Ryan Gross
G47667332
CSCI 6917
Guided Research Grad I
MidTerm Project Report

As a refresher, my research topic is the application of Large Language Models (LLM) in prediction of maritime vessel trajectories. To start, I read the original transformers paper [1] to gain some familiarity with terminology that would inevitably be referenced in later papers. Reading this paper provided confidence that trajectory prediction would be achievable with transformers. This is because the problem can be framed as a sequence to sequence (seq2seq) problem. In the same way that English can be translated to French by providing English as input and French as output, vessel tracks can be split in half, where the first half is input and second half is the output the LLM should predict. After searching the web I found the TrAISformer paper [2]. This paper took OpenAI's open source GPT-2 code base, and modified it specifically for the purposes of maritime vessel trajectory prediction. This was an important find as it is trying to solve the exact same problem I am trying to solve. The main drawback of the TrAISformer approach is that it is a custom solution, with code in the network specific to AIS data columns. This makes the solution problem specific, such that it can't be pivoted to similar problems like ground vehicle prediction. With this in mind my paper aims to build on the TrAISformer paper and investigates whether newer iterations of the GPT network can perform the same task with similar or better performance, with training rather than code modification.

The benefits of building on the TrAISformer paper is I can borrow their performance benchmarking, and dataset. Starting with measuring performance, it would be hard for a reader to know whether prediction performance is good or bad. The TrAISformer author already put in the time to compare his solution with other common seq2seq solutions such as LSTMs. This provides my work with a performance baseline, created by a third party, which can provide readers with some level of context for the results from my work. Additionally, this paper provides me with a dataset. This is helpful because it allows me to perform an apples-to-apples comparison of performance, and it saves me time on data cleaning and preparation. The TrAISformer dataset is a set of AIS data pulled from the Dutch government. The following data cleaning steps were applied to the dataset.

- Remove AIS messages with unrealistic speed values (SOG $\geq$ 30 knots);
- Remove moored or at-anchor vessels;
- Remove AIS observations within 1 nautical mile distance to the coastline;
- Split non-contiguous voyages into contiguous ones. A contiguous voyage [3], [4] is a voyage whose the maximum interval between two consecutive AIS messages is smaller than a predefined value, here 2 hours;
- Remove AIS voyages whose length is smaller than 20 or those that last less than 4h;
- Remove abnormal messages. An AIS message is considered as abnormal if the empirical speed (calculated by dividing the distance traveled by the corresponding interval between the two consecutive messages) is unrealistic, here above 40 knots;
- Down-sample AIS trajectory data with a sampling rate of 10-minute;
- Split long voyages into shorter ones with a maximum sequence length of 20 hours.

With cleaned data in hand and a general understanding of transformers I wanted to get a better understanding of how prompt engineering works for LLMs. One fear I had was that GPT 3.5 would take a set of trajectory points as inputs, and it would try to reply with English sentences rather than return an array of output points. To better understand prompt engineering I read a paper [5] from Open AI demonstrating various ways to teach GPT to respond with structured responses rather than English sentences. This paper also highlighted how GPT can learn the structure in just a few examples, commonly referred to as few-shot learning. This led me to read more about few-shot learning for LLMs [6]. After reading this paper I hypothesized that even with a few examples GPT 3.5 would be able to make predictions that would be better than random selection of values. To test this theory I created an OpenAI account, and created a python script to call the OpenAPI endpoints. I manually pulled some vessel tracks from the data source, split them in half, and put them into files. Using prompt engineering and few shot learning I fed five example inputs and outputs into the system, and then a sixth input, and prompted for the output for track number six. Unfortunately the input prompt size exceeded the maximum prompt input limit of 4096 characters. Therefore I had to reduce my few-shot learning from five examples to only two. Fortunately the resulting output was in the correct format, an array of points. The only issue is that GPT cut-off the output such that the array wasn't closed, and it ended with a latitude, but didn't include the longitude. After thinking about this for a while I noticed that the track sizes had differing lengths, causing GPT to not know how long its output should be. To fix this I decided to make all inputs and outputs 18 points long. This fixed the cut-off output issue, and GPT returned a well structured array of points, exactly 18 points long. This was a pivotal moment for the project, showing that the overall approach will work, and now it's just a matter of how well it will work.

