Industry Paper: Classification of vessel activity in streaming data

Ioannis Kontopoulos
kontopoulos@hua.gr
Department of Informatics and Telematics, Harokopio University
Athens, Greece
MarineTraffic
Athens, Greece

Konstantinos Chatzikokolakis
MarineTraffic
Athens, Greece
konstantinos.chatzikokolakis@marinetraffic.com

Konstantinos Tserpes
Department of Informatics and Telematics, Harokopio University
Athens, Greece
tserpes@hua.gr

Dimitris Zissis
MarineTraffic
Athens, Greece
Department of Product and Systems Design Engineering, University of the Aegean
Syros, Greece
dzissis@marinetraffic.com

ABSTRACT

In this paper we motivate the need for real-time vessel behaviour classification and describe in detail our event-based classification approach, as implemented in our real-world industry strong maritime event detection service at MarineTraffic.com. A novel approach is presented for the classification of vessel activity from real-time data streams. The proposed solution splits vessel trajectories into multiple overlapping segments and distinguishes the ones in which a vessel is engaged in trawling or longlining operation (e.g. fishing activity) from other segments that a vessel is simply underway from its departure towards its destination. We evaluate the effectiveness of our tool on real-world data, demonstrating that it can practically achieve high accuracy results. We present our results and findings intended for both researchers and practitioners in the field of intelligent ship tracking and surveillance.

CCS CONCEPTS

• **Computing methodologies** → **Machine learning approaches**; **Distributed algorithms**; • **Computer systems organization** → **Real-time system architecture**.

KEYWORDS

distributed processing, machine learning, vessel monitoring, AIS

ACM Reference Format:
Ioannis Kontopoulos, Konstantinos Chatzikokolakis, Konstantinos Tserpes, and Dimitris Zissis. 2020. Industry Paper: Classification of vessel activity in streaming data. In *The 14th ACM International Conference on Distributed and Event-based Systems (DEBS '20), July 13–17, 2020, Virtual Event, QC, Canada.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3401025.3401763

DEBS '20, July 13–17, 2020, Virtual Event, QC, Canada

ACM ISBN 978-1-4503-8028-7/20/07…$15.00
https://doi.org/10.1145/3401025.3401763

1 INTRODUCTION

The upsurge in mobility data volume has attracted researchers' interest in data-driven knowledge discovery and distributed data processing techniques. Discovering patterns in huge surveillance datasets is of paramount importance for delivering accurate insights on vessels' activities at sea. Today, almost all vessels worldwide are required to carry an Automatic Identification System (AIS) transponder. AIS is a global tracking system that allows vessels to be aware of vessel traffic in their vicinity and to be seen by that traffic. Through this tracking system vessels broadcast information about their location (i.e., GPS coordinates) and behaviour (e.g., speed, course, etc.), as well as information about their characteristics such as vessel size, draught and destination. Although AIS was initially designed for safety purposes and intended to assist officers on board and maritime authorities to monitor vessels' mobility, it soon became apparent that accessing vessels' mobility data can provide useful insights for the identification of illegal activities or abnormal behaviour. Such a use case is hosted by MarineTraffic.com[1] where AIS data is used to monitor vessels and extract meaningful information from their transmitted positions through an anomaly detection toolkit [9]. The toolkit consumes AIS data in real-time to search for anomalies such as deviating or abnormal vessel behaviour [52]. This toolkit can be further extended to support the classification of patterns from the trajectory data. By employing trajectory classification techniques, it is possible to classify or match a vessel mobility pattern to a set of predefined labels. Such a label can be the vessels' fishing activity during which they tend to perform characteristic patterns in their trajectories.

Consequently, such an extension of the anomaly detection toolkit can be used for the immediate identification of Illegal, Unreported and Unregulated (IUU) fishing activities in prohibited areas or nature protection areas which has gained much attention the recent years. A study showed that 640,000 tonnes of ghost gear is left in

[1] https://www.marinetraffic.com/anomaly-detection

the world's oceans each year, which entangles and kills around 136,000 turtles, whales, seals, birds, and other sea animals[2]. These animals end up suffering long and painful deaths. The US government has established the Seafood Import Monitoring Program to address the issue[3]. The Food and Agriculture Organization (FAO) of the United Nations has commissioned a study of IUU fishing activities to determine whether the Organisation should provide guidance for the future estimation of IUU fishing activities[4]. Furthermore, specific fishing activities (e.g., trawling) and fishing gear are strongly linked to the indigenous fauna, thus a mechanism that identifies such operations is of utmost importance. For that reason, several techniques have been developed that take advantage of the spatiotemporal features of trajectories.

In spatiotemporal analysis, data mining includes trajectory pattern clustering, frequent pattern discovery, trajectory classification, forecasting and outlier detection. Trajectory classification is a process of creating a model that can match the mobility pattern of an object to a specific label based on certain decision criteria [27]. Though several trajectory classification solutions have been proposed and applied in many mobility applications, less focus has been given in the maritime domain and specifically in classifying a vessel's type [7, 8, 27, 49], characterising shipping operation areas (e.g., fishing areas [31]), or discovering vessel activity such as fishing [32, 43], search and rescue operations [19, 46] based on its trajectory.

However, many of those methods assume a-priori availability of the whole dataset (i.e., batch processing), have limited availability of ground truth data and use features that make them inapplicable to streaming applications. More specifically the training data are specific to limited vessel types (e.g., only fishing) or the features are linked to trajectory information such as departure and arrival port, that can be discovered only through batch analysis of historical records (i.e., after the completion of a voyage).

Although it is possible to perform event detection in streaming data, this has been realised only for simple events such as proximity events and route deviations [9], potential vessel spoofing [22] or intentional AIS switch-off [21]. Performing complex event analysis (such as distinguishing vessels' activities at sea) poses significant challenges when applied in streaming data (e.g., limited execution time, memory consumption, data cleaning, etc.).

This paper presents a novel approach for classifying vessel activity in large volumes of AIS data streams. Each day approximately 46GB of AIS data are generated from up to 200,000 vessels worldwide, the velocity of which can reach up to 16,000 events per second. For this reason, our proposed classification method exploits the benefits of Akka[5] (which is an open source lightweight framework) in order to develop a distributed system that can efficiently handle large volumes of data and predict vessels' activities in real-time based on their data streams. Our system relies on the well-known Lambda architecture [30] in an attempt to balance latency and throughput for the 4Vs of the Big Data (i.e., Volume, Velocity, Veracity and Variety). More specifically, a 'batch-processing' layer is responsible to train the classification model, which is then used to distinguish the vessel activities in the 'stream-processing' layer at which data continuously arrive from vessels at high rates. Our approach is tested against vessels sailing at the seas of Northern Europe in 2018 and is capable of effectively distinguishing between trawling and longlining operations with high accuracy of over 90%.

The rest of the paper is structured as follows. The next section (Section 2) refers to the related work in the field of trajectory classification and anomaly detection. Section 3 describes in detail the proposed methodology for the classification of fishing activities. Specifically, in Section 3.1, a brief analysis of fishing vessel behaviour in such activities is presented. Section 3.2 presents how the fishing behaviour is processed to create the classification model, while Section 3.3 presents the two layers of the approach, namely the 'batch-processing layer' and the 'real-time processing layer'. Finally, Section 4 illustrates the experimental evaluation of our approach and Section 5 concludes the merits of our work.

2 RELATED WORK

Data mining techniques have been widely used to tackle the problems of anomaly detection and trajectory classification. Both problems require a classifier that is trained on several trajectory data indicating certain behaviours and then try to classify any future trajectory data that exhibit similar behaviour to a set of predefined labels. Another approach of detecting anomalies or complex behaviours is the research field of Complex Event Processing (CEP), where a set of predefined rules or patterns created by experts is given to the system, which later tries to identify such patterns in streams of events.

Several systems have been developed for the purposes of CEP [2, 11, 12, 14, 36]. The idea behind these systems is the use of a formal and expressive language which experts use to write patterns. Most systems use a SQL-like query language [11, 12, 14], while others employ logic-based rules [2] to describe complex events. These patterns are then matched against streams of events, usually in real-time, in order to produce a set of higher level events, called Complex Events. The field of CEP has attracted much attention the recent years and as such several well-known, open-source frameworks have implemented a CEP language such as Apache Flink[6]. Apache Flink offers a library, called FlinkCEP, which allows the user to perform CEP over distributed and streaming data.

