Identifying Risk-Prone Behavior of Seafarers by Using Explainable AI

Nicolas Fouqué 2002154
Master's degree Programme in Information Technology, double degree INSA Rennes
Supervisors: Sepinoud Azimi Rashti, Sébastien Lafond
Faculty of Science and Engineering
Åbo Akademi University
2020-2021

Abstract

Despite the advancements in technologies for maritime navigation, maritime accidents are still a big problem in the industry. Extensive research has been done to study these accidents and try to find solutions to avoid them, but no research has tried to apply Explainable AI approaches to navigational data.

The goal of this thesis is to produce a predictive deep learning model to study navigational data and use attention mechanisms to identify seafarer behaviors which could lead to accidents during the ship's navigational operations.

Keywords: Machine Learning, Deep Learning, LSTM, Explainable AI, Attention Layers, Maritime Navigation, Maritime Accidents.

Abbreviations: AI: Artificial Intelligence, AIS: Automatic Identification Systems, IMO: International Maritime Organization, MSC: Maritime Safety Committee, MMSI: Maritime Mobile Service Identity, SOG: Speed Over Ground, COG: Course Over Ground, ANN: Artificial Neural Networks, RNN: Recurrent Neural Network, LSTM: Long Short-Term Memory, XAI: Explainable AI .

Acknowledgements

I would like to thank my supervisors, Sepinoud Azimi-Rashti and Sébastien Lafond, for providing me with the opportunity to study this subject and for accompanying me all along. In addition, I am grateful to Johanna Salokannel from Aboa Mare and Florent Nicolas from HELCOM, as they were a great help when it came to domain-specific knowledge about maritime navigation. I would also like to thank all the other master's students I worked with, in particular Otto Lindfors and Jimmy Fagerholm; our exchanges have been very insightful and have helped me many times, and you often allowed me to catch up on some concepts that would have taken me much longer to grasp. Finally, I am very grateful to Daniel Neil, who personally took time to help me understand the original implementation of the neural network to tailor it to my needs.

Contents

Introduction

In the past years, maritime navigation is a sector that has kept growing, with a world fleet today that is almost three times what it was at the beginning of the century (Fig.1.1).

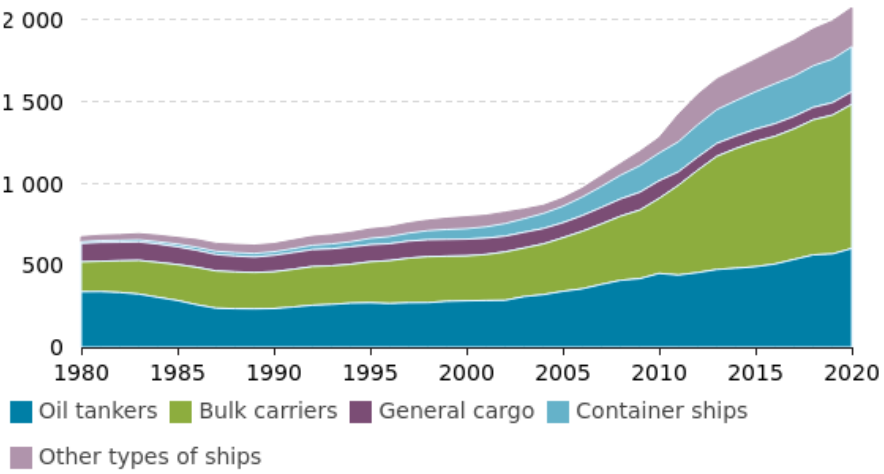

Figure 1.1: Size of the World Fleet for merchant ships above 100 Gross Tonnage per primary vessel type (in Millions of dead-weight tons). Source : [1]

With the number of ships sailing globally increasing, accidents are increasing too, and while the numbers are not growing at the same speed, the growing size of the ships means that the scale of the accidents is becoming larger too.

In the last decade, navigation in the Baltic Sea alone represented 15% of cargo transport worldwide[2], and has resulted in about 150 shipping accidents every year[3], causing financial losses, but more importantly costing human lives and causing dramatic environmental issues through pollution. Maritime safety is a very important topic for all neighboring countries and extensive research is led to try to reduce the number of accidents.[4]

1.1 Motivation

The goal of this thesis is to suggest a new approach for reviewing maritime accidents, using LSTM Neural Networks to analyze navigational data where ships involved in an accident have been isolated, to build a deep learning model capable of predicting seafaring behaviors that might lead to accidents (referred to as risk-prone behaviors) and using attention mechanisms to extract the features of said behaviors and translate them into general understanding, possibly allowing us to pass that information to sailors as behaviors to avoid when at sea.

1.2 Thesis outline

This introductory chapter presents general information about the thesis, its background as well as the purpose of the study presented. The next chapter will be dedicated to describing the concepts used to carry out this work. Chapter 3 will give details about the data sources used for this study and the resulting dataset to train the model, and then describe the implementation and architecture of the network. The results of the attention mechanisms will be studied in Chapter 4 and their validity will be discussed in Chapter 5 which will serve as conclusion and summarize this study's findings.

Theory

The goal of this chapter is to present the concepts of maritime navigation monitoring used in this study, and the relevant actors in this field, as well as present the techniques used to create our system.

2.1 Maritime navigation data

This section will be presenting the kind of navigational data used in this study and the organism providing this data to us.

2.1.1 AIS Data

Automatic Identification Systems (AIS) are a standard of maritime navigation identification. The International Maritime Organisation requires that every ship carries equipment capable of sending standard information to the surrounding ships and to coastal authorities, as well as receive this information from other ships themselves. This regulation is in effect for cargo ships (300 or more gross tonnage for international transport and 500 or more gross tonnage for non-international transport) as well as all passenger ships (regardless of their size). The European Union also requires all ships longer than 15 meters to comply with these regulations.

These regulations only enforce the use of vessel-based AIS transceivers, which should be sending information continuously, and at least comply with the specifications of "Class A transceivers"; Very High Frequency (VHF) transceivers composed of a VHF transmitter, two VHF Time-Division Multiple Access (TDMA) receivers, one VHF Digital Selective Calling (DSC) that use the sensors of the ship to collect the information (excepted for the time synchronization which is assured via an internal time base, synchronized to a global navigation satellite system (e.g. GPS) receiver).[5]

The data sent by Class A transceivers is separated into two parts. One is sent every 2 to 10 seconds, and it contains the fields (detailed in table 2.1). This one does not have a full timestamp; for completeness, it must be combined with the second

Vessel Maritime Mobile Service Identity (MMSI)	A unique identification number that is different from the IMO number
Navigation status	E.g. "at anchor", "under way using engine(s)", "not under command", etc.
Rate of turn	right or left, from 0 to 720 degrees per minute.
Longitude	to 0.0001 arcminutes
Latitude	to 0.0001 arcminutes
Speed Over Ground (SOG)	Speed of the ship when removing the effects of the currents; in knots : 0.1-knot (0.19 km/h) resolution from 0 to 102 knots (189 km/h)
Course Over Ground (COG)	Direction in which the ship is moving when taking into account the effects of the current; relative to true north to 0.1°
True heading	0 to 359° (for example from a gyro compass)
True bearing at own position	0 to 359°
UTC seconds	The seconds field of the UTC time when these data were generated

Table 2.1: Standard Class A AIS position report[6]

IMO ship identification number	A seven digit number that remains unchanged upon transfer of the ship's registration to another country
Radio call sign	International radio call sign, up to 7 characters, assigned to the vessel by its country of registry
Name	20 characters to represent the name of the vessel
Type of ship/cargo	
Dimensions of ship	to nearest meter
Location of positioning system's antenna on board the vessel	in meters aft of bow and meters port or starboard
Type of positioning system	such as GPS, DGPS or LORAN-C.
Draft of the ship	vertical distance between the waterline and the bottom of the hull; between 0.1–25.5 meters
Destination	max. 20 characters
ETA (estimated time of arrival) at destination	UTC month/date hour:minute
High precision time request	Optional field a vessel can use to request other vessels provide a high precision UTC time and datestamp

Table 2.2: Standard Class A AIS static data report[6]

one which is sent every 6 minutes and contains more static data about the ship (as detailed in table 2.2).

Class B transceivers send the same kind of data, but they limit their standard position report to MMSI, time, SOG, COG, longitude, latitude and true heading, and send it at a different rate depending on the SOG (every 3 minutes for speeds under 2 knots, and every 30 seconds for greater speeds), the same data presented in Table 2.1 is sent as Extended Position Report upon request from coast stations. The static data report uses the same 6 minutes interval as Class A and brodcasts MMSI, boat name, ship type, call sign, dimensions and equipment vendor id.

The fact that these transceivers are mandatory means that in theory, when collecting the signals in a set area, we should have complete information about the boats navigating through it during the observation period.

However, ultimately, it is up to the observer to decide what part of the data is to be saved in their database. Not all of them register the complete information, and it is often limited to the parts of the data that are common between all classes of AIS, so we might be limited depending on what features our data sources have decided to keep.

This data is only partly openly available online. Actually, the Maritime Safety Committee (MSC) has been condemning the publication of AIS data to the World Wide Web and encourages its Member Governments to actively discourage it as well. The MSC argues that the public availability of such data is detrimental to the safety objectives of the organization and contrary to its safety measures.[7]

Despite that, there are many applications using either openly available AIS data or data provided by safety organisms. These application range from vessel tracking via data mining[8] or visualizations[9] to navigator behaviors pattern mining[10].

2.1.2 HELCOM

The *Baltic Marine Environment Protection Commission*, also known as the *Helsinki Commission* or *HELCOM* is an organization that cooperates with organizations governed by the signers of the *Helsinki Convention on the Protection of the Marine Environment of the Baltic Sea Area*[1].

HELCOM has various missions around maritime environment protection, based on the Baltic Sea Action Plan, and its vision of *"A healthy Baltic Sea environment,*

[1]The 9 countries with a coast on the Baltic Sea : Denmark, Estonia, Finland, Germany, Latvia, Lithuania, Poland, Russia and Sweden, plus the European Union signing as a separate entity

with diverse biological components functioning in balance, resulting in good environmental/ecological status and supporting a wide range of sustainable human economic and social activities". Among its numerous missions, HELCOM works with the International Maritime Organization to monitor shipping activities in the Baltic sea which could have an environmental impact , and this implies aggregating data from all member countries for reporting, among which we can find AIS data.

2.2 Machine Learning

This section lays out what we describe as Machine Learning and presents the different techniques used in this study.

2.2.1 Predictive Model and Supervised Learning

In Supervised Machine Learning, we study a system with a set of variables, called *inputs*, that have an influence on one or more other variables, called *outputs*[11].

In the typical definition of Artificial Intelligence, a program is presented with a set of inputs and we create the rules that it has to follow to determine the correct output (e.g. the actions to take in a particular situation). Instead, in supervised learning, a program is presented with a set of inputs and their associated outputs and it then has to infer the rules that support the system that is studied[12].

