min_orientation_10min	0.001732	0.000217	0.000000	0.000175	0.000311	0.000092	0.000551	0.000081	0.001134
rolling_std_lat_10min	0.001986	0.000923	0.000277	0.000304	0.001050	0.001693	0.002610	0.001890	0.002638
rolling_std_lon_10min	0.041917	0.036271	0.014115	0.009524	0.008402	0.015425	0.012348	0.013977	0.015958
rolling_std_speed_10min	0.047841	0.034510	0.036771	0.034221	0.021904	0.018965	0.015612	0.010851	0.009045
rolling_std_orientation_10min	0.005520	0.000837	0.000779	0.000000	0.002865	0.002452	0.001601	0.001501	0.001540
rolling_median_lat_10min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000252	0.000000	0.000000	0.000000
rolling_median_lon_10min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
rolling_median_speed_10min	0.000000	0.000000	0.000251	0.000000	0.000000	0.000000	0.000000	0.000000	0.001581
rolling_median_orientation_10min	0.000331	0.000396	0.000412	0.000180	0.000797	0.000650	0.001538	0.000682	0.000250
rolling_range_lat_10min	0.001387	0.000799	0.000178	0.000000	0.000337	0.000000	0.000000	0.000824	0.000000
rolling_range_lon_10min	0.028123	0.037164	0.042796	0.044629	0.049139	0.047006	0.046560	0.043119	0.033788
rolling_range_speed_10min	0.011716	0.020794	0.020964	0.017681	0.014358	0.013947	0.017133	0.014963	0.016211
rolling_range_orientation_10min	0.001131	0.002371	0.001078	0.000512	0.001188	0.000756	0.003435	0.004718	0.003994
rolling_rel_max_lat_10min	0.051509	0.078405	0.078471	0.078233	0.069223	0.069741	0.059630	0.052620	0.055147
rolling_rel_max_lon_10min	0.000956	0.000000	0.000000	0.000000	0.000322	0.000370	0.000042	0.000366	0.000205
rolling_rel_max_speed_10min	0.024380	0.044223	0.050431	0.054604	0.050838	0.048786	0.041313	0.034453	0.033728
rolling_rel_max_orientation_10min	0.020574	0.014079	0.008330	0.008563	0.011152	0.009874	0.006981	0.008109	0.007694
rolling_rel_min_lat_10min	0.002643	0.000515	0.000678	0.000194	0.000623	0.001124	0.000870	0.000000	0.000049
rolling_rel_min_lon_10min	0.045314	0.070754	0.068219	0.062289	0.061283	0.055047	0.060929	0.064102	0.056244
rolling_rel_min_speed_10min	0.008291	0.019065	0.015949	0.017412	0.018581	0.019069	0.017399	0.015790	0.015944
rolling_rel_min_orientation_10min	0.014631	0.018050	0.019280	0.022791	0.015906	0.012121	0.009849	0.011983	0.008402
rolling_mean_lat_60min	0.000000	0.000000	0.000326	0.000625	0.000169	0.000223	0.000161	0.000251	0.001003
rolling_mean_lon_60min	0.000742	0.000687	0.000315	0.000370	0.000383	0.000492	0.000000	0.000060	0.000642
rolling_mean_speed_60min	0.000000	0.000000	0.000215	0.000085	0.000267	0.000688	0.000343	0.001514	0.004689
rolling_mean_orientation_60min	0.015815	0.022074	0.016037	0.017917	0.016053	0.013710	0.015776	0.013908	0.015142
rolling_max_lat_60min	0.000000	0.000473	0.000320	0.000278	0.000405	0.000111	0.000000	0.000131	0.000000
rolling_max_lon_60min	0.000000	0.000214	0.000000	0.000000	0.000220	0.000000	0.000000	0.000000	0.000000
rolling_max_speed_60min	0.006265	0.007889	0.006167	0.005411	0.000840	0.001192	0.000306	0.000624	0.000529
rolling_max_orientation_60min	0.008490	0.007855	0.008729	0.010742	0.010268	0.010237	0.010838	0.010529	0.010234
rolling_min_lat_60min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
rolling_min_lon_60min	0.000000	0.000207	0.000380	0.000382	0.000000	0.000752	0.001321	0.001326	0.001203
rolling_min_speed_60min	0.000621	0.000000	0.000840	0.000526	0.002625	0.002867	0.003439	0.004275	0.010197
rolling_min_orientation_60min	0.000000	0.000000	0.000414	0.000000	0.000480	0.002210	0.003685	0.002810	0.001815
rolling_std_lat_60min	0.003930	0.002721	0.002163	0.001162	0.001415	0.001162	0.003021	0.002988	0.004106
rolling_std_lon_60min	0.018203	0.028175	0.029523	0.029470	0.027621	0.019664	0.016851	0.016729	0.018694
rolling_std_speed_60min	0.061668	0.112475	0.132659	0.118959	0.113364	0.109898	0.088020	0.080334	0.082937
rolling_std_orientation_60min	0.000466	0.000225	0.000000	0.000000	0.000094	0.000000	0.000000	0.000824	0.001102
rolling_median_lat_60min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
rolling_median_lon_60min	0.000000	0.000000	0.000634	0.000000	0.000152	0.000000	0.000000	0.000000	0.000087
rolling_median_speed_60min	0.001257	0.001498	0.000000	0.000000	0.000146	0.000000	0.000518	0.000404	0.000207
rolling_median_orientation_60min	0.004418	0.002399	0.001936	0.001685	0.001573	0.002229	0.001388	0.000811	0.001179
rolling_range_lat_60min	0.000414	0.000935	0.000991	0.000447	0.001037	0.001628	0.001366	0.001667	0.004237
rolling_range_lon_60min	0.006069	0.013685	0.014573	0.013890	0.014658	0.010605	0.014168	0.013854	0.011825
rolling_range_speed_60min	0.016768	0.030924	0.047621	0.058003	0.070368	0.076524	0.085881	0.097721	0.093258
rolling_range_orientation_60min	0.000745	0.000000	0.000000	0.000367	0.000165	0.000000	0.000347	0.001097	0.001595
rolling_rel_max_lat_60min	0.003206	0.001051	0.001704	0.000225	0.000326	0.000262	0.001277	0.001351	0.000862
rolling_rel_max_lon_60min	0.001747	0.001486	0.000423	0.000186	0.000396	0.000000	0.000000	0.000457	0.000487
rolling_rel_max_speed_60min	0.002434	0.002307	0.001001	0.001497	0.000470	0.000421	0.000829	0.000200	0.000640
rolling_rel_max_orientation_60min	0.007702	0.008291	0.008755	0.005525	0.006712	0.002547	0.002445	0.001643	0.001375
rolling_rel_min_lat_60min	0.007733	0.006099	0.004364	0.003866	0.002012	0.003987	0.003596	0.006479	0.004553
rolling_rel_min_lon_60min	0.032952	0.042820	0.062170	0.071956	0.067195	0.069000	0.071391	0.062227	0.064759
rolling_rel_min_speed_60min	0.003373	0.000744	0.001221	0.000259	0.000637	0.000992	0.001158	0.001873	0.007601
rolling_rel_min_orientation_60min	0.011565	0.011144	0.008308	0.006686	0.010868	0.012996	0.007808	0.010573	0.007994

