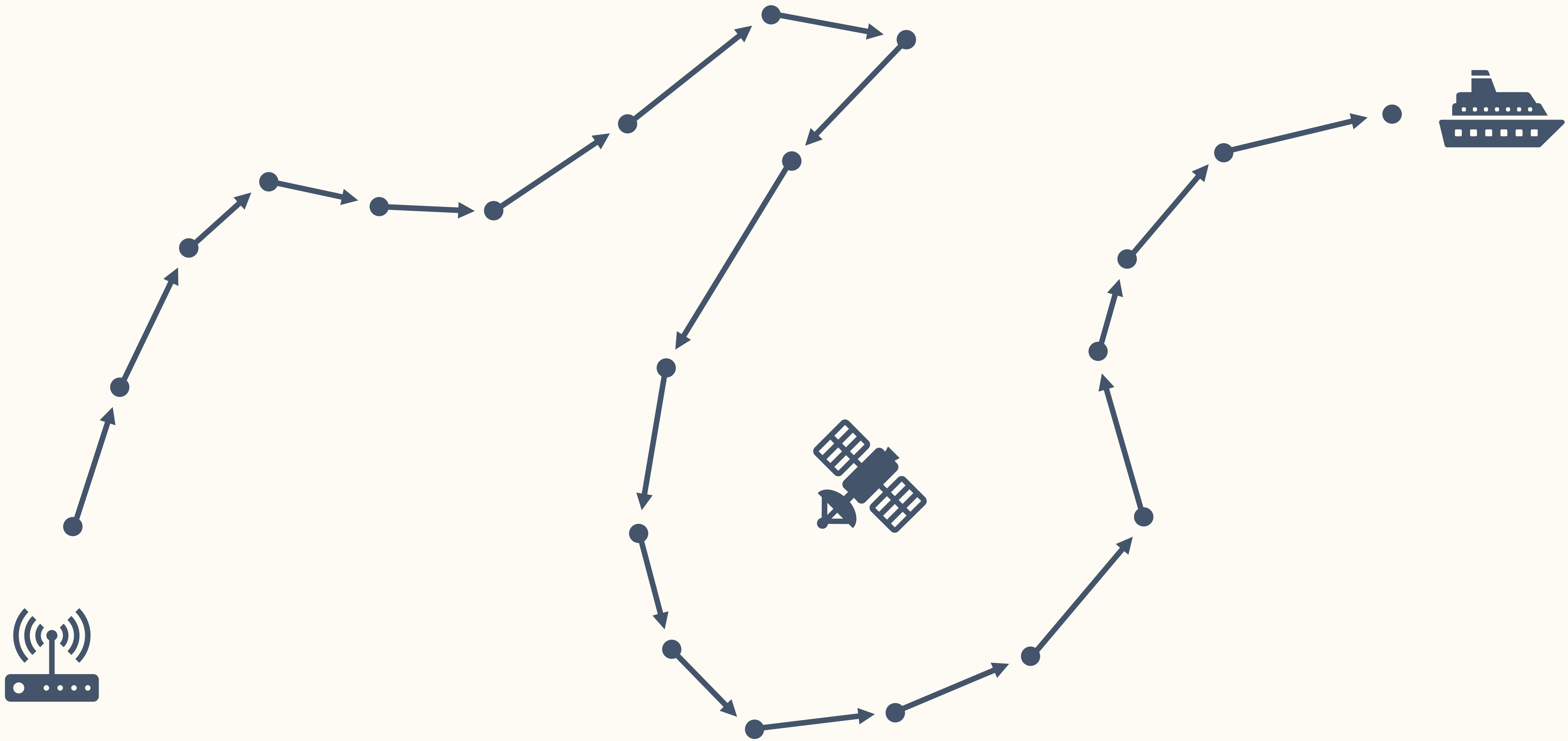
# Using **neural networks** to model the behavior in vessel trajectories

# Programme

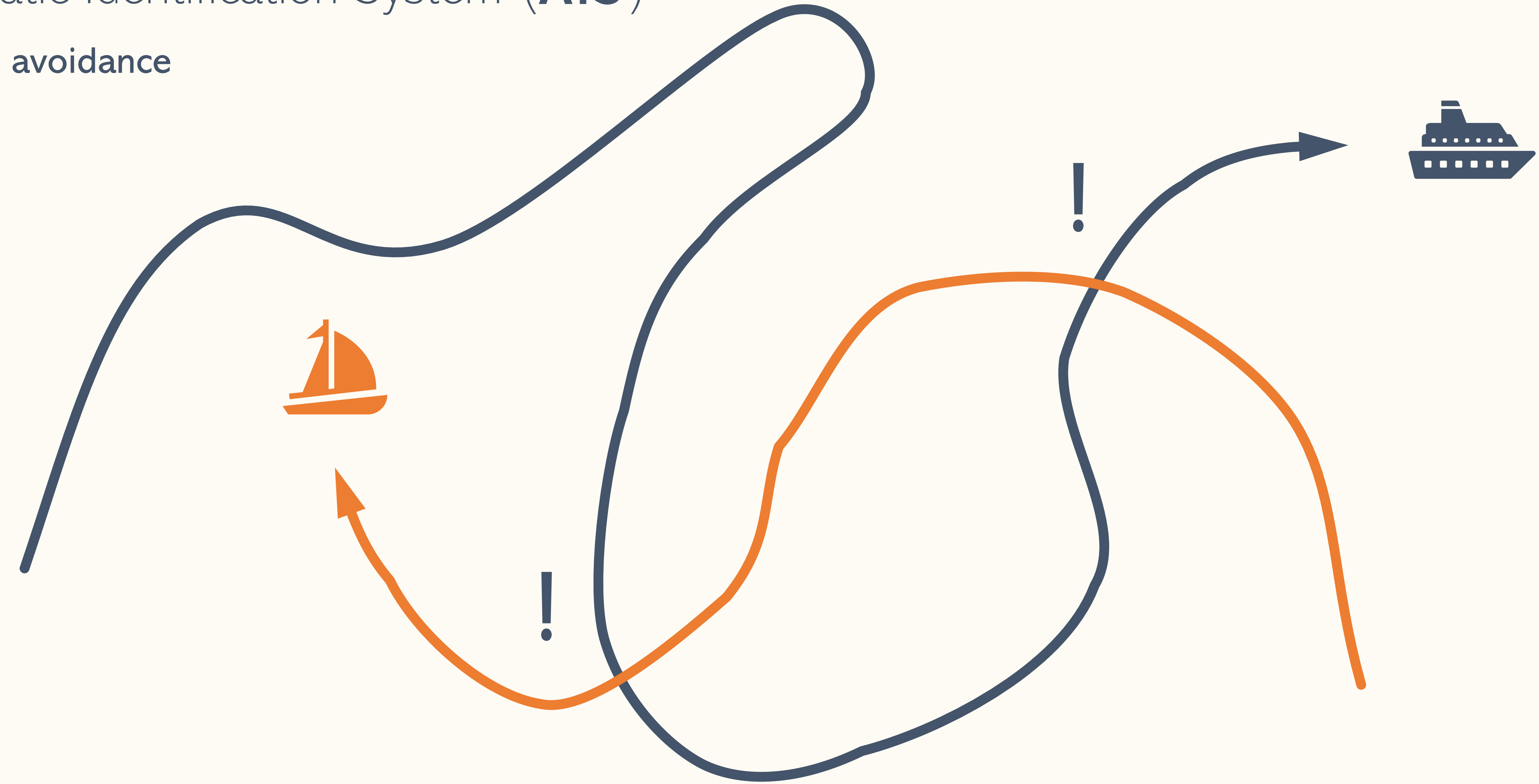
[1] terminology & relevance
[2] research question
[3] experiments & results
[4] conclusions

# Vessel trajectories



# Automatic Identification System (**AIS**)

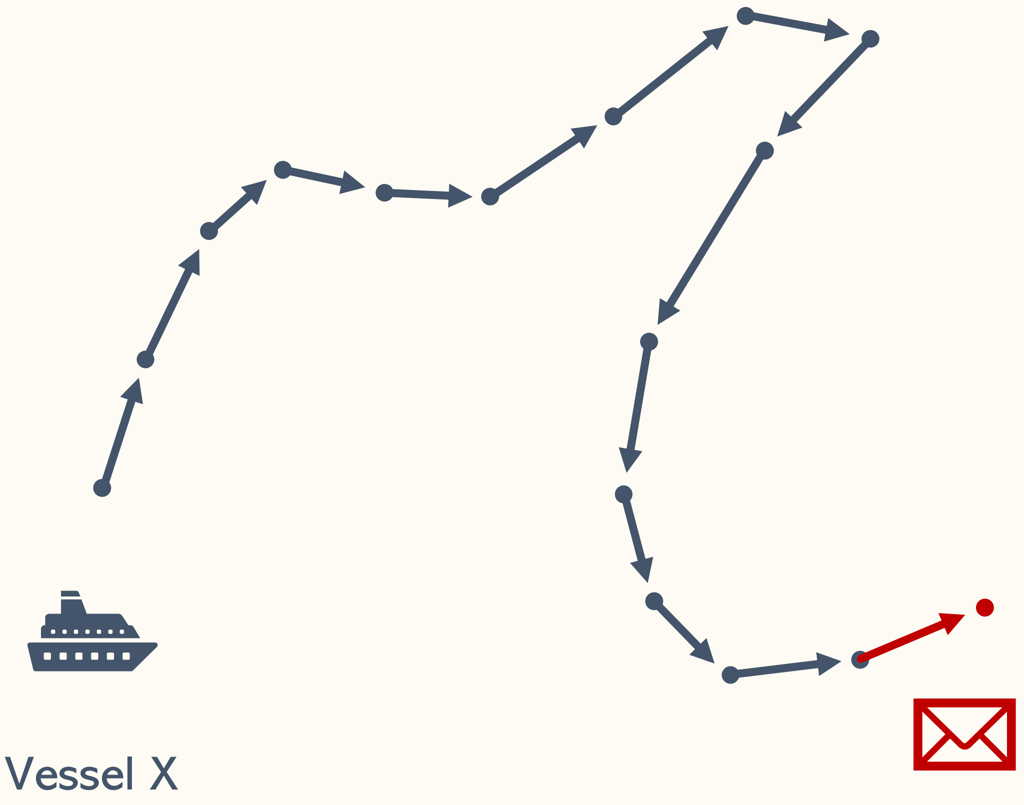
## Collision avoidance



# Automatic Identification System (**AIS**)

- Global
- Laws enforce AIS
- Coverage is good
- Additional information:
  - Longitude
  - Latitude
  - Speed
  - Rate of turn
  - Vessel ID
  - Vessel type
  - Etc.

# Automatic Identification System (**AIS**)

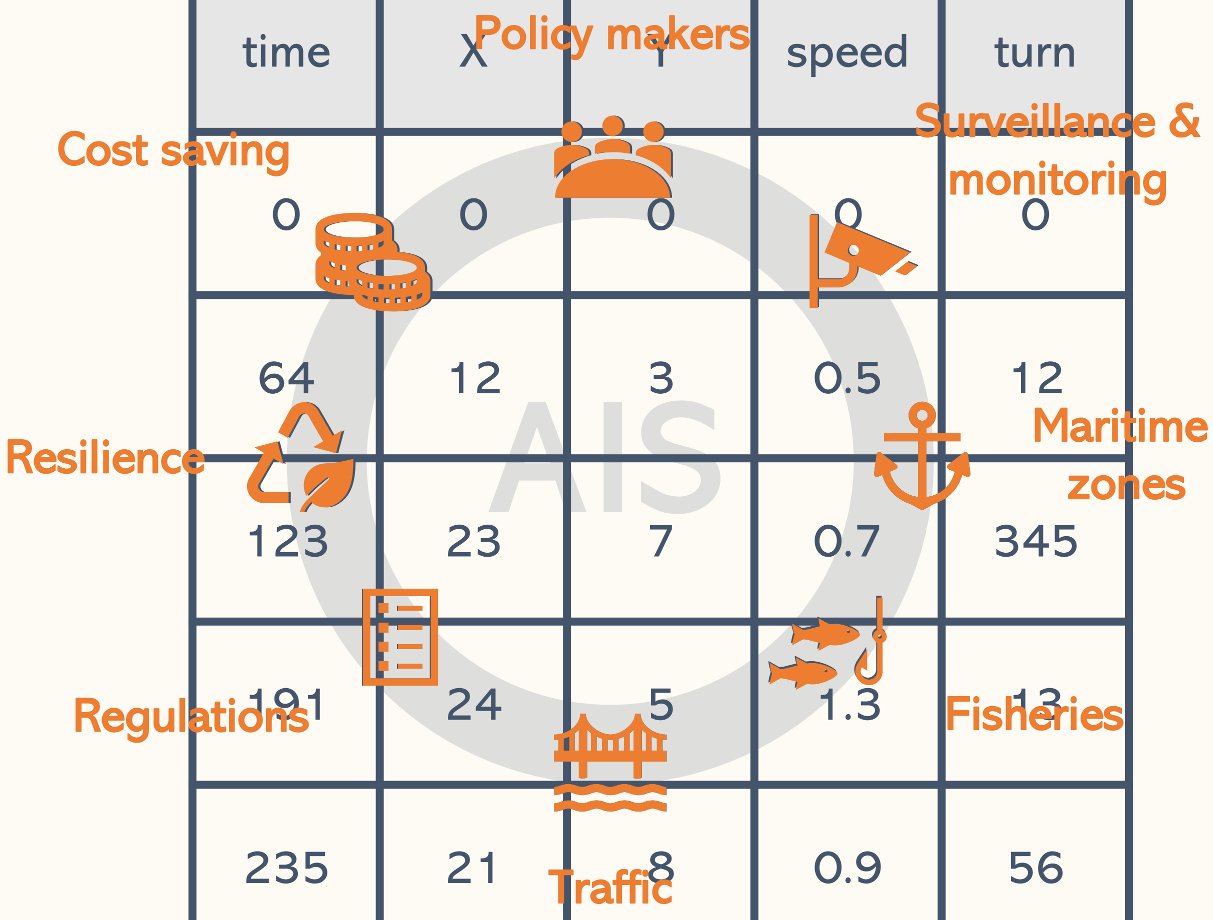



Vessel X

| time | X | Y | speed | turn |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 64 | 12 | 3 | 0.5 | 12 |
| 123 | 23 | 7 | 0.7 | 345 |
| 191 | 24 | 5 | 1.3 | 13 |
| **235** | **21** | **8** | **0.9** | **56** |