This set of found rules is called the model, and it is then tested by trying to apply its rules to a new set of inputs. The model provides a *prediction* based on the inputs, and we compare them to the *true outputs*. The ability of the model to provide accurate predictions consistently is most often the metric upon which we judge the quality of a model.

The term *supervised learning* is used in opposition to *unsupervised learning*, where no examples are needed; this can be applied for tasks where we let the networks find the characteristics of the inputs to differentiate them (e.g. clustering, anomaly detection...) or to reproduce them (e.g. content generation, speech synthesis...).

Supervised learning algorithms are not too different from the regression analysis that can be found in statistical modeling, but different methods have different ways of combining the inputs that go beyond what even nonlinear regression can do.

2.2.2 Deep Learning

This next section will be dedicated to presenting Deep Learning, from its most basic form to the highly complex systems that will be used in this study.

2.2.2.1 Neural Networks

Among supervised learning methods, *Artificial Neural Networks* (simply called *neural networks* hereafter), are the ones closest to nonlinear regression.

The name Neural Network conveys the initial motivation behind the creation of this technique; the goal is to replicate the way a human brain works, with its billions of neurons and synapses performing highly complex calculations.

Neural networks are composed of units called *cells*, *artificial neurons* or just *neurons* for simplicity; these neurons are linked to each other in layers, and each neuron in a layer applies *combinational weights* to the information coming from neurons on the previous layers to create new outputs[13].

This is to emulate the way neurons work in the human brain, and the high level of interconnectivity is important here.

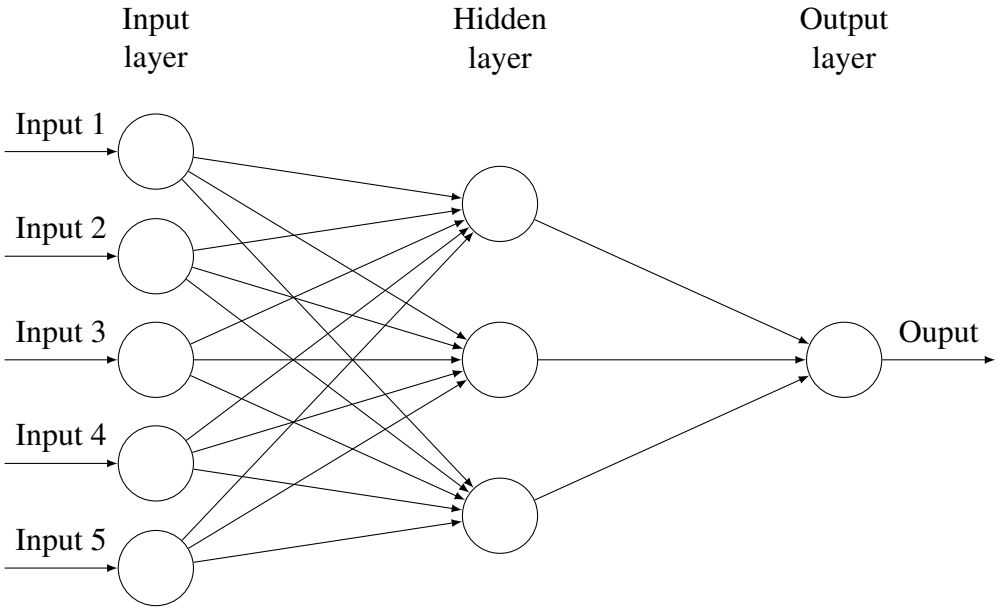

Figure 2.1: Representation of a simple Feed-Forward Neural Network.

The kind of network represented in Figure 2.1 is called a *feed-forward* network, as the information is always passed to the next layer, its layers are *dense*, fully connected to the ones after (i.e. all neurons in a layer are connected to all neurons of the following layer) with those weighted connections (detailed in Fig. 2.2).

$$y(x) = f(wx + b) \tag{2.1}$$

A layer between the inputs and the outputs is commonly called a *hidden layer* because it is not explicitly accessed. The weights w and biases b (Eq. 2.1) are adjusted during the training so that they can be combined with the input vector x to

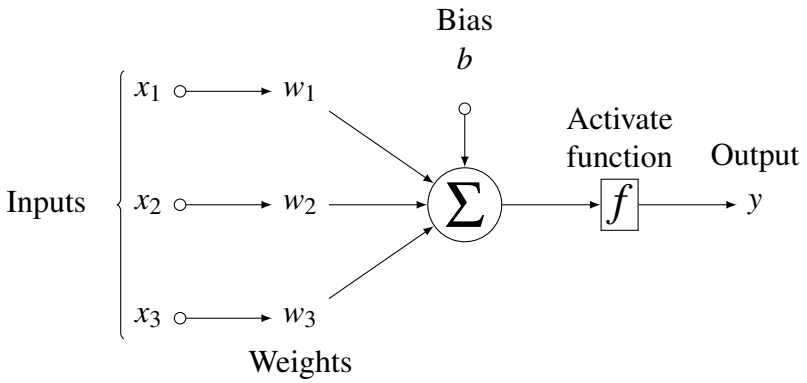

Figure 2.2: Representation of the connections between neurons using notations from Eq. 2.1

obtain an output y closer to the true result. The function applied to this combination is called the *activation function*. Some functions that are commonly used are the *rectified linear units (ReLU)*, the *logistic sigmoid* (σ) and *hyperbolic tangent (tanh)*.

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

$$\sigma(x) = \frac{1}{1+e^{-x}} \tag{2.3}$$

$$tanh(x) = \frac{e^{x}-e^{-x}}{e^{x}+e^{-x}} \tag{2.4}$$

2.2.2.2 Training process

The training of a neural network is done in several iterations called *epochs*. During every epoch, the training data is passed through the network and the outputs are evaluated. The evaluation is based on a metric called the *loss* which is usually a measure of the error of the model that we want to minimize (although sometimes it can be an other kind of measure that has to be maximized).

The minimization (or maximization) of the loss is done through an optimization algorithm that uses the loss to modify the weights in the networks starting from the last ones (this is the process of *backpropagation*).

$$g_t = \bigtriangledown L(W_t) \tag{2.5}$$
$$W_{t+1} = W_t - lr * g_t$$

Different optimization algorithms have different update rules using the loss and the previous weights and are affected by different hyperparameters. Equation 2.5 shows the update rule for Stochastic Gradient Descent[14], which is considered one of the simplest optimization algorithms[15]. It uses the gradient of the loss and has a single hyperparameter lr, its learning rate that many gradient-based optimizers also have, which determines how fast the model is going to converge on the optimal solution (a compromise is to be made between a low learning rate, which would make the training longer, and a high learning rate that might prevent the model from stabilizing on its optimal solution.).

$$v_0 = 0 \; ; \; v_{t+1} = \gamma * v_t + g_t \tag{2.6}$$
$$W_{t+1} = W_t - lr * v_{t+1}$$

$$\alpha_{t-1} = \sum_{i=1}^{t-1} \bigtriangledown g_i^2 \; ; \; lr_t = \frac{lr}{\sqrt{\alpha_{t-1} + \varepsilon}} \tag{2.7}$$
$$W_{t+1} = W_t - lr_t * g_t$$

$$eda_{t-1} = \gamma * eda_{t-2}((1-\gamma) * g_{t-1}^2) \tag{2.8}$$
$$lr_t = \frac{lr}{\sqrt{eda_{t-1} + \varepsilon}}$$
$$W_{t+1} = W_t - lr_t * g_t$$

Other gradient-based optimizers improve on this method by adding parameters to converge faster, such as Stochastic Gradient Descent with momentum (Eq. 2.6,[16]), or look into an adaptive learning rate such as Adagrad (Eq. 2.7,[17]) or AdaDelta (Eq2.8,[18]).

One of the most widely used optimizers based on these methods is Adam [19], which uses the first and second order of moment (i.e. g_t and its square) in the decay rate (Eq. 2.9) for a more complex tuning (since they are affected by separate hyperparameters β_1 and β_2.

$$
\begin{aligned}
m_t &= \beta_1 * m_{t-1} + (1-\beta_1) * g_t \\
v_t &= \beta_2 * v_{t-1} + (1-\beta_2) * g_t^2 \\
m_t^* &= \frac{m_t}{1-\beta_1^t} \\
v_t^* &= \frac{v_t}{1-\beta_2^t} \\
W_t &= W_{t-1} - lr * \frac{m_t^*}{\sqrt{v_t^* + \varepsilon}}
\end{aligned}
\tag{2.9}
$$

2.2.2.3 Deep Neural Networks

A neural network is called *deep* when it uses several hidden layers successively, allowing for a very high complexity in the combination of the inputs, especially when different functions are applied between each layer. The number of different layers between the inputs and outputs is the *depth* of the network.

In the few years since their introduction, deep learning models have imposed themselves as a staple of machine learning. They have been looked at for all kinds of applications and often taken as a silver bullet for tasks that we didn't think possible before. This is because they have shown high performance approaching or surpassing human accuracy on supervised tasks like image recognition[20][21], language translation[22], as well as unsupervised task like text generation[23] or image generation[24].

However to obtain the results they obtained, these methods have greatly expanded on the basic architecture of deep neural networks.

2.3 Recurrent Neural Network

Recurrent Neural Networks are a class of neural networks where, in addition to the traditional output (vertical outputs h in Fig. 2.3), the information of one unit is also passed along inside the same layer (horizontal outputs in Fig. 2.3). This property gives it the ability to use its internal state as a memory of sorts and exploit part of the information from previous states, which allows for better treatment of information that comes in the form of sequences, such as phrases, time series, audio or video inputs.

When unrolled, we can see that each cell receives information from all its pre-

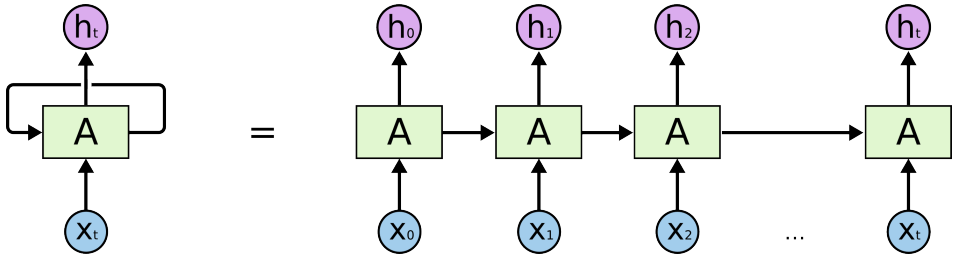

Figure 2.3: Recurrent Neural Network architecture showed in its compact and unrolled representation. Source: [25]

decessors, in addition to the normal input. In this form, we can assimilate each cell as a layer with a very specific way of combining the data from previous cells, which makes the RNNs analogous to a deep neural network, which means that it can be trained in the same way.

