Trajectory Prediction Network for Future Anticipation of Ships

Pim Dijt  
University of Amsterdam  
pim.dijt@student.uva.nl

Pascal Mettes  
University of Amsterdam  
p.s.m.mettes@uva.nl

ABSTRACT
This work investigates the anticipation of future ship locations based on multimodal sensors. Predicting future trajectories of ships is an important component for the development of safe autonomous sailing ships on water. A core challenge towards future trajectory prediction is making sense of multiple modalities from vastly different sensors, including GPS coordinates, radar images, and charts specifying water and land regions. To that end, we propose a Trajectory Prediction Network, an end-to-end approach for trajectory anticipation based on multimodal sensors. Our approach is framed as a multi-task sequence-to-sequence network, with network components for coordinate sequences and radar images. In the network, water/land segmentations from charts are integrated as an auxiliary training objective. Since future anticipation of ships has not previously been studied from such a multimodal perspective, we introduce the Inland Shipping Dataset (ISD), a novel dataset for future anticipation of ships. Experimental evaluation on ISD shows the potential of our approach, outperforming single-modal variants and baselines from related anticipation tasks.

CCS CONCEPTS
• Computing methodologies → Multi-task learning.

KEYWORDS
trajectory prediction, future anticipation

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

1 INTRODUCTION
This work focuses on predicting future locations of ships from a multimodal perspective. Future anticipation has recently gained traction in various tasks, including action anticipation [49], live video search [6], and human trajectory anticipation [1, 14, 23, 44, 48].

Figure 1: Future trajectory ship prediction. The available data collected on a ship are spatiotemporal GPS coordinates derived from the AIS (blue), radar images (red) and ENC charts (green). We propose a new multimodal network that combines the sources to predict future trajectories (orange).

current works focus on past coordinate locations to enable such a prediction [4, 31, 32], this work advocates the multiple modalities needed to anticipate future trajectories of ships.

When navigating ships, information from multiple auxiliary sensors are at hand, as shown in Figure 1. First, this includes the Automatic Identification System (AIS), which provides rough coordinate locations of nearby ships. Second is radar inputs from RADAR to scan the direct area around the ship. A third input comes from Electronic Navigational Charts (ENC) that detail which parts are water and land for a region. Each source provides different information, hence making sense of all sources in a unified manner enables strong spatio-temporal trajectory anticipation.

We propose a Trajectory Prediction Network, a novel deep network that integrates coordinates, radar images, and segmentation charts from previous observations to predict sequences of future coordinates. The network is a multi-task sequence-to-sequence model implemented as a recurrent convolutional neural network. Previous observation from ship coordinates are combined with radar images to predict future trajectories. Furthermore, the charts are used as supervision for an auxiliary segmentation task. Since future anticipation is inherently uncertain, we also model the uncertainty of the predictions within our network. To get the most out of the combined network, we show the importance of reorienting the network to perform prediction for each ship in a region.

Since multimodal ship trajectory prediction is a novel task, we introduce the Inland Shipping Dataset (ISD), a new dataset that enables us to tackle the problem. The dataset contains 33315 trajectories from 620 ships, along with information from all three
modalities. Experiments on this dataset shows that our approach obtains state-of-the-art results, outperforming single-modal solutions, heuristic approaches, and recent approaches from related anticipation tasks. To further promote research into this new task, the code and dataset can be obtained by contacting the first author.

2 RELATED WORK

2.1 Maritime data and its applications
With the introduction of AIS in 2003, new opportunities arose to improve current maritime surveillance systems and shift to more intelligent maritime navigation systems. AIS is a system where vessels broadcast their current location, course and speed to surrounding vessels. Coordinates from this system has been at the core of numerous applications which can be categorized in four parts as described in [42]:

- Anomaly detection, which aims to identify suspicious behaviour of target ships which could cause hazardous situations. Early anomaly detection models include trajectory clustering methods [34] and Gaussian processes [47] while recent implementations also feature deep learning methods [26–28].
- Trajectory prediction, where future position of a target ship is predicted. Applications of this range from early physical models [4] to learning methods such as Kalman Filters [32] and hybrids [31].
- Collision avoidance, where the risk of a collision is determined by the ship domain (safe zone) which can be static [45] or dynamic [40] which can be used for a collision assessment system [17].
- Path planning, a save path after a hazardous situation is observed with graph methods [9] or evolutionary algorithms which evolve new paths under certain constraints [37].