Figure 44: Classifier feature importances (gain) resulting from random sampling

	10	21	32	43	54	65	76	87	98
Latitude	0.012385	0.022893	0.033198	0.043988	0.044699	0.035504	0.035194	0.032531	0.028266
Longitude	0.019106	0.024555	0.031782	0.037037	0.034970	0.039879	0.038306	0.037634	0.042082
SOG	0.003004	0.009614	0.007651	0.006024	0.004496	0.003107	0.003065	0.002031	0.001238
COG	0.004364	0.019745	0.017881	0.016197	0.017011	0.011085	0.009266	0.007349	0.008631
time_elapsed_s	0.033112	0.029037	0.020040	0.014901	0.016712	0.010295	0.011041	0.009384	0.007518
expanding_mean_lat_full	0.000572	0.012333	0.019086	0.029404	0.030464	0.026403	0.024741	0.021944	0.017863
expanding_mean_lon_full	0.001512	0.005329	0.004958	0.006844	0.008169	0.012233	0.012625	0.004864	0.008901
expanding_mean_speed_full	0.002134	0.004983	0.006943	0.002838	0.001531	0.001199	0.001536	0.001028	0.001520
expanding_mean_orientation_full	0.000078	0.004712	0.006963	0.007124	0.008735	0.008279	0.010691	0.009203	0.007316
expanding_max_lat_full	0.000578	0.002183	0.001889	0.004132	0.004512	0.006049	0.010520	0.011923	0.013803
expanding_max_lon_full	0.001251	0.016456	0.018362	0.016421	0.019751	0.021122	0.019571	0.021439	0.021083
expanding_max_speed_full	0.000907	0.003529	0.004566	0.002103	0.002154	0.001378	0.001489	0.001299	0.000959
expanding_max_orientation_full	0.001110	0.001445	0.001924	0.001124	0.002887	0.002462	0.003290	0.005100	0.004494
expanding_min_lat_full	0.000228	0.000949	0.006801	0.017104	0.023454	0.021900	0.026470	0.021082	0.018135
expanding_min_lon_full	0.001329	0.001113	0.002497	0.003562	0.005770	0.008187	0.006775	0.006646	0.005107
expanding_min_speed_full	0.000165	0.000140	0.000467	0.000840	0.001866	0.001006	0.000289	0.000555	0.000195
expanding_min_orientation_full	0.001199	0.001149	0.001195	0.001111	0.001156	0.000797	0.000515	0.000774	0.000732
expanding_std_lat_full	0.049115	0.035424	0.019756	0.010139	0.006099	0.004599	0.004245	0.003485	0.003003
expanding_std_lon_full	0.041056	0.040480	0.031726	0.029652	0.023810	0.024572	0.031444	0.030067	0.036331
expanding_std_speed_full	0.000196	0.001081	0.000660	0.002595	0.003104	0.003361	0.003289	0.002226	0.002413
expanding_std_orientation_full	0.016901	0.031217	0.038808	0.037457	0.033011	0.027915	0.019001	0.017692	0.015418
expanding_median_lat_full	0.000763	0.001165	0.002185	0.003419	0.003012	0.004053	0.003002	0.002525	0.002867
expanding_median_lon_full	0.000021	0.000659	0.000784	0.001138	0.003704	0.008619	0.012759	0.015486	0.014641
expanding_median_speed_full	0.000269	0.003435	0.005059	0.003458	0.004449	0.003473	0.003307	0.002853	0.002670
expanding_median_orientation_full	0.003185	0.001339	0.005018	0.008157	0.003941	0.005252	0.005755	0.003460	0.001830
expanding_range_lat_full	0.020049	0.026378	0.023216	0.010095	0.007317	0.005569	0.004474	0.003319	0.002403
expanding_range_lon_full	0.056577	0.047653	0.058500	0.067577	0.058492	0.057061	0.054659	0.059066	0.064123
expanding_range_speed_full	0.004165	0.003410	0.003065	0.001933	0.001524	0.002299	0.001479	0.000894	0.001007
expanding_range_orientation_full	0.000112	0.000211	0.001233	0.001523	0.002713	0.002027	0.003080	0.003422	0.003557
expanding_rel_max_lat_full	0.004921	0.008202	0.009707	0.017691	0.014900	0.017205	0.016783	0.017830	0.015929
expanding_rel_max_lon_full	0.004336	0.007694	0.008869	0.005961	0.008031	0.007031	0.006781	0.005100	0.004950
expanding_rel_max_speed_full	0.003257	0.008890	0.003922	0.006517	0.005421	0.004782	0.004668	0.004034	0.003920
expanding_rel_max_orientation_full	0.001734	0.001720	0.002152	0.002847	0.004485	0.004500	0.004013	0.004611	0.004651
expanding_rel_min_lat_full	0.000345	0.007457	0.007186	0.005246	0.006937	0.004784	0.004699	0.004046	0.003168
expanding_rel_min_lon_full	0.001680	0.003096	0.012641	0.014800	0.014504	0.013945	0.010749	0.007630	0.006172
expanding_rel_min_speed_full	0.001809	0.004334	0.002473	0.002789	0.002122	0.002398	0.001900	0.003697	0.001797
expanding_rel_min_orientation_full	0.000418	0.001850	0.001440	0.002187	0.001917	0.003840	0.002550	0.003417	0.002895
rolling_mean_lat_10min	0.000183	0.007001	0.006958	0.012580	0.017568	0.020152	0.019841	0.015507	0.017935
rolling_mean_lon_10min	0.000000	0.000534	0.004083	0.006616	0.008905	0.005442	0.009148	0.004572	0.006568
rolling_mean_speed_10min	0.000762	0.002209	0.001062	0.000276	0.001077	0.001203	0.001236	0.001292	0.000626
rolling_mean_orientation_10min	0.012204	0.008583	0.015855	0.021608	0.011738	0.019542	0.024603	0.027905	0.026179
rolling_max_lat_10min	0.000001	0.000926	0.001652	0.000951	0.000952	0.000232	0.000443	0.000410	0.000490
rolling_max_lon_10min	0.000000	0.000000	0.002096	0.001758	0.002388	0.001513	0.003118	0.002337	0.002853
rolling_max_speed_10min	0.000000	0.001020	0.000067	0.000122	0.000759	0.000735	0.001034	0.001117	0.000544
rolling_max_orientation_10min	0.014396	0.017830	0.020145	0.022250	0.023369	0.028581	0.027178	0.031162	0.034349
rolling_min_lat_10min	0.000000	0.000000	0.000223	0.000582	0.000231	0.000052	0.000894	0.000013	0.000288
rolling_min_lon_10min	0.000000	0.000033	0.001542	0.004557	0.003081	0.016145	0.012065	0.025534	0.033119
rolling_min_speed_10min	0.001042	0.001454	0.001057	0.001150	0.003255	0.005610	0.002477	0.002845	0.001560
rolling_min_orientation_10min	0.005786	0.004483	0.005473	0.007305	0.009766	0.003547	0.005030	0.004931	0.006457
rolling_std_lat_10min	0.000052	0.010973	0.005439	0.006088	0.005869	0.006393	0.005837	0.003414	0.001686