2.3.1 Long Short-Term Memory

Long Short-Term Memory networks are an upgrade of the classic RNN[26]. They have a *memory cell* that has a more complex structure than regular neurons. It includes the *input gate* that updates the cell state values, the *output* gate that selects the parts of the cell state to output, and a *forget gate*. This gate allows the internal memory to select irrelevant information from the previous cells to be forgotten.[27]

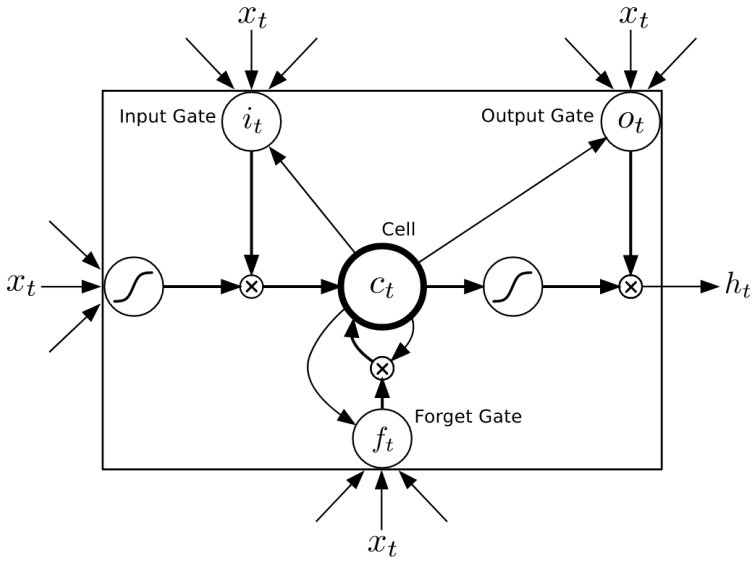

Figure 2.4: Architecture of the LSTM cell as defined in [28].

Figure 2.4 shows the design of the LSTM cell with the notation commonly found when laying down the equation behind the activation function vectors of the cell

state.

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i) \tag{2.10}$$

$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f) \tag{2.11}$$

$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_t + b_o) \tag{2.12}$$

$$c_t = f_t c_{t-1} + i_t tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \tag{2.13}$$

$$h_t = o_t tanh(c_t) \tag{2.14}$$

Equations 2.10-2.12 show the use of a logistic sigmoid ruling the updates of the different gates, each using its own set of weights W and bias b that are learned during the training. Equations 2.13 and 2.14 denote the two types of hidden cell states of the LSTM. We can see that h_t, which is a regularization of c_t through a hyperbolic tangent and a combination with o_t, is used by the next cell in conjunction with the bare c_t.

LSTMs have been known to solve the problem of *vanishing gradients* that regular RNNs can suffer from[29] as the cell state accumulates activities over time and derivatives of the error are summed[30].

Whereas RNNs get the full information from past states indiscriminately, LSTM's selective memory also makes it better at identifying actions and changes that can have an impact much later (long-term dependencies)[31].

In recent years, LSTMs have been widely used in the context of Natural Language Processing, due to their high performance in prediction and classification tasks over long sequences, but those properties can be seen in a much wider field of problems, and many methods like ours look to apply LSTMs to other domains.

Other networks have been developed with designs analogous to the LSTM, like the Gated Recurrent Unit (GRU)[32] , which also has a forget gate but no output gate (which means their cell state and hidden state are one and the same), and have been used for their lower number of parameters yielding higher learning speed.

2.3.2 Phased LSTM

The Phased LSTM is defined by Neil et al. (2016)[33] as a further extension of the LSTM that adds a new *time gate*, k_t (as shown in Figure 2.5). This gate has an opened and a closed state. When it is open, the cell functions as usual, but when it is closed, the cell state is not updated by c_t and h_t.

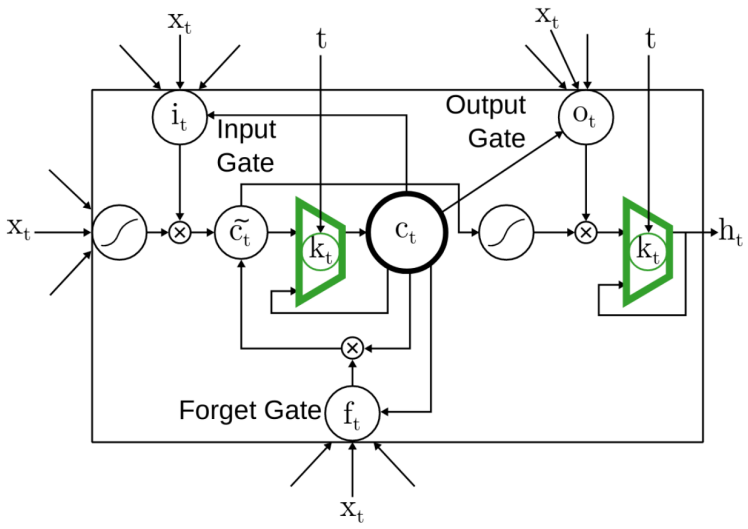

Figure 2.5: Architecture of a Phased LSTM cell in the same style as [28] showing where the new k_t gate takes place in the LSTM cell. Source: [33]

$$\phi_t = \frac{(t-s) mod \tau}{\tau} \tag{2.15}$$

$$k_t = \begin{cases} \frac{2\phi_t}{r_{on}}, \text{ if } \phi_t < \frac{1}{2} r_{on} \\ 2 - \frac{2\phi_t}{r_{on}}, \text{ if } \frac{1}{2} r_{on} < \phi_t < r_{on} \\ \alpha \phi_t, \text{ otherwise} \end{cases} \tag{2.16}$$

As shown in Eq. 2.16, what determines if the gate is opened or closed is a rhythmic oscillator that is independent of the rest of the model. This relationship with the oscillator has its own three parameters that can be learned during training : τ which is different for each neuron, s the phase shift of the oscillator and r_{on} the openness ratio. Figure 2.6 shows how these parameters apply graphically.

The parameter α used in the closed phase is the *leak rate* that allows some important information to be propagated despite the closed gate.

$$\tilde{c_j} = f_j c_{j-1} + i_j \sigma(x_j W_{xc} + h_{j-1} W_{hc} + b_c) \tag{2.17}$$

$$c_j = k_j \tilde{c_j} + (1 - k_j) c_{j-1} \tag{2.18}$$

$$\tilde{h_j} = o_j \sigma(\tilde{c_j}) \tag{2.19}$$

$$h_j = k_j \tilde{h_j} + (1 - k_j) h_{j-1} \tag{2.20}$$

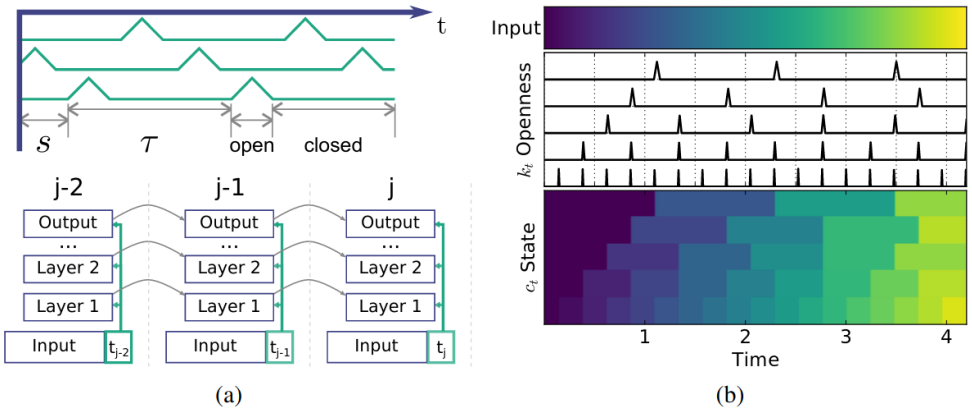

Figure 2.6: Diagram of Phased LSTM behaviour. **(a)** Top: The rhythmic oscillations to the time gates of 3 different neurons; the period τ and the phase shift s is shown for the lowest neuron. The parameter r_{on} is the ratio of the open period to the total period τ. Bottom: Note that in a multilayer scenario, the timestamp is distributed to all layers which are updated at the same time point. **(b)** Illustration of Phased LSTM operation. A simple linearly increasing function is used as an input. The time gate k_t of each neuron has a different τ, identical phase shifts, and an open ratio r_{on} of 0.05. Note that the input (top panel) flows through the time gate k_t (middle panel) to be held as the new cell state c_t (bottom panel) only when k_t is open. Source (including caption): [33]

For the Phased LSTM, since it can be applied to irregularly sampled time points, there is a new notation t_j. The new cell states equations use the short notation $c_j = c_{t_j}$ for cell states a time t_j (and $c_{j-1} = c_{t_{j-1}}$).

2.4 Explainable AI

The problem with a Phased LSTM, as with any kind of Deep Learning method, is that it is what is commonly called a *"black box"*. Contrary to a classic regression, the weights in a Deep Neural Network cannot be simply traced back to produce an explanation of the decision-making process.

Some machine learning models are developed only for prediction, so they can rely on deep learning only for its high performance, but in some cases, the models are built for applications of critical importance where we cannot afford to not understand the inner workings of the system.

Explainable AI has already proved it could be used to develop solutions in such applications like in the medical field[34] where the results can be reassessed by health professionals, or to bring interpretability to the computer vision used for autonomous driving[35]; two sectors where the safety of human life is on the balance.

Explainable AI designates the methods that aim to produce explainability for a system that does not have it by default.

2.4.1 Attention Mechanisms

Attention was first introduced by Bahdanau in 2014 [36] as a way for a translation model to decide what parts of the word sequence to focus on when doing the prediction. It was a parallel mechanism that was used in conjunction with an encoder-decoder to separate the information and make it easier to process.

Subsequent studies have looked into attention mechanisms and modified them to apply them to different kinds of tasks, such as translation[37] [38], document classification[39], or even image captioning[40]. Attention generally designates techniques that aim at enhancing certain parts of the input data.

Later works describe an attention mechanism that is an additional subnetwork that works in parallel of the main network (in this case a RNN)[41]. Among the input sequence elements, it selects the ones that will be used to update the RNN. The subnetwork learns to generate *attention weights* that are then combined to the original inputs and taken into account by the predictive part of the network. The most common method of attention mechanisms is using *dot-product attention*, where the weights and inputs are combined through a dot-product.

To generate proper weights, the outputs of the subnetwork are passed through a *softmax* function (Eq. 2.21).

$$\begin{aligned} &letx \in \mathbb{R}^K \\ &softmax(x) = \frac{e^{x_i}}{\sum_{j=1}^{K} e^{x_j}} \end{aligned} \tag{2.21}$$

The softmax function converts the outputs into values between 0 and 1, so that they can be interpreted as probabilities, while preserving the relative weight and making the weight's sum add up to 1, so we can use them as an indication of what part of the input the model should be focused on.

Methodology

The goal of this chapter is to describe the different processes that are necessary to carry out the objectives presented in section 1.1.