After successfully applying prompt engineering to get the correct output format, the next step was to measure accuracy and visualize the results. This will provide a base level of accuracy for GPT 3.5, which should only improve through fine-tuning. Figure 1 shows a visualization of the vessel trajectory, along with the LLM's prediction. The solid red line is the input vessel track, the dashed green line is the LLM prediction, and the dashed red line is the correct trajectory. As we can see the LLM picked up on the average distance between points and just continued that out into the future, continuing the input line. The reality is that the boat made some turns, but the prediction from the LLM is reasonable. In fact I would argue if a person was given the same input line and asked to draw an output line, they would have done the same thing. With training, the hope is that the LLM will pick up on where boats make turns. Maybe there is a piece of land that the boat was moving around, preventing it from taking a straight path. With training the LLM should pick up on that.

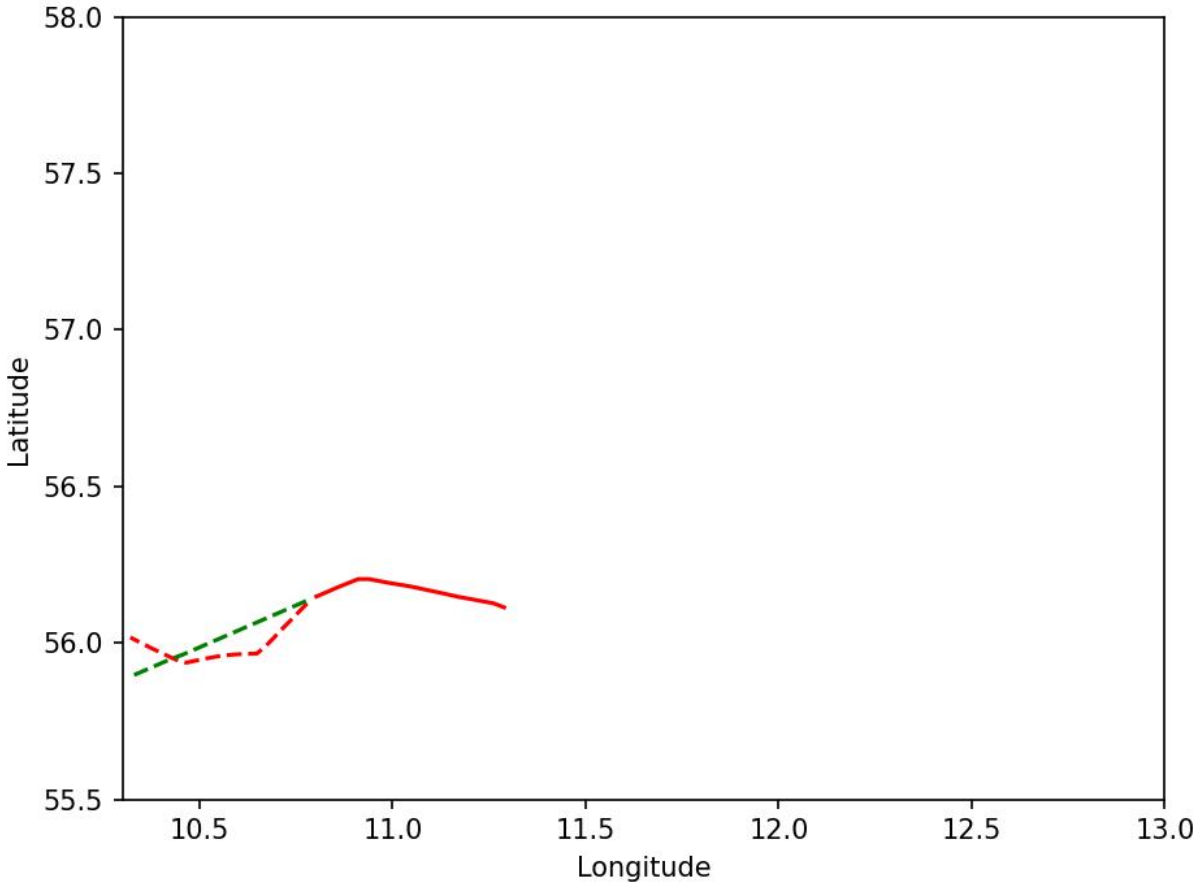

Figure 1. The red line represents a randomly selected vessel track. The track was split in half, where the first half in solid red was fed to the LLM as input, and the second half is the correct answer for the trajectory. The green is the LLM prediction.