Complex Event Processing is not absent in the maritime domain [37, 38, 40, 45] where the early detection of abnormal or illegal vessel behaviour is of interest to the authorities. Authors in [37] present a system which compresses streaming AIS messages to meaningful low-level events, called Simple Events. Simple events are then used to build higher, complex events with the use of the Event Calculus, a logical language for event reasoning. Authors later expand their patterns [38, 39] to support more complex events designed for the maritime domain, such as rendezvous of two or more vessels, fishing and loitering. Tsogas et al. [45] developed a CEP engine, called TRITON, able to consume messages from various sources – Maritime radars, Long Range Identification and Tracking (LRIT) systems, AIS and Earth Observation satellites – and provide in near real-time, complex events describing encounters

[2] https://www.worldanimalprotection.org/illegal-fishing-threatens-wildlife
[3] https://www.worldwildlife.org/threats/illegal-fishing
[4] http://www.fao.org/iuu-fishing/tools-and-initiatives/iuu-fishing-estimation-and-studies/en/
[5] https://akka.io/
[6] https://flink.apache.org/

at sea, drifting, entering/exiting areas of interest and deviations of usual routes. Similarly, authors in [6] developed a distributed CEP system able to identify complex events such as AIS hijacking, engine malfunction or ship collisions.

Although CEP systems provide an expressive way to describe events, it is not always straightforward to understand when or how an event occurred, even by experts. For that reason, data analysis and classification techniques need to be employed in order to infer knowledge. The studies on anomaly detection and trajectory classification provide several approaches for the identification of vessel behaviour. In the field of anomaly detection, several studies have been conducted in the maritime domain. In [24], authors created a Gaussian Mixture Model (GMM) in order to identify anomalies in the trajectory data. Later [25], they compared the GMM with a Kernel Density Estimator (KDE), evaluated the proposed methodologies and showed that there is no significant difference in terms of classification performance. Pallotta et al [34] tried to model the vessel behaviour by using the DBSCAN algorithm to create route objects and waypoints. The resulted traffic model was then used to detect vessel movements that deviate from normality. Similarly, other works tried to model the maritime traffic in order to detect anomalies using unsupervised learning [1, 26, 29]. However, our approach is more similar to the trajectory classification problem. In trajectory classification, a set of trajectories with predefined labels is used as a training set in order to classify any future trajectory or set of trajectories to a specific label from the training set. Similarly, a common methodology in anomaly detection, is the use of a set of outliers or anomalous trajectories as a baseline for classification algorithms. The goal is to classify a future instance or trajectory as anomalous or not based on the training data.

In the field of trajectory classification, many works have focused on analysing the behaviour of the moving objects of interest. Several studies have used trajectories from Vessel Monitoring System (VMS) data to classify fishing activity [4, 17, 47, 50]. Huang et al. [17] tackle the problem of fishing vessel type identification based on only VMS trajectories. To do so, they extract trajectory features that are used in machine learning schemes of XGBoost in order to classify fishing vessels into nine types, achieving a classification accuracy of 95.42%. However, since the usage of AIS became compulsory for vessels and its positional transmission rate is much greater compared to the VMS, research studies have shifted towards the analysis of AIS data. In [31], the authors identified the moves and stops of fishing vessels in a specific area. To do so, they used a combination of algorithms, namely CB-SMoT [35] and DB-SMoT [41], which are able to take into account the speed variation of the trajectory and the direction of the trajectory respectively. Then, they used the DBSCAN algorithm to extract clusters indicating dense areas of fishing activity. Souza et al. [43] analysed the behaviour of fishing vessels using three different types of gear, namely trawlers, longliners and purse-seiners. To distinguish fishing activity between gear types, they created different classification models per activity, in order to identify for each fishing activity the segments of the trajectory during which the vessels were engaged in fishing. The main drawback of this methodology is that it does not use a universal classifier for all fishing activities and the gear type is not always given by the AIS messages. Following the footsteps of [43], authors in [18] presented early promising results of classification performance with the use of neural networks and autoencoders. They evaluated their approach in 10 longline fishing vessels and compared their methodology with other classifiers such as Random Forests and SVMs. Similarly, authors in [10] use the DBSCAN algorithm to extract Points of Interest (POI) in the trajectories and create features from these points. Later, they use these features to train a classifier and achieve high-accuracy results. Finally, in [42], General Hidden Markov Models (GHMM) and Structural Hidden Markov Models (SHMM) are combined with a Genetic Algorithm (GA) in an attempt to classify trajectories. Their approach has been tested in two surveillance datasets, MIT car [51] and T15 [16], yielding promising results.

Despite the fact that there are many methodologies to detect fishing behaviour, less works have focused on real-time stream processing of events in the maritime domain [3, 5, 28, 33]. These works focus on solving the ACM DEBS 2018 Grand Challenge. Authors in [5] treat the problem of destination port prediction as a classification problem. Similarly, Lin et al. [28] use a deep neural network which is fed with features extracted from AIS messages, in order to predict the Estimated Time of Arrival (ETA) of vessels. For the same problem, authors in [33] propose a method of spatial grid for the representation of trajectories as a sequence of historical locations. Consequently, a sequence-to-sequence model is trained to predict future locations. Bachar et al. [3], present Venilia, a methodology for online continuous training using Markov predictive models.

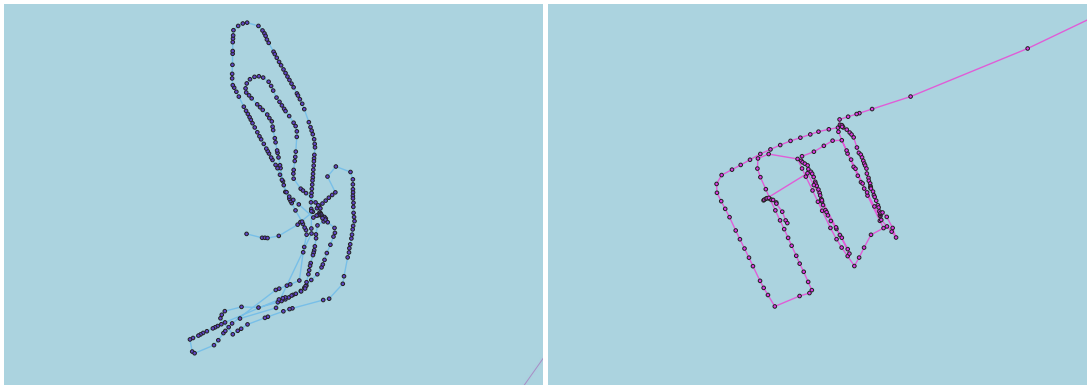
The main difference of our approach is three fold. First, we use a pre-trained classifier in order to detect fishing activities in real-time, contrary to the aforementioned methods of trajectory classification. This is achieved by the use of a two-layer approach in our architecture, the 'batch-processing layer' and the 'stream-processing layer' where the first one is responsible for training a classifier and the second one is responsible for creating features online, making our proposed architecture able to scale to a global dataset. Moreover, none of the work in the literature creates a connection between the length of the trajectory in terms of temporal windows and the classification performance. Patterns start to form properly only after several hours have passed, thus making the distinction between activities more clear. Finally, all of the aforementioned approaches use binary classification in order to detect fishing trajectory segments from non-fishing trajectory segments and do not focus on the multi-class classification of the activities, e.g., distinguish trawling from longlining activity.

3 PROPOSED METHODOLOGY

In this section we describe the investigated fishing methods (i.e., trawling and longlining) and the vessels' behaviour and mobility patterns when engaged in fishing activities. Furthermore, we thoroughly detail the features selected for the proposed classification model and introduce a lambda-architecture scheme in which the classification model can be applied for online fishing activity detection.

3.1 Fishing patterns

Two different fishing methods have been studied in our work, namely trawling and longlining. Specialised gear equipment is used in each fishing method, leading to different mobility behaviour



(a) The movement pattern of a trawling trajectory. (b) The movement pattern of a longlining trajectory.