3.1 Data processing

This section will detail the shape of the data used in this study and the different processes needed to obtain a dataset suited to our Machine Learning task.

3.1.1 HELCOM's navigational dataset

The dataset used for navigation data is a dataset made available to us by HELCOM. It is referred to in the rest of the study as the *maritime navigation dataset*. This dataset contains files listing all signals collected in HELCOM member countries all across the Baltic Sea. There is a file for each month from January 2009 to December 2019.

The features that have been kept in this dataset are the MMSI of the ship and timestamp of the data point, the latitude, longitude, speed over ground, course over ground, and draft at the moment of the recordings, as well as the IMO number and dimensions of the ship extracted from the second recording.

The main challenge with these data comes from the unevenness of the sampling rate; the points in our dataset are very unevenly distributed, sometimes separated by one or two minutes, and on other occasions, there can be 15 minutes between two points.

The reason why shorter intervals may appear comes from the fact that HELCOM is a cooperation of many institutions that do not record the data in exactly the same way, and the data we have are reconstructed on top of those from the two types of signals described in section 2.1, in addition to the mix of Class A and Class B transceivers (with their adaptive sampling rate).

The reason for longer intervals is the potential data loss that can occur if the AIS transmitter is not sending the information properly (either due to a technical issue

or if it was voluntarily turned off by the crew), or if the data is not received properly (most likely due to weather conditions). In the Baltic Sea Area, maritime observers only work with base stations. There are no satellites, since the base stations are supposedly good enough to cover the area, but these base stations also have technical limitations depending on the weather, maintenance, etc..

The last potential explanation for the timing irregularities is that the data we are working with have already been processed by HELCOM when merging databases from all member countries and harmonizing inputs, which includes discarding signals that contain wrong information (e.g. position on land, wrong IMO number, wrong MMSI, etc.).

In the end, less than half of the data actually follow the standard 6-minute cycle, and even these ones have irregularities, be it one missing point or some points being 1-2 minutes late.

3.1.2 Maritime Accidents Database

The data sources presented in the previous section do not contain a simple indicator : whether the ship was involved in an accident. Although we could probably identify certain accidents by analyzing strange behaviors in the draft of the ship, we will not be able to catch all of them, so we need external data source on maritime accidents.

3.1.2.1 Source

The data we used relating to shipping accidents are extracted from an openly accessible database from HELCOM containing data from all the shipping accidents in the Baltic Sea since 1989 (this database being updated every few years, only data up to 2017 were available at the time of this study).

However, this database, like others of the same kind, suffers from problems of underreporting[42]. This means that numerous accident reports are missing some fields, and that includes some fields that we would need to identify the ships involved and trace those accidents to the navigation behaviors that caused them.

3.1.2.2 Integration

Since the navigation dataset only has data from as far back as 2009, we only considered accidents that were more recent than this. The maritime dataset was also reduced to 2009-2017 to make sure that we only take time periods where we have information on accidents.

To clean the accident dataset, we keep only the records that contain an indication of when and where an accident occured and what ships were involved. From this information, we can look inside the navigation dataset. If we have a record of a ship navigating at that time under that identification, we can take the data before the accident and decide on a time period where we consider the behavior as risk-prone. For this study, we took the estimate of one hour leading to the accident.

As a matter of fact, only a few accident records add the identification numbers of the involved ships. A total of 512 ships were properly identified out of 1297 accidents during the studied time period (note that this includes some accidents involving two ships), 286 of which could be traced back to their behavior before the accident.

We used the indications of where the accidents occured to identify *danger areas* and used this to filter our pool of behaviors. This is to ensure that even the behaviors we select that do not imply an accident are in zones where it is relevant to observe them. The estimation of a danger area is 5km around the location where an accident has taken place.

3.1.3 Behavior Sequences

The navigation dataset in its original form is a set of data points following each other in a pseudo-continuous way. It can be ordered by timestamps and separated by boat. However, the danger areas are the critical point in the data processing; they give us a metric upon which we can determine how to cut the sequences (that would otherwise have been cut arbitrarily). A sequence starts when a boat enters a potentially dangerous area and ends when the boat has an accident or exits the area.

3.1.3.1 Coordinates projection

To allow for better generalization despite the low number of examples in the dataset, the geographical coordinates are replaced by a projection on a relative area. Entering in a danger area marks the beginning of a sequence, and the coordinates at this entry mark the origin point of the sequence. The coordinates for the rest of the sequence are replaced with relative coordinates with the distance and angle from the origin point to the current point.

As a measure of the angle, we choose the absolute bearing of the trajectory. In maritime navigation, the absolute bearing is the angle between the magnetic north and an object observed from the vessel. An other way to use it, and what we are

using here, is taking the origin point of the sequence as point of observation, and as observed object the current point, this is the initial bearing of the trajectory. As the vessel moves the absolute bearing will change constantly, which is why for this coordinate system we only consider the initial bearing of the trajectory, which is enough (combined to the point of origin and the distance) to find the current coordinates.

Furthermore, to avoid big jumps in value when going from 359° to 0°, the bearing will be denoted by its sine and cosine, which as a side-effect, will make them analogous to a projection of the latitude and longitude in the relative coordinate system (since the bearing is the angle between the trajectory and the north, the cosine represents the latitudinal movement and the sine represents the longitudinal part).

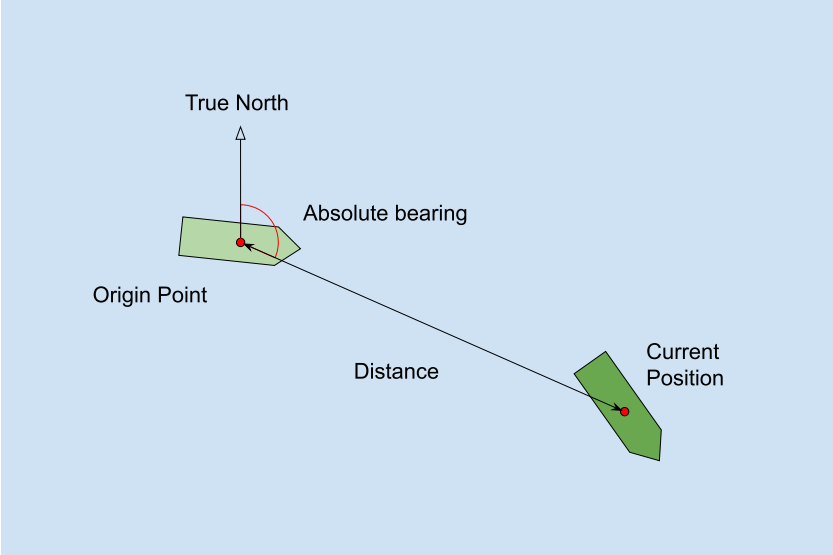

Figure 3.1: Visual representation of the projected coordinates

3.1.4 Final dataset

Table 3.1 summarizes the ten features selected for the final dataset; seven of them are from the original data and three have been derived from it.

The final data have been split into three datasets : the *training*, *validation* and *testing* datasets. Each dataset is composed of 50% of sequences containing a risk-prone behavior (as defined in section 3.1.2.2) and 50% of sequences showing only regular behaviors. This is a proportion that has been selected because, with a lower proportion of risk-prone behaviors, the number of false negatives increases; as risk-prone behaviors become negligible to the model. The training, validation and testing

datasets containing 70%, 15% and 15% respectively of the total sequences containing risk-prone behaviors, the sequences with regular behavior are selected randomly among the big pool of behaviors linked to the danger areas.

Distance	Distance from the origin point to the current point (in km)
Bearing sine	$sin(b_i)$ where b_i is the absolute bearing from the origin point to the current point, i.e. the sinus of the angle between the trajectory (0° at origin)
Bearing cosine	$cos(b_i)$ where b_i is the absolute bearing from the origin point to the current point (0° at origin)
Speed Over Ground (SOG)	Speed of the ship when removing the effects of the currents; in knots : 0.1-knot (0.19 km/h) resolution from 0 to 102 knots (189 km/h)
Course Over Ground (COG)	Direction in which the ship is moving when taking into account the effects of the current; relative to true north to 0.1°
Draft of the ship	vertical distance between the waterline and the bottom of the hull; between 0.1–25.5 meters
Bow dimension	Distance between the AIS transmitter and the bow (front) of the boat in meters
Port side dimension	Distance between the AIS transmitter and the port side (left) of the boat in meters
Starboard side dimension	Distance between the AIS transmitter and the starboard side (right) of the boat in meters
Stern dimension	Distance between the AIS transmitter and the stern (back) of the ship in meters

Table 3.1: Features selected for the predictive model

3.2 Implementation of the predictive model

This section will present the decisions that have been made concerning the modeling and implementation of our classification model.

3.2.1 Predicting over Irregular sequences

Recently, LSTMs have been widely used in the context of Natural Language Processing, due to their high performance in prediction and classification tasks over long sequences, but those properties can be seen in a much wider field of problems, and many methods like ours look to apply LSTMs to other domains.

The choice of an LSTM for our problem is motivated by the same kind of objectives. Since, in our navigational data, we have consecutive recordings of the situation of the ships, the behavior of a ship comes in the form of a sequence of those recordings, which LSTMs should be the right technology to work with.

However, the understanding of language does not require to take into account the timing of the words, only their place in the sentence; this is a property that we have when dealing with real-world navigational data. And as described in section 3.1.1, the sampling rate of our data is very uneven which is bad for the LSTM, that sees the sequences as a series of input without considering the link between them, which is akin to implicitly assuming the data to be evenly sampled.

This is why we have chosen a Phased LSTM for our model. The original publication on Phased LSTM showed a better performance than regular LSTMs on non-uniformly sampled data[33], and this property has proven useful in certain domains, like the medical domain, where studies used Phased LSTM to solve issues introduced by the event-based sampling of Electronic Health Records [43].

3.2.2 Architecture of the network

In our case, the Phased LSTM is used to build a predictive model that will identify the behaviors that are considered risk-prone.

The phased LSTM implementation we have chosen makes use of the PyTorch library for Python[44]. PyTorch is a library that provides a framework to implement various machine learning and neural networks models. The implementation of the phased LSTM cell is left untouched from the updated implementation provided by

Daniel Neil, with our model built around the phased LSTM layer provided (after adapting it to our input data).

Our problem is to discriminate between regular behaviors and risk-prone behaviors, so it is a binary classification problem, where the risk-prone behaviors represent our ***positive*** class. As with a logistic regression model, we only have one cell on the output layer (i.e. for each step of the behavior, a single value between 0 and 1 is returned) : a value closer to 0 indicates a regular behavior, a value closer to 1 indicates a risk-prone behavior. To that end, we chose a sigmoid function as the output activation function to confine the results between 0 and 1.

In the implementation we have chosen, the input is separated into two vectors, the regular input bX consists of the ten features we have selected as relevant to the behavior and will be fed to the network as x, after going through the attention mechanism. A separate vector bT contains the timestamps associated with these data points and will indicate the values of t_j for the *time gate* k_t.