To measure loss over time we simply look at the distance between the prediction points and the actual points. Since the earth is a sphere, we need to use the Haversine distance [x], which calculates distance between points on a sphere. The data has 10 minute samples, so 6 samples represent an hour, and with 18 total points we are predicting 3 hours into the future.

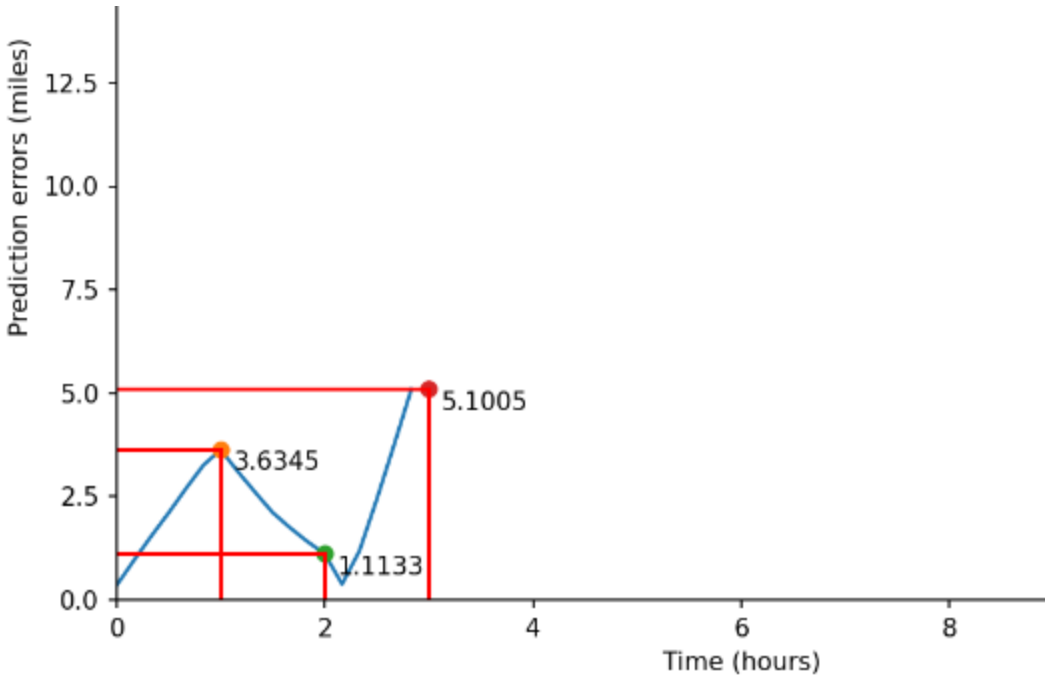

Figure 2. Prediction error over time for a single vessel trajectory prediction.

We can see from Figure 2 that the error after 3 hours is about 5 nautical miles. To get a better sense of the accuracy we will need to run many more tests and take the average, but as a first test this is actually pretty good. Figure 3 shows the error over time from many algorithms in the TrAISformer paper, and 7 of 9 algorithms greater than 5nm of error after 3 hours.