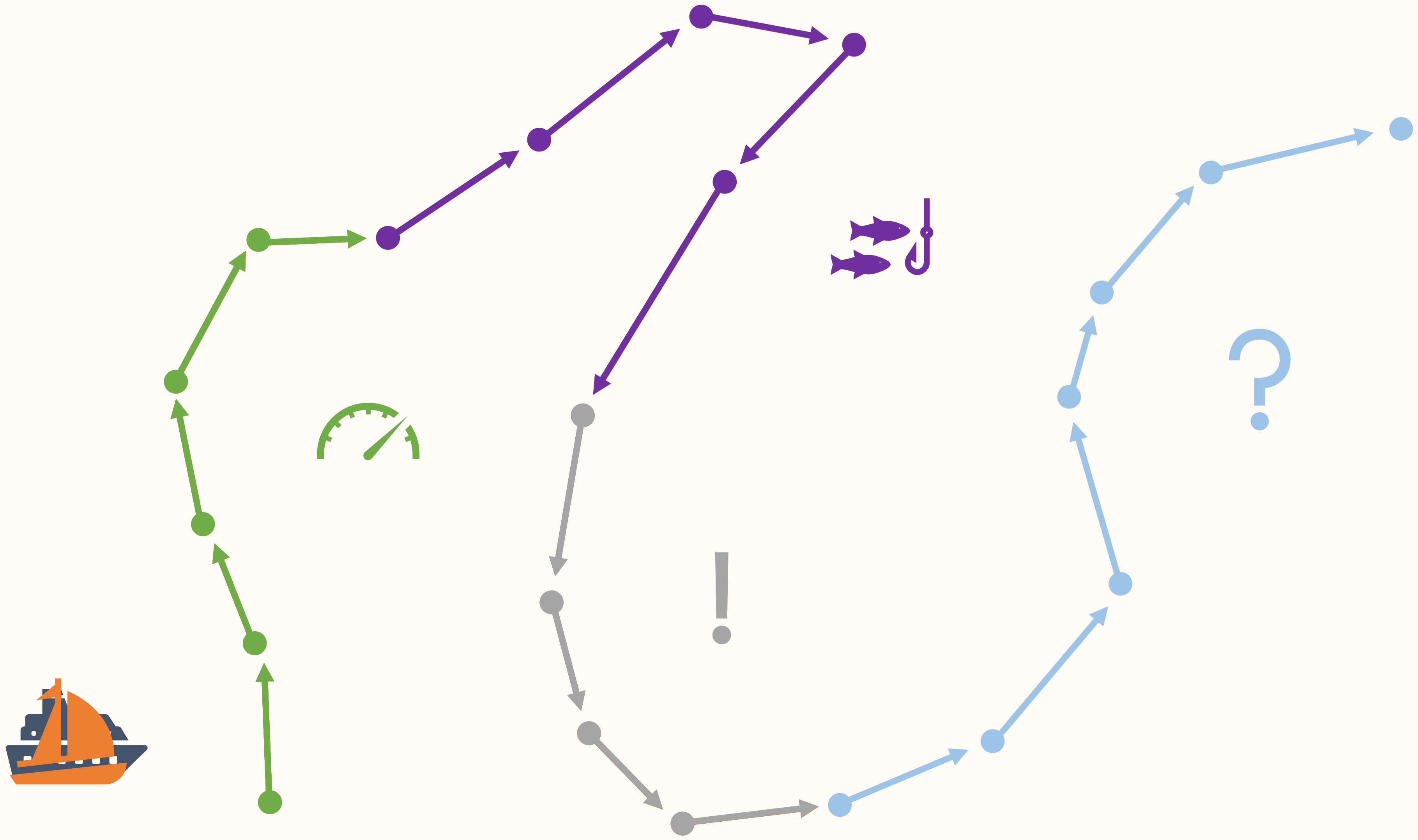
# Why **analyze** this information?

| time | X | Y | speed | turn |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 64 | 12 | 3 | 0.5 | 12 |
| 123 | 23 | 7 | 0.7 | 345 |
| 191 | 24 | 5 | 1.3 | 13 |
| 235 | 21 | 8 | 0.9 | 56 |



# Some examples

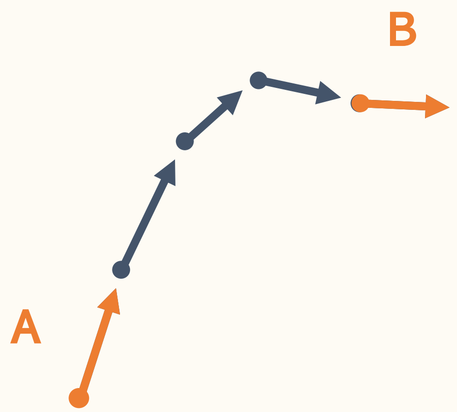
- Classification
- Segmentation
- Forecasting
- …



# **How** to analyze this information?

## We use algorithms

turning right or left?




## Rule-based algorithm

calculate cross product:

$$\mathbf{A} \times \mathbf{B}$$

set rules:

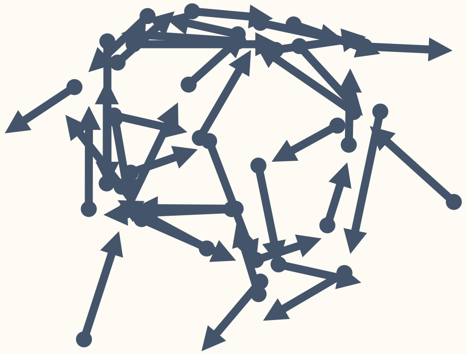
IF $\mathbf{A} \times \mathbf{B} < 0$

THEN turning right

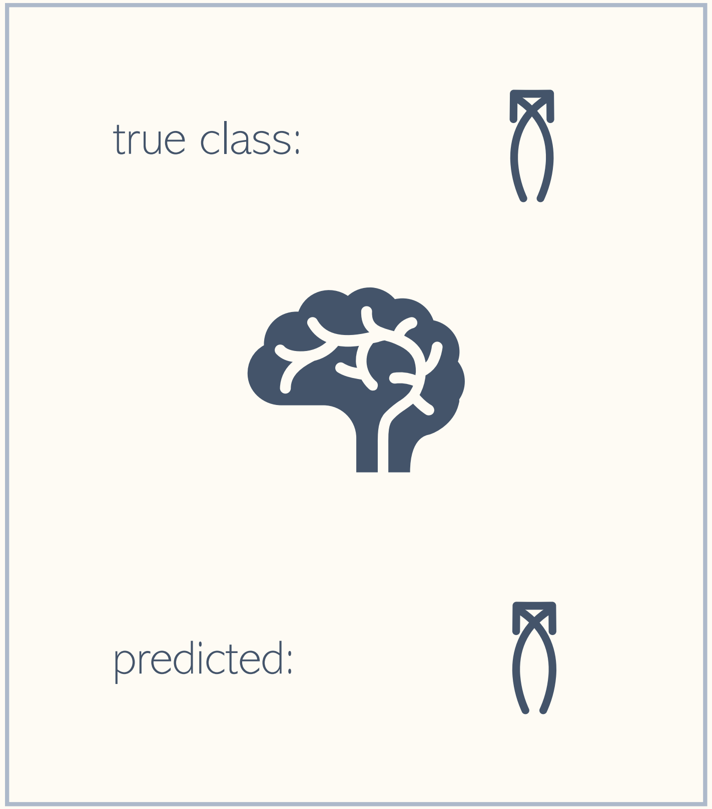
IF ...

# **How** to analyze this information?

## We use algorithms



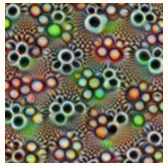
## Deep learning



# **How** to analyze this information?

## We use algorithms

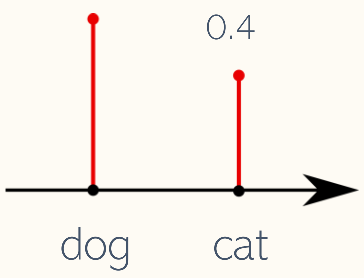
dog or cat?

[1]  →    → **dog**

input | layer 1 | layer 2 | layer 3 | output

$v_{in} = \langle v_1, v_2, \ldots, v_n \rangle \rightarrow v_1 = \boldsymbol{f_1}(v_{in}) \rightarrow v_2 = \boldsymbol{f_2}(v_1) \rightarrow v_3 = \boldsymbol{f_3}(v_2) \rightarrow$

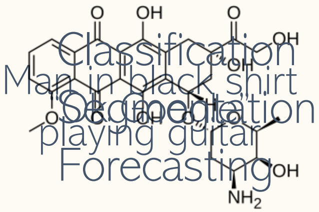
$\sigma(Mv + b)$



# **How** to analyze this information?

## We use algorithms

Neural network (**NN**)




- Classification
- Segmentation
- Forecasting

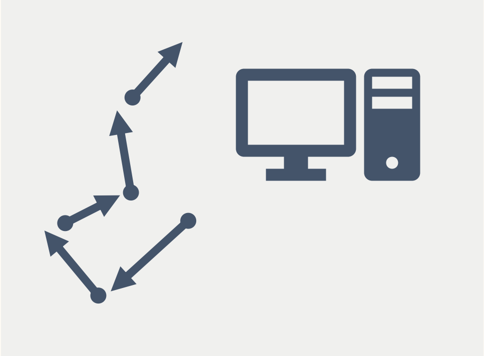
Man in black shirt playing guitar

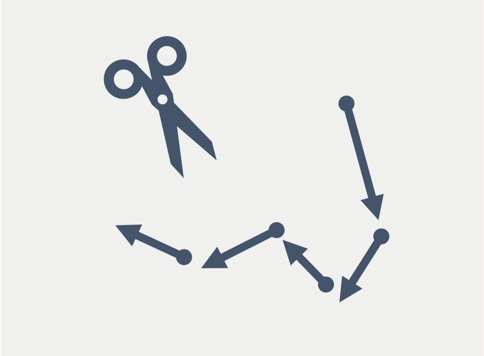
[2] [3] [4]

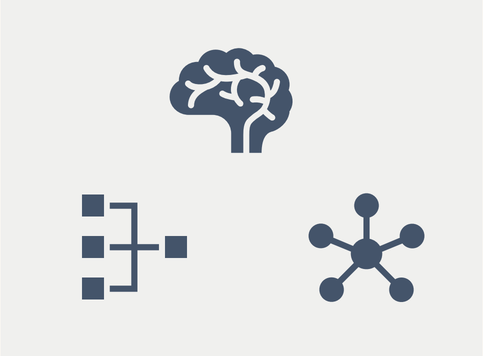
# Research question

To what extent can **neural networks** contribute to modeling the behavior in vessel trajectories?

- What is a good way to represent a trajectory for a neural network?
- How should the trajectories be segmented?
- What type of neural network is fit to do this?
- How well does it work?







# Data set & tools

### data

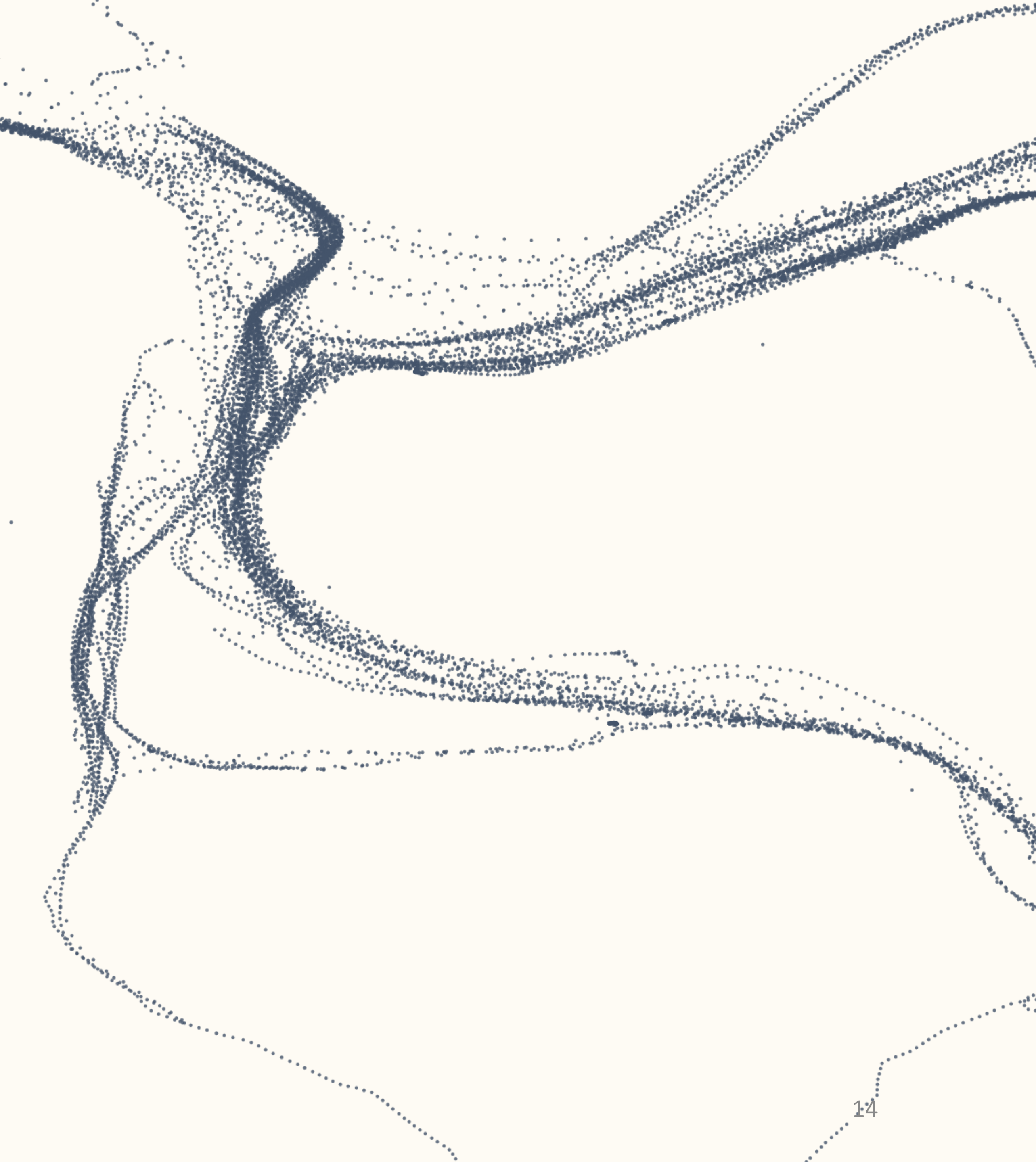
- global AIS 2016 - 2017
- coastal receivers
- 1 billion messages