3.2.3 Training Specifics

The model's loss function is the binary cross-entropy, which is a standard measure of how close the results are to the expected outputs when doing binary classification.

$$\begin{aligned} l(x,y) &= mean(L) \text{ where } L = \{l_1, l_2, ..., l_N\}^\top \\ l_n &= -[y_n \cdot log(x_n) + (1-y_n) \cdot log(1-x_n)] \end{aligned} \tag{3.1}$$

Equation 3.1 defines the binary cross-entropy between vectors x and y, the model's predictions and the true values (i.e. usually called the input and the target), by calculating the error between each predicted term and the expected term, reduced to their mean for the error over the whole batch.

For this model we have chosen the Adam optimizer[19] which is widely used in deep learning and has been proven as a reliable optimizer[15]. A learning rate of 0.0001 has been found to be ideal, because the small size of our dataset prevents the model from converging smoothly, using a learning rate lower than the defaults slightly helped to stabilize it.

The model was trained with an early stopping clause; this ensures that the training stops when the loss on the validation set stops decreasing. Since the validation

set does not contain the data used for training, the ability of the model to give consistently accurate predictions on it is representative of its ability to generalize the problem. Any decrease of the loss on the training set that is not reflected with the validation set would be synonymous with overfitting, i.e. the model learning patterns specific to the training model that are not useful in other data.

3.3 Explainable AI

In their Survey[45], Rojat et al. define seven potential purposes for Explainable AI (XAI) for time series, putting **trustworthiness** at the center (i.e. the underlying purpose of every XAI approach). This definition aligns with our objectives, as we have mentioned several times the critical aspect of the maritime safety domain, and we need a model that we can trust if we want to protect human lives.

Figure 3.2: Knowledge graph relating all the purposes of explain-ability methods for time series. Source: [45]

Of the seven potential purposes of XAI, we have already covered one : **confidence** (the ability of the model to assess the quality of its own predictions); as we have decided to work with an output in the form of probabilities, this already represents the confidence of the model in each prediction, and helps avoiding false alarms.

The main purpose of looking into XAI in our case is **explainability**. The goal of

the study is not only to classify which behaviors are risk-prone, but also to identify what sets those behaviors apart from the rest, so that we can provide useful insights to seafarers in order to reduce the risks taken at sea. This is why we need an XAI approach.

There are many approaches that can be qualified as XAI, but to add explainability to the Phased LSTM model, the one we have selected is Attention Mechanisms.

3.3.1 Attention Mechanisms

Attention Mechanisms are a combination of techniques that aim to direct the attention of the model on a certain part of the input, often by applying weights that change the way the inputs are taken into consideration by the model.

Those attention weights, while they cannot be regarded as a straight explanation of the way the model takes a decision [46][47], can still provide insight into how to look at the results[48].

Despite that, they are widely used in the field of NLP because they can give a human-interpretable intuition that is easy to verify a posteriori; but other domains might be able to take advantage of that kind of human interpretation verification as well, as long as there are people who have the necessary knowledge to interpret it.

This is why, in this study, the interpretation obtained from the attention mechanisms is meant to be presented to experts on maritime navigation, who will determine what parts of the identified behavior are actually at fault, if any.

3.3.2 Implementation of Attention-Based Phased LSTM

The biggest difference between our network and a typical neural network with attention is the lack of an embedding layer, as an embedding serves better to describe categorical value and not something continuous like our data.

As our attention mechanism, we chose a LSTM layer, as we figured that the attention mechanism would profit from the recognition of long-term dependencies that it provides, but the use of an additional Phased LSTM would imply the introduction of a second oscillator that might conflict with the main one, which is why we chose to make the attention mechanism time agnostic.

The attention LSTM was not made bidirectional because the model is seen as a predictive model so we do not want to give it access to future information.

The attention LSTM layer takes the original inputs and gives 10 outputs for each

data point (as many as the number of features our dataset has). These values then pass through a softmax function, as described in section 2.4.1, to obtain attention weights that can be assimilated to probabilities.

The softmax is applied over the features in each datapoint, so that the features will receive attention when doing the prediction for the point, but the points themselves keep the same weight inside the sequence, since we are outputting a prediction for each data point. Furthermore, the relative importance of different points in a sequence is already managed by the time gate.

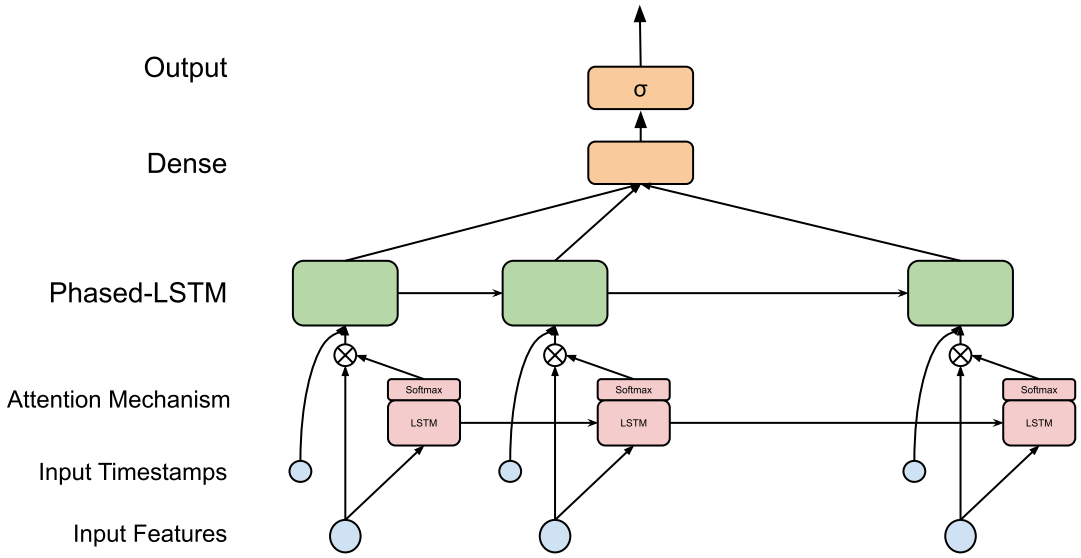

Figure 3.3: Final architecture of our Neural Network.

3.4 Evaluation of the Model

For the purpose of evaluating the model, we will review standard performance measures for classification : precision, recall and f-measure (also known as F_1 score), as defined in Equations 3.2-3.4.

$$precision = \frac{true\ positives}{true\ positives + false\ positive} \tag{3.2}$$

$$recall = \frac{true\ positives}{true\ positives + false\ negatives} \tag{3.3}$$

$$F_1 = \frac{2}{\frac{1}{precision} + \frac{1}{recall}} = 2 * \frac{precision * recall}{precision + recall} \tag{3.4}$$

The f-measure is the harmonic mean between the precision and the recall, so it can be seen as a compromise between the two other metrics. The idea of using a harmonic mean instead of a standard one is to make it so that the score will plummet if any of the two is too low. However, we will pay specific attention to the recall, which can also be seen as the *sensitivity* of the model (i.e. a higher recall is a lower chance of missing a risk-prone behavior).

This metric is very important in a domain involving the security of people, as we prefer to investigate a behavior and find that there was actually nothing wrong with it, instead of letting a risk-prone behavior pass us by.

This is why, while the main metric of judgement will be the f-measure, if we need to differentiate between two models with seemingly the same performance, we will prefer the one with a higher recall.

Results

The goal of this chapter is to present the results of this study in terms of machine learning architecture as well as the findings pertaining to maritime safety.

4.1 Predictive Model evaluation

This section will be showing the results obtained with our predictive model and evaluate them according to the criteria established in section 3.4.

4.1.1 Model output

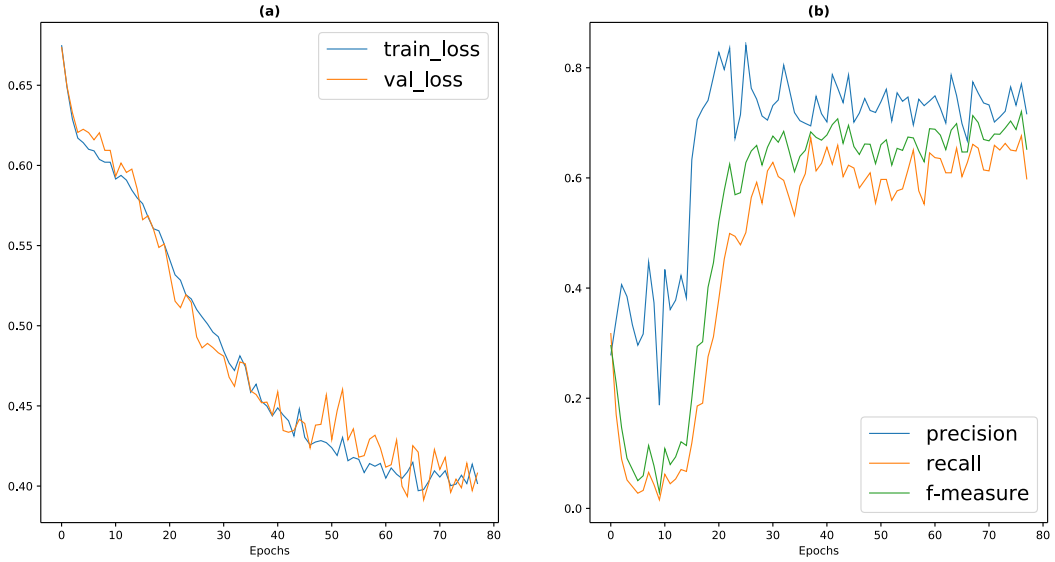

Figure 4.1: **(a)** Evolution of the losses during the training. **(b)** Evolution of the performance measures on the validation set during the training

Figure 4.1 shows the convergence of the loss and the evolution of the performance metrics during the training (ten extra epochs were left to show the stagnation that was removed by the early stopping). We can see that the loss decreases progressively for both datasets, and as we could expect the loss variations on the validation dataset are not as smooth as on the training dataset, that is most likely due to the

small size of the dataset which implies that a few errors can have a big impact on the overall performance.