Figure 1: The movement patterns of fishing activities.

while fishing. Understanding the nature of each operation will give insights on the characteristics distinguishing one activity from the other.

Trawling: Fishing vessels engaged in trawling activity use a fishing net located in the stern of the boat, called trawl, which is dragged through the water. The net is typically pulled by one or more fishing vessels, either on the sea floor (i.e., bottom trawling) or mid-water (i.e., mid-water or pelagic trawling), although single-boat trawling is usually the case. In single-boat trawling the spread of the net depends on the trawl doors, also called "otter boards" and act as wings. To keep the wings steady the vessel must be travelling at a constant speed and for that reason, during trawling, vessels usually sail with lower steady speeds.

Longlining: During this type of activity vessels set multiple fishing lines with baited hooks attached to them, called snoods. The length of the fishing lines can reach up to a kilometer, while the total length of all the fishing lines in the entire activity can reach up to several kilometers [44]. The lines can be deployed either near the surface (i.e., pelagic longline) or along the sea floor (i.e., demersal longline). While setting the lines, vessels travel at their steaming speed or slightly less and they maintain a constant speed. When all lines are set, they are left in the water for several hours and the vessel drifts slowly with them. To retract the lines, vessels follow the same procedure as when setting the lines. The process of setting the lines, waiting and retracting the lines can be repeated several times before returning to a port.

In both fishing methods vessels maneuver and make frequent turns to remain in the same fishing area of interest. However, longlining is characterised by long straight-line trajectories followed by sudden turns. This leads to less maneuvers compared to trawling when the vessels are observed over the same amount of time. At this point, it should be noted that both fishing activities may take up to several hours or even days.

To understand better the behaviour of the fishing activities, their movement patterns have been unveilled in Figures 1, 2 and 3. Figure 1 depicts two movement pattern paradigms of the studied fishing activities. In Figure 1a the trajectory of a trawling vessel is illustrated and shows that the trajectory is characterised by frequent and irregular turns. Similarly, in Figure 1b, which illustrates the trajectory of a longlining vessel, it is apparent that the trajectory, despite the irregular and frequent turns, included also long and straight lines.

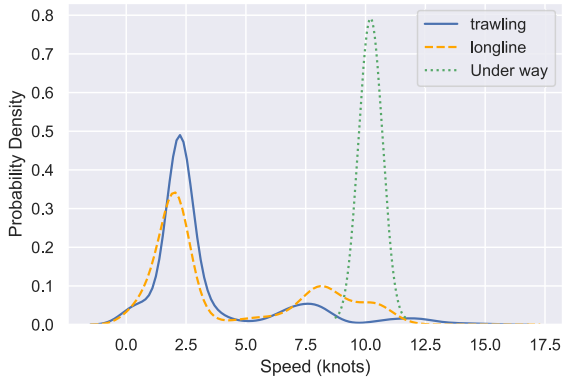


Figure 2: Speed distribution of fishing vessels.

Figure 2 illustrates the activities' speed kernel density distribution. The distribution was calculated from the AIS messages transmitted by the fishing vessels during January and February of 2018. The blue lines refer to the trawling activity, the orange dashed lines refer to the longlining activity and the green dotted lines refer to non-fishing activity. From Figure 2 we can observe that both trawlers and longliners follow similar speed patterns. Both types of fishing activity have two peaks in their speed distribution, the first one being in the range of 1 to 3 knots and the second one being in the range of 7 to 10 knots. A closer inspection though, reveals that the probability density of trawlers' speed being in the first peak is higher than the probability density of longliners' speed in the same peak. A similar pattern can be observed for the second peak as well but with the longliners' speed now having higher probability density. Furthermore, in longlining activity, the first peak is slightly

shifted to the left and the second peak to the right compared to the trawling activity. This can be explained because longliners have higher speeds when setting the lines and the duration of the process takes up a large part of the longlining activity. Moreover, they have lower speeds (range of 1 to 3 knots), compared to the trawlers, after setting the lines and before retracting them, which indicates that they remain stationary drifting along with the lines. Finally, Figure 2 shows that when vessels travel from the ports to the fishing areas and vice versa, they have higher speeds (range of 9 to 12 knots).

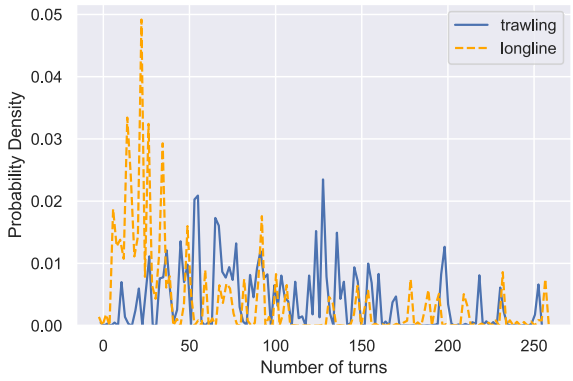


Figure 3: Distribution of number of turns.

Figure 3 illustrates the kernel density distribution of the number of turns per fishing activity. The blue line represents the trawling activity and the orange dashed line represents the longline activity. It can be observed that there is not a clear distinction between the two fishing activities. However, a closer look reveals that probability density of trawling activity follows a bimodal distribution with two modes at 50 turns and 125 turns while the probability density of longliners is a right skewed normal distribution. This phenomenon indicates that, during the longline activity, vessels tend to turn less compared to the trawling vessels. This is due to the fact that longliners, as already explained in the previous paragraphs, have long straight-line trajectories.

3.2 Classification of Trajectory Patterns

To create a proper classification model, several features that are capable to capture the observed behaviour described in Section 3.1 have been selected. Authors in [17] presented a set of features based on speed for the classification of various fishing vessel types with the use of XGBoost classifiers on VMS data, yielding results of high accuracy. Among these features, the average speed and its standard deviation are of high importance. Following their footsteps, we selected some of their features and we extended them to be able to capture all of the trajectory characteristics described in the previous Section. The features selected do not require batch analysis of data and can be computed online over streaming data coming in the Akka system in real-time. The selected features fall into three dimensions; vessels' speed, vessels' drifting and turn frequency.

Features based on speed: Fishing vessels when engaged in fishing activity tend to maintain a constant speed without any significant deviations. Therefore, the *Average Speed* and the *Standard Deviation* of speed play an important role in the identification of the activity.

- *Average Speed:* The average speed during a vessel's trajectory indicates the value of speed most vessels have during the fishing activity, which, based on Figure 1a, revolves around 2.5 knots, especially in the trawling activity.
- *Standard Deviation:* The standard deviation is able to reveal whether the speed is constant or not during a vessel's trajectory. A standard deviation close to 0 indicates the steadiness of the speed.

Features based on drifting: The drifting can be inferred from the difference between the course over ground (Cog) and the heading of the vessel. The course over ground represents the actual direction the vessel has along its path, while the heading represents the direction of the vessel's bow. Figure 4 visualises an example of drifting. The two of them might differ due to the effects of wind, tide or currents of the sea. Two features that may indicate such behaviour are the *Average Drifting* and the *Standard Deviation of Drifting*.

- *Average Drifting:* Excessive drifting is indicative of the behaviour longline vessels have after they set the lines.
- *Standard Deviation of Drifting:* The standard deviation of drifting indicates changes in drifting behavior of a vessel's trajectory. There are segments in longliners' trajectory where the vessels turn their engines off and drift. This behaviour can be identified by the large values of the standard deviation of drifting.

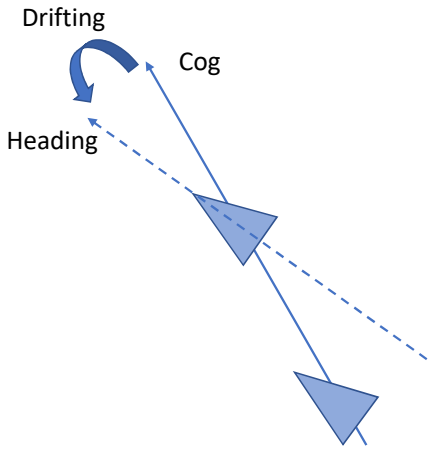


Figure 4: Example of drifting.