rolling_std_lon_10min	0.020146	0.016780	0.015523	0.009539	0.006173	0.011202	0.006218	0.005928	0.004499
rolling_std_speed_10min	0.026716	0.011947	0.004853	0.005558	0.005532	0.005774	0.004794	0.004513	0.004225
rolling_std_orientation_10min	0.001113	0.003790	0.006453	0.006913	0.012260	0.009271	0.005113	0.007501	0.006684
rolling_median_lat_10min	0.000112	0.000873	0.001186	0.006165	0.006933	0.005191	0.007785	0.005641	0.005459
rolling_median_lon_10min	0.000000	0.000298	0.000521	0.000909	0.001440	0.004812	0.003351	0.003146	0.004431
rolling_median_speed_10min	0.000000	0.000262	0.000000	0.000357	0.000000	0.000268	0.000000	0.000147	0.000056
rolling_median_orientation_10min	0.001177	0.000724	0.003894	0.006140	0.006513	0.004031	0.005600	0.004919	0.006225
rolling_range_lat_10min	0.000000	0.000322	0.003979	0.001209	0.002043	0.002076	0.000920	0.001198	0.001342
rolling_range_lon_10min	0.013744	0.017579	0.018743	0.022171	0.023611	0.019958	0.016205	0.017145	0.019633
rolling_range_speed_10min	0.032571	0.013140	0.005900	0.004702	0.004062	0.003992	0.002959	0.002086	0.001683
rolling_range_orientation_10min	0.000000	0.003940	0.002399	0.004410	0.003944	0.002404	0.002859	0.002075	0.001334
rolling_rel_max_lat_10min	0.061720	0.062786	0.069536	0.064027	0.071017	0.065097	0.061136	0.063901	0.070671
rolling_rel_max_lon_10min	0.000431	0.002951	0.000539	0.000000	0.000703	0.000192	0.000000	0.000000	0.000008
rolling_rel_max_speed_10min	0.015564	0.021455	0.016339	0.012133	0.011120	0.008741	0.006023	0.006551	0.005486
rolling_rel_max_orientation_10min	0.011981	0.020740	0.012626	0.009881	0.008503	0.006572	0.004676	0.005226	0.005136
rolling_rel_min_lat_10min	0.000382	0.003641	0.004780	0.002163	0.002422	0.003130	0.000530	0.000557	0.000134
rolling_rel_min_lon_10min	0.136071	0.058375	0.044765	0.037180	0.034178	0.032070	0.035145	0.032280	0.036510
rolling_rel_min_speed_10min	0.025173	0.018830	0.013830	0.011180	0.007207	0.004954	0.005546	0.004748	0.004864
rolling_rel_min_orientation_10min	0.007242	0.009008	0.006728	0.008070	0.007310	0.006396	0.007460	0.004720	0.004182
rolling_mean_lat_60min	0.000636	0.002671	0.005553	0.007435	0.008563	0.016096	0.016936	0.022890	0.027085
rolling_mean_lon_60min	0.000000	0.001172	0.001679	0.003623	0.002116	0.004259	0.005600	0.007442	0.004443
rolling_mean_speed_60min	0.000000	0.002557	0.005038	0.003821	0.006468	0.013641	0.015662	0.013230	0.012073
rolling_mean_orientation_60min	0.010652	0.014240	0.014948	0.024601	0.015136	0.014124	0.020551	0.028341	0.025881
rolling_max_lat_60min	0.000000	0.001427	0.000790	0.002631	0.007665	0.007391	0.006886	0.008062	0.006096
rolling_max_lon_60min	0.000000	0.000000	0.000177	0.002376	0.003197	0.006477	0.005175	0.005022	0.003453
rolling_max_speed_60min	0.000980	0.002792	0.001377	0.000725	0.000543	0.000432	0.000491	0.000587	0.000458
rolling_max_orientation_60min	0.004846	0.008222	0.005154	0.006844	0.006647	0.008017	0.010936	0.010458	0.011166
rolling_min_lat_60min	0.000197	0.000817	0.004612	0.010986	0.011610	0.020178	0.016507	0.021822	0.018933
rolling_min_lon_60min	0.000958	0.003328	0.006176	0.009052	0.010161	0.011376	0.006597	0.005847	0.004353
rolling_min_speed_60min	0.000423	0.004207	0.010407	0.002873	0.006223	0.003784	0.005061	0.003958	0.002444
rolling_min_orientation_60min	0.000064	0.000888	0.002324	0.002963	0.005564	0.007374	0.005925	0.004243	0.002985
rolling_std_lat_60min	0.001880	0.004709	0.005488	0.003459	0.003849	0.003940	0.005754	0.003071	0.002508
rolling_std_lon_60min	0.020618	0.033640	0.021118	0.011658	0.005322	0.004950	0.007007	0.011826	0.005882
rolling_std_speed_60min	0.116626	0.050603	0.044764	0.033018	0.029419	0.027051	0.022860	0.019481	0.018744
rolling_std_orientation_60min	0.002315	0.002988	0.000683	0.002197	0.000956	0.000916	0.002007	0.001958	0.001141
rolling_median_lat_60min	0.000000	0.000000	0.000392	0.001376	0.004202	0.004121	0.008376	0.009150	0.008630
rolling_median_lon_60min	0.000000	0.002451	0.001177	0.001705	0.003989	0.006815	0.015121	0.016300	0.027197
rolling_median_speed_60min	0.000183	0.000158	0.000639	0.000446	0.001348	0.000696	0.001432	0.001779	0.001831
rolling_median_orientation_60min	0.001399	0.006793	0.005365	0.010177	0.016035	0.015446	0.027271	0.034792	0.046854
rolling_range_lat_60min	0.000396	0.001002	0.007416	0.002981	0.004024	0.003579	0.004329	0.003946	0.003271
rolling_range_lon_60min	0.011630	0.008610	0.011739	0.012683	0.012639	0.015674	0.011775	0.011771	0.004963
rolling_range_speed_60min	0.065848	0.050355	0.040368	0.027581	0.027939	0.026537	0.024158	0.025006	0.025124
rolling_range_orientation_60min	0.005557	0.001823	0.000714	0.000548	0.000517	0.000314	0.000679	0.000924	0.000704
rolling_rel_max_lat_60min	0.002313	0.003658	0.008072	0.009056	0.009372	0.006360	0.007098	0.005259	0.004385
rolling_rel_max_lon_60min	0.002436	0.000213	0.002134	0.001381	0.004652	0.002252	0.000955	0.001160	0.000558
rolling_rel_max_speed_60min	0.001906	0.001194	0.001993	0.001497	0.002449	0.004426	0.004926	0.005919	0.005014
rolling_rel_max_orientation_60min	0.003859	0.005076	0.006852	0.002336	0.003890	0.002339	0.002798	0.003702	0.002416
rolling_rel_min_lat_60min	0.000501	0.002106	0.003303	0.005206	0.004058	0.004333	0.002879	0.002363	0.002065
rolling_rel_min_lon_60min	0.043937	0.040470	0.026946	0.020901	0.010611	0.009417	0.009198	0.007359	0.006071
rolling_rel_min_speed_60min	0.000770	0.005327	0.007197	0.009041	0.007752	0.006844	0.006629	0.006280	0.004621
rolling_rel_min_orientation_60min	0.008515	0.012454	0.016886	0.012380	0.011624	0.008041	0.005525	0.005298	0.004413