### hardware

- remote server
- database system

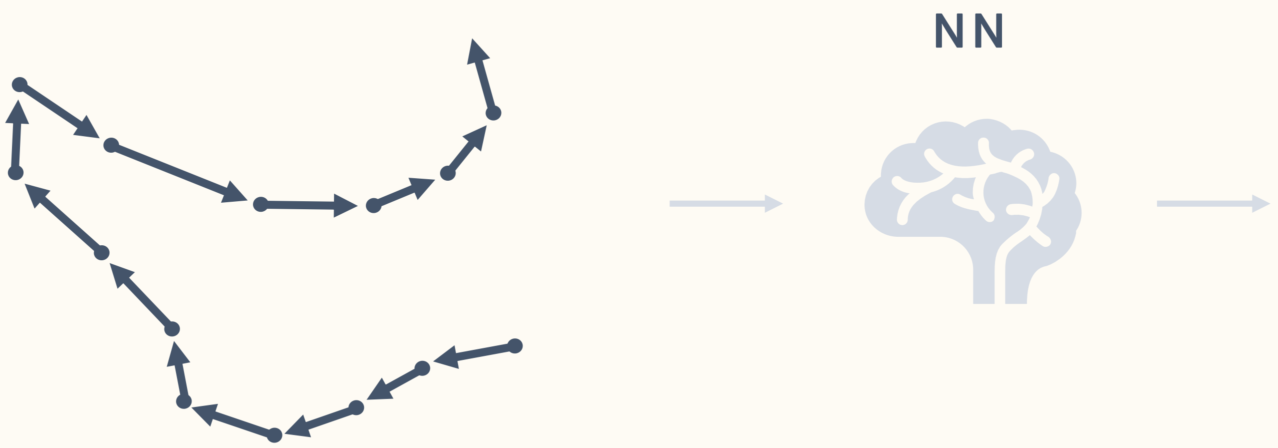
### software

- Python – **programming**
- PostgreSQL – **database system**
- Keras – **deep learning**



# Some **experiments**

1. Classifying vessel type based on trajectory




| class | vessel type |
|---|---|
| 1 | Cargo |
| 2 | Tanker |
| 3 | Fishing |
| 4 | Tug |
| 5 | Passenger |

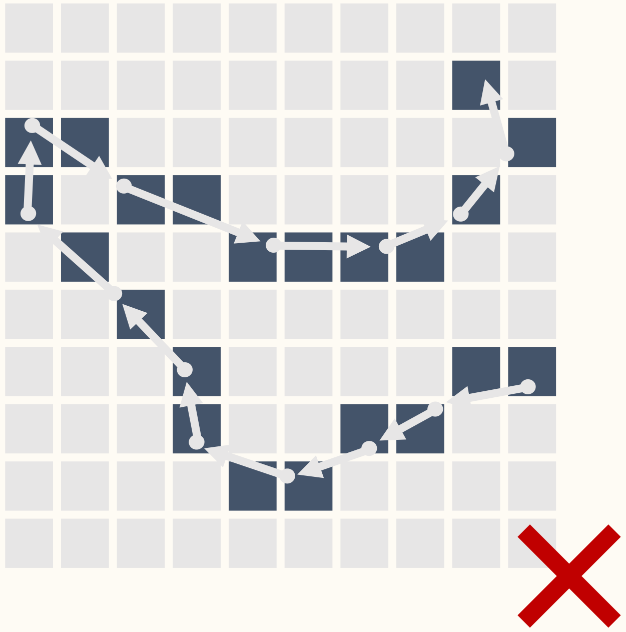
- Vessel type is included in AIS messages
- These types are most prevalent
- **Objective**: find out if NN can learn from movement data
- What if wanted classes are not included in AIS?
    - Use small set to train NN

# Data **representation**

1. Classifying vessel type based on trajectory

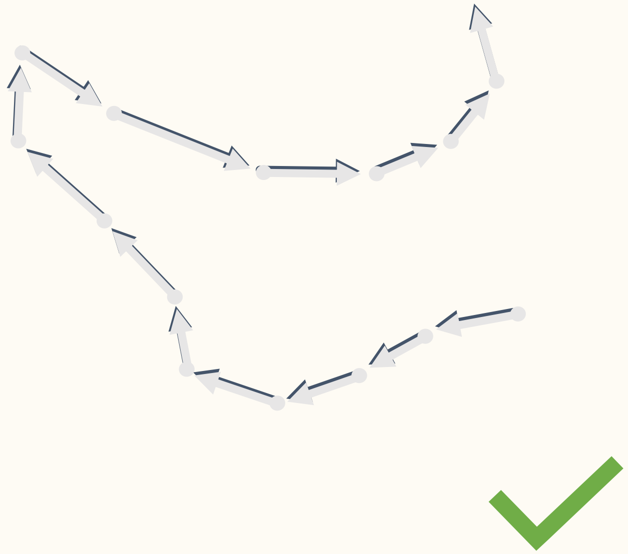
**Raw data**

- coordinates
- attributes (speed, turn)
- bias included



**Image**

- coordinates as pixels
- attributes in channels
- waste of space



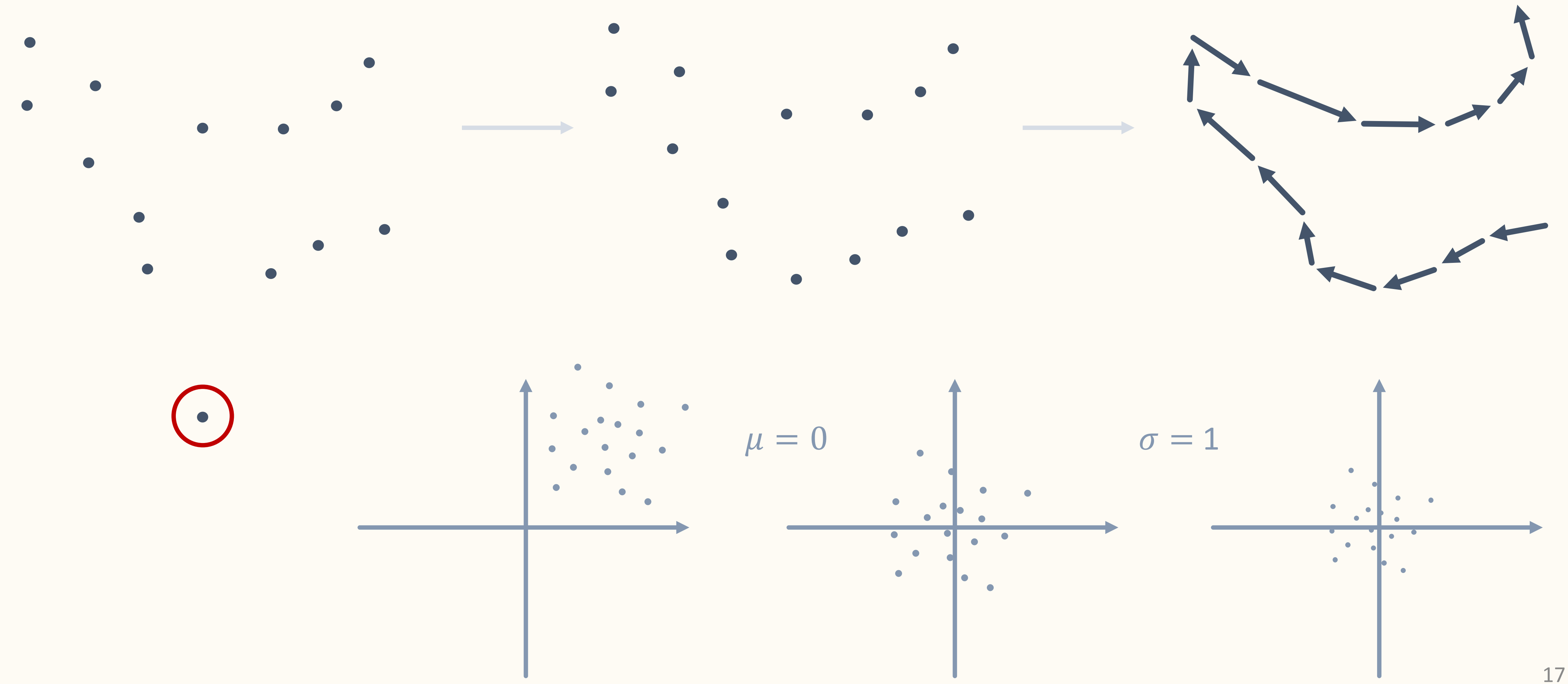
**Sequence**

- relative coordinates
- attributes (speed, turn)
- good solution

# Data **preprocessing**

**1.** Classifying vessel type based on trajectory

1. remove outliers
2. extract relative coordinates
3. standardization



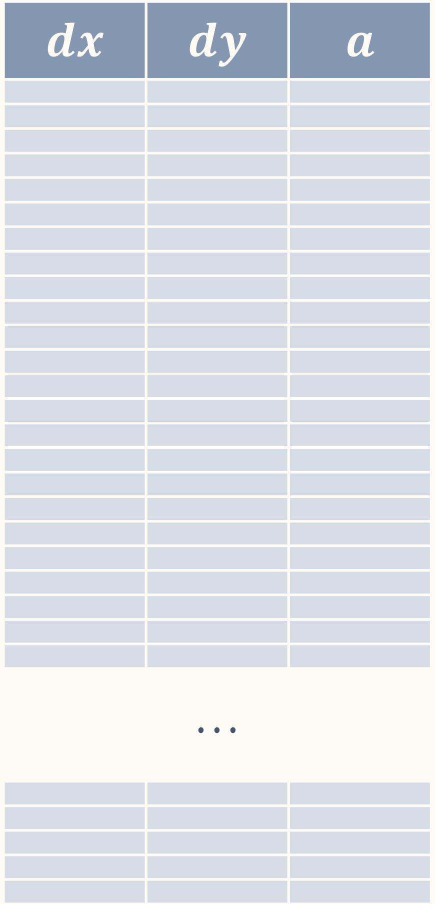
# Data set **compilation**

true class: **cargo**

predicted: **tug**

1. Classifying vessel type based on trajectory

**Trajectory per vessel**




- Sample length of 2,000
- Retrieve several samples based on N
- Non overlapping
- Random position

$L = 2{,}000$

- Equal amount of samples per class
- Distinct vessels per partition

$N$ *to* $150{,}000$

**partition: ~55,000 samples**

**70 % training**

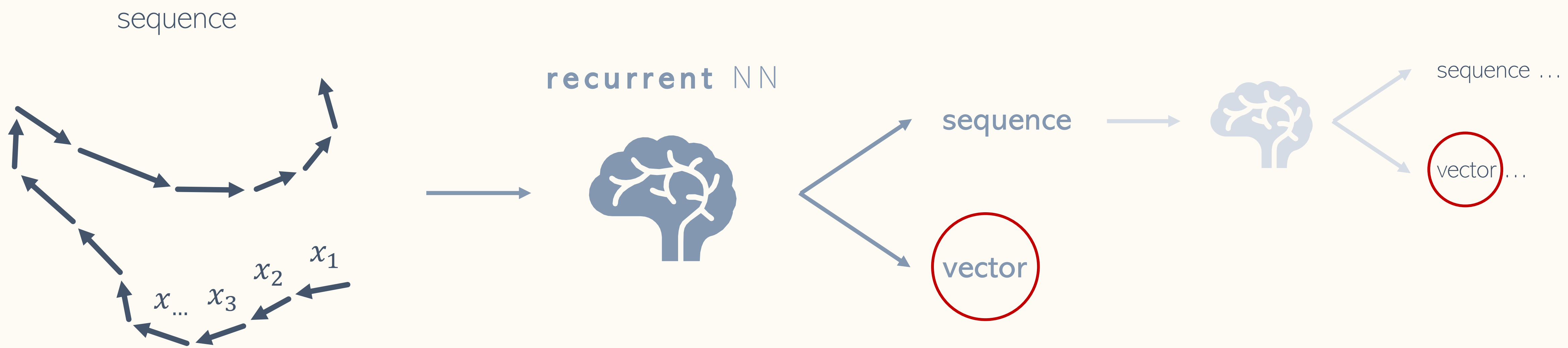
| | |
|---|---|
| Cargo | ~7,700 |
| Tanker | ~7,700 |
| Fishing | ~7,700 |
| Tug | ~7,700 |
| Passenger | ~7,700 |