All aforementioned applications are developed with AIS data streams alone and are primarily used for ocean traffic. Other maritime systems rely on radar data collected on the own ship. Automatic Radar Plotting Aid [38] is a system which automatically identifies and tracks moving parts on the radar. Despite advancements in the field [24, 25], classifying these moving parts remaining challenging. The addition of AIS to radar data has shown to be vital for the classification of vessels [18]. The integration of AIS and radar data has also been investigated to classify collision risk [7] and to perform path planning [41], where the latter additionally uses ENC. In this work, we take inspiration from the complementary nature of all three sources to tackle the problem of anticipation of future ship trajectories.

2.2 Trajectory prediction
Driven by application in autonomous driving, trajectory prediction in deep networks has recently gained traction as research problem. By design, autonomous systems require anticipation to make them safe to use. Future trajectory prediction is commonly addressed as a single-modal problem. Social-LSTM for example performs human trajectory prediction from coordinates only by introducing a social pooling to facilitate interaction with neighbouring trajectories [1]. Social pooling has similarly been adopted in follow-up works, resulting in state-of-the-art performance [2, 10, 14, 44]. Such approaches do not model the interactions a trajectory can have with the environment. Several works have therefore integrated such interactions using images and occupancy grid maps [20, 36, 43, 50]. These models are commonly evaluated on datasets with static scenes. Other methods consider a moving camera either in an on-board ego fashion [5, 51] or with a bird-eye view [16, 46] to predict pedestrian or vehicle trajectories. We show that in a maritime context, coordinates alone do not provide the full picture. We propose a new network to combine modalities specific to ships, namely coordinates, radar, and segmentation charts.

3 TRAJECTORY PREDICTION NETWORK

3.1 Problem description
In this paper, we aim to solve the problem of trajectory prediction of target ships on the water. We are given a dataset \( X = \{ (p^{(i)}, R^{(i)}, C^{(i)}, y^{(i)}) \}_{i=1}^{K} \) with K samples. Each sample consists of a historical trajectory \( p \), a sequence of radar images \( R \), a sequence of chart images \( C \) and a future trajectory \( y \). A historical trajectory \( p \) of the last \( N \) timestamps comprises of polar coordinates of a target ship at any of those timestamps such that \( p = \{ p_1, \ldots, p_N \} \in \mathbb{R}^{N \times 2} \). These polar coordinates relate to the current speed and course of a target ship (relative to the true north), where at timestamp \( i \) the speed if given by \( p_{1,i} \) and the course by \( p_{2,i} \). The historical trajectories make use of polar coordinates as they give more explicit information about the motion of a target ship compared to Cartesian coordinates. The radar sequence \( R = \{ r_1, \ldots, r_N \} \) and chart sequence \( C = \{ c_1, \ldots, c_N \} \) are both sequences of images giving information of the surroundings with identical dimension such that \( R, C \in \mathbb{R}^{N \times W \times H} \). A future trajectory \( y \) of the upcoming M timestamps comprises of the Cartesian coordinates of a target ship at any of those timestamps such that \( y = \{ y_1, \ldots, y_M \} \in \mathbb{R}^{M \times 2} \). The future trajectory consist of Cartesian coordinates as it enables a direct evaluation of the models in terms of meters.

We seek to optimize a model which takes as input past trajectory \( p \) and radar sequence \( R \) and predicts a future trajectory \( y \). Since chart information might not always be available in practice, our approach does not require the chart sequence \( C \) during inference, so that the model only uses data captured in present time. This way the model can still function in uncharted areas. To evaluate the performance of our approach, a second dataset \( T = \{ (p^{(i)}, R^{(i)}, y^{(i)}) \}_{i=1}^{L} \) of L samples is used.

3.2 Network modules
To solve the problem of trajectory prediction, we propose a novel multi-task sequence-to-sequence model implemented as a recurrent convolutional neural network. Our approach can be divided into three modules. Below, we outline the different components, how they are integrated in a unified network, and how we optimize this network. An overview of our approach is shown in Figure 2.