Figure 45: Classifier feature importances (gain) resulting from uncertainty sampling

	10	21	32	43	54	65	76	87	98
Latitude	0.006708	0.028752	0.026133	0.026407	0.032059	0.024305	0.023088	0.020174	0.021062
Longitude	0.006403	0.018011	0.013780	0.009643	0.009520	0.012740	0.014265	0.014954	0.018359
SOG	0.001411	0.011974	0.006355	0.004826	0.004558	0.002759	0.001928	0.002129	0.001924
COG	0.012085	0.017858	0.010108	0.007486	0.007046	0.005311	0.006945	0.008093	0.008265
time_elapsed_s	0.046431	0.027477	0.023433	0.020522	0.019389	0.020005	0.021402	0.018694	0.020300
expanding_mean_lat_full	0.003093	0.011367	0.010531	0.020052	0.023943	0.024221	0.027757	0.031588	0.027574
expanding_mean_lon_full	0.000750	0.006035	0.004716	0.005347	0.008870	0.008163	0.005786	0.008009	0.008567
expanding_mean_speed_full	0.000420	0.004556	0.004073	0.005826	0.004447	0.003951	0.002572	0.001731	0.001894
expanding_mean_orientation_full	0.002366	0.004528	0.004100	0.007620	0.006290	0.007849	0.007159	0.009235	0.012389
expanding_max_lat_full	0.000559	0.006189	0.006998	0.009644	0.008091	0.012525	0.011919	0.010361	0.006544
expanding_max_lon_full	0.004216	0.019076	0.015423	0.008053	0.008816	0.006844	0.005003	0.006265	0.008434
expanding_max_speed_full	0.001724	0.005206	0.004707	0.002218	0.002712	0.002027	0.001194	0.001826	0.000934
expanding_max_orientation_full	0.001329	0.001122	0.003736	0.004499	0.002332	0.002142	0.001076	0.000920	0.001014
expanding_min_lat_full	0.000000	0.003724	0.003494	0.005514	0.006376	0.008544	0.010029	0.010080	0.011405
expanding_min_lon_full	0.001865	0.001807	0.006100	0.004760	0.003463	0.003623	0.005314	0.004181	0.003964
expanding_min_speed_full	0.000000	0.000651	0.000770	0.002749	0.003218	0.004586	0.005441	0.002270	0.001025
expanding_min_orientation_full	0.000000	0.002400	0.004027	0.002413	0.001589	0.001694	0.000952	0.000787	0.000604
expanding_std_lat_full	0.021234	0.026151	0.016885	0.008716	0.006982	0.005452	0.003741	0.004006	0.003989
expanding_std_lon_full	0.053307	0.051234	0.042840	0.040000	0.043213	0.053625	0.060139	0.067094	0.064754
expanding_std_speed_full	0.000094	0.001046	0.003410	0.003304	0.004705	0.004997	0.005267	0.006343	0.006815
expanding_std_orientation_full	0.028876	0.052549	0.067457	0.054367	0.055614	0.053257	0.049787	0.044612	0.042896
expanding_median_lat_full	0.001125	0.000842	0.003213	0.004580	0.005309	0.003537	0.001921	0.003304	0.003921
expanding_median_lon_full	0.000897	0.003364	0.004161	0.004058	0.001889	0.002034	0.002090	0.001658	0.002468
expanding_median_speed_full	0.002052	0.007106	0.002642	0.003602	0.003794	0.003910	0.004021	0.004257	0.004044
expanding_median_orientation_full	0.001944	0.004946	0.008754	0.008181	0.006823	0.004826	0.003992	0.003233	0.002568
expanding_range_lat_full	0.005125	0.015555	0.020379	0.018031	0.012467	0.014175	0.008546	0.009105	0.003806
expanding_range_lon_full	0.029315	0.043878	0.040451	0.048386	0.055171	0.067228	0.078752	0.077993	0.081835
expanding_range_speed_full	0.000000	0.003578	0.004073	0.002430	0.001466	0.002913	0.002600	0.001304	0.001482
expanding_range_orientation_full	0.001421	0.002784	0.004028	0.002743	0.002950	0.001868	0.001333	0.001297	0.001482
expanding_rel_max_lat_full	0.000870	0.010610	0.019262	0.029492	0.025058	0.024427	0.023024	0.023006	0.019323
expanding_rel_max_lon_full	0.002086	0.005981	0.004729	0.005629	0.008728	0.005303	0.005146	0.007526	0.007113
expanding_rel_max_speed_full	0.007393	0.005618	0.009809	0.007330	0.009652	0.008293	0.006587	0.006065	0.004587
expanding_rel_max_orientation_full	0.000000	0.006913	0.005802	0.005822	0.002471	0.001863	0.001743	0.001507	0.001988
expanding_rel_min_lat_full	0.000413	0.000786	0.002242	0.004554	0.004550	0.004734	0.004823	0.002187	0.003022
expanding_rel_min_lon_full	0.001185	0.005639	0.004493	0.006864	0.003914	0.003907	0.005649	0.003952	0.005856
expanding_rel_min_speed_full	0.003037	0.002872	0.004354	0.005092	0.004759	0.005992	0.003479	0.003600	0.005063
expanding_rel_min_orientation_full	0.000537	0.002155	0.004231	0.006113	0.006722	0.005004	0.004090	0.003363	0.007535
rolling_mean_lat_10min	0.001445	0.004800	0.012336	0.010285	0.009884	0.008884	0.008225	0.008711	0.011615
rolling_mean_lon_10min	0.000000	0.000816	0.003874	0.002323	0.000997	0.000746	0.001218	0.002192	0.000925
rolling_mean_speed_10min	0.000152	0.002914	0.003292	0.000637	0.001484	0.000988	0.000521	0.000714	0.000896
rolling_mean_orientation_10min	0.006114	0.014480	0.005966	0.006787	0.008469	0.006359	0.007342	0.009508	0.006398
rolling_max_lat_10min	0.000000	0.000986	0.000547	0.000376	0.000916	0.000737	0.001132	0.001279	0.000539
rolling_max_lon_10min	0.000000	0.000091	0.000401	0.000346	0.000121	0.000362	0.001053	0.000664	0.000226
rolling_max_speed_10min	0.000000	0.000510	0.002332	0.000888	0.000366	0.000254	0.000342	0.000384	0.000227
rolling_max_orientation_10min	0.005981	0.010445	0.013958	0.009631	0.009156	0.012613	0.013876	0.017583	0.013347
rolling_min_lat_10min	0.000000	0.000437	0.000285	0.000363	0.000175	0.000138	0.000204	0.000528	0.000340
rolling_min_lon_10min	0.000000	0.000799	0.001882	0.001169	0.001691	0.001443	0.001655	0.001317	0.001931
rolling_min_speed_10min	0.000000	0.003001	0.004679	0.004036	0.002837	0.003602	0.002267	0.001601	0.001141
rolling_min_orientation_10min	0.000046	0.001181	0.003731	0.007463	0.002577	0.002689	0.002818	0.001202	0.001743
rolling_std_lat_10min	0.000327	0.006681	0.005105	0.005654	0.004298	0.004202	0.004970	0.001556	0.001756