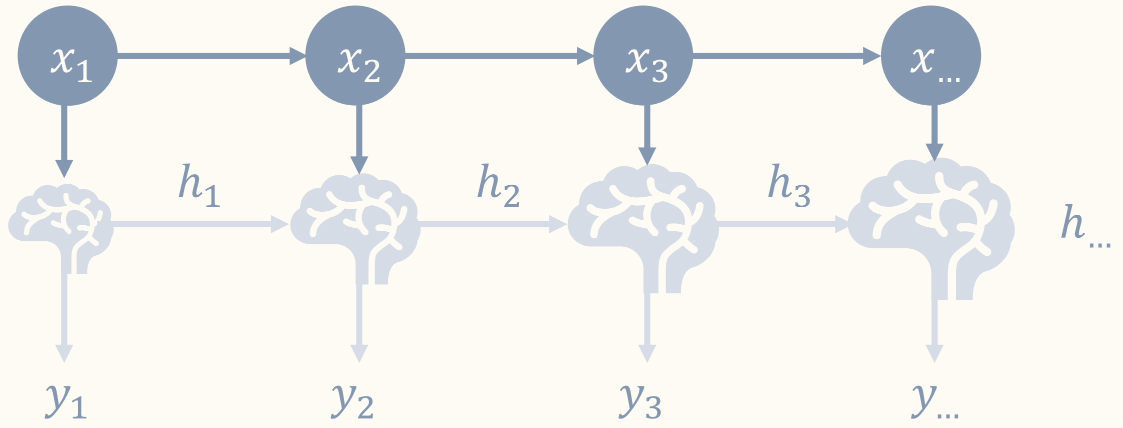
**15 % validation**

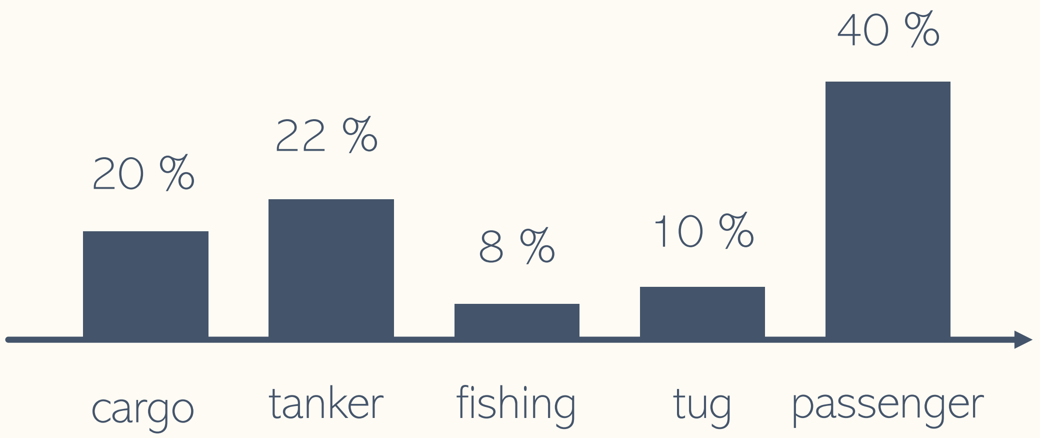
**15 % testing**

# Configurating **neural network**

1. Classifying vessel type based on trajectory

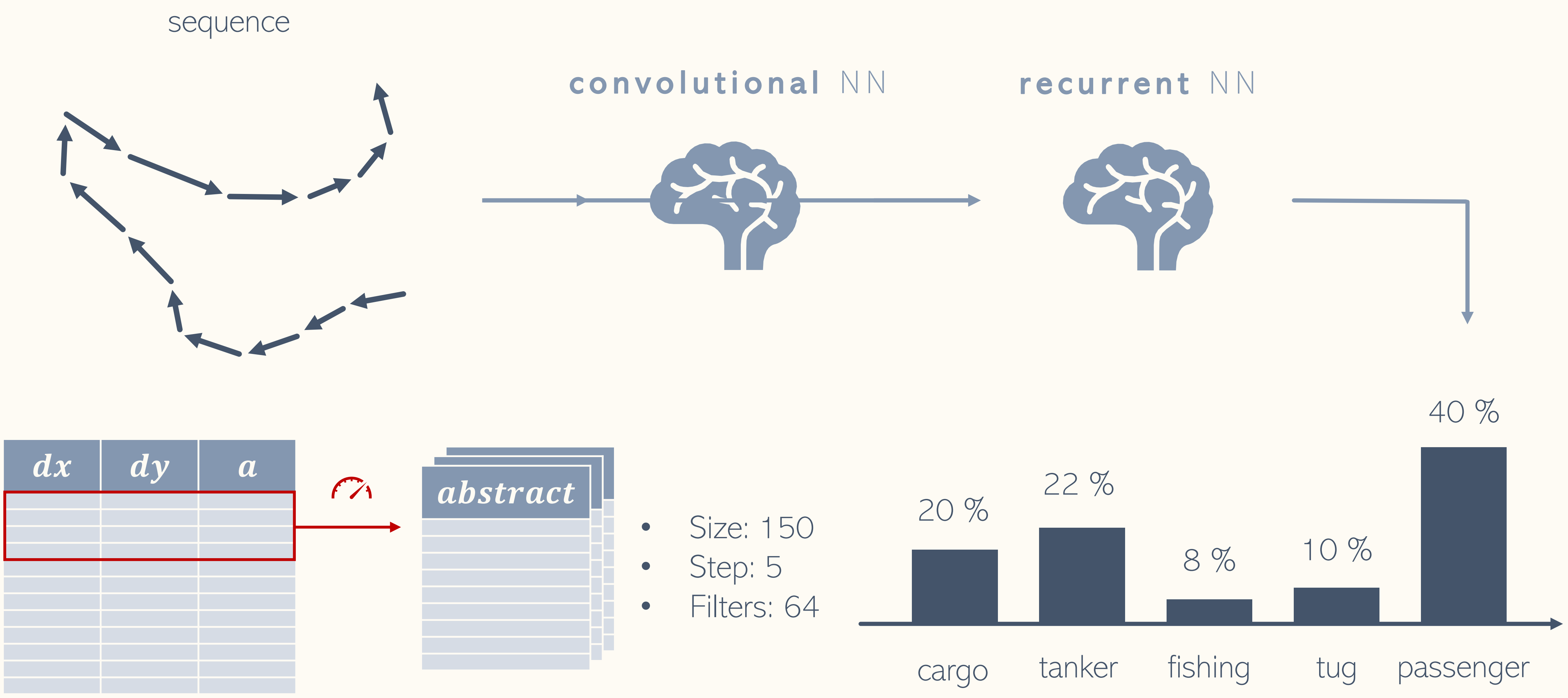



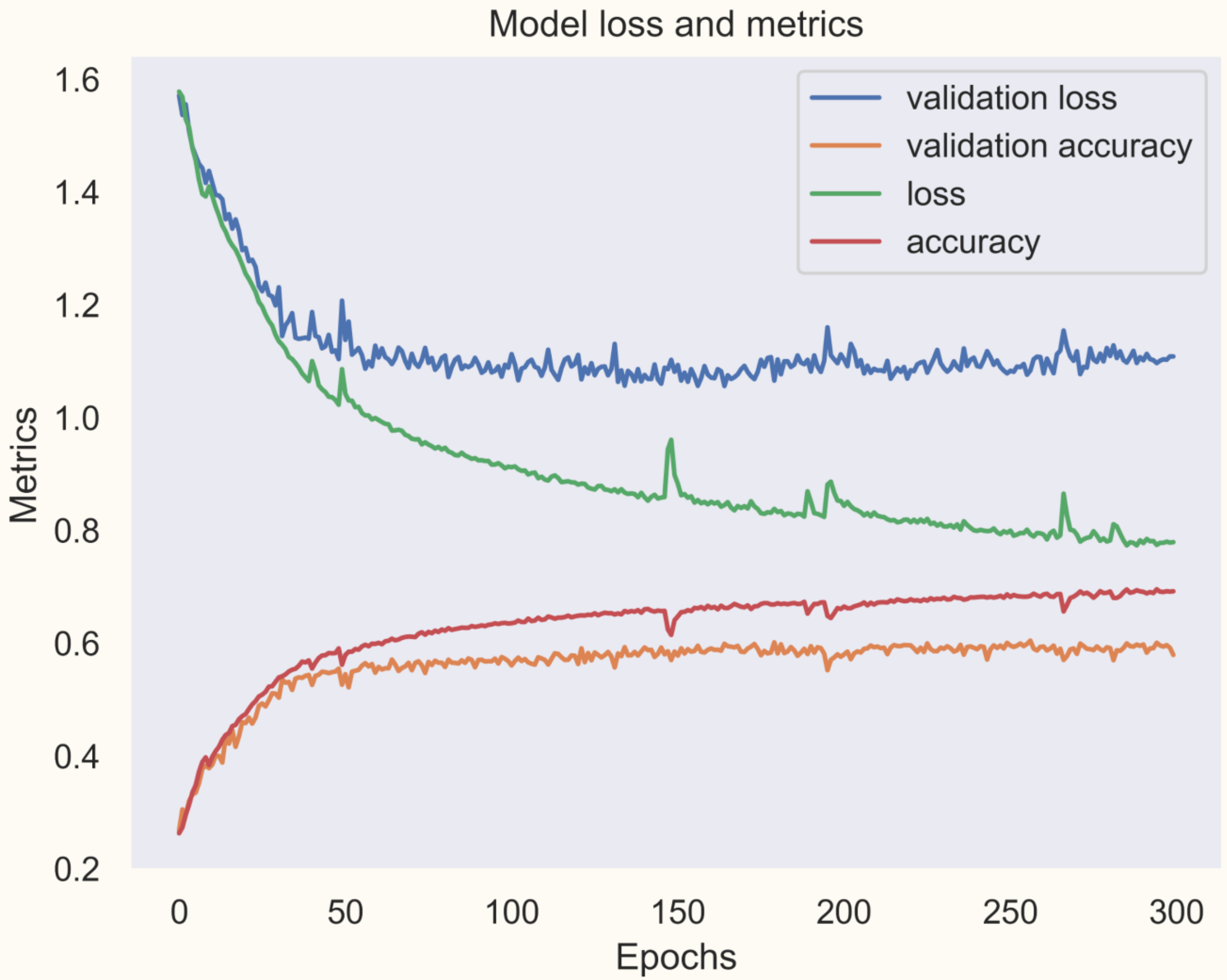
# Configurating **neural network**

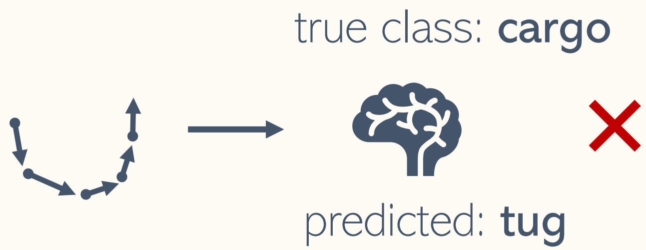
1. Classifying vessel type based on trajectory