Features based on turns: Frequent turns are apparent on fishing vessels regardless of the fishing method (i.e., trawling or longlining). Although vessels during both activities seem to have a similar behaviour, three features that are able to reveal hidden characteristics of the patterns have been selected, namely *Standard Deviation of Cog*, *Number of Turns* and *Accumulated Angle*.

- *Standard Deviation of Cog:* This feature indicates that the vessel does not maintain a steady course, thus it does not move in a straight line.
- *Number of Turns:* As the name suggests, it shows the number of turns the vessel made during a trajectory.
- *Accumulated Angle:* When the vessel starts turning, either right or left, it has a certain Cog c_1. Again, when the vessel stops turning, the Cog has another value c_2. Each turn t is terminated when another turn of the opposite direction starts (e.g., a left turn stops when a right turn starts) or when the vessel maintains a steady course. The difference between these two values is the angle of the turn $a = c_1 - c_2$. The sum of all angles of all turns is the *Accumulated Angle* $A = \sum_{i=1}^{i=t} a_i$. This feature indicates how much the vessel turned during its trajectory.

The next step, after all features have been extracted from representative trawling, longlining and under way trajectories, is to use an algorithm able to effectively classify unseen trajectories based on the features given as a training set. To this end, we chose the Random Forest (RF) classification algorithm due to its high performance results in the maritime domain [5, 8] and because RFs combine the predictions of many Decision Trees into one model, thus they are less prone to overfitting [23]. Furthermore, a Random Forest classifier is less computationally expensive creating a good basis for online predictions or online re-training. Finally, Random Forests require far less data than other state of the art algorithms such as Neural Networks and it is easier to interpret their predictions based on the features [13], which in an industry setting, such an interpretation gives meaning to interested stakeholders.

3.3 Real-time stream processing

Low latencies and high throughput in a streaming system that supports fast decision making are of utmost importance. Specifically, events must be predicted in real-time, meaning the moment a new message is consumed, a prediction must be provided with a sub-second latency. In practice, websites such as MarineTraffic, are flooded by more than 16,000 AIS messages per second received from over 4,000 terrestrial AIS receivers (without satellite receivers in the equation). These volumes of data originate from almost 200,000 vessels globally, totalling in a more than 50GB per day increase rate. Therefore, an architecture needs to be developed that is able to balance latency and throughput. To this end, we used the λ-architecture developed in previous work [9] which achieves sub-second latencies and high throughput. This architecture allows the reduction of the performance cost of the online computations by taking into account pre-computed results. Specifically, the λ-architecture relies on two layers, the "batch-processing layer" and the "stream-processing layer". The former is responsible for creating pre-computed views of data which in our case is the classification model. The latter is responsible for processing streaming events and providing views into the most recent data by taking advantage of the views from the batch-processing layer, hence predictions based on the classification model. To this end, we extended the batch-processing layer of the previously developed architecture by adding a step responsible for constructing the classification model. In this section, both the "batch-processing layer" and the "stream-processing layer" are described.

Batch-processing layer: This layer is responsible for the construction of the Random Forest classification model. For the training of the model, ground truth trajectories are required. Thus, this layer consumes as input already labeled trajectories, namely trawling, longlining and underway. Subsequently, features are extracted per trajectory and the RF model is created and saved to an "xml" file which is then consumed by the stream-processing layer. The Statistical Machine Intelligence and Learning Engine (SMILE)[7] library was used for the implementation. SMILE is a Scala library that supports a wide range of supervised and unsupervised learning algorithms. Internally, it uses a "DataFrame" structure similar to the one used in the Pandas[8] python library. The benefits of this library is its performance in terms of speed and memory consumption and its compatibility with our existing architecture which is implemented in Scala. According to a third party benchmark[9], SMILE outperforms R, Python, Spark, H2O and xgboost significantly.

Stream-processing layer. In order to achieve high-throughput, low-latency performance, the stream-processing layer is implemented in the Akka[10] framework which takes advantage of the Actor model [48]. Actors in the Akka framework exploit the concurrency capabilities of threads, they are lightweight and millions of actor instances can be deployed in a single machine since they have a small memory footprint (i.e., 2.5 million actors per GB of heap). To assist the implementation of the classification of fishing activity, one type of actor from the architecture is used, namely the Vessel Actor.

The Vessel Actor is responsible for consuming AIS messages originating from a single vessel and classifying parts of its trajectory based on the model created from the batch-processing layer. Thus, each vessel has a dedicated actor which is created after the first AIS message has been received. The actor remains idle and acts only when a new message is received. To classify parts of a trajectory of a vessel, each actor takes into account features extracted from an event-based window of user-defined length. To reduce memory consumption, features are extracted online, thus the need of storing a batch of AIS messages is eliminated. To demonstrate the online feature extraction, we present the calculation of the moving average and the standard deviation of speed. The average value of speed is given by Formula 1:

$$\overline{x}_n = \frac{\sum_{i=1}^{i=n} x_i}{n} \tag{1}$$

which can be broken down to the average of $n - 1$ speed values plus a speed value of a newly received AIS message x_n, where n is the total number of messages received in our window:

$$\overline{x}_n = \frac{(\sum_{i=1}^{i=n-1} x_i) + x_n}{n} \tag{2}$$

Since the average value equals to the sum divided by the total count:

[7] https://haifengl.github.io/
[8] https://pandas.pydata.org/
[9] https://github.com/szilard/benchm-ml
[10] https://akka.io/

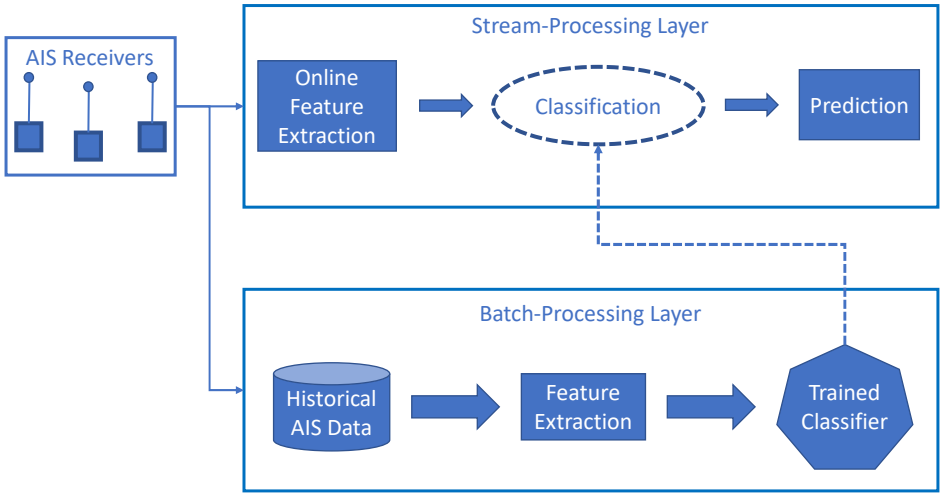


Figure 5: System architecture.

$$\bar{x}_{n-1} = \frac{\sum_{i=1}^{i=n-1} x_i}{n-1} \Rightarrow \sum_{i=1}^{i=n-1} x_i = \bar{x}_{n-1}(n-1) \tag{3}$$

we can substitute Equation 3 to Equation 2 which results in:

$$\bar{x}_n = \frac{(n-1)\bar{x}_{n-1} + x_n}{n} \Rightarrow \bar{x}_n = \bar{x}_{n-1} + \frac{x_n - \bar{x}_{n-1}}{n} \tag{4}$$

where $\bar{x}_{n-1}$ is the average speed value of $n-1$ AIS messages. Upon the arrival of the n-th AIS message, we calculate $\frac{x_n - \bar{x}_{n-1}}{n}$ and add it to the average speed value of $n-1$ AIS messages using Equation 4. When the window closes, all values are reset to zero and the online calculation of the average value starts over. To calculate the standard deviation in a streaming fashion, we employ B. P. Welford's method [20]. Similarly to the calculation of the moving average, we need to reach in a state where we have the standard deviation of $n-1$ values and we need to recalculate for the n^{th} value. To begin with, the formula of variance, which is the square of standard deviation, is given by Equation 5:

$$s^2 = \frac{\sum_{i=1}^{i=n}(x_i - \bar{x}_n)^2}{n-1} \tag{5}$$

Then, we can multiply both sides by $n-1$ and define the first part of the equation as d_n^2:

$$(n-1)s^2 = \sum_{i=1}^{i=n}(x_i - \bar{x}_n)^2 \Rightarrow d_n^2 = \sum_{i=1}^{i=n}(x_i - \bar{x}_n)^2 \tag{6}$$