Convolutional module. We first combine each radar image \( R_i \in \mathbb{R}^{W \times H} \) with its corresponding coordinate \( p_i \in \mathbb{R}^2 \) by transform the radar image into radar features \( x_i \in \mathbb{R}^K \). We do this with the convolutional module which is a Convolutional Neural Network.
which takes as input a sequence of historical coordinates and radar recurrent module we incorporate a convolutional layers are designed in the same way as the ob-olutional layers and concatenates intermediate feature maps of the encoder to the decoder. The segmentation module is a Semantic Segmentation Network. This module out-
sults are used recurrently to predict future coordinates.

Recurrent module. To make the approach temporally dynamic, we incorporate a recurrent module as a Recurrent Neural Network which takes as input a sequence of historical coordinates and radar images. As recurrent cells we use Long Short-Term Memory cells which has had a wide range of success in sequence-to-sequence tasks in natural language processing [8, 13]. The recurrent module is divided in an encoder and a decoder, where the decoder produces the predicted future trajectory. The result is a sequence-to-sequence model, where sequences of combined coordinate and radar repre-
sentations are used recurrently to predict future coordinates.

Segmentation module. Lastly, we incorporate a segmentation module in a Semantic Segmentation Network. This module out-
puts a binary segmentation mask of the radar images. The models uses parameter sharing with the convolutional layers in the convolutional module, after which the upsample layers output the binary segmentation mask. The upsample layers are designed with the same principles as used in U-Net [35] which uses transposed con-
volutional layers and concatenates intermediate feature maps of the encoder to the decoder. The segmentation module introduces an auxiliary task for the network to learn transforming it to a multitask-learning network.

Multi-task optimization. For the main trajectory prediction task, we use the mean squared error loss where the error is the Euclidean distance. For a training example $i$, the trajectory loss is defined for $M$ timestamps as:

$$L_y = \frac{1}{M} \sum_{i=1}^{M} \|\hat{y}_i - y_i\|^2. \quad (1)$$

For the auxiliary semantic segmentation task, we use a binary cross-entropy loss function. The loss is averaged over the $N$ semantic segmentation outputs produced by the segmentation module. The semantic loss function is therefore as follows:

$$L_C = \frac{1}{N} \sum_{i=1}^{N} BCELoss(\hat{C}_i, C_i). \quad (2)$$

The above loss calculates the pixel-wise binary cross-entropy loss of a predicted binary segmentation mask. We combine the two losses in a combined loss inspired by [19]. We add a learnable noise parameter $\sigma$ for each task resulting in the following loss:

$$L_{total} = \frac{1}{2\sigma_1^2} L_y + \frac{1}{2\sigma_2^2} L_C + \log \sigma_1 \sigma_2. \quad (3)$$

By adding these parameters to the loss function, it is possible to learn the task weights during the optimization process. Using this setup we circumvent costly procedures for finding optimal task weights by learning them on the fly instead.

Uncertainty estimation. Additionally we are also interested in modeling the epistemic uncertainty [11]. It is important to know how certain the model is when using the predictions for naviga-
tional purposes, as wrong predictions could lead to accidents. To model the epistemic uncertainty we extend our implementation to a Bayesian Neural Network. We use Dropout as a Bayesian ap-
proximation as proposed in [12] to approximate the posterior. This method uses the dropout layers in a neural network during inference to sample multiple outcomes and uses these outcomes to quantify the uncertainty of the model.

3.3 Determining the point of view

The origin and orientation of the coordinate systems used for the historical (polar) trajectory $\rho$ and future trajectory $y$ dictate how these trajectories are represented. The historical trajectory uses the current location of the target ship as origin and the course is relative to the true north. In its raw form, the future trajectory are derived from GPS coordinates which have null island as origin and also have the true north as reference orientation. For the model to perform well it needs to learn the semantics of these trajectories, such as “going straight”, “left turn” or “right turn”. With the origin at null island, two future trajectories which are semantically the same are expressed in different coordinates if they occur on different parts of a river. The same problem arises for the orientation towards the true north, where two semantically equal trajectories are represented differently when going in the opposite directions. This complicates the problem as the model has to learn the same semantics at different parts of the river.

To resolve the orientation problem, we use the last known course of the target ship $\rho_{N,2}$ as the reference orientation for the historical
trajectory. Furthermore, we use the last known Cartesian coordinate $c_N$ as reference origin for the future trajectory. As a result, regardless of where the target ship is in the world, future Cartesian coordinates are always in front of the target ship. More formally we apply the following transformations on the trajectories $\rho$ and $\gamma$:

$$\rho^* = \{R_{\rho u, x}(p_i)\}_{i=1}^N,$$

$$\gamma^* = \{R_{\rho y, x}(y_i - c_N)\}_{i=1}^M,$$

where $R_x(\cdot)$ is used to denote rotating a point $x$ degrees around the origin. After these transformation semantically similar trajectories will be represented in the same way. These transformations effectively change the point of view to that of the target ship associated with the trajectory.