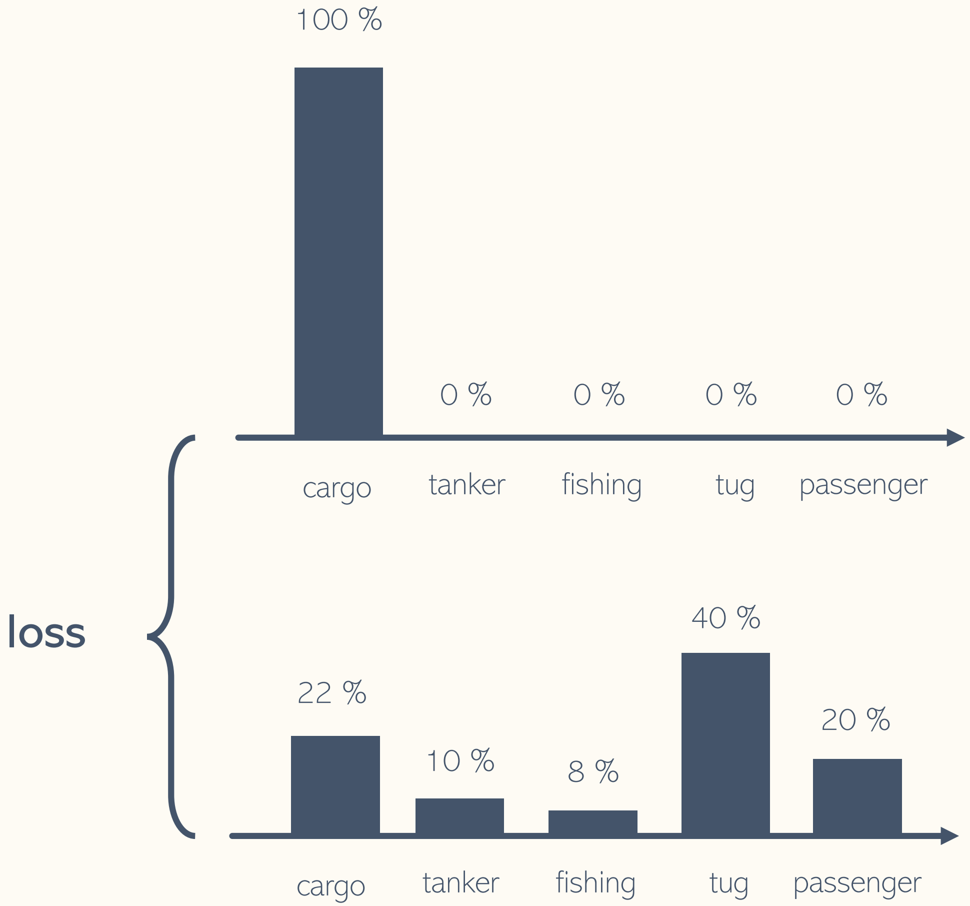


# Training **neural network**

1. Classifying vessel type based on trajectory



# Results of **test set**

## 1. Classifying vessel type based on trajectory

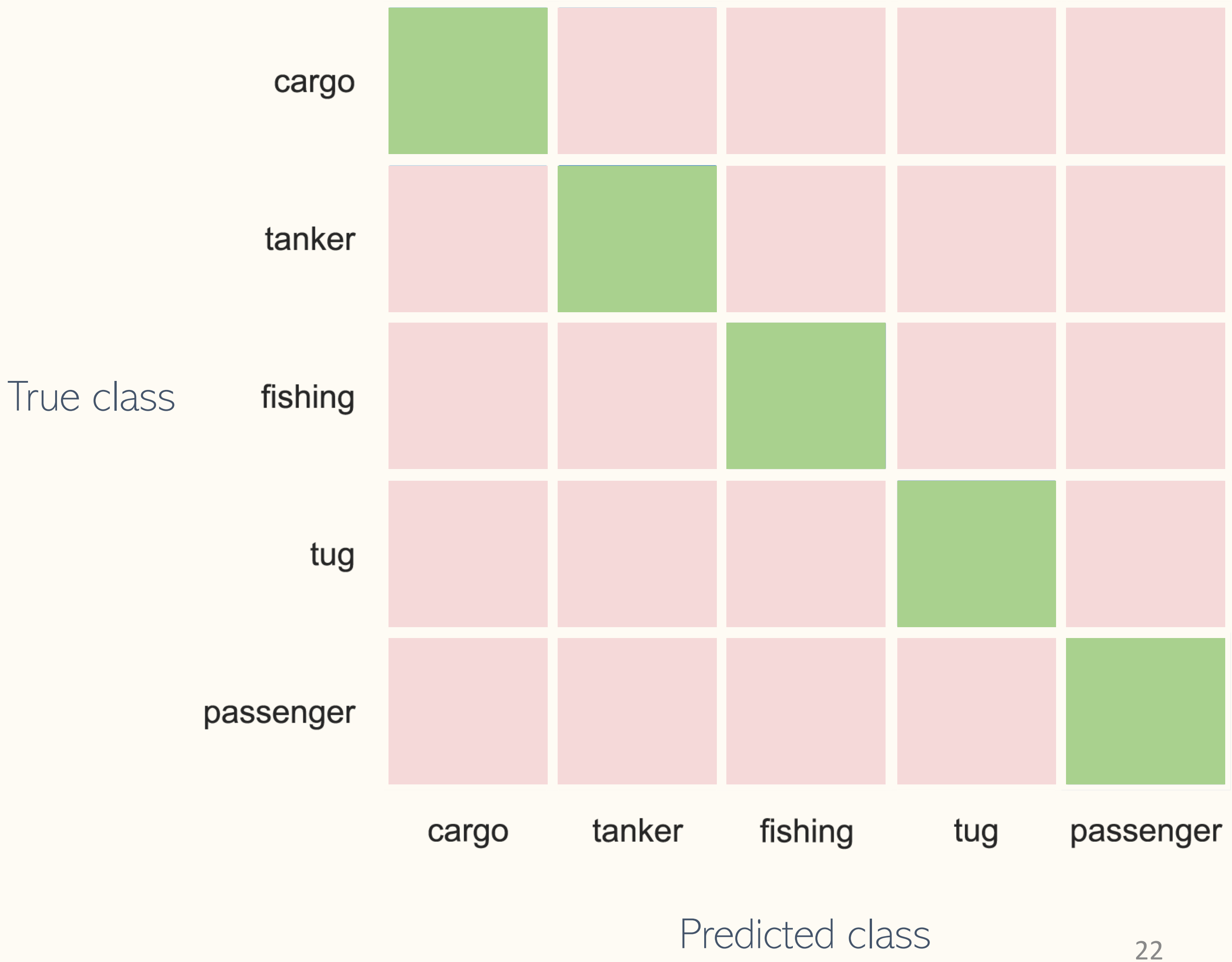
### metrics

| | |
|---|---|
| Loss | 1.10 |
| Accuracy | 0.57 |
| Precision | 0.55 |
| Recall | 0.27 |
| F1-score | 0.37 |

$$\frac{TP}{TP + FP}$$

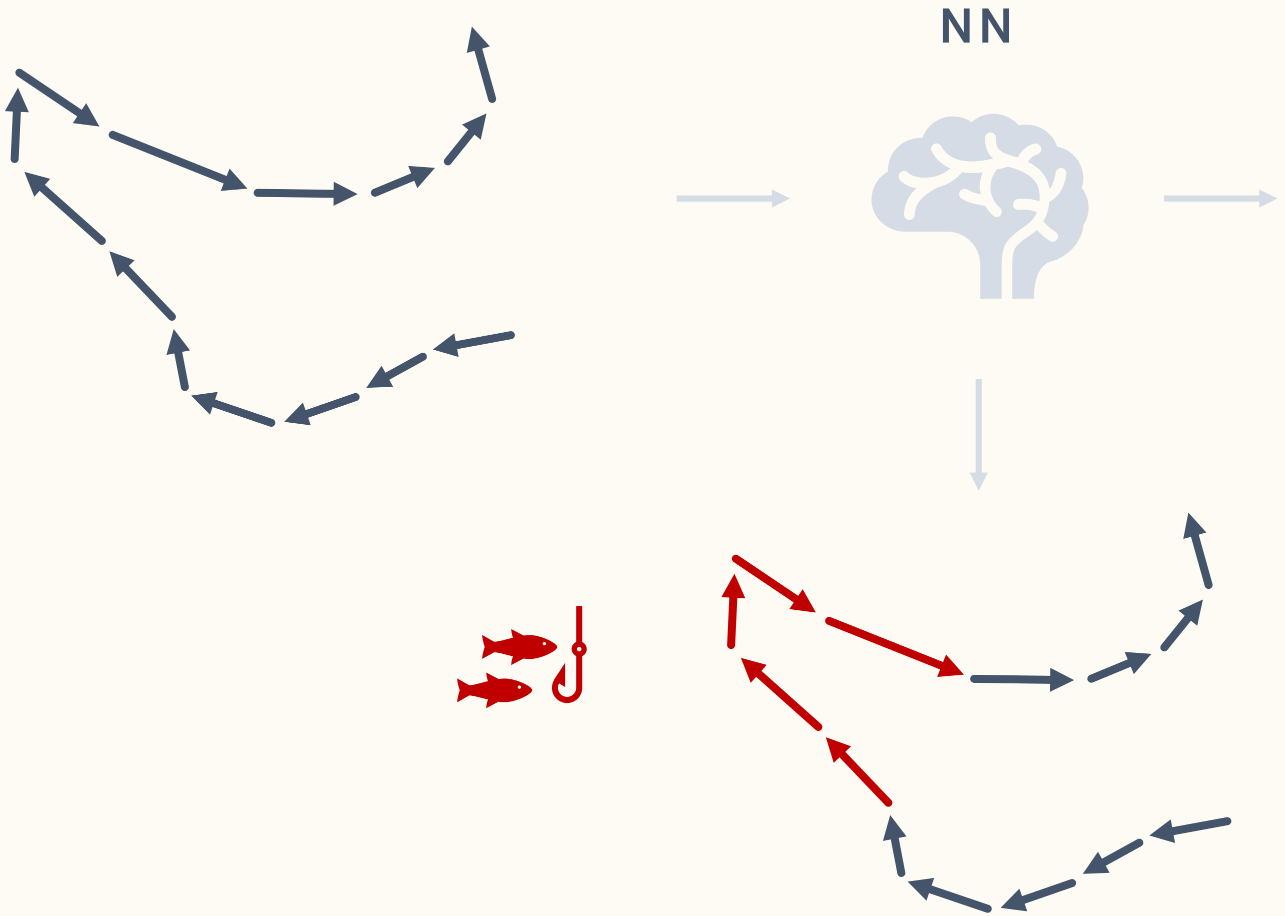
$$\frac{TP}{TP + FN}$$

### confusion matrix



# Some **experiments**

2. Detecting fishing activity based on trajectory (**fishing vessels**)

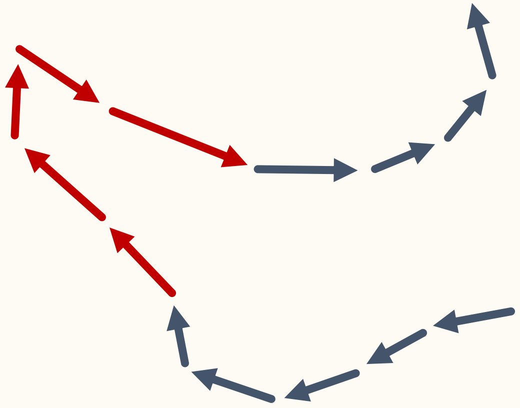


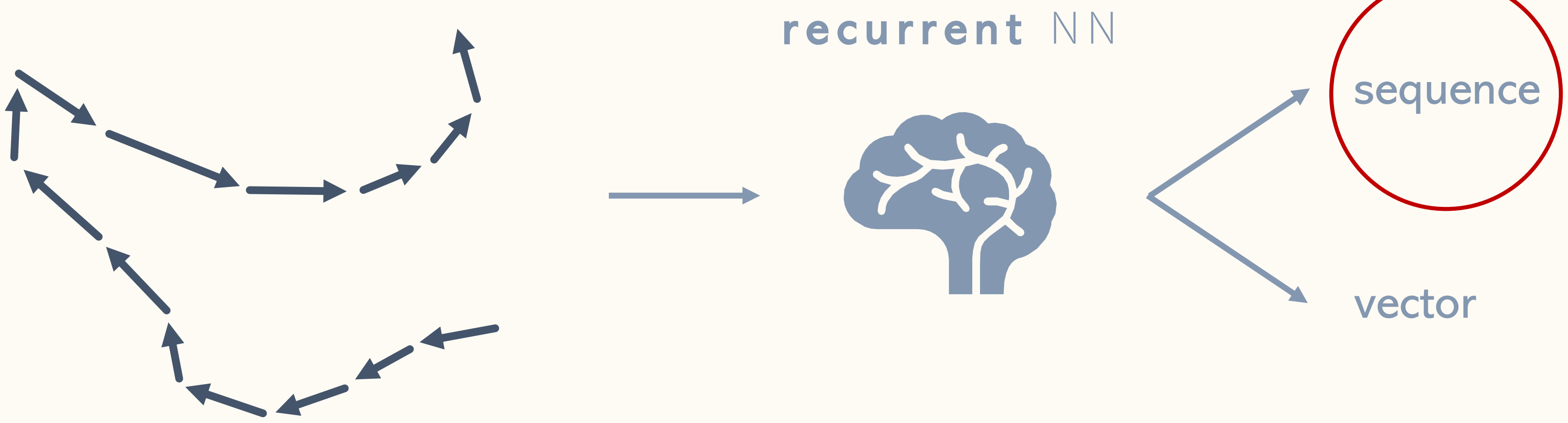

| class | vessel type | speed |
|---|---|---|
| 1 | Dredge | 0.2-1.5 |
| 2 | Beam trawler | 2.5-5 |
| 3 | Beam 255 kW+ | 2.5-7.5 |
| 4 | Otter trawler | 2-4 |

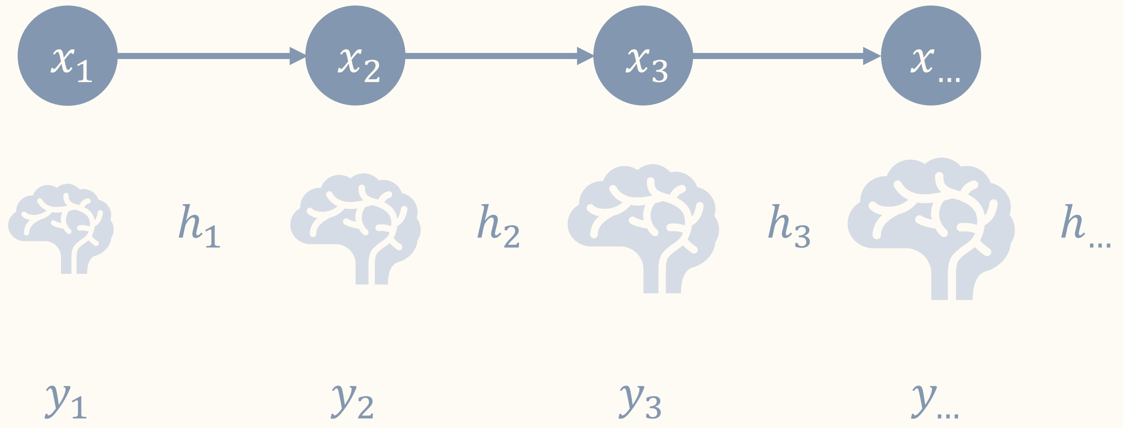
- vessels classified by Niels Hintzen
- points labeled on basis of speed
- 384 training samples
- 64 test samples
- Length of 2000 points per sample