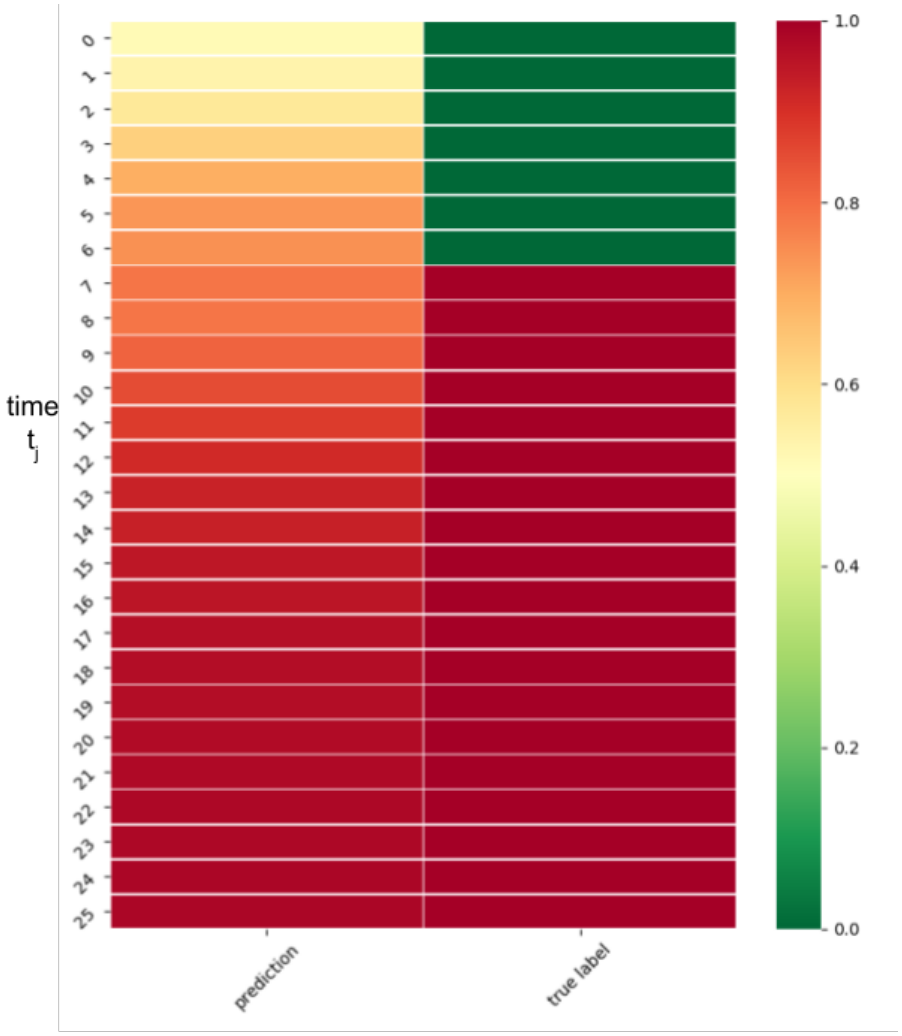

Figure 4.2: Example of a model prediction over the sequence.

Figure 4.2 shows an example of the output we get when predicting over a whole whole sequence, as described in section 3.2.2 for each point in the sequence we get an output that can be assimilated to a probability of being risk-prone. This sequence of probabilities is visualized as a heatmap with the probability as the color's intensity. The beginning of the sequence (at the top) is usually not too far from 0.5 as the network does not yet have enough information to give a very confident prediction. As the sequence progresses, the model tends to get more confident in its predictions.

4.1.2 Performance comparison with standard networks

The final model's performance measures are shown in Table 4.1, the model was tested on the same classification task with and without using its attention weights,

and other recurrent neural networks were tested to compare their performance on the same task. The models chosen for comparison were a standard LSTM, and a GRU.

Model	**Precision**	**Recall**	**F1-Score**
Phased-LSTM with Attention	67%	62%	64%
Phased-LSTM without Attention	64%	64%	64%
Standard LSTM	19%	27%	22%
Gated Recurrent Unit (GRU)	19%	21%	20%

Table 4.1: Performance results of the predictive model and other standard models for the classification of the testing dataset

For a fair comparison, the two models without a time gate received an additional feature representing the difference between the timestamp of the current point and the timestamp of the point before it (contrary to the Phased LSTM's time gate, which receives the timestamps as-is, but feeding the timestamps as-is to these networks seems to produce too much noise and prevents them from giving any results).

For the sake of those metrics, any probability over 0.5 is considered a 1 (i.e. a classification of risk-prone behavior) and values under 0.5 are considered a 0 (classification as regular behavior).

The first thing to note is that our model largely outperforms the other two models in terms of precision and recall. The conventional models do not seem to be able to generalize the risk-prone behaviors.

The second thing is that the difference between the phased LSTM with and without attention is not as pronounced as we had expected. In other domains, attention weights have been shown to increase the performance of neural networks on certain tasks[40][38], but our attention mechanism does not impact the performance too much (in fact it slightly decreases the recall, which is an important metric as described in section 3.4). This is most likely due to the fact that our attention mechanism is based on a regular time-agnostic LSTM.

4.1.3 Results observations

One thing we can notice when looking at the predictions of our model, is that the model rarely gives an abrupt change from one point to the next. This could be an

indication that the model is not able to precisely pinpoint the moment a behavior becomes risk-prone. This is easily explained by the fact that in our original dataset, the beginning of a risk-prone behavior was decided to be one hour before the accident, as described in section 3.1.2. This estimation means that, even in the training data, the marked beginning of the risk prone behavior was not necessarily significant, and the fact that the model was not able to learn this is actually a good sign that it did not learn dependencies where they should not be.

However, it also means that the model has a lot of "inertia": Figure 4.3 shows an example of a sequence where after a long history of (correctly classified) regular behavior the probability of being risk prone increases again, but not fast enough to cross the threshold of 0.5.

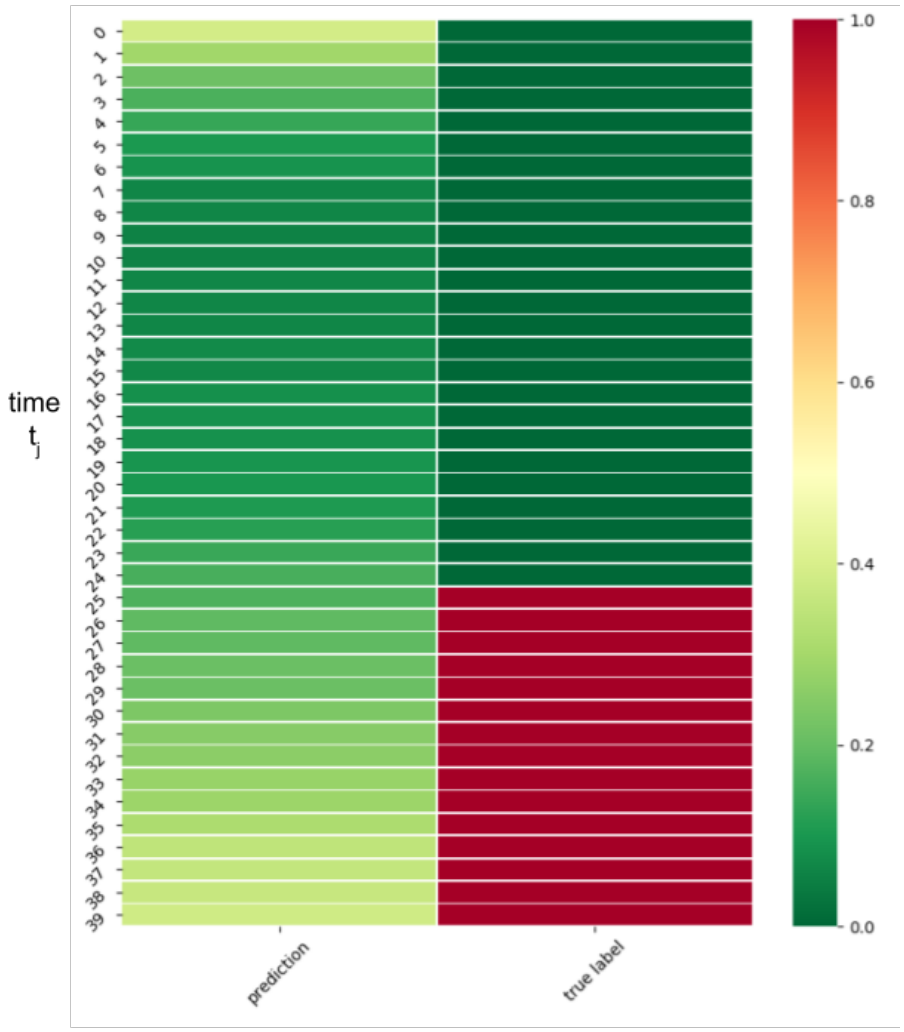

Figure 4.3: Another example of prediction

4.2 Attention mechanism interpretation

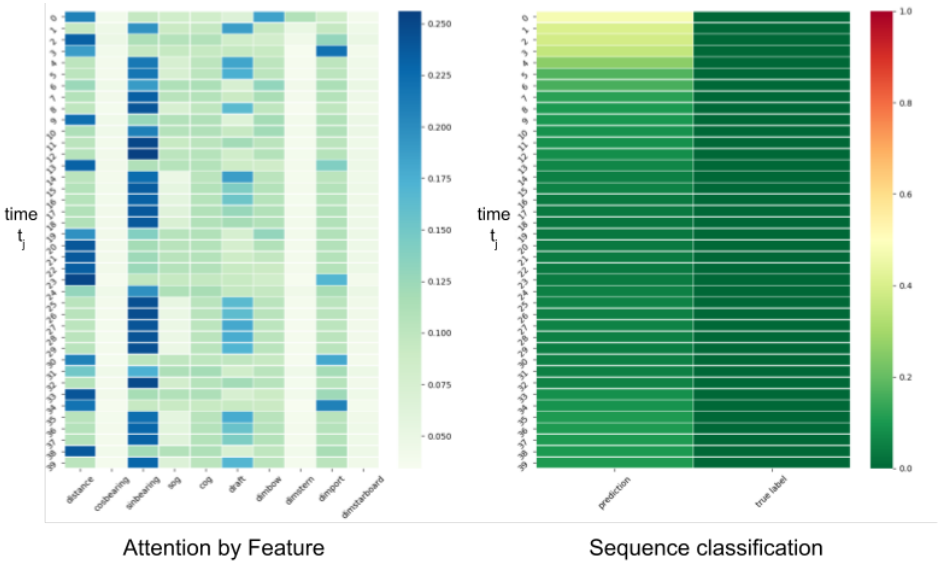

Figure 4.4: Visualization of the attention weights next to the visualization of the risk classification.