The same change in point of view is applied to the radar images. The last known coordinate of the target ship is used as the centre of the image, and it is rotated to the last known orientation of the target ship. Furthermore, instead of using the whole radar image we make a crop to only show the relevant information for the target ship. This crop only shows the part of the river the target ship could reach in two minutes time with a speed of 15 knots, which is about the maximum speed we have observed. The procedure removes a lot of unnecessary information about regions which are impossible to reach by either being too far away or not having water. The transformation process is visualized in figure 3. This transformation effectively creates a view of the surroundings from the target ship, using the radar of the own ship.

4 EXPERIMENTAL SETUP

4.1 Dataset

For this paper a new dataset was created by collecting all available sensor data from a container ship used for inland shipping. A total of 44 hours of sailing is used in this dataset. This sailing took place in March 2018. The final dataset consists of 620 tracked target ships. The dataset contains the following three data sources:

AIS messages. Ships are required by law to send AIS messages to notify other ships of their current GPS location, course and speed. We project the GPS locations to a flat surface with the European cartographic standard (EPSG: 3035) where we can easily calculate distances in terms of meters. The transmission rate of AIS messages can vary depending on the speed of the target ships and whether they are changing course or not. Only AIS messages from target ships who are currently sailing and are in range of the radar are included.

Radar images. Marine radars are used to detect target ships and land obstacles in the surroundings of the own ship. Each 2.5 seconds such a radar image is created. The range of the radar can be varied but only images with a radius of 1500 meters are included (most commonly used). The images are captured with a resolution of 1800 by 1800 pixels. The radar transformations as shown in figure 3 are done by centering and rotating a crop of 650 by 650 pixels to the location and orientation of the target ship. To put more emphasis to the front view we shift the crop 225 pixels in the direction of the target ship. The cropped images are further resized to a resolution of 288 by 288 pixels.

ENC images. Electronic Navigational Charts contain information necessary for safe navigation. In this research we make use of the land and water information. The ENC are maintained by the government and can be downloaded for free from the website of The Dutch Ministry of Infrastructure and Water Management. In this research we use ENC from the Netherlands, Belgium and Germany. We apply the same transformation on the ENC images as on the radar images.

Trajectory preprocessing steps. As the AIS transmission rate and radar capture rate differ the data needs to be synchronized. We do this by interpolating the AIS messages between the radar images. The resulting trajectories are further divided into sub-trajectories to create trainable samples. These sub-trajectories are created with a step size of 3 between consecutive subsets of the trajectories. Subsets of length 55 are used, where the first five timestamps are used as input to the model, and the last 50 as a target making the model predict roughly two minutes in the future. The dataset will be made available to encourage research into multimodal ship trajectory prediction.

4.2 Implementation details

Dataset split. Six hours of sailing is used as a test-set, identical for all implementations. The remaining dataset is split up as 90% train and 10% validation sets. This split is done before the target ship trajectories are divided into sub-trajectories to avoid having consecutive sub-trajectories in different sets. These splits are done times with different random seeds. All inputs and targets normalized to have mean 0 and standard deviation of 1.

Model details. The recurrent module has 2 layers, a hidden dimension of 128 and a dropout rate of 0.5 is used between the layers. The convolutional module consists of the YOLOv3-tiny architecture [33] up until MaxPool is used 5 times followed by fully connected

---

1. https://epsg.io/3035
2. https://vaarweginformatie.nl/frp/main/#/page/infra_enc
layers of size 2048, 512 and 128 neurons respectively. The first two FC layers use a ReLU activation function and dropout layers with a dropout rate of 0.5 and the last FC layer is followed by a Sigmoid activation function. The segmentation module (in particular upsampling layers) are designed with the same principles as U-Net [35] by using transposed convolution layer to upsample from the bottleneck. The models mentioned in this research are all implemented using PyTorch [29]. We will release our code for reproducibility.

Optimization. All the models are optimized with Adam [21] with a learning rate of $1 \times 10^{-3}$ and weight decay [22] of $1 \times 10^{-6}$. The learning rate is halved if no improvement in the validation error is observed for 10 epochs. The models are trained for a total of 100 epochs.