rolling_std_lon_10min	0.002557	0.021964	0.018490	0.011975	0.012077	0.008972	0.009635	0.003964	0.003361
rolling_std_speed_10min	0.043172	0.017167	0.014449	0.009238	0.004702	0.003530	0.003173	0.002568	0.006067
rolling_std_orientation_10min	0.003188	0.009199	0.010782	0.015174	0.013187	0.012847	0.007636	0.005427	0.006008
rolling_median_lat_10min	0.000000	0.000484	0.001099	0.002364	0.002377	0.002664	0.002228	0.002718	0.002177
rolling_median_lon_10min	0.000000	0.000000	0.000053	0.000061	0.000415	0.000733	0.000412	0.001219	0.001024
rolling_median_speed_10min	0.000000	0.000151	0.000219	0.000260	0.000047	0.000314	0.000058	0.000195	0.000081
rolling_median_orientation_10min	0.000000	0.000554	0.002187	0.002687	0.005025	0.004630	0.001287	0.001854	0.001186
rolling_range_lat_10min	0.000000	0.002254	0.006831	0.004938	0.003390	0.003532	0.003393	0.003792	0.002483
rolling_range_lon_10min	0.005139	0.011379	0.013224	0.022610	0.031925	0.035430	0.039827	0.025705	0.022398
rolling_range_speed_10min	0.029810	0.021015	0.009489	0.006127	0.004689	0.002529	0.002015	0.002198	0.004473
rolling_range_orientation_10min	0.001107	0.001518	0.008932	0.006985	0.005427	0.003543	0.003775	0.002589	0.001816
rolling_rel_max_lat_10min	0.108242	0.050341	0.060426	0.077859	0.104392	0.118940	0.130743	0.146808	0.159198
rolling_rel_max_lon_10min	0.000130	0.000000	0.000553	0.000000	0.000011	0.000000	0.000000	0.000000	0.000000
rolling_rel_max_speed_10min	0.038571	0.031179	0.025575	0.023574	0.012806	0.013228	0.011838	0.013041	0.011260
rolling_rel_max_orientation_10min	0.010104	0.011599	0.013027	0.011085	0.009641	0.006924	0.007221	0.006140	0.006147
rolling_rel_min_lat_10min	0.000791	0.001622	0.002145	0.003712	0.003755	0.002558	0.002404	0.002742	0.001383
rolling_rel_min_lon_10min	0.180582	0.073077	0.042497	0.029480	0.028890	0.027402	0.033403	0.031096	0.027223
rolling_rel_min_speed_10min	0.029578	0.010779	0.010922	0.009118	0.007244	0.006337	0.006492	0.005623	0.007073
rolling_rel_min_orientation_10min	0.008244	0.004931	0.005691	0.009042	0.008597	0.006567	0.007321	0.007653	0.005639
rolling_mean_lat_60min	0.000000	0.003126	0.007111	0.013440	0.017047	0.017240	0.019190	0.018259	0.020749
rolling_mean_lon_60min	0.000000	0.001226	0.001193	0.001521	0.000545	0.000226	0.000166	0.000213	0.000320
rolling_mean_speed_60min	0.000000	0.001063	0.006488	0.005223	0.005469	0.008365	0.004375	0.008796	0.008130
rolling_mean_orientation_60min	0.007655	0.020180	0.017513	0.012892	0.014375	0.015774	0.012840	0.019049	0.017493
rolling_max_lat_60min	0.000000	0.000721	0.004530	0.005457	0.002256	0.001796	0.003403	0.003903	0.005404
rolling_max_lon_60min	0.000000	0.000000	0.000000	0.000524	0.000000	0.000658	0.000590	0.000000	0.000095
rolling_max_speed_60min	0.000256	0.000355	0.000476	0.002021	0.000353	0.001796	0.001317	0.001783	0.000488
rolling_max_orientation_60min	0.003824	0.005851	0.005677	0.015299	0.009422	0.010560	0.008629	0.010744	0.010091
rolling_min_lat_60min	0.000000	0.000915	0.004358	0.009697	0.011205	0.012247	0.014455	0.012841	0.014145
rolling_min_lon_60min	0.001568	0.001851	0.005101	0.004724	0.004665	0.004144	0.005109	0.005511	0.004840
rolling_min_speed_60min	0.001010	0.005068	0.006613	0.005280	0.005682	0.005128	0.005101	0.004232	0.005873
rolling_min_orientation_60min	0.000226	0.001527	0.003217	0.002698	0.003840	0.003433	0.003749	0.003300	0.003534
rolling_std_lat_60min	0.002432	0.006785	0.009255	0.006559	0.008180	0.013454	0.008220	0.008869	0.008823
rolling_std_lon_60min	0.004544	0.006702	0.008527	0.005041	0.003346	0.001873	0.002677	0.005229	0.006988
rolling_std_speed_60min	0.105935	0.055113	0.034604	0.029676	0.029772	0.022389	0.014688	0.013352	0.011322
rolling_std_orientation_60min	0.001107	0.000309	0.001376	0.005380	0.001772	0.001079	0.000790	0.000740	0.001399
rolling_median_lat_60min	0.000000	0.000071	0.002507	0.005334	0.006132	0.004469	0.004697	0.005222	0.007173
rolling_median_lon_60min	0.000000	0.000000	0.000967	0.000000	0.000852	0.000853	0.001627	0.001694	0.001923
rolling_median_speed_60min	0.000251	0.000398	0.003071	0.001213	0.002298	0.002092	0.000728	0.001223	0.001274
rolling_median_orientation_60min	0.001502	0.005739	0.003241	0.007732	0.007372	0.006493	0.015474	0.009470	0.011418
rolling_range_lat_60min	0.000000	0.004691	0.005173	0.004977	0.005106	0.005936	0.007168	0.008172	0.008031
rolling_range_lon_60min	0.000384	0.002313	0.003533	0.005752	0.004922	0.004402	0.004907	0.005717	0.004619
rolling_range_speed_60min	0.059434	0.034191	0.042510	0.040373	0.046677	0.044603	0.043692	0.050890	0.051997
rolling_range_orientation_60min	0.001663	0.000623	0.000719	0.000662	0.001088	0.001019	0.001434	0.001389	0.002595
rolling_rel_max_lat_60min	0.004696	0.010538	0.013282	0.012561	0.009358	0.006631	0.004927	0.004182	0.004643
rolling_rel_max_lon_60min	0.001063	0.000085	0.001413	0.001720	0.002262	0.002301	0.000777	0.000534	0.000000
rolling_rel_max_speed_60min	0.002436	0.003761	0.006631	0.007274	0.010970	0.008540	0.008506	0.006981	0.007675
rolling_rel_max_orientation_60min	0.002232	0.002779	0.004011	0.003924	0.003230	0.004532	0.004861	0.003520	0.002883
rolling_rel_min_lat_60min	0.003229	0.007699	0.012113	0.008116	0.004388	0.002578	0.001851	0.001531	0.002173
rolling_rel_min_lon_60min	0.057666	0.037365	0.016710	0.024917	0.022552	0.019848	0.016509	0.016554	0.013735
rolling_rel_min_speed_60min	0.000373	0.009196	0.010366	0.009638	0.009716	0.010543	0.009721	0.010191	0.011657
rolling_rel_min_orientation_60min	0.008081	0.019720	0.022042	0.014649	0.013444	0.012284	0.009438	0.008981	0.006755