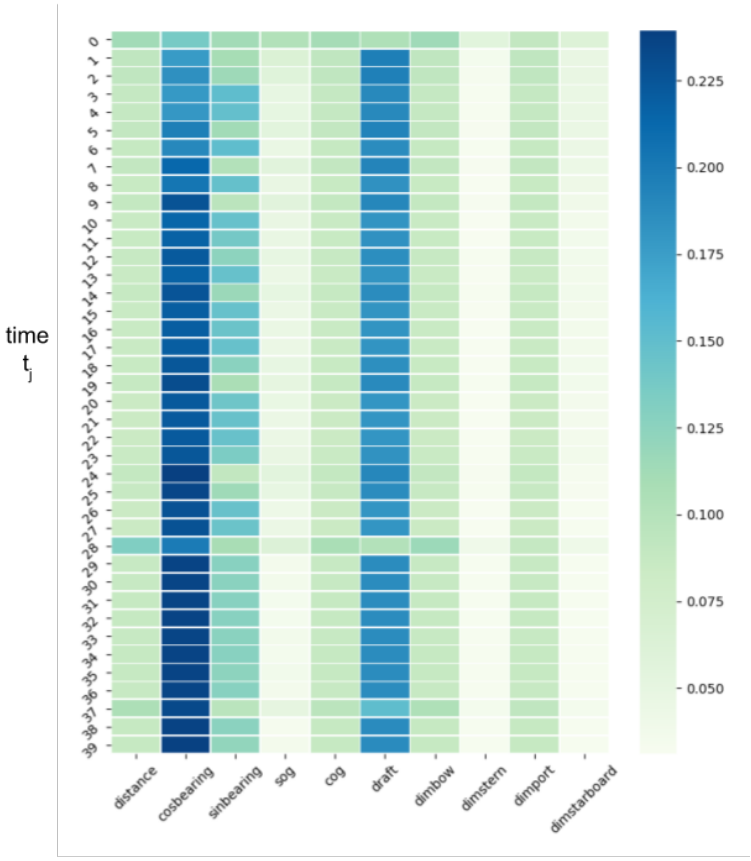

Figure 4.5: A second example of weights visualization for a more stable sequence

When we get the predictions for a sequence, we can extract the attention weights applied to the input. Figure 4.4 shows these attention weights as a heatmap, where each line shows the attention applied on each feature of each data point. A more intense shade of blue indicates a higher attention.

On Fig. 4.4 we can see a sequence where the attention alternates between the distance traveled and the longitudinal part of the movement. Other sequences, like the one shown on Fig. 4.5, have attention weights that stay relatively the same all across the sequence (here focused in the latitudinal movement and the draft of the ship).

4.2.1 Attention observations

Once we have made our predictions for the whole testing dataset, we can observe the tendencies of the weights. Figure 4.6 represents the distribution of the weights across all predictions.

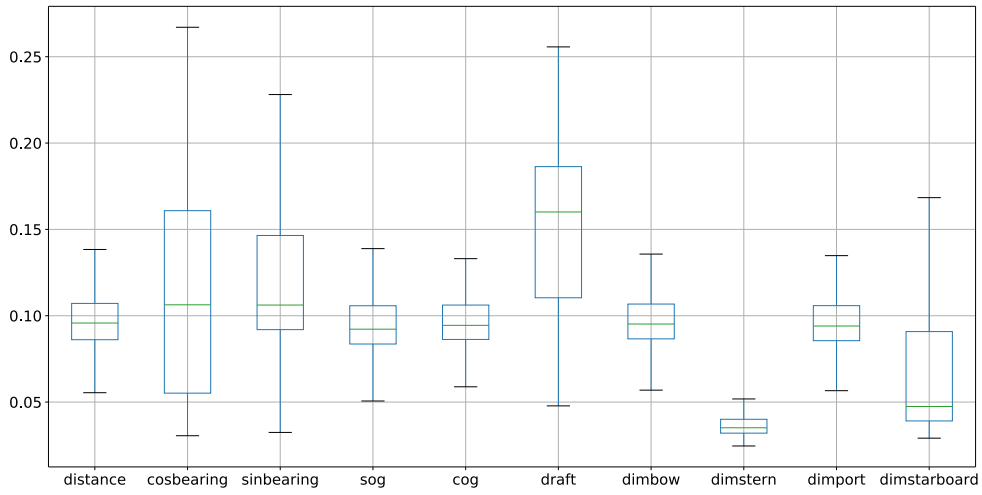

Figure 4.6: Boxplot of the attention weights for the whole testing dataset. (note that since we have 10 features and the attention weights are made to sum up to 1, 0.10 is our medium attention)

We can note that certain dimensions of the ship tend to receive lower attention in general, this effect is particularly strong for the distance between the transceiver and the stern of the ship. A possible explanation for this is that the AIS transceiver is often posted at the back of the ship, which means that the distance behind it is very often not indicative of the actual dimensions of the ship, and thus not relevant to the analysis of the ship's movement.

This was briefly considered as an indication that this feature could be removed

from the dataset, but there were enough outliers (shown on Fig. 4.7), to justify keeping it as it could prove relevant in cases where the transceiver is located in the middle of the ship. The same logic can be applied to the distance to the starboard side.

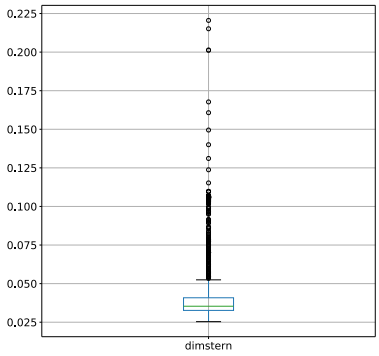

Figure 4.7: Boxplot of the weights applied on the the distance between the transceiver and the stern showing outliers.

The other features that tend to receive a particular amount of attention are the sine and cosine of the bearing, while on average they receive a regular 10% of attention, their attention has a much higher variance than the other features. This seems to be characteristic of a phenomenon that we can observe of Figure 4.5, as the network is rarely giving attention to both direction at the same time, and especially on Figure 4.4 often there is only one of the direction that is observed so closely.

Finally, the feature with the highest average attention is the draft of the ship. The draft of the ship is an important variable carrying multiple information at once, as it is a kind of height dimension for the ship but also an indicator of the sea conditions, so it is understandable that the model focuses on it more often than the other features.

Figure 4.8 shows that this discrepancy is even more noticeable in risk-prone behaviors. The draft of the ship is almost the only feature to have a sensibly different attention distribution depending on the true label (with the longitudinal movement, which tends to have attention values closer to the median and less spread toward higher values). Knowing that it tends to have a higher attention on average tells us that it might be worth observing the draft more closely than any other variables when we will be reviewing the behaviors manually, and the absence of a strong attention might be an information in and of itself.

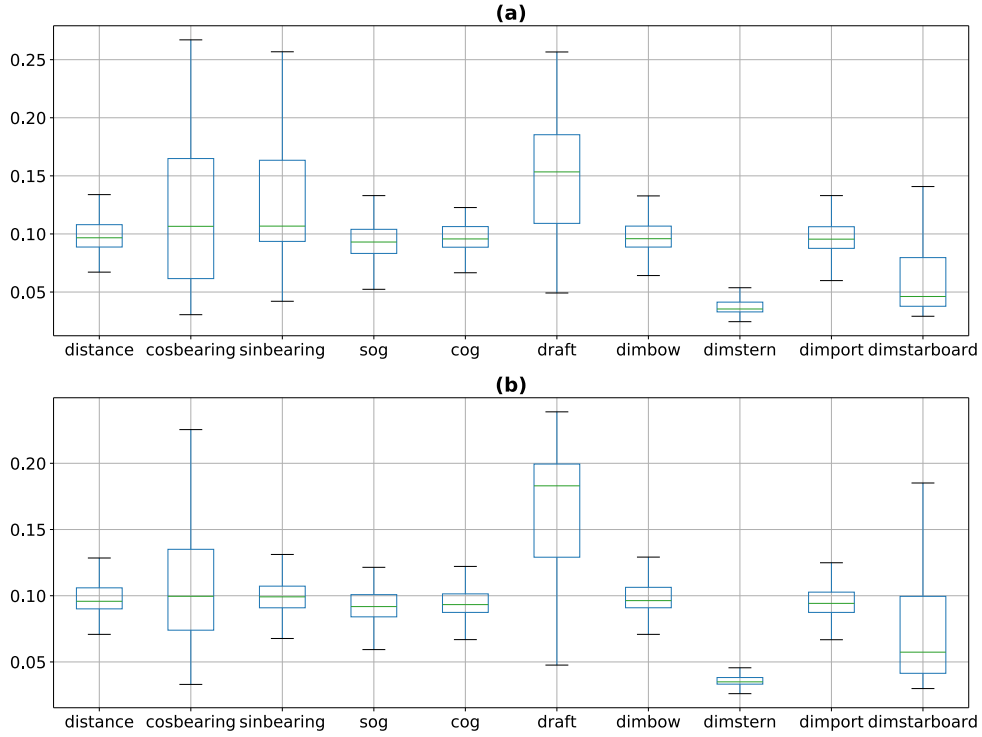

Figure 4.8: Boxplot of the attention weights for behaviors with true labels **(a)** regular behavior and **(b)** risk-prone behavior.

Discussion

This chapter's goal is to bring an external discussion on the results and their validity, as well as provide a conclusion for this whole study.

5.1 Threats to validity

This sections aims to address the possible threats to the validity of this study and its results.

5.1.1 Conclusion validity threats

Some aspects of the study threaten the validity of its results. The first hindrance during the setting of this experiment is the small number of examples we have for the risk-prone class. Only having 286 risk-prone behaviors to constitute our dataset not only reduces the ability of the network to generalize, but it forced us to introduce a major sampling bias in the data to make sure that the network wouldn't just discard said behaviors.

The same can be said for the criteria upon which we have decided to classify a behavior as risk-prone. Using the report of an accident as the basis seems sound, but big estimates were made by extrapolating what qualifies as a risk behavior from the record of an accident. Notably, we assumed for every boat that their behavior one hour before having an accident is a risk-prone behavior. Secondly, the *danger areas* are defined as 5km around a place where an accident has occurred, despite the fact that some of those zones are located in or close to ports, which should be very different from a zone in the middle of the sea.

Potential solutions to this specific issue will be addressed in section 5.2.

Moreover, a compromise had to be made between removing all accidents happening at port and removing accidents linked to a technical failures, as both could not be done without reducing the data to an unusable size. The choice was made to filter out the accidents at port, as AIS Data does not have the granularity necessary to observe the small movements made inside of a port (accidents on port approach

were still kept). The accidents related to technical failures were kept on the assumption that the technical failure can influence the behavior of the ship in a way that is visible from the data we have.

However, these issues have just been accepted as the reality of the maritime safety sector, and unless new data is formed with the specific intent of classifying risk-prone behaviors (possibly including risk-prone behaviors not directly leading to accidents), the dataset cannot be expanded too fast. As we cannot retrieve earlier AIS data and we are not hoping for an increase in shipping accidents in the Baltic Sea, we can only hope for better accident reporting.

5.1.2 Social threats

One upside of having data that was gathered a posteriori from a general monitoring database, is that we can safely assume that there are no social threats of any kind to the validity of our study. Seafarers know that their behaviors are monitored all the time but that is always the case independently of this study, and the database was built for HELCOM's own internal reporting so we know that our conjectures didn't have an impact on the data collection.

5.2 Improvements to the Data

This section describes the different shortcomings in the data used for this study, as well as possible improvements that could give be made to the data to give better results with the same model implementation.

5.2.1 Navigational Data

It is unmistakable that if this work were to be expanded on in future studies, a dataset would have to be created specifically for the task at hand. The ideal form of this dataset would a have much shorter sampling rate. Although an even one is not necessary thanks to the Phased LSTM architecture; a sampling rate that is shorter on average would make the behavior sequences more detailed and most probably give a better view of the ships' movements.