4.3 Evaluation

Absolute Trajectory Error. To evaluate how well a model performs in predicting trajectories we use the Absolute Trajectory Error (ATE) which are commonly used in SLAM systems [39]. The ATE is the Root Mean Squared Error (RMSE) of a trajectory. This is computed by averaging the squared (euclidean) distance between the predicted and the ground truth coordinates at every timestamp in the future trajectory and taking the root. We prefer the use of RMSE over Mean Average Error (MAE) used in other methods as the RMSE proportionally gives larger errors to outliers, in this case large distances. Because the safety of both the own ship and target ship can be compromised by these large errors we prefer a metric which reflects this.

Error @ k. The ATE gives a good indication of how well a model is performing in the form of a single number metric. However, the ATE is sensitive to when the start of the error in the trajectory occurs. For example, when the course of the predicted trajectory is wrong in the beginning of the prediction the error of the trajectory is larger compared to an error in the course later in the trajectory. To get a better understanding of how the error of a trajectory develops we also use the Error @ k (E@k). For each trajectory we compute the error at every five future timestamps.

5 RESULTS

In our experiments, we first perform ablation studies on our own network and the importance of the right viewpoint. Second, we perform a comparative evaluation to several baselines. Third, we perform qualitative and uncertainty analyses to better understand our approach.

5.1 Ablation Studies

Determining the point of view. An important component for trajectory prediction is the point of view for ships. To evaluate its importance, we start from our network using only the coordinates, ignoring the radar stream and segmentation loss. Only the mean squared error loss is used. We compare to two baseline viewpoints:

- None: Not changing the point of view for other ships and thus using raw coordinates.
- Own ship: Using the point of view of the own ship to pre-process the data. We use the same methods as described in

<table>
<thead>
<tr>
<th>Point of view</th>
<th>ATE</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>440203.37 ± 725.00</td>
</tr>
<tr>
<td>Own ship</td>
<td>148.77 ± 3.37</td>
</tr>
<tr>
<td>Target ship</td>
<td>20.22 ± 0.08</td>
</tr>
</tbody>
</table>

Table 1: Comparison of different point of views. Using the point of view of the target ship for which the future prediction is performed results in the lowest error.

<table>
<thead>
<tr>
<th>Coordinates</th>
<th>Radar Images</th>
<th>Chart Images</th>
<th>ATE</th>
</tr>
</thead>
<tbody>
<tr>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>20.22 ± 0.08</td>
</tr>
<tr>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>76.81 ± 3.21</td>
</tr>
<tr>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>17.13 ± 0.09</td>
</tr>
<tr>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>64.01 ± 2.85</td>
</tr>
</tbody>
</table>

Table 2: Combining data sources. The lowest ATE is in bold, showing that the configuration with all the data sources performs best. Furthermore the configurations which use coordinates outperform other configurations.

Section 3.3 to achieve this change in point of view to the own ship.

The results of these experiments are shown in Table 1. Using no point of view at all gives poor results. This is to be expected as without any point of view each trajectory is completely unique, which makes it extremely difficult for the model to find patterns and generalize. Using the own ship as point of view already results in highly reduced error rates, since as a lot semantically similar trajectories are represented in a similar fashion. Using the point of view of the target ship best aligns ships in different locations and directions, resulting in the lower error. We will therefore use the target ship point of view throughout the rest of the experiments.

Combining data sources. To evaluate the complementary nature of the different data sources, we evaluate the error rates when coordinates, radar images, and chart images interact in our Trajectory Prediction Network. We investigate multiple configurations. Note that the chart images are used as supervision for an auxiliary segmentation task, which means that they can only be used in combination with the radar images. The results of the investigation is shown in Table 2. The Table shows that the combination of all data sources performs best compared to other combinations. The use of coordinates as one of the signals is vital to obtain good scores, highlighting that the historical coordinates are very important in the task of trajectory prediction.

Besides combining data sources during training, the models can also be trained separately and their prediction can be combined. We test this by averaging the predictions of the separately trained models. The possible configurations for this are combining a coordinate trained model with either a radar trained model or a radar and chart trained model. The ATE of these combinations are 42.39 and 36.69 respectively. This is worse than using the coordinate trained model on its own, showing that simply combining data sources is
5.2 Comparative Evaluation

Next, we compare our results to external baselines. We compare to two deterministic baselines and the Social Ways model, state-of-the-art in trajectory prediction of pedestrians. The reported ATE for the Social Ways model is the last checkpoint before the model diverged at epoch 65. Our approach outperforms all baselines, showing its effectiveness for ship trajectory prediction.