Later, we can apply the following identity to Equation 6:

$$(a-b)^2 = a^2 - 2ab + b^2 \tag{7}$$

which results in:

$$d_n^2 = \sum_{i=1}^{i=n}(x_i^2 - 2x_i\bar{x}_n + \bar{x}_n^2) \Rightarrow d_n^2 = \sum_{i=1}^{i=n} x_i^2 - 2\bar{x}_n \sum_{i=1}^{i=n} x_i + \bar{x}_n^2 \sum_{i=1}^{i=n} 1 \tag{8}$$

Similarly to the calculation of the moving average, since we know that $\sum_{i=1}^{i=n} 1 = n$ and that the total equals to the mean times the count, we get the following:

$$d_n^2 = \sum_{i=1}^{i=n} x_i^2 - 2n\bar{x}_n^2 + n\bar{x}_n^2 \Rightarrow d_n^2 = \sum_{i=1}^{i=n} x_i^2 - n\bar{x}_n^2 \tag{9}$$

After reaching Equation 9, we can get the value of d^2 for the first $n-1$ values, resulting in:

$$d_{n-1}^2 = \sum_{i=1}^{i=n-1} x_i^2 - (n-1)\bar{x}_{n-1}^2 \tag{10}$$

By subtracting Equation 10 and Equation 9 and after a few rearrangements to the equations we have the resulting equation:

$$d_n^2 = d_{n-1}^2 + (x_n - \bar{x}_n)(x_n - \bar{x}_{n-1}) \tag{11}$$

As with Equation 4 of the moving average, we have a relation which allows us to calculate the new d^2 value by adding an increment, $(x_n - \bar{x}_n)(x_n - \bar{x}_{n-1})$, to its previous value, d_{n-1}^2. We can retrieve the variance by dividing d_n^2 with $n-1$, which subsequently gives us the standard deviation s_n, since s_n is the square root of the variance:

$$s_n^2 = \frac{d_n^2}{n-1} \Rightarrow s_n = \sqrt{\frac{d_n^2}{n-1}} \tag{12}$$

Again, when a window is complete, all values are reset and the calculations start over for the next window. After all features have been extracted, they are fed to the RF model which outputs a decision along with its probability, indicating the activity the vessel performs at the current window. Figure 6 illustrates the online calculation of the average speed. At each incoming AIS message a new average is calculated. The new average speed calculated based on the last AIS message, represents the average speed of the entire window. When the last AIS message of the window is received, the classification is triggered. Finally, Figure 5 illustrates the

system architecture. The batch-processing layer extracts features from the historical AIS data and trains a classifer. The stream-processing layer consumes AIS messages from the receivers and makes a prediction by using the pre-trained classifier generated from the batch-processing layer.

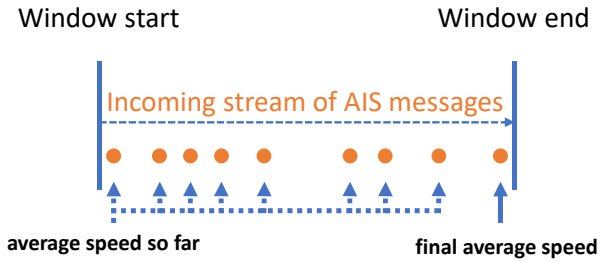


Figure 6: Online calculation of the average speed.

4 EXPERIMENTAL RESULTS

This section describes the dataset used to train and test the Random Forest classifier and its achieved classification performance. The achieved latency performance of the distributed and streaming architecture described in Section 3.3 has already been discussed in previous work [9].

4.1 Dataset Description

The dataset that was used was provided by MarineTraffic[11] and contains AIS messages during a two-month period, from January 1^{st}, 2018 to February 28^{th}, 2018. The dataset covers the seas of Northern Europe, specifically the Irish sea, the North sea and the Baltic sea and contains high quality AIS information without gaps of information. The AIS transmits two kinds of messages, positional and static[12]. Positional messages transmit information about the vessel's location, speed, heading and navigational status. The navigational status, which is manually inserted by the vessel's crew is an identifier that indicates the vessel's activity (e.g., 1 indicates that the vessel was anchored when the message was received). Static messages transmitted include information regarding the vessel's name, its dimensions, the location of its on board positioning system antenna and its destination. The destination denotes the port or area that the vessel is headed but it may also denote the activity the vessel is currently engaged in. Again, this kind of information is set manually by the crew. The AIS messages used for our ground truth dataset contain fishing activities that have been extracted from vessels which have set their navigational status to 7, which indicates "fishing activity", and by vessels which have set their destination to *Trawling* or *Longlining* for the corresponding activity. These fishing trajectories are then segmented to fishing or non-fishing segments. The total number of AIS messages sums up to 61, 050. Table 1 shows the number of AIS messages per activity. Although, the number of messages per activity varies, the number of events per activity remains the same. This is due to the transmission rate of the AIS protocol. When vessels travel at high speeds, the frequency can get as high as one message every two seconds, while the lowest frequency can get as low as one message every 3 minutes when the vessels are not moving. Since the underway vessels travel at much higher speeds, the number of AIS messages is also increased. Another factor affecting the AIS transmision rate is the vessel's turn frequency. Since trawling activity is characterised by frequent turns, the number of AIS messages will be higher compared to the longlining activity but not higher than the messages of the underway activity since the speed during trawling is much lower.

[11] https://www.marinetraffic.com
[12] https://help.marinetraffic.com/hc/en-us/articles/205426887-What-kind-of-information-is-AIS-transmitted-

Table 1: Number of AIS messages per activity.

Activity	# AIS messages
Trawling	16, 110
Longline	8, 484
Underway	36, 456

4.2 Experimental Evaluation

In this section we provide the evaluation results of the proposed classification scheme for the identification of trawling and longlining activities. Firstly we evaluate how various hyperparameters of Random Forests influence the performance and accuracy of the proposed classifier providing well-known metrics (e.g., f1-score, accuracy etc.) for multiple Random Forest configurations. Then, we evaluate the robustness of the classifier when applied in various temporal window lengths. Finally, we compare the classification performance of Random Forests against other well-known classifiers. In our first series of experiments we used the complete trajectories[13] of the dataset from which features are extracted following the methodology described in Section 3.2. Then, these features are fed to a Random Forest classifier, tuning in each experiment a different hyperparameter of the forest. The first hyperparameter selected is the number of trees of the forest. We use three distinct configurations for the number of trees, $numTrees = 10, 50, 100$, and run 5-fold cross validation per setting. The idea behind the Random Forest algorithm is to optimise the prediction of an instance by averaging the predictions of multiple decision trees. Moreover, for each configuration of *numTrees*, we tested three distinct configurations of another hyperparameter of Random Forest, the maximum depth, setting its value to 2, 5 and 10 respectively. This parameter controls the size of the trees, specifically defining the maximum number of levels each tree in the forest can have. Table 2 shows the results of each experimental setting. From the results, we observe that the configuration with the highest classification performance is $numTrees = 100$ and $maxDepth = 10$ that achieves f1-score of 93.52% and accuracy of 95.33%. Therefore, for the rest of the experiments we used the best setting from Table 2.

Next, we measured the importance of each feature for our classifier and visualised the results in Figure 7. It can be seen that all the features are approximately of equal importance with the exception of the average speed and the standard deviation of speed. This means that all of the features contribute equally to the decision

[13] The length of the trajectories can last from few hours to several days.