5.2.2 Accidents

The accident database suffers from the same problem of incompleteness. As we noted in the threats to validity, we needed to make big estimates and generalizations, when in reality each accident is unique and should be reviewed independently, at least when it comes to constituting the training data.

On that subject what is needed to improve the quality of the dataset would be a closer cooperation with maritime experts, from the very beginning when constituting the data.

5.2.3 Additional Features

An other addition to give the model better insight into the behaviors to determine their risk-prone factor would be to incorporate in the data some of the information that the seafarers use themselves when taking decisions.

One such information would be the weather conditions at the time of recording, as this has a major influence on the navigation, and some major irregularities could be considered completely normal under bad weather while still indicating a dangerous behavior in normal conditions. Weather information is so important for maritime situational awareness that it has recently been studied as a potential addition to AIS[49].

5.3 Conclusion

We have proposed an architecture that allows us to predict the risk of accident over irregularly sampled data, using attention mechanisms to extract a potential explanation of this risk. Although the performance of the predictive model was hindered by several shortcomings in the dataset, the results are satisfactory in that they demonstrate the ability of the model to generalize over new data samples.

Thus, this thesis has laid the groundwork for future studies to use the same architecture for the task of identifying risk-prone behaviors by seafarers. Future works should also include a visualization support to facilitate the transfer of knowledge to maritime experts without any knowledge of Artificial Intelligence and we have good hopes that it can be used for all sorts of Explainable AI approaches over irregularly sampled time series.

References

[1] *Unctad handbook of statistics 2020 - merchant fleet*, 2020. [Online]. Available: https://stats.unctad.org/handbook/MaritimeTransport/MerchantFleet.html.

[2] J. Häkkinen and A. Posti, "Overview of maritime accidents involving chemicals worldwide and in the baltic sea," *Maritime Transport & Shipping-Marine Navigation and Safety at Sea Transportation, CRC Press, Taylor and Frances Group, Abingdon, Oxford*, pp. 15–25, 2013.

[3] F. Nicolas, A. Bakhtov, M. Helavuori, and D. Shinoda, "Report on shipping accidents in the baltic sea from 2014 to 2017," HELCOM, Tech. Rep., 2018. [Online]. Available: https://helcom.fi/media/publications/Report-on-shipping-accidents-in-the-Baltic-Sea-from-2014-to-2017.pdf.

[4] K. Kulkarni, F. Goerlandt, J. Li, O. V. Banda, and P. Kujala, "Preventing shipping accidents: Past, present, and future of waterway risk management with baltic sea focus," *Safety science*, vol. 129, p. 104 798, 2020.

[5] *Iec61993-2 : Maritime navigation and radiocommunicationequipment and systems –automatic identification systems (ais)*. [Online]. Available: https://gmdsstesters.com/downloads/docs/IEC61993.pdf.

[6] H. M. Perez, R. Chang, R. Billings, and T. L. Kosub, "Automatic identification systems (ais) data use in marine vessel emission estimation," in *18th Annual International Emission Inventory Conference*, vol. 14, 2009, e17.

[7] Anonymous, *International maritime organization - ais transponders*. [Online]. Available: https://www.imo.org/en/OurWork/Safety/Pages/AIS.aspx.

[8] F. Deng, S. Guo, Y. Deng, H. Chu, Q. Zhu, and F. Sun, "Vessel track information mining using ais data," in *2014 International Conference on Multisensor Fusion and Information Integration for Intelligent Systems (MFI)*, 2014, pp. 1–6.

[9] F. Natale, M. Gibin, A. Alessandrini, M. Vespe, and A. Paulrud, "Mapping fishing effort through ais data," *PloS one*, vol. 10, no. 6, e0130746, 2015.

[10] M. Hansen, T. Jensen, T. Lehn-Schiøler, K. Melchild, F. Rasmussen, and F. Ennemark, "Empirical ship domain based on ais data," *Journal of Navigation*, vol. 66, pp. 931 –940, 2013.

[11] T. Hastie, R. Tibshirani, and J. Friedman, "Overview of supervised learning," in *The elements of statistical learning*, Springer, 2009, pp. 9–41.

[12] F. Chollet *et al.*, *Deep learning with Python*. Manning New York, 2018, vol. 361.

[13] S.-C. Wang, "Artificial neural network," in *Interdisciplinary computing in java programming*, Springer, 2003, pp. 81–100.

[14] H. Robbins and S. Monro, "A stochastic approximation method," *The annals of mathematical statistics*, pp. 400–407, 1951.

[15] D. Choi, C. J. Shallue, Z. Nado, J. Lee, C. J. Maddison, and G. E. Dahl, "On empirical comparisons of optimizers for deep learning," *CoRR*, vol. abs/1910.05446, 2019. arXiv: 1910.05446. [Online]. Available: http://arxiv.org/abs/1910.05446.

[16] B. T. Polyak, "Some methods of speeding up the convergence of iteration methods," *Ussr computational mathematics and mathematical physics*, vol. 4, no. 5, pp. 1–17, 1964.

[17] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization.," *Journal of machine learning research*, vol. 12, no. 7, 2011.

[18] M. D. Zeiler, "ADADELTA: an adaptive learning rate method," *CoRR*, vol. abs/1212.5701, 2012. arXiv: 1212.5701. [Online]. Available: http://arxiv.org/abs/1212.5701.

[19] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[20] A. Karpathy, "What i learned from competing against a convnet on imagenet," *Andrej Karpathy Blog*, vol. 5, pp. 1–15, 2014.

[21] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. H. S. Torr, "Conditional random fields as recurrent neural networks," *CoRR*, vol. abs/1502.03240, 2015. arXiv: 1502.03240. [Online]. Available: http://arxiv.org/abs/1502.03240.

[22] S. P. Singh, A. Kumar, H. Darbari, L. Singh, A. Rastogi, and S. Jain, "Machine translation using deep learning: An overview," in *2017 International Conference on Computer, Communications and Electronics (Comptelix)*, 2017, pp. 162–167.

[23] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners," 2019.

[24] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *arXiv preprint arXiv:1511.06434*, 2015.

[25] C. Olah, "Understanding lstm networks," 2015.

[26] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997. eprint: `https://doi.org/10.1162/neco.1997.9.8.1735`. [Online]. Available: `https://doi.org/10.1162/neco.1997.9.8.1735`.

[27] F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: Continual prediction with lstm," 1999.

[28] A. Graves, "Generating sequences with recurrent neural networks," *arXiv preprint arXiv:1308.0850*, 2013.

[29] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," in *International conference on machine learning*, PMLR, 2013, pp. 1310–1318.

[30] G. Abdullayeva, "Application and evaluation of lstm architectures for energy time-series forecasting," M.S. thesis, University of Tartu, 2019.

[31] A. M. Schaefer, S. Udluft, and H.-G. Zimmermann, "Learning long-term dependencies with recurrent neural networks," *Neurocomputing*, vol. 71, no. 13-15, pp. 2481–2488, 2008.

[32] K. Cho, B. van Merrienboer, Ç. Gülçehre, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," *CoRR*, vol. abs/1406.1078, 2014. arXiv: `1406.1078`. [Online]. Available: `http://arxiv.org/abs/1406.1078`.

[33] D. Neil, M. Pfeiffer, and S.-C. Liu, "Phased lstm: Accelerating recurrent network training for long or event-based sequences," *arXiv preprint arXiv:1610.09513*, 2016.

[34] E. Choi, M. T. Bahadori, J. A. Kulas, A. Schuetz, W. F. Stewart, and J. Sun, "Retain: An interpretable predictive model for healthcare using reverse time attention mechanism," *arXiv preprint arXiv:1608.05745*, 2016.

[35] M. Hofmarcher, T. Unterthiner, J. Arjona-Medina, G. Klambauer, S. Hochreiter, and B. Nessler, "Visual scene understanding for autonomous driving using semantic segmentation," in *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*, W. Samek, G. Montavon, A. Vedaldi, L. K. Hansen, and K.-R. Müller, Eds. Cham: Springer International Publishing, 2019, pp. 285–296, ISBN: 978-3-030-28954-6. [Online]. Available: https://doi.org/10.1007/978-3-030-28954-6_15.

[36] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[37] M. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," *CoRR*, vol. abs/1508.04025, 2015. arXiv: 1508.04025. [Online]. Available: http://arxiv.org/abs/1508.04025.

[38] P.-Y. Huang, F. Liu, S.-R. Shiang, J. Oh, and C. Dyer, "Attention-based multimodal neural machine translation," in *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, Berlin, Germany: Association for Computational Linguistics, Aug. 2016, pp. 639–645. [Online]. Available: https://www.aclweb.org/anthology/W16-2360.

[39] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, and E. Hovy, "Hierarchical attention networks for document classification," in *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies*, 2016, pp. 1480–1489.

[40] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhutdinov, R. Zemel, and Y. Bengio, *Show, attend and tell: Neural image caption generation with visual attention*, 2016. arXiv: 1502.03044 [cs.LG].

[41] D. Bahdanau, J. Chorowski, D. Serdyuk, P. Brakel, and Y. Bengio, "End-to-end attention-based large vocabulary speech recognition," in *2016 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, IEEE, 2016, pp. 4945–4949.

[42] M. Grabowski, Z. You, Z. Zhou, H. Song, M. Steward, and B. Steward, "Human and organizational error data challenges in complex, large-scale systems," *Safety Science*, vol. 47, no. 8, pp. 1185–1194, 2009.

[43] S.-J. Bang, Y. Wang, and Y. Yang, *Phased-lstm based predictive model for longitudinal ehr data with missing values*, 2020.

[44] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, *et al.*, "Pytorch: An imperative style, high-performance deep learning library," *arXiv preprint arXiv:1912.01703*, 2019.

[45] T. Rojat, R. Puget, D. Filliat, J. Del Ser, R. Gelin, and N. Díaz-Rodríguez, "Explainable artificial intelligence (xai) on timeseries data: A survey," *arXiv preprint arXiv:2104.00950*, 2021.

[46] S. Jain and B. C. Wallace, "Attention is not explanation," *CoRR*, vol. abs/1902.10186, 2019. arXiv: 1902.10186. [Online]. Available: http://arxiv.org/abs/1902.10186.

[47] S. Serrano and N. A. Smith, "Is attention interpretable?" *CoRR*, vol. abs/1906.03731, 2019. arXiv: 1906.03731. [Online]. Available: http://arxiv.org/abs/1906.03731.

[48] S. Wiegreffe and Y. Pinter, "Attention is not not explanation," *CoRR*, vol. abs/1908.04626, 2019. arXiv: 1908.04626. [Online]. Available: http://arxiv.org/abs/1908.04626.

[49] B Tetreault and G. Johnson, "Sharing ships' weather data via ais," *TransNav: International Journal on Marine Navigation and Safety of Sea Transportation*, vol. 14, 2020.