# Configurating **neural network**

## 2. Detecting fishing activity based on trajectory



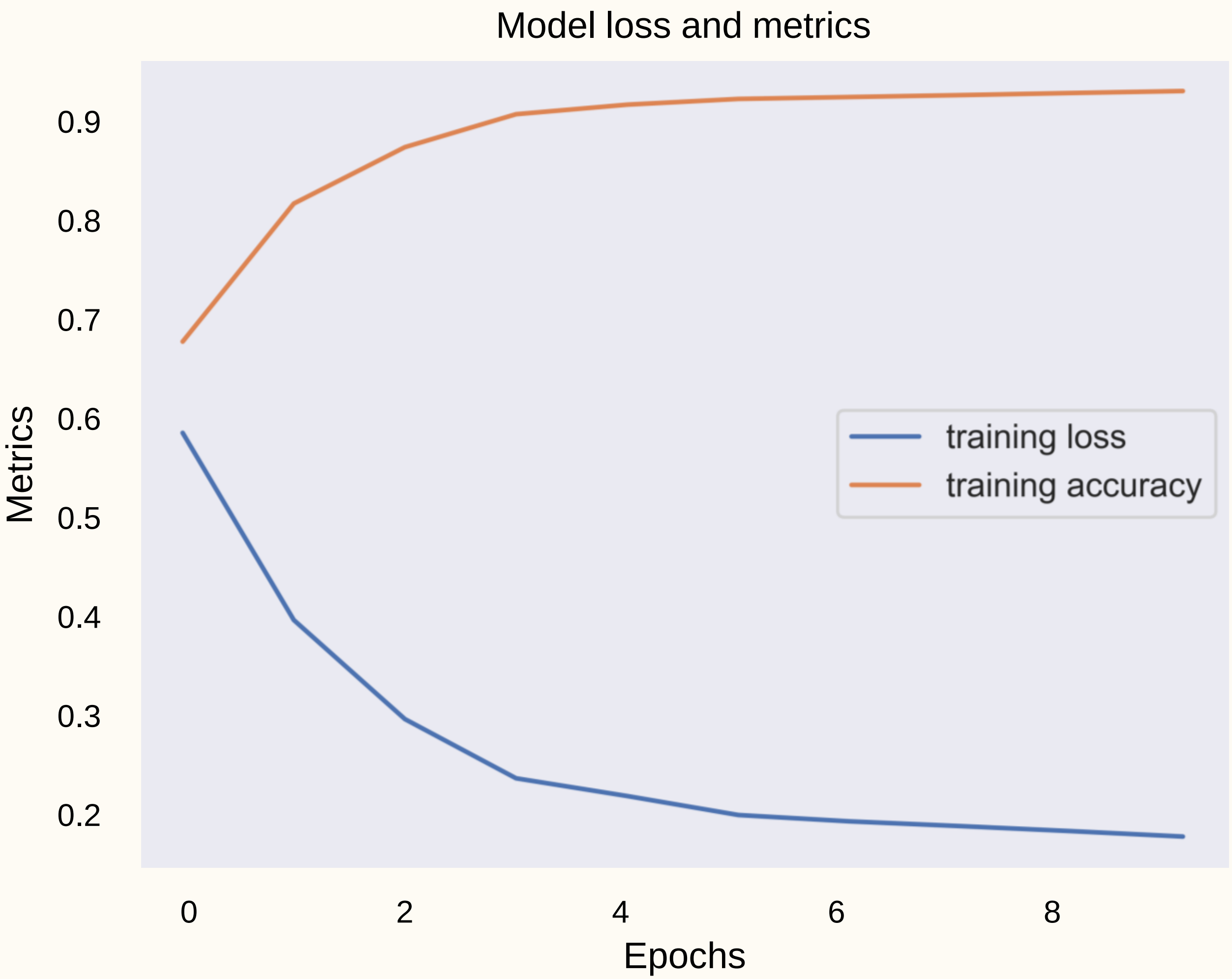


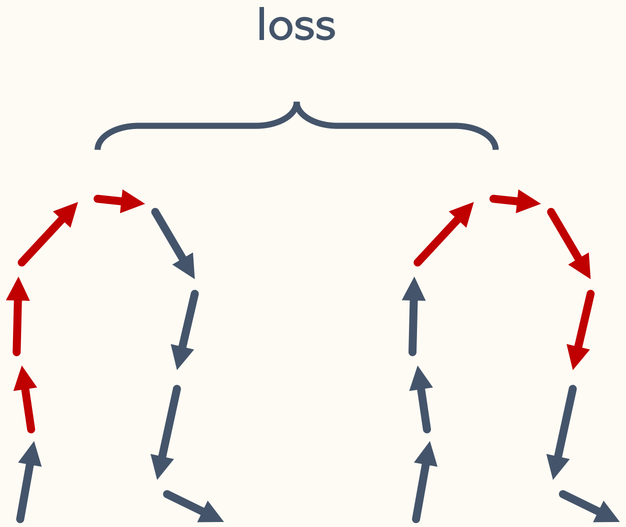




| $P(fishing)$ |
|---|
| 0.1 |
| 0.6 |
| 1.0 |
| 1.0 |
| 1.0 |
| 0.7 |
| 0.2 |
| 0.0 |
| 0.0 |

# Training **neural network**

## 2. Detecting fishing activity based on trajectory




| ***Predicted*** | ***True*** |
|---|---|
| 0.0 | 0.0 |
| 1.0 | 0.0 |
| 1.0 | 0.0 |
| 1.0 | 1.0 |
| 1.0 | 1.0 |
| 0.0 | 1.0 |
| 0.0 | 1.0 |
| 0.0 | 0.0 |
| 0.0 | 0.0 |

# Results of **test set**

## 2. Detecting fishing activity based on trajectory

| Metrics | $dx/dy$ |
|---|---|
| Loss | 0.22 |
| Accuracy | 0.93 |
| Precision | 0.86 |
| Recall | 0.95 |
| F1-score | 0.90 |

Sequence of movement vectors

| $speed$ |
|---|
| 0.14 |
| 0.95 |
| 0.91 |
| 0.98 |
| 0.94 |

Sequence of speed

| $rate\ of\ turn$ |
|---|
| 0.21 |
| 0.91 |
| 0.83 |
| 0.97 |
| 0.89 |

Sequence of rates of turn

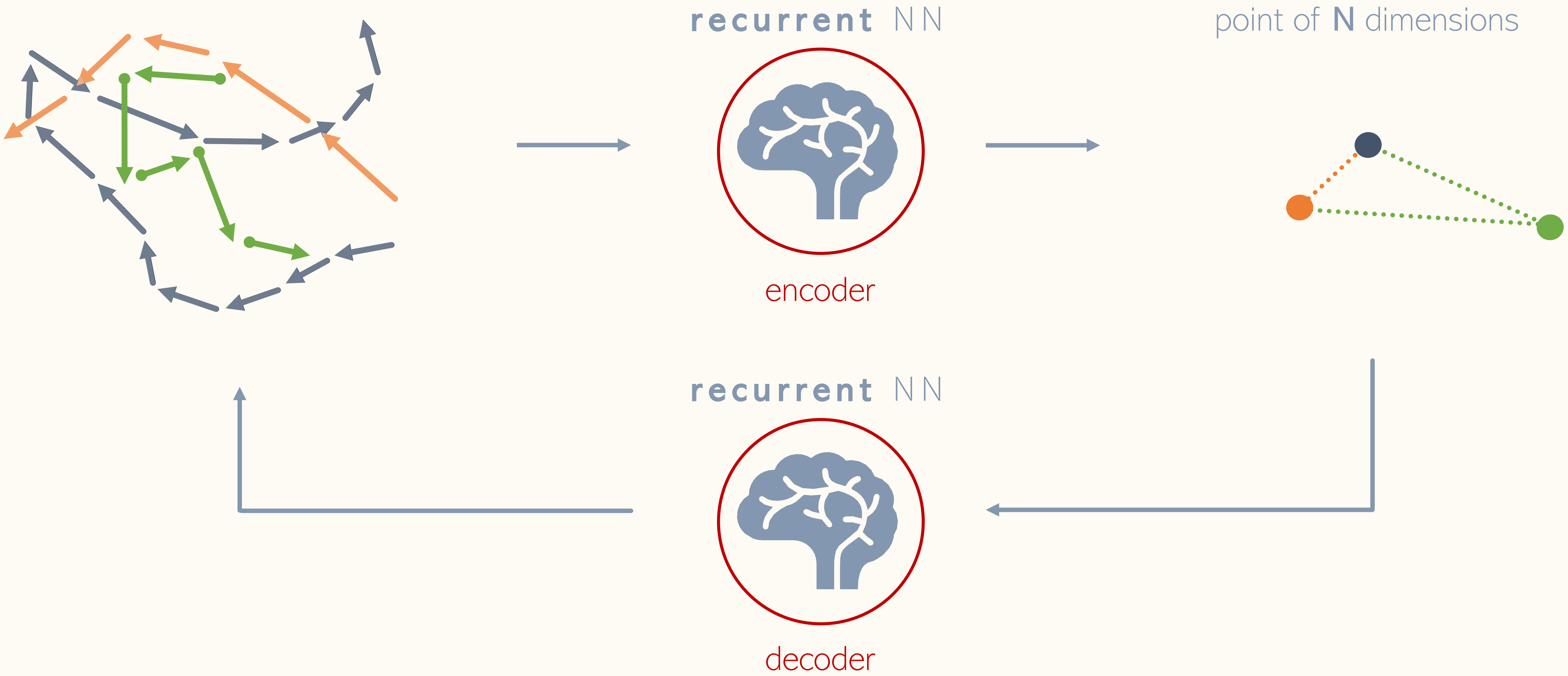
| $bathymetry$ |
|---|
| 0.66 |
| 0.62 |
| 0 |
| 0 |
| NaN |