Table 2: Classification results per RF hyperparameter.

number of trees	maximum depth	Accuracy	Precision	Recall	F1-score
10	2	72 %	64.02 %	68.05 %	65.97 %
50	2	74.66 %	62.07 %	65.01 %	63.5 %
100	2	77 %	64.5 %	66.22 %	65.34 %
10	5	89.66 %	84.76 %	88.24 %	86.47 %
50	5	90.66 %	88.13 %	89.81 %	88.96 %
100	5	91.33 %	85.64 %	88.22 %	86.91 %
10	10	92.66 %	89.02 %	88.75 %	88.88 %
50	10	93.33 %	87.71 %	91.91 %	89.76 %
100	10	95.33 %	93.91 %	93.15 %	93.52 %

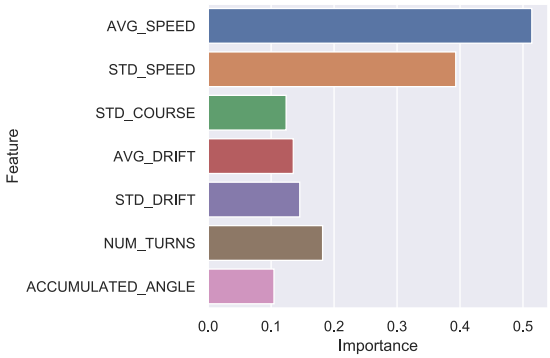


Figure 7: Feature importances.

of the classifier, but the features that contain the speed factor are more decisive.

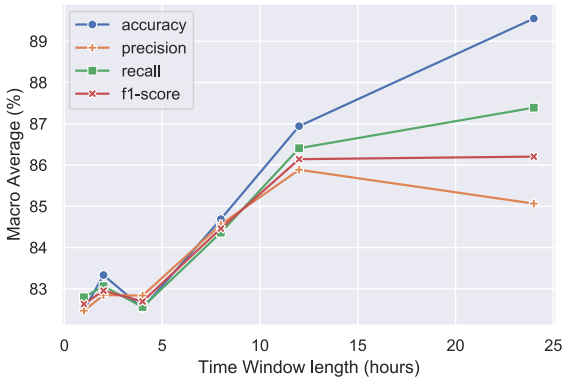


Figure 8: Macro average results per temporal size.

Table 3: Macro average results per temporal size.

Hours	Accuracy	Precision	Recall	F1-score
1	82.48 %	82.47 %	82.79 %	82.63 %
2	83.33 %	82.84 %	83.06 %	82.85 %
4	82.53 %	82.83 %	82.55 %	82.69 %
8	84.68 %	84.55 %	84.36 %	84.46 %
12	86.94 %	85.88 %	86.4 %	86.14 %
24	89.54 %	85.06 %	87.38 %	86.2 %

Afterwards, we evaluated the performance of our classifier when applied in time windows of different length. More specifically, we segmented each trajectory into temporal segments of 1, 2, 4, 8, 12 and 24 hours. This means that a fishing activity, e.g. trawling is now segmented into more parts, according to the temporal size.

For each temporal size we performed 5-fold cross validation, keeping 80% of the trajectories as a training set and the rest 20% as a test set. The macro average results after the cross validation are presented in Table 3 and Figure 8. From the results, we can observe that the accuracy is increased as the time window length is increased. Furthermore, it can be seen that from the one-hour window length to the four-hour, the accuracy does not show any significant change, while from the four-hour window length to the twenty-four-hour, the accuracy increases abruptly, from 82.53% to 89.54% (i.e., a 7.01% increase). This can be explained due to the fact that a fishing pattern may take hours to form. The features of each trajectory start to become distinguishable from one another after several hours have passed. From our experiments, we can see that the time threshold with which a pattern in the fishing trajectory is formed needs to be at least four hours which explains the increase in the accuracy.

Finally, we compared the Random Forests classifier against three other well-known classifiers, namely Gradient Boosted Trees (GBT), Linear Discriminant Analysis (LDA), and Logistic Regression, on the same set of features. To this end, we performed 5-fold cross validation, similar to the previous series of experiments of Table 3 and set the temporal size of the trajectories to a fixed length of 24 hours, the maximum length of the trajectories that yields the best classification performance. For the Gradient Boosted Trees we used the same hyperparameters to the Random Forests, since both

Table 4: Macro average results per classifier.

Classifier	Accuracy	Precision	Recall	F1-score
Random Forests	89.54 %	85.06 %	87.38 %	86.2 %
Gradient Boosted Trees	90.98 %	86.77 %	87.83 %	87.29 %
Linear Discriminant Analysis	81.4 %	69.13 %	77.71 %	73.17 %
Logistic Regression	79.71 %	76.19 %	75.59 %	75.89 %

of these classifiers use multiple decision trees to make a prediction, thus making a more direct comparison between the two. For the rest of the classifiers, we used the default hyperparameters that are provided by the Smile library. Table 4 presents the macro average results for each classifier. According to Table 4, Gradient Boosted Trees slightly outperform the Random Forests, yielding the best classification performance (a F1-score of 87.29 %), while both the LDA and the Logistic Regression perform weakly, achieving a F1-score of 73.17 % and 75.89 %, respectively. Despite the fact that Gradient Boosted Trees present a higher classification performance, they are computationally more expensive and are harder to fine-tune [15], making them the least favorable option for a real-time streaming system. A more in-depth comparison between the classifiers by fine-tuning the classifiers' hyperparameters will be conducted in the future for a more thorough evaluation.

Three studies in the literature are comparable to our own methodology in terms of classification performance [17, 18, 43]. Authors in [17] present a set of features that is used from XGBoost classifiers achieving an accuracy of approximately 97%. Despite their high classification accuracy, several features are best suited for the region of China and are not applicable to other regions, making it unsuitable for global fishing detection. Moreover, their methodology extracts features based on entire trajectories of nine fishing vessel types, thus their goal is to identify the vessel type and not the vessel activity. Souza et al. [43] create three different classifiers for the detection of trawlers, longliners and purse seiners respectively, achieving a median accuracy of 83%, 84% and 97% for each activity correspondingly. The main drawback of their approach is the need of three separate classifiers, each one performing a binary classification task of fishing and non-fishing activity for each vessel type, compared to our methodology where it acts as a universal classifier of fishing activity. Finally, authors in [18], use autoencoders to detect whether a longliner vessel is engaged in fishing activity, achieving an accuracy of 85%. Similarly to [43], they perform a binary classification task instead of the multi-class classification task of our approach.

5 CONCLUSION

In this work we presented a methodology for the classification of fishing activities in a streaming fashion. Specifically, the fishing behaviour was analysed which led to the selection of a specific set of features able to describe and characterise two fishing activities, trawling and longlining. Furthermore, an approach of online feature extraction and classification was presented which eliminates the need of storing streaming window batches in memory. Experimental evaluation showed the analysis of the hyperparameter tuning of our classifier along with the importance of the selected features. Moreover, the evaluation of our approach demonstrated the increased classification performance and the way the temporal size of the trajectory affects the classification accuracy. Finally, a comparison between four classifiers was made on the same set of features, demonstrating that, in our case, decision-tree based classifiers yield a better classification performance.

As a future work, we intend to extend our methodology to cover more fishing activities, by incorporating more features to our classifier. Furthermore, to support more fishing activities, more classification schemes will be investigated and tested, in terms of their classification performance. Finally, we aim at developing a methodology which will retrain the suggested classifier online with the use of newly received data.

ACKNOWLEDGMENTS

This work has received funding from the European Union's Horizon 2020 Research and Innovation programme under grant agreement No 825070.

REFERENCES

[1] Virginia Fernandez Arguedas, Giuliana Pallotta, and Michele Vespe. 2018. Maritime Traffic Networks: From Historical Positioning Data to Unsupervised Maritime Traffic Monitoring. *IEEE Transactions on Intelligent Transportation Systems* 19 (2018), 722–732.

[2] Alexander Artikis, Marek Sergot, and Georgios Paliouras. 2015. An Event Calculus for Event Recognition. *IEEE Transactions on Knowledge and Data Engineering* 27 (04 2015), 895–908.