Figure 46: Classifier feature importances (gain) resulting from QBC sampling

	10	21	32	43	54	65	76	87	98
Latitude	0.003855	0.013931	0.012865	0.026614	0.034394	0.034331	0.030492	0.029225	0.025880
Longitude	0.002740	0.014660	0.011606	0.009622	0.018616	0.034937	0.038717	0.036042	0.039801
SOG	0.001032	0.013200	0.004358	0.003748	0.001868	0.001614	0.001799	0.002301	0.001826
COG	0.006377	0.031052	0.016143	0.026603	0.014902	0.011662	0.013476	0.013989	0.012680
time_elapsed_s	0.000000	0.001385	0.000585	0.003871	0.003089	0.005693	0.008989	0.012204	0.010503
expanding_mean_lat_full	0.000000	0.003647	0.003823	0.011314	0.026402	0.026717	0.024573	0.021369	0.017705
expanding_mean_lon_full	0.000301	0.003976	0.000249	0.002204	0.005032	0.012093	0.015282	0.010758	0.005706
expanding_mean_speed_full	0.000207	0.003546	0.001613	0.000486	0.001687	0.000914	0.001247	0.001796	0.003569
expanding_mean_orientation_full	0.000000	0.000110	0.001453	0.001163	0.001097	0.003325	0.007002	0.007039	0.006004
expanding_max_lat_full	0.000091	0.000011	0.014754	0.011415	0.011881	0.008212	0.007561	0.007369	0.013295
expanding_max_lon_full	0.003661	0.010170	0.004978	0.011095	0.027910	0.029118	0.030966	0.023362	0.023692
expanding_max_speed_full	0.000000	0.000205	0.001959	0.001965	0.001096	0.001689	0.001631	0.001879	0.001639
expanding_max_orientation_full	0.000000	0.004180	0.003261	0.002714	0.003272	0.004512	0.006853	0.004943	0.009700
expanding_min_lat_full	0.000000	0.000000	0.000939	0.009915	0.011865	0.011393	0.010897	0.008694	0.008984
expanding_min_lon_full	0.000000	0.000000	0.001331	0.001954	0.001903	0.001545	0.002437	0.003721	0.002436
expanding_min_speed_full	0.000000	0.000000	0.001398	0.001299	0.000281	0.000740	0.000612	0.000027	0.000245
expanding_min_orientation_full	0.000000	0.018861	0.002742	0.002361	0.002148	0.001024	0.002083	0.002131	0.002190
expanding_std_lat_full	0.006188	0.028711	0.016593	0.021740	0.011257	0.009661	0.006666	0.003780	0.002986
expanding_std_lon_full	0.009119	0.024826	0.002365	0.003826	0.005360	0.008521	0.009795	0.014552	0.018717
expanding_std_speed_full	0.000463	0.001606	0.002448	0.001102	0.002393	0.001605	0.001921	0.001599	0.001432
expanding_std_orientation_full	0.026951	0.094926	0.132198	0.086104	0.052200	0.037217	0.028799	0.025078	0.020735
expanding_median_lat_full	0.000000	0.000856	0.001139	0.000938	0.002505	0.004073	0.004279	0.007420	0.003233
expanding_median_lon_full	0.000000	0.000522	0.000437	0.001120	0.001479	0.003081	0.006952	0.004042	0.006875
expanding_median_speed_full	0.000000	0.003391	0.003836	0.001934	0.002639	0.002582	0.002877	0.002424	0.002298
expanding_median_orientation_full	0.001226	0.002370	0.000482	0.002164	0.002344	0.001676	0.002012	0.002803	0.002461
expanding_range_lat_full	0.007286	0.020038	0.038661	0.026626	0.017638	0.011151	0.004128	0.003056	0.003289
expanding_range_lon_full	0.013868	0.016267	0.016357	0.030564	0.061679	0.075105	0.075629	0.084556	0.085291
expanding_range_speed_full	0.000578	0.001636	0.001063	0.001348	0.000892	0.001485	0.000908	0.001607	0.001735
expanding_range_orientation_full	0.000488	0.000754	0.001299	0.003133	0.002833	0.004614	0.006059	0.004410	0.004237
expanding_rel_max_lat_full	0.000187	0.002085	0.007782	0.010981	0.013657	0.016626	0.019032	0.018392	0.018090
expanding_rel_max_lon_full	0.000679	0.001755	0.004030	0.005491	0.005474	0.006039	0.005228	0.005712	0.005450
expanding_rel_max_speed_full	0.000395	0.000976	0.000841	0.002298	0.002602	0.002521	0.004590	0.004786	0.005676
expanding_rel_max_orientation_full	0.000000	0.001479	0.001854	0.001375	0.006964	0.009164	0.006727	0.005882	0.008442
expanding_rel_min_lat_full	0.000158	0.001424	0.000508	0.002657	0.001146	0.001520	0.002329	0.001098	0.002777
expanding_rel_min_lon_full	0.000000	0.007707	0.004882	0.004294	0.004890	0.008107	0.003320	0.005382	0.008170
expanding_rel_min_speed_full	0.000532	0.005161	0.003480	0.002070	0.002596	0.002141	0.003278	0.003331	0.006500
expanding_rel_min_orientation_full	0.000169	0.000000	0.000562	0.002272	0.002479	0.004448	0.004570	0.005117	0.005873
rolling_mean_lat_10min	0.000000	0.000000	0.001886	0.007828	0.005653	0.006886	0.008575	0.013488	0.011917
rolling_mean_lon_10min	0.000000	0.000000	0.000256	0.000469	0.000813	0.000037	0.000893	0.002365	0.003788
rolling_mean_speed_10min	0.000000	0.008449	0.003044	0.004933	0.000465	0.000718	0.001139	0.001257	0.001028
rolling_mean_orientation_10min	0.019163	0.004328	0.003804	0.005684	0.005440	0.012703	0.011976	0.020809	0.016898
rolling_max_lat_10min	0.000000	0.000000	0.000030	0.000260	0.001314	0.000293	0.000779	0.000378	0.000407
rolling_max_lon_10min	0.000000	0.000000	0.000000	0.000000	0.000151	0.000348	0.001025	0.001236	0.000837
rolling_max_speed_10min	0.000166	0.003231	0.000000	0.000045	0.000013	0.000005	0.000098	0.000284	0.000853
rolling_max_orientation_10min	0.010506	0.005819	0.003115	0.004832	0.009722	0.014305	0.012805	0.016157	0.013571
rolling_min_lat_10min	0.000000	0.000000	0.000000	0.000000	0.000162	0.000357	0.000813	0.000639	0.000172
rolling_min_lon_10min	0.000000	0.000000	0.000000	0.000075	0.000531	0.000774	0.001850	0.005033	0.008062
rolling_min_speed_10min	0.000893	0.002553	0.002482	0.010720	0.007272	0.006388	0.007266	0.010117	0.009976
rolling_min_orientation_10min	0.000000	0.002821	0.001261	0.004694	0.004861	0.004612	0.006441	0.006083	0.009896
rolling_std_lat_10min	0.000165	0.002686	0.002068	0.007191	0.011471	0.008183	0.001901	0.002253	0.002890
rolling_std_lon_10min	0.013972	0.003711	0.005256	0.006917	0.003145	0.004017	0.003747	0.007389	0.005370
rolling_std_speed_10min	0.027030	0.025275	0.014733	0.008413	0.006892	0.007503	0.007735	0.008173	0.007658
rolling_std_orientation_10min	0.003239	0.006192	0.001927	0.006736	0.012000	0.011162	0.011131	0.013801	0.007470
rolling_median_lat_10min	0.000000	0.000000	0.000873	0.000546	0.002294	0.003574	0.002314	0.005926	0.006417
rolling_median_lon_10min	0.000000	0.000219	0.000500	0.003266	0.002316	0.004157	0.004512	0.004483	0.002278
rolling_median_speed_10min	0.000000	0.000000	0.000887	0.000324	0.000079	0.000000	0.000000	0.000000	0.000019
rolling_median_orientation_10min	0.005496	0.001638	0.001392	0.001470	0.003393	0.003425	0.004151	0.007320	0.007767
rolling_range_lat_10min	0.000000	0.001741	0.000000	0.000265	0.003615	0.002310	0.001700	0.001690	0.000755
rolling_range_lon_10min	0.067015	0.031403	0.012023	0.019149	0.012930	0.012072	0.012618	0.013845	0.018208
rolling_range_speed_10min	0.012347	0.011628	0.018302	0.011006	0.008127	0.004621	0.004122	0.004665	0.002983
rolling_range_orientation_10min	0.001333	0.002274	0.001390	0.003205	0.005618	0.006073	0.003957	0.003695	0.003042
rolling_rel_max_lat_10min	0.080829	0.088586	0.119120	0.067117	0.033844	0.031434	0.033249	0.049467	0.054942
rolling_rel_max_lon_10min	0.000000	0.000000	0.000000	0.000000	0.000045	0.000018	0.000060	0.000154	0.000041
rolling_rel_max_speed_10min	0.043623	0.033488	0.014999	0.011762	0.010818	0.007701	0.007544	0.007228	0.007607
rolling_rel_max_orientation_10min	0.063584	0.026763	0.014786	0.009281	0.009449	0.009070	0.006683	0.005595	0.005188
rolling_rel_min_lat_10min	0.000000	0.000000	0.000011	0.000000	0.000088	0.000274	0.000215	0.001080	0.001343
rolling_rel_min_lon_10min	0.161865	0.081491	0.149622	0.157305	0.132647	0.100280	0.088902	0.079029	0.068158
rolling_rel_min_speed_10min	0.015659	0.019339	0.013483	0.013383	0.005780	0.005572	0.005817	0.005989	0.004817
rolling_rel_min_orientation_10min	0.002203	0.003248	0.003807	0.006541	0.006521	0.006026	0.004973	0.006690	0.006025
rolling_mean_lat_60min	0.000000	0.000000	0.001362	0.002900	0.006083	0.009946	0.011398	0.017954	0.017419
rolling_mean_lon_60min	0.000000	0.000510	0.001245	0.001817	0.004890	0.001771	0.003313	0.005772	0.006289
rolling_mean_speed_60min	0.000000	0.000588	0.001809	0.004926	0.005537	0.003820	0.006855	0.014303	0.018340
rolling_mean_orientation_60min	0.014854	0.021545	0.008435	0.010241	0.008373	0.008933	0.011230	0.007256	0.010027
rolling_max_lat_60min	0.000000	0.000000	0.000513	0.002577	0.002688	0.004293	0.008543	0.007401	0.007692
rolling_max_lon_60min	0.000000	0.000000	0.000000	0.000000	0.000215	0.000000	0.000133	0.000327	0.000487
rolling_max_speed_60min	0.004240	0.000303	0.001018	0.000547	0.001070	0.000288	0.000595	0.000591	0.000776
rolling_max_orientation_60min	0.004241	0.031305	0.019583	0.007756	0.016542	0.013733	0.017166	0.016866	0.018192
rolling_min_lat_60min	0.000000	0.000000	0.000000	0.000875	0.003987	0.009147	0.015781	0.013586	0.014614
rolling_min_lon_60min	0.000000	0.000219	0.001240	0.002249	0.006900	0.006274	0.008018	0.009460	0.010020
rolling_min_speed_60min	0.000930	0.004286	0.003185	0.003604	0.006312	0.007470	0.012524	0.004957	0.006435
rolling_min_orientation_60min	0.000093	0.001780	0.004292	0.005206	0.005458	0.005329	0.008277	0.008634	0.009684
rolling_std_lat_60min	0.006180	0.021703	0.025254	0.024354	0.012354	0.006970	0.003017	0.003280	0.001753
rolling_std_lon_60min	0.031800	0.007593	0.006606	0.006389	0.004606	0.006838	0.004022	0.005219	0.001651
rolling_std_speed_60min	0.204093	0.052925	0.040995	0.030838	0.017491	0.010712	0.007699	0.006936	0.004895
rolling_std_orientation_60min	0.000000	0.006142	0.003287	0.001060	0.002850	0.001496	0.001665	0.002159	0.001703
rolling_median_lat_60min	0.000000	0.000000	0.000433	0.003000	0.006454	0.007431	0.010779	0.011481	0.012065
rolling_median_lon_60min	0.000000	0.000000	0.000000	0.000024	0.000124	0.000853	0.003767	0.006252	0.016986
rolling_median_speed_60min	0.001897	0.000000	0.001459	0.000000	0.000613	0.000216	0.000038	0.000042	0.004549
rolling_median_orientation_60min	0.004851	0.005022	0.003884	0.007650	0.011832	0.020354	0.022047	0.015202	0.011593
rolling_range_lat_60min	0.000877	0.001154	0.003072	0.007618	0.017410	0.020072	0.014718	0.006919	0.004570
rolling_range_lon_60min	0.000395	0.002529	0.002245	0.003388	0.003933	0.003482	0.006083	0.005378	0.008926
rolling_range_speed_60min	0.061684	0.089563	0.080584	0.080008	0.110114	0.115537	0.103778	0.081164	0.073997
rolling_range_orientation_60min	0.000000	0.000018	0.000398	0.001608	0.000520	0.000376	0.000000	0.000352	0.000777
rolling_rel_max_lat_60min	0.001207	0.001280	0.010358	0.015610	0.015369	0.012007	0.010119	0.008862	0.005930
rolling_rel_max_lon_60min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000024	0.000000	0.000268	0.000006
rolling_rel_max_speed_60min	0.003863	0.000390	0.000295	0.000666	0.002651	0.003772	0.004048	0.006403	0.006740
rolling_rel_max_orientation_60min	0.009077	0.011204	0.022701	0.012967	0.005932	0.003863	0.002293	0.002130	0.002377
rolling_rel_min_lat_60min	0.000000	0.000995	0.004318	0.011159	0.011819	0.013326	0.011462	0.007514	0.005354
rolling_rel_min_lon_60min	0.026960	0.015329	0.013160	0.008339	0.002946	0.004352	0.006822	0.006857	0.007761
rolling_rel_min_speed_60min	0.006529	0.011355	0.009903	0.013515	0.013269	0.011825	0.009843	0.008626	0.008104
rolling_rel_min_orientation_60min	0.000000	0.004778	0.006040	0.008913	0.007942	0.006429	0.006041	0.005230	0.005487