### Table 3: Comparison of our approach with baselines

<table>
<thead>
<tr>
<th>Methods</th>
<th>ATE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Social Ways [2]</td>
<td>295.72</td>
</tr>
<tr>
<td>Average course and speed</td>
<td>33.59</td>
</tr>
<tr>
<td>Average displacement</td>
<td>24.98</td>
</tr>
<tr>
<td>Our approach</td>
<td><strong>17.13</strong></td>
</tr>
</tbody>
</table>

Intuitively, the first baseline simply uses the average spatial difference from the past coordinate observations, while the second baseline does the same for direction and speed. Hence the second baseline is a polar version of the first baseline.

**State-of-the-art baseline.** We also compare our method with Social Ways as proposed in [2]. Social Ways uses GANs to generate pedestrian trajectories and has state-of-the-art performance in the trajectory prediction dataset BIWI Walking Pedestrians [30]. Social Ways uses all trajectories in present time together with social features between these trajectories to jointly generate new trajectories. The discriminator then predicts if the generated trajectories are fake or not. The BIWI Walking Pedestrians dataset is created with a still camera observing multiple pedestrians over time. As Social Ways predicts all trajectories simultaneously, we use the point of view of the own ship for the trajectories, instead of either of the target ships. This creates similar circumstances to the BIWI dataset by simulating a still camera pointed at the own ship in bird-eye view.

**Results.** The results of the comparative evaluation are shown in Table 3. The Table shows that our proposed approach improves upon the baselines. In the context of ship trajectory prediction, training the GANs as outlined by social ways did not result in converging models. The training proved to be unstable, resulting in divergence for the hyperparameter settings as described in [2], as well as an additional grid search of hyperparameters. Social Ways achieves state-of-the-art results in pedestrian trajectory prediction in a still camera setting. However, our proposed dataset uses the point of view of the own ship. This results in a moving camera and thus a changing environment. Furthermore, we evaluate our models with an input length of five and output length of fifty, while Social Ways uses an input length of eight and output length of twelve. Such factors likely contribute to the inability to converge training. Overall, the results show that models optimized for the trajectory prediction for pedestrians are not trivially transferable to trajectory prediction for ships. The heuristic baselines show that the problem can also not be solved by interpolation, a multimodal network solution is needed to generalize better.

**Architectural designs.** We have also investigated the effect of different architectures for the convolutions and segmentation modules. Using a ResNet-34 [15] as convolutional network and a SegNet [3] as the segmentation decoder resulted in an ATE of 17.28 compared to 17.13. The fact these numbers are close indicate that the specific choice of architecture is not a vital component, as long as they are combined adequately.

**Heuristic baselines.** We use two deterministic baselines to further assess the performance of the models. The baselines are extrapolation methods and thus need no training data. The baselines used in this research are as follows:

- **Average Displacement**: extrapolate with average $\Delta x$ and $\Delta y$ of consecutive timestamps.
- **Average Course and Speed**: extrapolate with average course and speed of consecutive timestamps.

![Figure 4: Plots showing E@k with a one standard deviation confidence interval of our approach (left) and the E@k for the x-coordinates (middle) and y-coordinates (right). The super-linear growth of error in course (middle) combined with the linear growth of error in speed (right) results in an overall super-linear growth of error in distance (left).](image-url)
5.3 Analysis

Error rates over time. Figure 4 shows the E@k for the predicted coordinates in the trajectories, and we also split the errors for the x-coordinates and y-coordinates. Because we use the point of view of the target ship, the E@k for the x-coordinates can be interpreted as the error in predicting the course of the target ship. Similarly, the E@k for the y-coordinates can be interpreted as the error in predicting the speed of the target ship. The left figure shows that the total E@k increases in a super-linear speed. This can be further explained by decomposing the error between the two coordinates as shown in the middle and right figures. The E@k for the x-coordinates also increases in super-linear speed. A possible explanation for this is that the course can be changing (increasing or decreasing) or stable. If the magnitude of the estimation of the change in course is wrong, the error starts accumulating over time resulting in a super-linear growth. The error for the y-coordinates in the right figure grows in a linear speed. Opposite to the course, the speed of the target ships are very stable. Ships do not abruptly stop or accelerate while sailing on a river. Therefore almost all target ships have a close to constant speed while passing the own ship. The linear buildup in the E@k suggests that the predicted coordinates of the model indeed have equal space between them (constant speed). This speed is however not always the correct speed, resulting in the shown error. Together, these two phenomena cause the super-linear growth in the ME@k shown in the left figure.