[3] Moti Bachar, Gal Elimelech, Itai Gat, Gil Sobol, Nicolo Rivetti, and Avigdor Gal. 2018. Venilia, On-Line Learning and Prediction of Vessel Destination. In *Proceedings of the 12th ACM International Conference on Distributed and Event-Based Systems* (Hamilton, New Zealand) *(DEBS '18)*. Association for Computing Machinery, New York, NY, USA, 209–212. https://doi.org/10.1145/3210284.3220505

[4] Nicolas Bez, Emily Walker, Daniel Gaertner, Jacques Rivoirard, and Philippe Gaspar. 2011. Fishing activity of tuna purse seiners estimated from vessel monitoring system (VMS) data. *Canadian Journal of Fisheries and Aquatic Sciences* 68 (10 2011), 1998–2010. https://doi.org/10.1139/f2011-114

[5] Oleh Bodunov, Florian Schmidt, André Martin, Andrey Brito, and Christof Fetzer. 2018. Real-Time Destination and ETA Prediction for Maritime Traffic. In *Proceedings of the 12th ACM International Conference on Distributed and Event-Based Systems* (Hamilton, New Zealand) *(DEBS '18)*. Association for Computing Machinery, New York, NY, USA, 198–201. https://doi.org/10.1145/3210284.3220502

[6] Juan Boubeta-Puig, Inmaculada Medina-Bulo, Guadalupe Ortiz, and Germán Fuentes-Landi. 2012. Complex Event Processing Applied to Early Maritime Threat Detection. In *Proceedings of the 2nd International Workshop on Adaptive Services for the Future Internet and 6th International Workshop on Web APIs and Service Mashups* (Bertinoro, Italy) *(WAS4FI-Mashups '12)*. Association for Computing Machinery, New York, NY, USA, 1–4. https://doi.org/10.1145/2377836.2377838

[7] Konstantinos Chatzikokolakis, Dimitrios Zissis, Spiliopoulos, and Tserpes. 2019. A comparison of supervised learning schemes for the detection of search and rescue (SAR) vessel patterns. *GeoInformatica* (may 2019). https://doi.org/10.1007/s10707-019-00365-y

[8] Konstantinos Chatzikokolakis, Dimitrios Zissis, Giannis Spiliopoulos, and Konstantinos Tserpes. 2018. Mining Vessel Trajectory Data for Patterns of Search and Rescue. In *Proceedings of the Workshops of the EDBT/ICDT 2018 Joint Conference (EDBT/ICDT 2018), Vienna, Austria, March 26, 2018 (CEUR Workshop Proceedings)*, Nikolaus Augsten (Ed.), Vol. 2083. CEUR-WS.org, 117–124. http://ceur-ws.org/Vol-2083/paper-18.pdf

[9] Konstantinos Chatzikokolakis, Dimitris Zissis, Marios Vodas, Giannis Spiliopoulos, and Ioannis Kontopoulos. 2019. A distributed lightning fast maritime anomaly detection service. In *OCEANS 2019 - Marseille*. IEEE, 1–8. https://doi.org/10.1109/OCEANSE.2019.8867269

[10] Buncha Chuaysi and Supaporn Kiattisin. 2020. Fishing Vessels Behavior Identification for Combating IUU Fishing: Enable Traceability at Sea. In *Wireless Personal Communications*. Springer. https://doi.org/10.1007/978-3-319-34111-8_4

[11] Gianpaolo Cugola and Alessandro Margara. 2010. TESLA: a formally defined event specification language. In *Proceedings of the Fourth ACM International Conference on Distributed Event-Based Systems, DEBS 2010, Cambridge, United Kingdom, July 12-15, 2010*, Jean Bacon, Peter R. Pietzuch, Joe Sventek, and Ugur Çetintemel (Eds.). ACM, 50–61. https://doi.org/10.1145/1827418.1827427

[12] Gianpaolo Cugola and Alessandro Margara. 2012. Processing Flows of Information: From Data Stream to Complex Event Processing. *Comput. Surveys* 44 (06 2012). https://doi.org/10.1145/2187671.2187677

[13] Richard Duda, Peter Hart, and David G.Stork. 2001. *Pattern Classification*. Wiley.

[14] Daniel Gyllstrom, Eugene Wu, Hee-Jin Chae, Yanlei Diao, Patrick Stahlberg, and Gordon Anderson. 2006. SASE: Complex Event Processing over Streams. *CoRR* abs/cs/0612128 (12 2006).

[15] Trevor Hastie, Robert Tibshirani, Jerome Friedman, and James Franklin. 2004. The Elements of Statistical Learning: Data Mining, Inference, and Prediction. *The Mathematical Intelligencer* 27 (11 2004), 83–85. https://doi.org/10.1007/BF02985802

[16] Weiming Hu, Xi Li, Guodong Tian, Stephen J. Maybank, and Zhongfei Zhang. 2013. An Incremental DPMM-Based Method for Trajectory Clustering, Modeling, and Retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35 (2013), 1051–1065.

[17] Haiguang Huang, Feng Hong, Jing Liu, Chao Liu, Yuan Feng, and Zhongwen Guo. 2019. FVID: Fishing Vessel Type Identification Based on VMS Trajectories. *Journal of Ocean University of China* 18 (04 2019), 403–412. https://doi.org/10.1007/s11802-019-3717-9

[18] Xiang Jiang, Daniel Silver, Baifan Hu, and Erico Souza. 2016. Fishing Activity Detection from AIS Data Using Autoencoders. In *Proceedings of the 29th Canadian Conference on Artificial Intelligence on Advances in Artificial Intelligence*. Springer, 33–39. https://doi.org/10.1007/978-3-319-34111-8_4

[19] Konstantinos Kapantais, Iraklis Varlamis, Christos Sardianos, and Konstantinos Tserpes. 2019. A Framework for the Detection of Search and Rescue Patterns Using Shapelet Classification. *Future Internet* 11 (09 2019), 192. https://doi.org/10.3390/fi11090192

[20] Donald E. Knuth. 1997. *The Art of Computer Programming, Volume 2 (3rd Ed.): Seminumerical Algorithms*. Addison-Wesley Longman Publishing Co., Inc., USA.

[21] Ioannis Kontopoulos, Konstantinos Chatzikokolakis, Dimitrios Zissis, Konstantinos Tserpes, and Giannis Spiliopoulos. 2020. Real-time Maritime Anomaly Detection: Detecting intentional AIS switch-off. *IJBDI - International Journal of Big Data intelligence* 7, 2 (2020), 85–96.

[22] Ioannis Kontopoulos, Giannis Spiliopoulos, Dimitris Zissis, Konstantinos Chatzikokolakis, and Alexander Artikis. 2018. Countering Real-Time Stream Poisoning: An Architecture for Detecting Vessel Spoofing in Streams of AIS Data. In *4th Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress(DASC/PiCom/DataCom/CyberSciTech)*. IEEE, 981–986. https://doi.org/10.1109/DASC/PiCom/DataCom/CyberSciTec.2018.00139

[23] Bart Lariviere and Dirk Van den Poel. 2005. Predicting Customer Retention and Profitability by Using Random Forests and Regression Forests Techniques. *Expert Systems with Applications* 29 (08 2005), 472–484. https://doi.org/10.1016/j.eswa.2005.04.043

[24] Rikard Laxhammar. 2008. Anomaly detection for sea surveillance. In *11th International Conference on Information Fusion*. IEEE, 1 – 8.

[25] Rikard Laxhammar, Göran Falkman, and E. Sviestins. 2009. Anomaly detection in sea traffic - A comparison of the Gaussian Mixture Model and the Kernel Density Estimator. In *2009 12th International Conference on Information Fusion, FUSION 2009*. IEEE, 756 – 763.

[26] Nicolas Le Guillarme and Xavier Lerouvreur. 2013. Unsupervised extraction of knowledge from S-AIS data for maritime situational awareness. In *Proceedings of the 16th International Conference on Information Fusion, FUSION 2013*. IEEE, 2025–2032.