Figure 47: Classifier feature importances (gain) resulting from density-weighted sampling

	10	21	32	43	54	65	76	87	98
Latitude	0.002444	0.001019	0.002780	0.007332	0.014612	0.025351	0.019668	0.019687	0.019245
Longitude	0.013922	0.003145	0.002242	0.005999	0.008523	0.009668	0.013801	0.017011	0.021980
SOG	0.004704	0.000508	0.001525	0.009166	0.010796	0.019154	0.018653	0.021325	0.017977
COG	0.022251	0.005006	0.005663	0.006105	0.017794	0.024573	0.029437	0.029267	0.022926
time_elapsed_s	0.097477	0.070119	0.059925	0.065626	0.052331	0.045722	0.042158	0.037767	0.038925
expanding_mean_lat_full	0.000610	0.000671	0.000262	0.001813	0.000000	0.001621	0.004866	0.003874	0.002077
expanding_mean_lon_full	0.001410	0.000000	0.000000	0.000000	0.000897	0.001625	0.001200	0.000564	0.002825
expanding_mean_speed_full	0.021330	0.005029	0.001821	0.005442	0.006977	0.011148	0.009510	0.012840	0.009909
expanding_mean_orientation_full	0.005045	0.001904	0.001654	0.002854	0.001916	0.005148	0.003157	0.006929	0.005106
expanding_max_lat_full	0.013267	0.008540	0.002259	0.000869	0.001012	0.002256	0.004771	0.003598	0.004872
expanding_max_lon_full	0.010955	0.003284	0.003880	0.003826	0.007014	0.007917	0.011309	0.010685	0.011033
expanding_max_speed_full	0.005927	0.002996	0.002107	0.001680	0.001950	0.002270	0.001985	0.004024	0.005883
expanding_max_orientation_full	0.004850	0.001321	0.001788	0.002693	0.000000	0.000819	0.002433	0.000968	0.001810
expanding_min_lat_full	0.001338	0.000000	0.000000	0.000000	0.000700	0.000301	0.000000	0.000000	0.000047
expanding_min_lon_full	0.013920	0.006713	0.000544	0.001486	0.000210	0.001467	0.003065	0.000890	0.001355
expanding_min_speed_full	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000015	0.000024	0.000003
expanding_min_orientation_full	0.003268	0.000572	0.000399	0.001463	0.000905	0.000227	0.001737	0.000144	0.000405
expanding_std_lat_full	0.063300	0.064098	0.067733	0.056530	0.061117	0.048645	0.048382	0.051889	0.046126
expanding_std_lon_full	0.034194	0.026027	0.027827	0.026863	0.024376	0.026505	0.033803	0.029799	0.035557
expanding_std_speed_full	0.007692	0.002775	0.002070	0.001154	0.001065	0.002419	0.002485	0.001049	0.003444
expanding_std_orientation_full	0.022963	0.014847	0.012543	0.013658	0.011021	0.013844	0.012237	0.014434	0.012712
expanding_median_lat_full	0.000000	0.000309	0.000360	0.000000	0.000000	0.000753	0.000462	0.001385	0.000755
expanding_median_lon_full	0.001132	0.000351	0.000673	0.000076	0.000000	0.000000	0.001027	0.001217	0.000079
expanding_median_speed_full	0.006238	0.006249	0.002870	0.002266	0.003802	0.003057	0.003110	0.004322	0.004181
expanding_median_orientation_full	0.004113	0.002615	0.002133	0.000571	0.000041	0.001282	0.000458	0.001397	0.003614
expanding_range_lat_full	0.006985	0.019752	0.024113	0.021895	0.022125	0.021390	0.021955	0.018684	0.022812
expanding_range_lon_full	0.017135	0.021819	0.028651	0.029052	0.032682	0.028904	0.028982	0.025634	0.029218
expanding_range_speed_full	0.005036	0.003879	0.001239	0.000451	0.000453	0.000115	0.002509	0.000460	0.001522
expanding_range_orientation_full	0.001857	0.000355	0.000905	0.000975	0.001075	0.000655	0.000621	0.001019	0.000217
expanding_rel_max_lat_full	0.001496	0.001252	0.000194	0.001552	0.000719	0.000610	0.003622	0.003954	0.007723
expanding_rel_max_lon_full	0.004192	0.003385	0.002176	0.000549	0.000001	0.000953	0.001815	0.002700	0.002688
expanding_rel_max_speed_full	0.018083	0.011793	0.011547	0.009703	0.008758	0.005091	0.005043	0.002662	0.003711
expanding_rel_max_orientation_full	0.009170	0.002081	0.001276	0.000090	0.000299	0.000519	0.000747	0.001509	0.000273
expanding_rel_min_lat_full	0.000710	0.000573	0.001115	0.000264	0.001123	0.002223	0.000977	0.001453	0.002986
expanding_rel_min_lon_full	0.001967	0.000000	0.000381	0.000000	0.005077	0.003354	0.001858	0.002274	0.003659
expanding_rel_min_speed_full	0.002661	0.000500	0.000283	0.000283	0.000293	0.002365	0.001138	0.000738	0.001463
expanding_rel_min_orientation_full	0.015772	0.004646	0.002278	0.002713	0.001947	0.001663	0.000474	0.001905	0.001515
rolling_mean_lat_10min	0.000000	0.000000	0.000000	0.000000	0.000284	0.000060	0.000209	0.000000	0.000276
rolling_mean_lon_10min	0.000399	0.000799	0.000300	0.001306	0.001861	0.000759	0.000485	0.000187	0.000873
rolling_mean_speed_10min	0.001138	0.001288	0.001043	0.000050	0.001592	0.000837	0.001773	0.001926	0.003067
rolling_mean_orientation_10min	0.013367	0.008202	0.004297	0.002226	0.004490	0.002852	0.004289	0.004549	0.006363
rolling_max_lat_10min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
rolling_max_lon_10min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
rolling_max_speed_10min	0.003575	0.001034	0.001023	0.001212	0.000000	0.000733	0.001199	0.000000	0.000000
rolling_max_orientation_10min	0.019220	0.011972	0.011255	0.006541	0.006698	0.007470	0.007663	0.008208	0.009043
rolling_min_lat_10min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
rolling_min_lon_10min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000252	0.000395
rolling_min_speed_10min	0.002640	0.000000	0.000692	0.000173	0.002075	0.000697	0.000903	0.002425	0.005517
rolling_min_orientation_10min	0.002225	0.000561	0.000000	0.000200	0.002244	0.000838	0.001941	0.001550	0.002977
rolling_std_lat_10min	0.001035	0.000927	0.000804	0.000000	0.000039	0.000626	0.001291	0.001979	0.002349
rolling_std_lon_10min	0.048212	0.029758	0.019846	0.014085	0.013112	0.010014	0.011573	0.011733	0.011990
rolling_std_speed_10min	0.039691	0.038500	0.029334	0.023836	0.017836	0.015727	0.019437	0.018801	0.010799
rolling_std_orientation_10min	0.001824	0.000350	0.000517	0.000480	0.000949	0.000565	0.000000	0.000871	0.000000
rolling_median_lat_10min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000534
rolling_median_lon_10min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000184	0.000000	0.000088	0.000104
rolling_median_speed_10min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000338	0.000066	0.001255	0.000000
rolling_median_orientation_10min	0.003067	0.001416	0.001056	0.000270	0.000000	0.001130	0.002368	0.001226	0.001919
rolling_range_lat_10min	0.000792	0.000488	0.000969	0.000000	0.000345	0.001652	0.001617	0.000925	0.001721
rolling_range_lon_10min	0.026976	0.028425	0.030834	0.034816	0.030505	0.033947	0.027840	0.025868	0.024598
rolling_range_speed_10min	0.019722	0.023398	0.020554	0.017103	0.014056	0.012380	0.014322	0.009814	0.008440
rolling_range_orientation_10min	0.002016	0.000474	0.000389	0.000000	0.000440	0.000574	0.000507	0.001415	0.001086
rolling_rel_max_lat_10min	0.048156	0.073493	0.073364	0.077389	0.069689	0.069210	0.065268	0.060054	0.053361
rolling_rel_max_lon_10min	0.001810	0.000000	0.000000	0.000000	0.000000	0.000000	0.000032	0.000000	0.000483
rolling_rel_max_speed_10min	0.040040	0.057570	0.058328	0.065322	0.072034	0.068315	0.053362	0.049318	0.046612
rolling_rel_max_orientation_10min	0.007955	0.010901	0.007841	0.007583	0.005960	0.006022	0.007725	0.007046	0.006480
rolling_rel_min_lat_10min	0.001270	0.000665	0.000183	0.000538	0.000379	0.000710	0.000000	0.001798	0.000105
rolling_rel_min_lon_10min	0.036497	0.069014	0.076165	0.066459	0.058378	0.054611	0.049982	0.052364	0.050112
rolling_rel_min_speed_10min	0.015241	0.026730	0.027901	0.034014	0.031903	0.024970	0.018604	0.011421	0.013933
rolling_rel_min_orientation_10min	0.013147	0.011201	0.012304	0.010893	0.006797	0.010466	0.011151	0.009277	0.008980
rolling_mean_lat_60min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000256	0.001124	0.001145	0.000849
rolling_mean_lon_60min	0.000000	0.000000	0.000000	0.000144	0.000000	0.000202	0.000952	0.001309	0.000447
rolling_mean_speed_60min	0.001303	0.000421	0.000000	0.000657	0.000172	0.000000	0.002696	0.000459	0.004476
rolling_mean_orientation_60min	0.010971	0.016151	0.014926	0.007961	0.010986	0.009240	0.011024	0.011420	0.009518
rolling_max_lat_60min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000112	0.000000	0.000000
rolling_max_lon_60min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
rolling_max_speed_60min	0.006940	0.007855	0.007641	0.003875	0.003287	0.001883	0.002106	0.001954	0.002491
rolling_max_orientation_60min	0.013448	0.014047	0.009870	0.014844	0.009585	0.008085	0.008793	0.012867	0.014022
rolling_min_lat_60min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000500	0.000000
rolling_min_lon_60min	0.000000	0.000000	0.000317	0.000064	0.000000	0.000209	0.000747	0.000203	0.000035
rolling_min_speed_60min	0.000000	0.000000	0.000000	0.000087	0.000000	0.003474	0.006034	0.009844	0.017480
rolling_min_orientation_60min	0.001872	0.000607	0.000000	0.000014	0.000000	0.000272	0.000928	0.001220	0.000564
rolling_std_lat_60min	0.000497	0.000267	0.000191	0.000487	0.001150	0.003280	0.003523	0.006086	0.003709
rolling_std_lon_60min	0.014920	0.031522	0.036774	0.030178	0.035619	0.025255	0.023990	0.025845	0.023506
rolling_std_speed_60min	0.038582	0.108298	0.130510	0.136727	0.126571	0.120122	0.108061	0.094794	0.092497
rolling_std_orientation_60min	0.000314	0.000552	0.000000	0.000000	0.000000	0.000125	0.000390	0.001688	0.002099
rolling_median_lat_60min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
rolling_median_lon_60min	0.000000	0.000000	0.000000	0.000427	0.000537	0.000275	0.000379	0.000000	0.000000
rolling_median_speed_60min	0.001278	0.000000	0.000455	0.000000	0.000351	0.000308	0.000364	0.000204	0.000000
rolling_median_orientation_60min	0.006953	0.003509	0.002574	0.002902	0.002344	0.001975	0.000523	0.001138	0.000705
rolling_range_lat_60min	0.000000	0.000510	0.000000	0.000192	0.000000	0.000000	0.000000	0.000523	0.001708
rolling_range_lon_60min	0.001592	0.005435	0.009378	0.010979	0.012088	0.012288	0.010721	0.009887	0.006997
rolling_range_speed_60min	0.013558	0.029431	0.043953	0.054284	0.062961	0.076560	0.084777	0.097334	0.095392
rolling_range_orientation_60min	0.000993	0.000000	0.000747	0.000386	0.000056	0.000000	0.000617	0.000000	0.000000
rolling_rel_max_lat_60min	0.003739	0.003565	0.003322	0.001734	0.001470	0.000852	0.001170	0.002001	0.000724
rolling_rel_max_lon_60min	0.003312	0.001048	0.000000	0.000109	0.000000	0.000000	0.000798	0.000862	0.000661
rolling_rel_max_speed_60min	0.009256	0.001659	0.001799	0.001047	0.000381	0.000910	0.000492	0.000797	0.000312
rolling_rel_max_orientation_60min	0.007750	0.007086	0.008529	0.006986	0.004625	0.005612	0.003066	0.004908	0.004226
rolling_rel_min_lat_60min	0.005772	0.004131	0.001397	0.000834	0.000877	0.003504	0.001316	0.002522	0.003610
rolling_rel_min_lon_60min	0.015189	0.034240	0.043286	0.053794	0.065454	0.054688	0.062297	0.065224	0.066671
rolling_rel_min_speed_60min	0.008828	0.008219	0.007801	0.004201	0.003192	0.004897	0.003738	0.003134	0.002085
rolling_rel_min_orientation_60min	0.009562	0.009574	0.010221	0.008156	0.008241	0.009282	0.010671	0.014315	0.015192

Figure 48: Classifier feature importances (gain) resulting from xPAL sampling

Appendix E Frequently selected trajectories

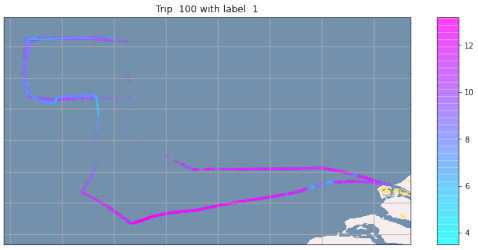

Figure 49: Looping trajectory frequently selected near the start of learning (uncertainty sampling)

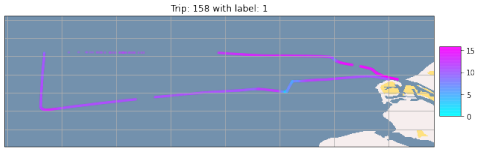

Figure 50: Looping trajectory frequently selected near the start of learning (uncertainty sampling)