Measuring uncertainty. We use dropout during inference to make a Bayesian approximation of the posterior. Using this technique, we can sample multiple predictions to estimate how certain our model is about them. Knowledge of this uncertainty can be important in situations which require caution (such as maneuvering a large container ship). Ideally, the model shows low uncertainty with good predictions (low error) and high uncertainty with bad predictions.

To test this, we sample one hundred predictions from the model and compute the uncertainty at different timestamps in the future. We compute the uncertainty by averaging separate standard deviations of the x and y-coordinates. We also show this for the Coordinate Network and the Radar network, and compare the uncertainty with the ME@k. The results are shown in Figure 5.

The figures show that, indeed, the uncertainty increases with the E@k. However, when only using the coordinate data the uncertainty grows linear and not super-linear as the E@k does. An explanation why the uncertainty better aligns with the E@k curve by adding radar and chart images is that the convolutional module contains dropout layers between the dense layers. This leads to more randomness in the samples and possibly a better posterior approximation.

Success and failure cases. Figure 6 shows several success and failure cases of our approach. The three left columns show various scenarios at each row for different data combinations at each column. It immediately becomes clear that by only using coordinates, without the context provided by the radar images, the model fails to follow these curves. This is no surprise as the model does not have the context to foresee this curve. Adding radar images and chart images indeed show significant improvements by following the curve and also show more uncertainty towards the end of the trajectory, where the chart 13.21 seem to provide some extra uncertainty. This is also reflected in figure 5, which shows that using all data gives the most uncertainty.

The right column shows three different scenarios at which our approach (and all other data combinations) fails. These scenarios cover inserting, overtaking and mooring in the river respectively. All these scenarios are very uncommon in the dataset and involve sharp steering movements. In general, a ship has slow parabolic type of trajectories. Furthermore, the second and third scenario also occurred while limited vision was available due to power lines crossing the river and a large distance respectively. The large distance makes that the resulting radar image with the point of view of the target ship has low detail. It is clear that our approach did not properly learn to handle these scenarios. Fortunately, our approach also shows more uncertainty, especially in the last two scenarios. This is desirable as future applications can reason with this uncertainty, together with the predictions.

Figure 5: Comparison between error and uncertainty. The E@k (Top) and the computed uncertainty (bottom) show similar trends when the radar and chart data is included.

![Error@k](image1)

(a) Error@k.

![Computed uncertainty](image2)

(b) Computed uncertainty.
6 CONCLUSION

In this work, we address the problem of future anticipation of ship trajectories from a multimodal perspective. We propose a novel multi-task sequence-to-sequence network, called Trajectory Prediction Network, which combines past coordinate locations and radar images to predict future trajectories. On top, we add a loss for predicting pixel-wise binary segmentations from water or land. To evaluate the proposed network, we introduce a new dataset. The dataset is created by collecting sensor data from an inland container ship. We show that determining the point of view of the trajectories can greatly reduce the complexity of the problem and increase performance. The addition of radar provides environmental context and makes it possible for the model to follow curved river banks. Lastly, using the segmentation masks from charts as training supervision for an auxiliary segmentation task further boosts the performance.

The resulting multi-task sequence-to-sequence network is able to accurately predict long-term future trajectories of inland ships. We show that our model outperforms heuristic baselines and a state-of-the-art method for pedestrian trajectories. We use dropout as a Bayesian approximation to quantify the epistemic uncertainty, which can be helpful for future applications. We observe that our model is able to make good predictions even with poor vision. However, some scenarios remain challenging, such as mooring and overtaking.

To overcome these challenges, a more explicit modeling of the dynamic environment might be needed. Currently, the surroundings of a target ship is simulated with the radar of the own ship. To improve social sailing maneuvers, such as overtaking, the trajectories of nearby ships should be taken into account. This has been proven to work for models which involve a lot of social interactions such as pedestrian trajectory models.

REFERENCES