[27] Jae-Gil Lee, Jiawei Han, Xiaolei Li, and Hector Gonzalez. 2008. Traclass: Trajectory Classification Using Hierarchical Region-Based and Trajectory-Based Clustering. *Proceedings of The Vldb Endowment - PVLDB* 1 (08 2008). https://doi.org/10.14778/1453856.1453972

[28] Chun-Xun Lin, Tsung-Wei Huang, Guannan Guo, and Martin D. F. Wong. 2018. MtDetector: A High-Performance Marine Traffic Detector at Stream Scale. In *Proceedings of the 12th ACM International Conference on Distributed and Event-Based Systems* (Hamilton, New Zealand) *(DEBS '18)*. Association for Computing Machinery, New York, NY, USA, 205–208. https://doi.org/10.1145/3210284.3220504

[29] Bo Liu, Erico Souza, and Marcin Sydow. 2014. Knowledge-based clustering of ship trajectories using density-based approach. In *Proceedings - 2014 IEEE International Conference on Big Data, IEEE Big Data 2014*. IEEE. https://doi.org/10.1109/BigData.2014.7004281

[30] Nathan Marz and James Warren. 2015. Big Data: Principles and best practices of scalable realtime data systems. *Manning Publications* (2015).

[31] Fabio Mazzarella, Michele Vespe, Dimitrios Damalas, and Giacomo Osio. 2014. Discovering vessel activities at sea using AIS data: Mapping of fishing footprints. In *FUSION 2014 - 17th International Conference on Information Fusion*. IEEE.

[32] Fabrizio Natale, Maurizio Gibin, Alfredo Alessandrini, Michele Vespe, and Anton Paulrud. 2015. Mapping Fishing Effort through AIS Data. *PLoS ONE* 10 (06 2015), e0130746. https://doi.org/10.1371/journal.pone.0130746

[33] Duc-Duy Nguyen, Chan Le Van, and Muhammad Intizar Ali. 2018. Vessel Destination and Arrival Time Prediction with Sequence-to-Sequence Models over Spatial Grid. In *Proceedings of the 12th ACM International Conference on Distributed and Event-Based Systems* (Hamilton, New Zealand) *(DEBS '18)*. Association for Computing Machinery, New York, NY, USA, 217–220. https://doi.org/10.1145/3210284.3220507

[34] Giuliana Pallotta, Michele Vespe, and Karna Bryan. 2013. Vessel Pattern Knowledge Discovery from AIS Data: A Framework for Anomaly Detection and Route Prediction. *Entropy* 15 (06 2013), 2218–2245. https://doi.org/10.3390/e15062218

[35] Andrey Tietbohl Palma, Vania Bogorny, Bart Kuijpers, and Luis Otavio Alvares. 2008. A Clustering-Based Approach for Discovering Interesting Places in Trajectories. In *Proceedings of the 2008 ACM Symposium on Applied Computing* (Fortaleza, Ceara, Brazil) *(SAC '08)*. Association for Computing Machinery, New York, NY, USA, 863–868. https://doi.org/10.1145/1363686.1363886

[36] Adrian Paschke and Alexander Kozlenkov. 2009. Rule-Based Event Processing and Reaction Rules. In *Rule Interchange and Applications*, Guido Governatori, John Hall, and Adrian Paschke (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 53–66.

[37] Kostas Patroumpas, Elias Alevizos, Alexander Artikis, Marios Vodas, Nikos Pelekis, and Yannis Theodoridis. 2017. Online event recognition from moving vessel trajectories. *GeoInformatica* 21, 2 (2017), 389–427. https://doi.org/10.1007/s10707-016-0266-x

[38] Manolis Pitsikalis, Alexander Artikis, Richard Dreo, Cyril Ray, Elena Camossi, and Anne-Laure Jousselme. 2019. Composite Event Recognition for Maritime Monitoring. In *Proceedings of the 13th ACM International Conference on Distributed and Event-Based Systems* (Darmstadt, Germany) *(DEBS '19)*. Association for Computing Machinery, New York, NY, USA, 163–174. https://doi.org/10.1145/3328905.3329762

[39] Manolis Pitsikalis, Ioannis Kontopoulos, Alexander Artikis, Elias Alevizos, Paul Delaunay, Jules-Edouard Pouessel, Richard Dreo, Cyril Ray, Elena Camossi, Anne-Laure Jousselme, and et al. 2018. Composite Event Patterns for Maritime Monitoring. In *Proceedings of the 10th Hellenic Conference on Artificial Intelligence* (Patras, Greece) *(SETN '18)*. Association for Computing Machinery, New York, NY, USA, Article 29, 4 pages. https://doi.org/10.1145/3200947.3201042

[40] Maria Riveiro, Göran Falkman, Tom Ziemke, and Håkan Warston. 2009. VISAD: An interactive and visual analytical tool for the detection of behavioral anomalies in maritime traffic data. *Proc SPIE* 7346 (05 2009). https://doi.org/10.1117/12.817819

[41] Jose Antonio M. R. Rocha, Valeria C. Times, Gabriel Oliveira, Luis O. Alvares, and Vania Bogorny. 2010. DB-SMoT: A direction-based spatio-temporal clustering method. In *2010 5th IEEE International Conference Intelligent Systems* (London, United Kingdom). IEEE, 114–119.

[42] Rajkumar Saini, Partha Roy, and Debi Dogra. 2017. A Segmental HMM based Trajectory Classification using Genetic Algorithm. *Expert Systems with Applications* 93 (10 2017). https://doi.org/10.1016/j.eswa.2017.10.021

[43] Erico Souza, Kristina Boerder, and Boris Worm. 2016. Improving Fishing Pattern Detection from Satellite AIS Using Data Mining and Machine Learning. *PLOS ONE* 11 (07 2016), e0158248. https://doi.org/10.1371/journal.pone.0158248

[44] Paul Tixier, Jade Vacquié-Garcia, Nicolas Gasco, Guy Duhamel, and Christophe Guinet. 2015. Mitigating killer whale depredation on demersal longline fisheries by changing fishing practices. *ICES Journal of Marine Science* 72 (04 2015), 1610–1620. https://doi.org/10.1093/icesjms/fsu137

[45] Manolis Tsogas, Polyzois Parthymos, Marios Moutzouris, Nektarios Patlakas, George Karagiannis, Antonis Kostaridis, and Dimitris Diagourtas. 2019. A Geospatial Complex Event Processing Engine for Abnormal Vessel Behavior Detection Suitable for Maritime Surveillance. In *1st Maritime Situational Awareness Workshop MSAW2019*.

[46] Iraklis Varlamis, Konstantinos Tserpes, and Christos Sardianos. 2018. Detecting Search and Rescue Missions from AIS Data. In *2018 IEEE 34th International Conference on Data Engineering Workshops (ICDEW)*. IEEE, 60–65.

[47] Youen Vermard, Etienne Rivot, Stéphanie Mahévas, Paul Marchal, and Didier Gascuel. 2010. *Identifying fishing trip behaviour and estimating fishing effort from VMS data using Bayesian Hidden Markov Models*.

[48] Vaughn Vernon. 2015. *Reactive Messaging Patterns with the Actor Model: Applications and Integration in Scala and Akka* (1st ed.). Addison-Wesley Professional.

[49] Gerben Klaas Dirk Vries and Maarten Someren. 2012. Machine learning for vessel trajectories using compression, alignments and domain knowledge. *Expert Systems with Applications* 39 (12 2012). https://doi.org/10.1016/j.eswa.2012.05.060

[50] Emily Walker and Nicolas Bez. 2010. A pioneer validation of a state-space model of vessel trajectories (VMS) with observers' data. *Ecological Modelling* 221 (08

2010), 2008–2017. https://doi.org/10.1016/j.ecolmodel.2010.05.007

[51] Xiaogang Wang, Keng Teck Ma, Gee Wah Ng, and W. Eric L. Grimson. 2008. Trajectory analysis and semantic region modeling using a nonparametric Bayesian model. In *2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2008), 24-26 June 2008, Anchorage, Alaska, USA*. IEEE Computer Society. https://doi.org/10.1109/CVPR.2008.4587718

[52] D. Zissis, K. Chatzikokolakis, G. Spiliopoulos, and M. Vodas. 2020. A Distributed Spatial Method for Modeling Maritime Routes. *IEEE Access* 8 (2020), 47556–47568.