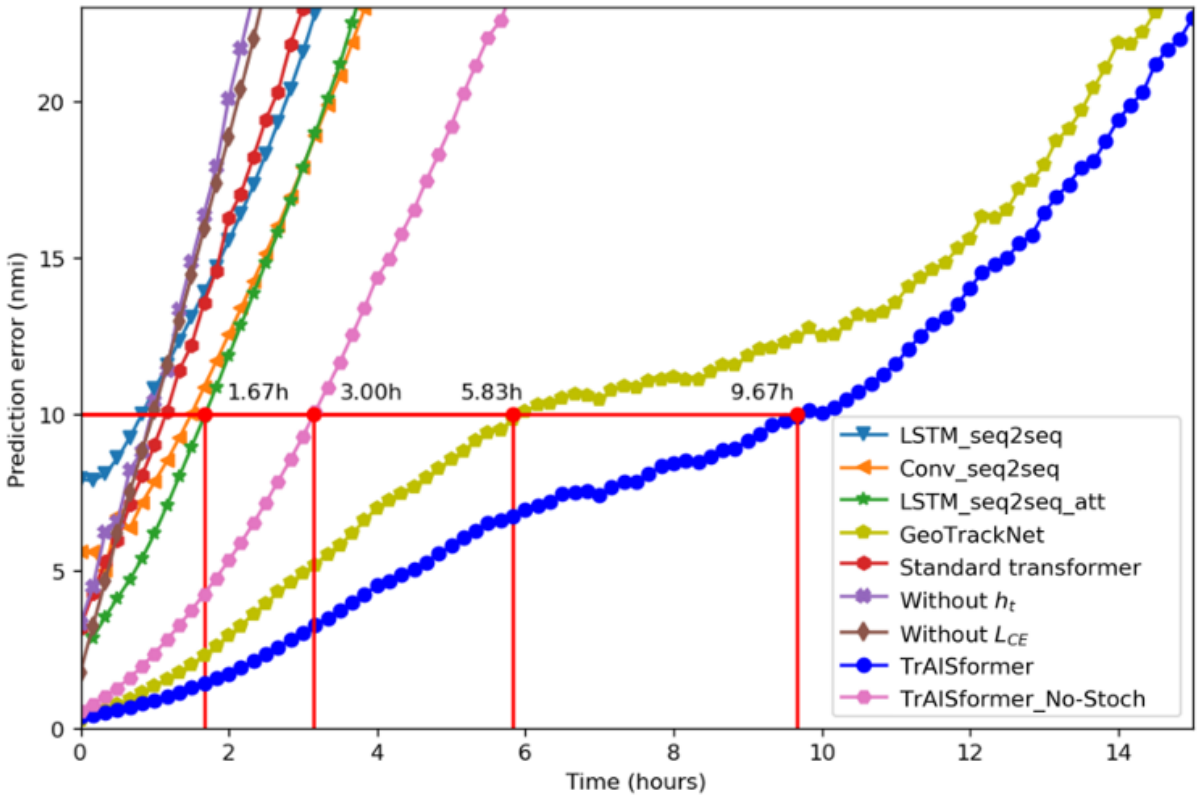

Figure 3. Many machine learning techniques attempting to predict vessel trajectories based on the same AIS training data. Results are measured in prediction error over time.

At the midterm of the project the following critical objectives have been met:

- ☑ ~~The LLM can output structured format.~~
- ☑ ~~The LLM picks up on patterns between points.~~
- ☑ ~~The LLM performs about as well as a human would perform with only a few examples.~~
- ☑ ~~Prediction examples can be visualized.~~
- ☑ ~~Accuracy can be visualized.~~

For the rest of the project the following critical items still need to be proven/completed:

- ☐ The LLM can pick up on patterns across multiple vessel trajectories (fine-tuning the model)
- ☐ Accuracy measurements need to be average over many predictions
- ☐ Place the final "fine-tuned" LLM loss of GPT 3.5 on the prediction error graphic from the TrAISformer paper.

References

[1] Vaswani, Ashish et al. "Attention is All you Need." NIPS (2017).
[2] Nguyen, Duong and Ronan Fablet. "TrAISformer-A generative transformer for AIS trajectory prediction." ArXiv abs/2109.03958 (2021): n. pag..

[3] D. Nguyen, R. Vadaine, G. Hajduch, R. Garello, and R. Fablet, “A Multi-task Deep Learning Architecture for Maritime Surveillance using AIS Data Streams,” in 2018 IEEE International Conference on Data Science and Advanced Analytics (DSAA), Oct. 2018.
[3] D. Nguyen, R. Vadaine, G. Hajduch, R. Garello, and R. Fablet, “GeoTrackNet-A Maritime Anomaly Detector using Probabilistic Neural Network Representation of AIS Tracks and A Contrario Detection,” IEEE Transactions on Intelligent Transportation Systems, Feb. 2021.
[5] Ouyang, Long et al. “Training language models to follow instructions with human feedback.” ArXiv abs/2203.02155 (2022): n. Pag.
[6] Brown, Tom B. et al. “Language Models are Few-Shot Learners.” ArXiv abs/2005.14165 (2020): n. Pag.