Sequence of ocean depth

# Some **experiments**

**3.** Representation learning with synthetic trajectories

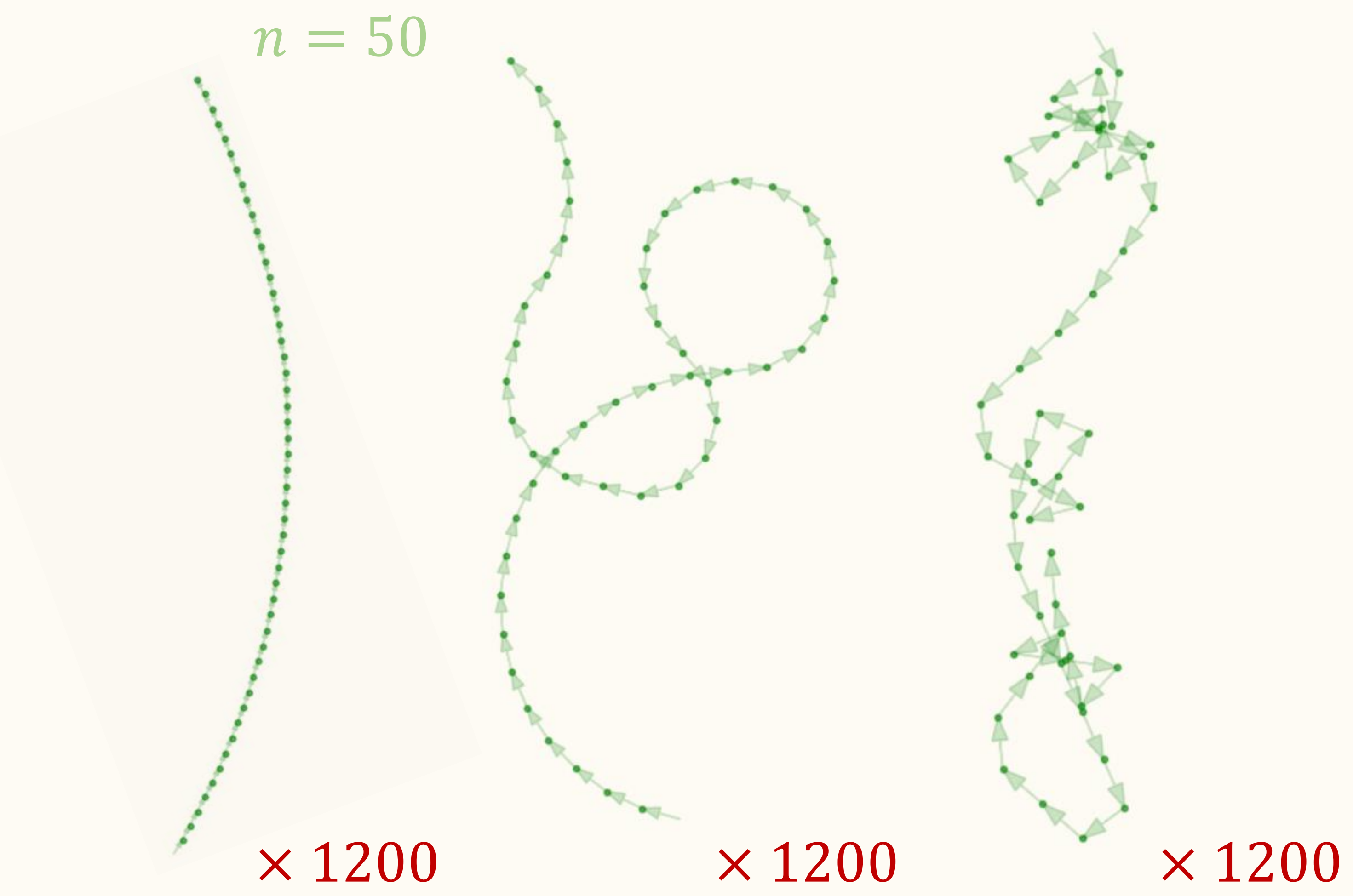
**When labeled data is not available!**



# Data set **compilation**

**3.** Representation learning with synthetic trajectories

1. arching
2. curly
3. random




**partition: 3,600 samples**

**80 % training**

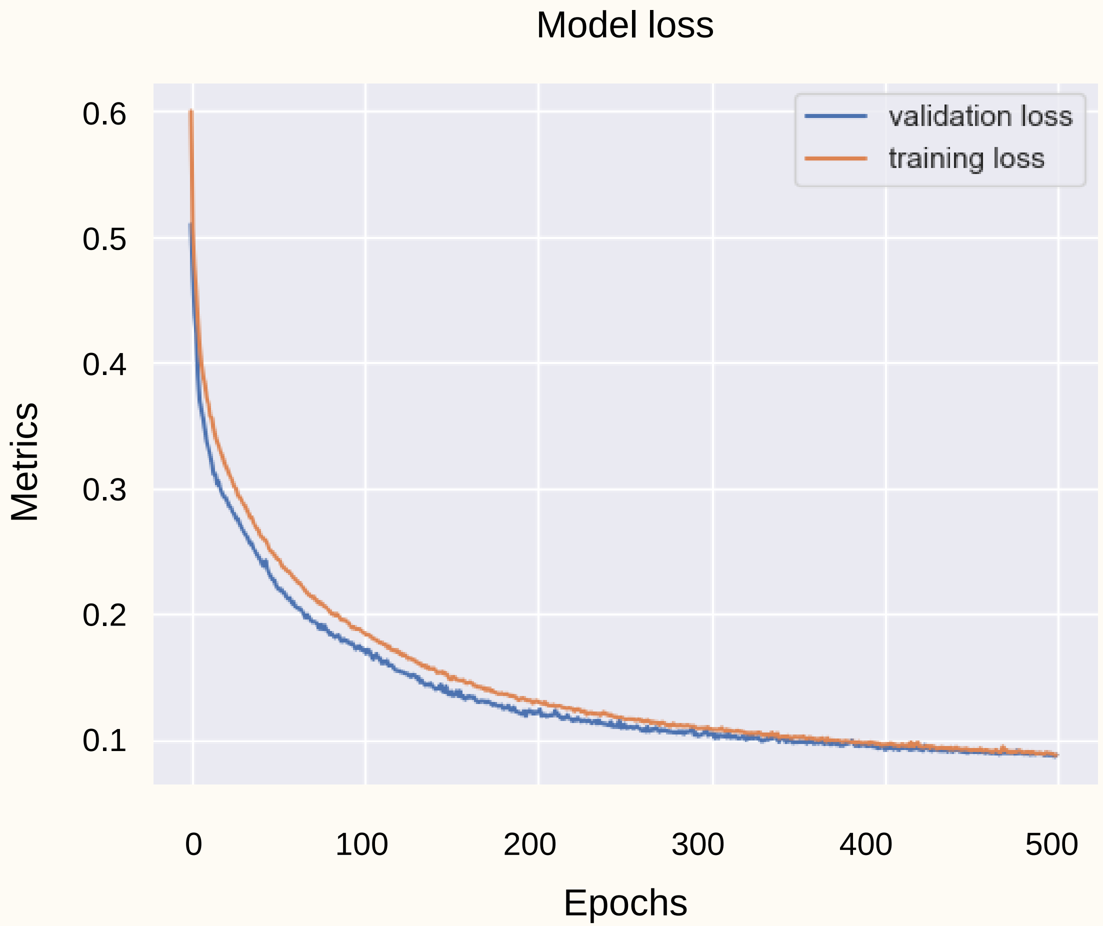
| | |
|---|---|
| Arching | 1,000 |
| Curly | 1,000 |
| Random | 1,000 |

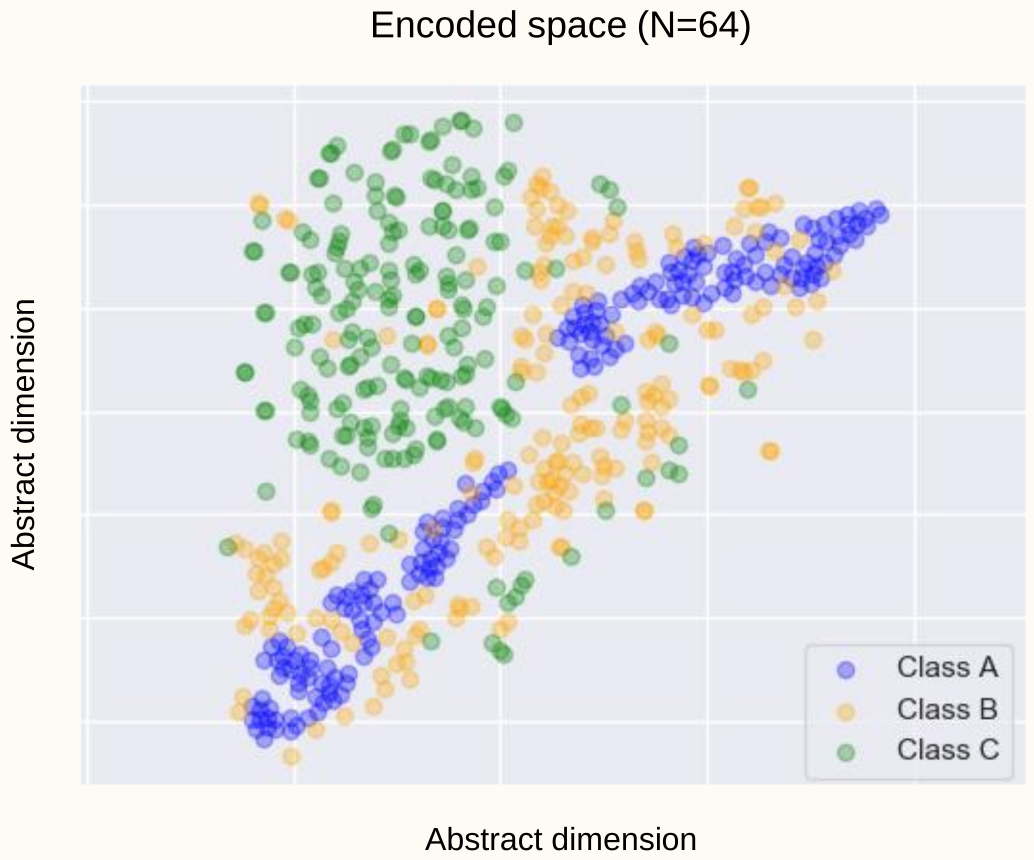
**20 % testing**

# Results of **test set**

**3.** Representation learning with synthetic trajectories

A: arching B: curly C: random



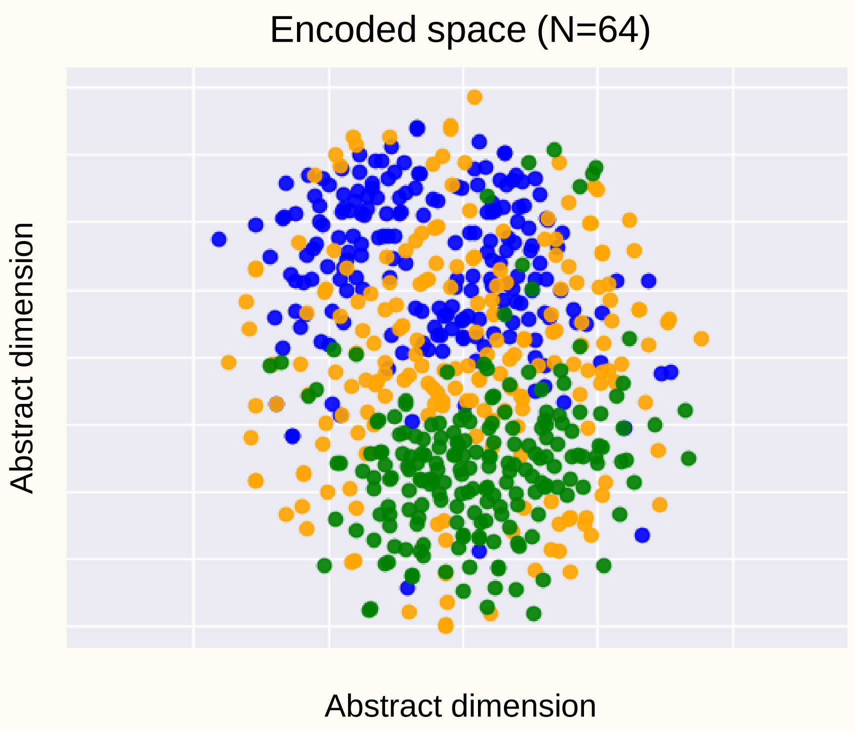




Used dimensionality reduction algorithm to get
From 64 to 2 dimensions

# Results of `test set`

## 3. Representation learning with synthetic trajectories

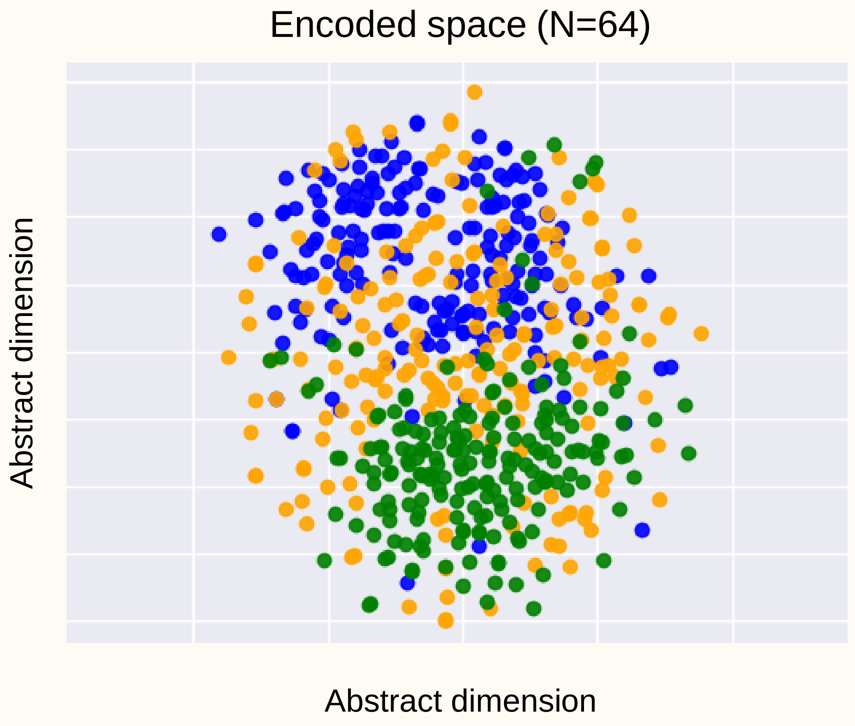
- Step further: **variational** model
- Put constraint on encoded space: make it **Gaussian**
- Encoded space is now **continuous**



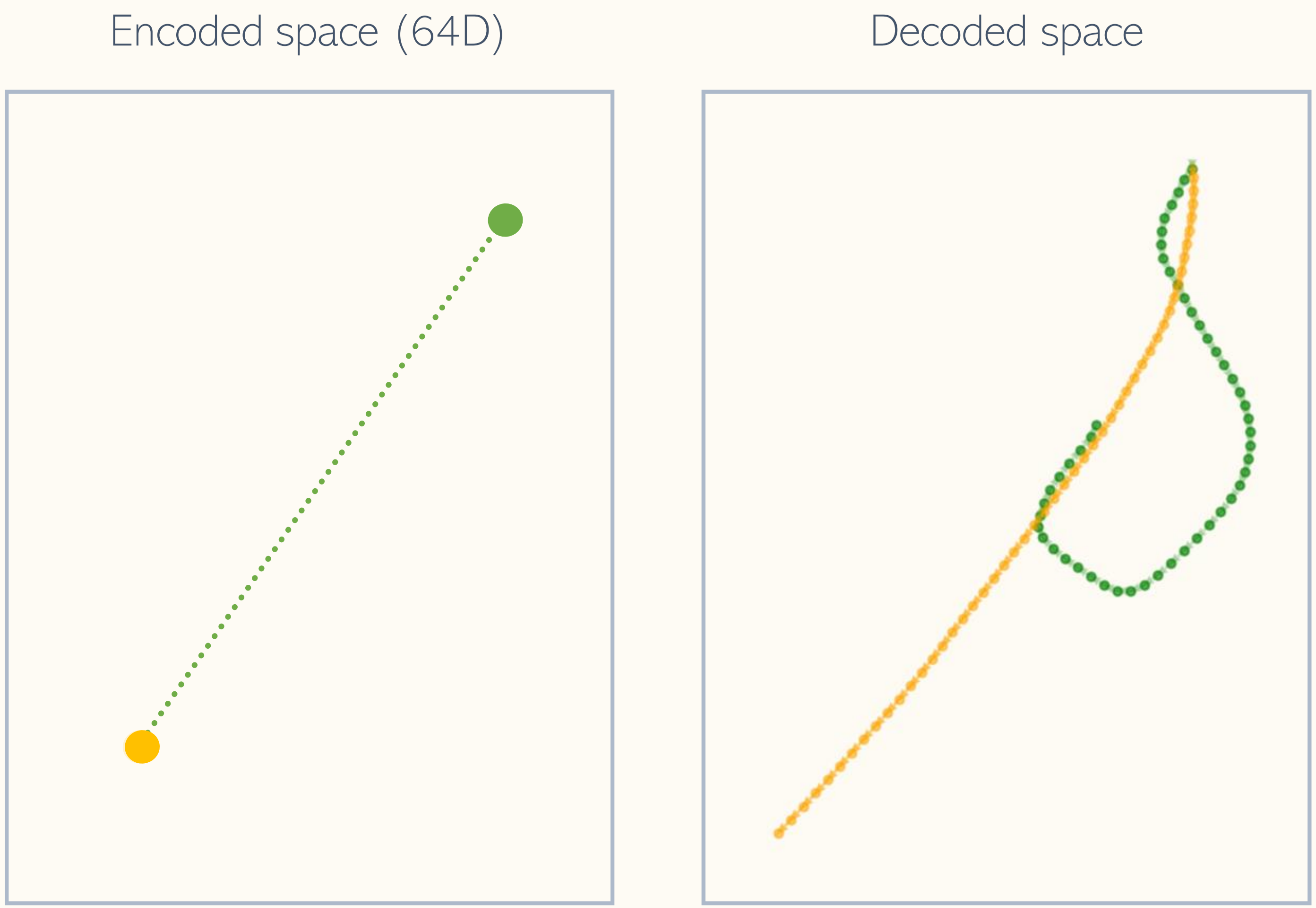
# Results of **test set**

## 3. Representation learning with synthetic trajectories

- Step further: **variational** mode
- Put constraint on encoded space: make it **Gaussian**
- Encoded space is now **continuous**