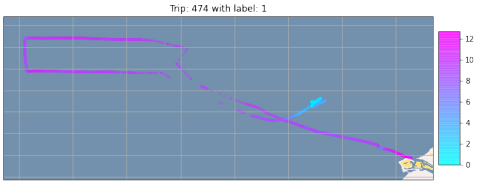

Figure 51: Looping trajectory frequently selected near the start of learning (QBC sampling)

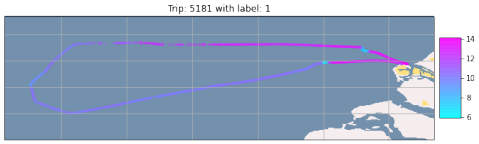

Figure 52: Looping trajectory frequently selected near the start of learning (density-weighted sampling)

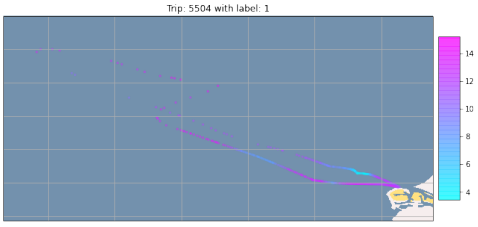

Figure 53: Looping trajectory frequently selected near the start of learning (density-weighted sampling)

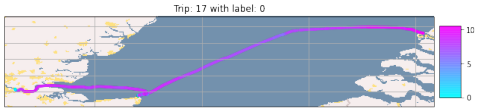

Figure 54: Non-looping trajectory frequently selected near the start of learning (QBC sampling)

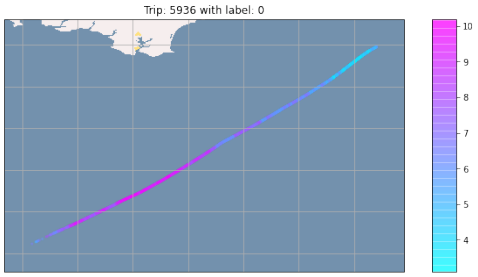

Figure 55: Non-looping trajectory frequently selected near the start of learning (density-weighted sampling)

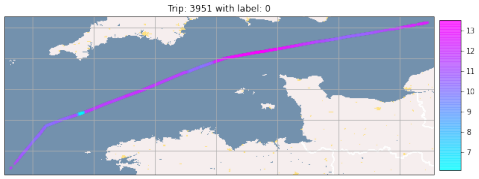

Figure 56: Non-looping trajectory frequently selected near the start of learning (xPAL sampling)

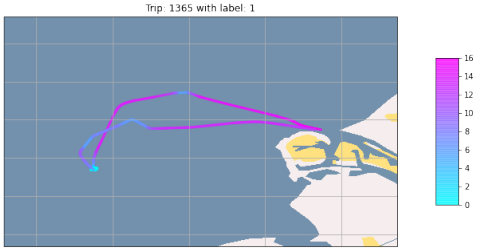

Figure 57: Looping trajectory frequently selected around the middle of learning (QBC sampling)

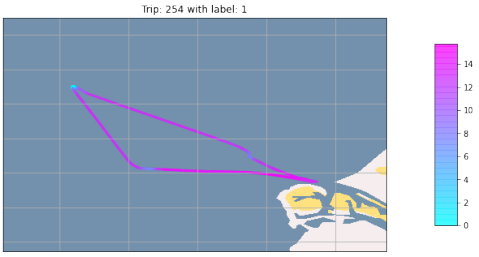

Figure 58: Looping trajectory frequently selected around the middle of learning (QBC sampling)

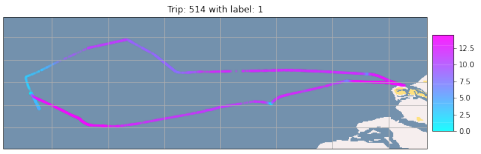

Figure 59: Looping trajectory frequently selected around the middle of learning (QBC sampling)

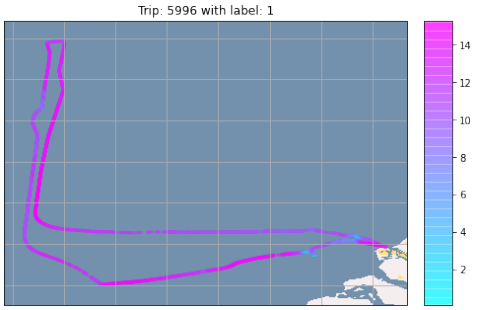

Figure 60: Looping trajectory frequently selected around the middle of learning (QBC sampling)

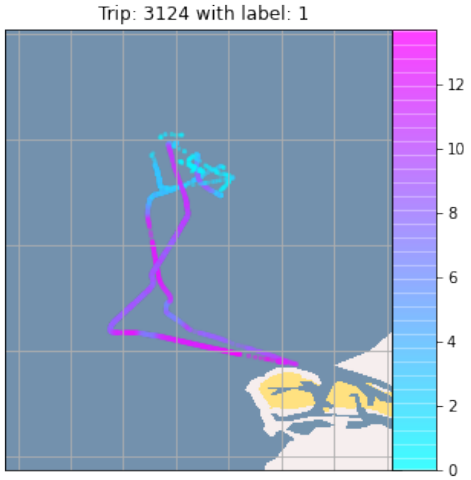

Figure 61: Non-looping trajectory frequently selected near the middle of learning (uncertainty$_s ampling$)

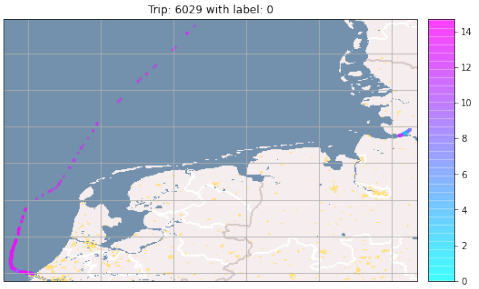

Figure 62: Non-looping trajectory frequently selected near the middle of learning (uncertainty sampling)

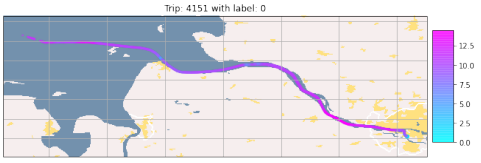

Figure 63: Non-looping trajectory frequently selected near the middle of learning (density-weighted sampling)

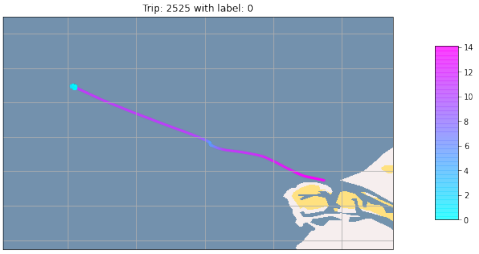

Figure 64: Non-looping trajectory frequently selected near the middle of learning (xPAL sampling)

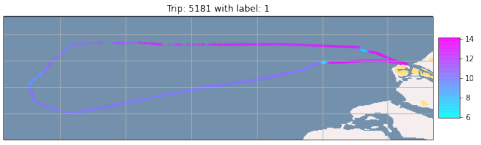

Figure 65: Looping trajectory frequently selected near the end of learning (QBC sampling)

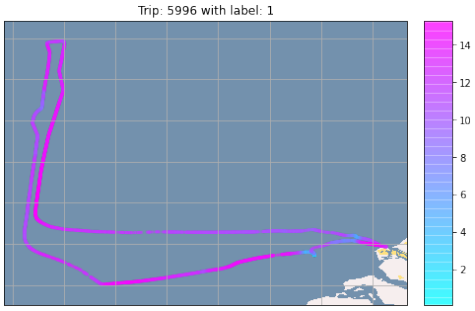

Figure 66: Looping trajectory frequently selected near the end of learning (QBC sampling)

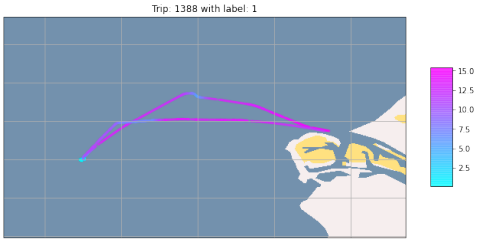

Figure 67: Looping trajectory frequently selected near the end of learning (xPAL sampling)

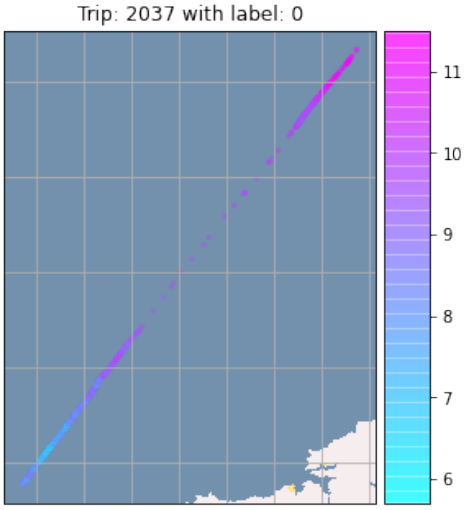

Figure 68: Non-looping trajectory frequently selected near the end of learning (xPAL sampling)