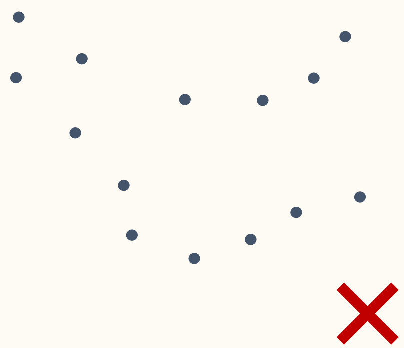

**Model has a generative aspect!**



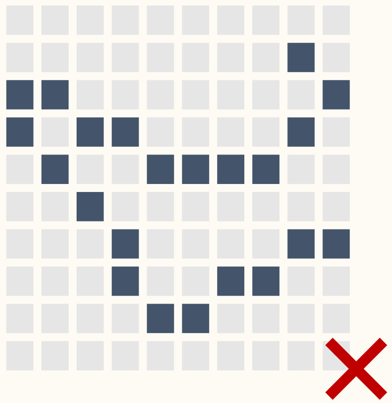
# Conclusions

1. What is a good way to represent a trajectory for a neural network?

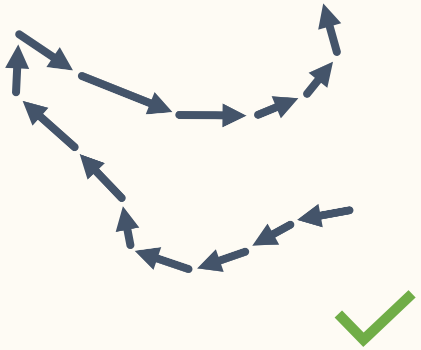
**Raw data**



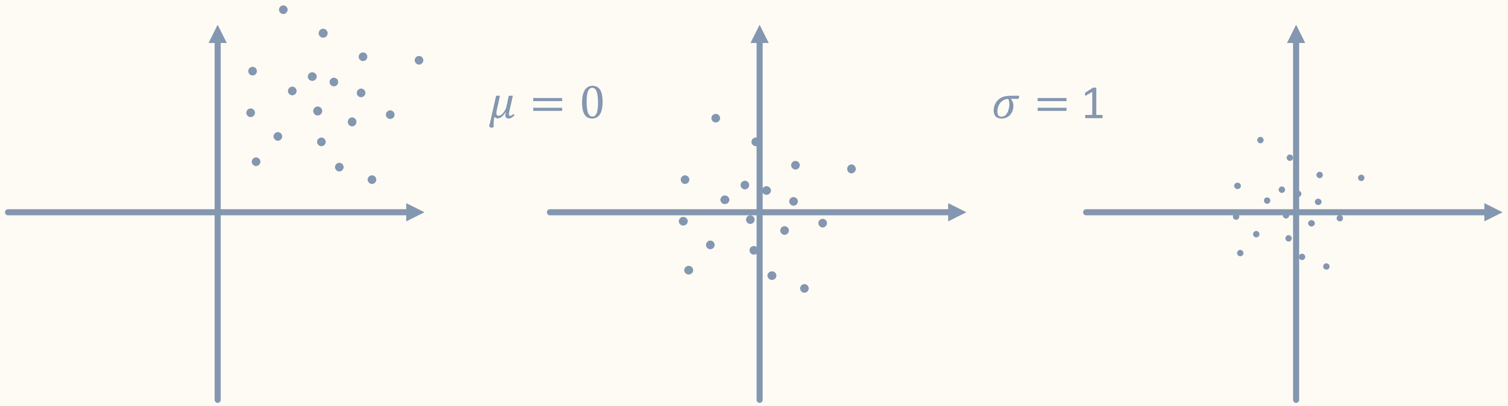
**Image**



**Sequence**



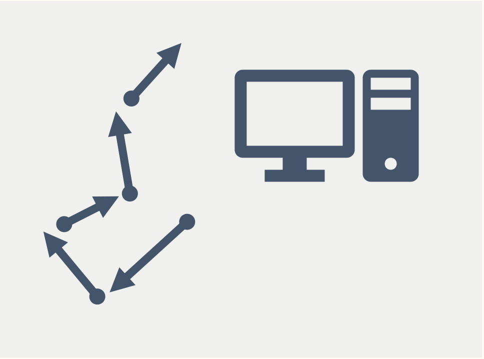
**Standardization**




- As sequence makes most sense
- Standardizing is a good idea
- Removing outliers
- Using relative movement vectors works best

**Recommendations**

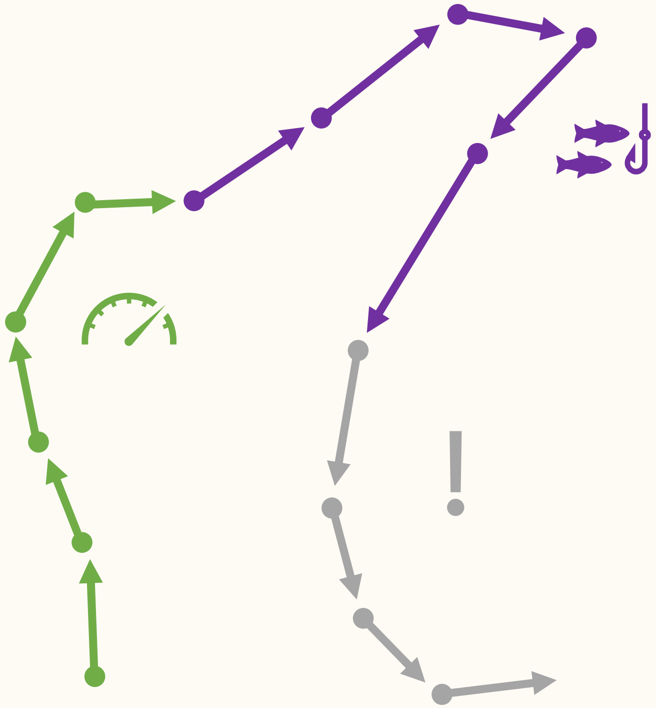
- Look at better ways to store the data
- Create single application



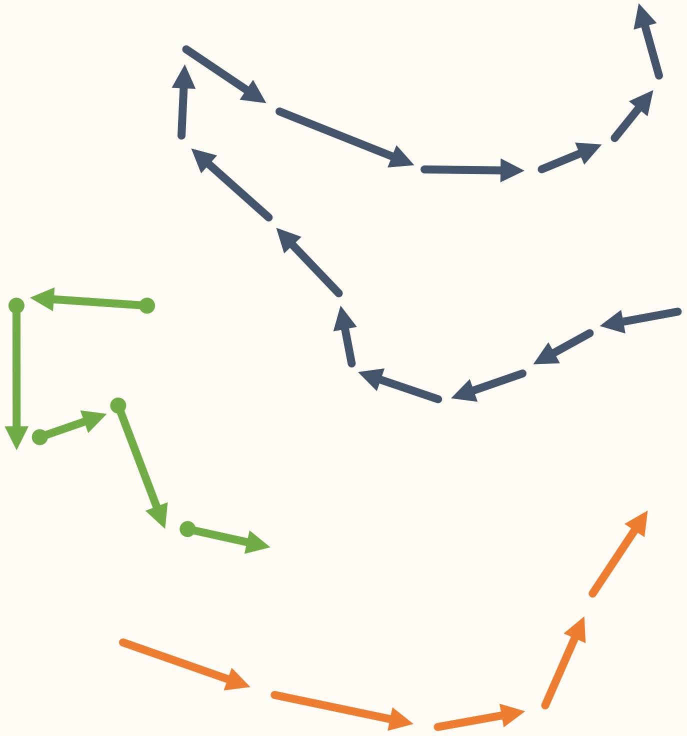
# Conclusions

2. How should the trajectories be segmented?

**Within trajectory classification**



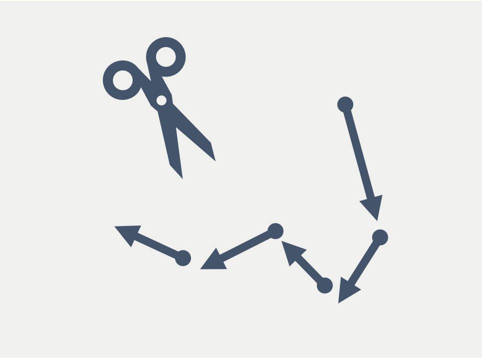
**Sub trajectory classification**



- Segmentation can be dealt with by NN
- If not, choose length as long as necessary
- Long samples take much longer to process, but encompass more information

**Recommendations**

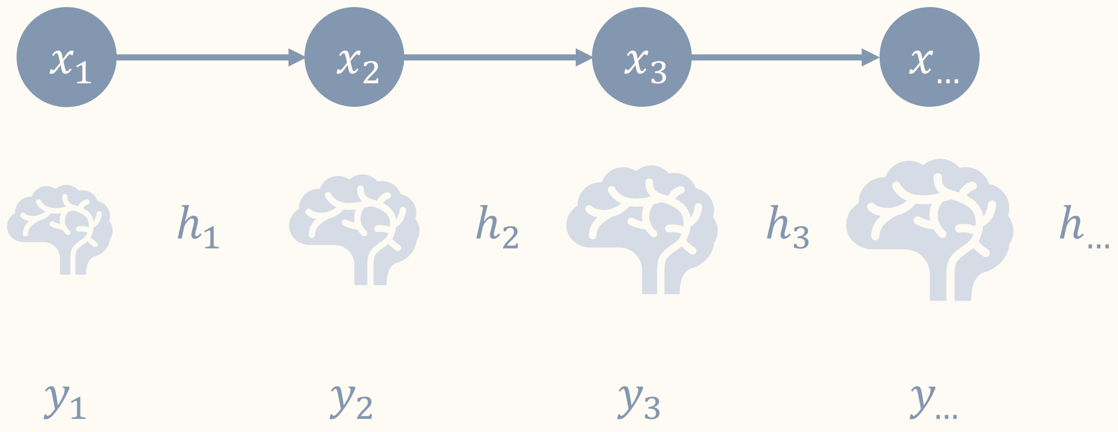
- Use time intervals as segmentation



# Conclusions

**3.** What type of neural network is fit to do this?

**recurrent** NN

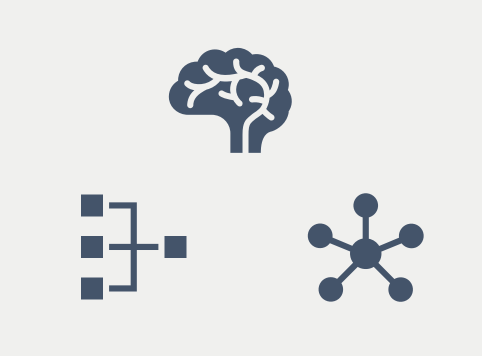



**convolutional** NN

- Recurrent NN's are a logical choice with sequences
- For vessel types it only worked together with convolutional NN
- Stacking multiple layers did not work well

**Recommendations**

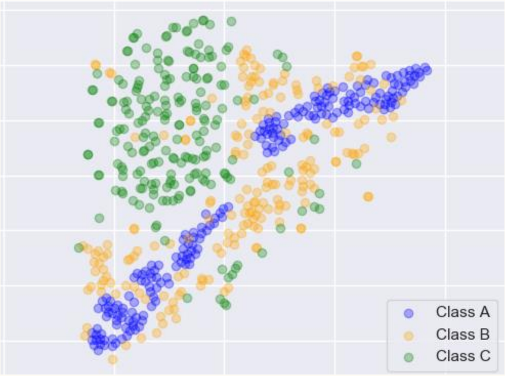
- Keep good logbooks
- Use grid search
- Use GPU



# Conclusions

**4.** How well does it work?

| | cargo | tanker | fishing | tug | passenger |
|---|---|---|---|---|---|
| cargo | 527 | 501 | 47 | 149 | 159 |
| tanker | 344 | 669 | 30 | 135 | 158 |
| fishing | 32 | 23 | 877 | 157 | 206 |
| tug | 117 | 157 | 61 | 800 | 234 |
| passenger | 66 | 77 | 41 | 125 | 923 |




- Explore use cases of generative model
  - Realistic simulations?

- In all experiments the NN is able to learn from the data
- Performance may increase based on different factors:
  - Better network design
  - Better data samples
  - Better data

**Recommendations**

- Talk to domain experts for performance criteria

Using **neural networks** to model the behavior in vessel trajectories

# References


[1] https://ai.googleblog.com/2017/11/feature-visualization.html

[2] https://towardsdatascience.com/image-captioning-in-deep-learning-9cd23fb4d8d2

[3] https://www.pinclipart.com/maxpin/Thwhxi/

[4] http://pngimg.com/imgs/miscellaneous/dna/