Exploring the potential of Machine Learning methods to predict energy consumption in shipping

D.F.G. Tijdeman

TU Delft

Royal HaskoningDHV
Enhancing Society Together

Exploring the potential of Machine Learning methods to predict energy consumption in shipping

by

D.F.G. Tijdeman

Thesis

to obtain the degree of Master of Science at the Department Hydraulic Engineering of Faculty Civil Engineering and Geosciences of the Delft University of Technology

to be defended publicly on Wednesday September 18, 2024 at 14:00 PM.

Student number:	4599020
MSc Track:	Civil Engineering
Committee members TU Delft:	Prof. dr. ir. M. van Koningsveld
	Dr. R. Taormina
	Ir. S.E. van der Werff
Committee member Royal Haskoning DHV:	Ir. R. van den Berg
Date:	September 3, 2024

TU Delft

Royal HaskoningDHV
Enhancing Society Together

Preface

This research is undertaken to complete the Hydraulic Engineering program at the faculty of Civil Engineering and Geosciences at the Delft University of Technology, to obtain a master's degree. The work, carried out at Royal Haskoning DHV, was motivated by the growing concern around global warming. With the outcomes, I hope to contribute my part towards a cleaner, greener world.

I would like to thank everyone from the Ports Consultancy teams at Royal Haskoning DHV for guiding me, making me feel welcome from the start, and sharing their practical knowledge with me. In particular, I would like to thank my daily supervisor, Robin van den Berg, for the guidance. Not only did you help me find my place at the office, but you were always available for a coffee to talk about the research or give me a pep talk. Special thanks to my supervisor at the University, Solange van der Werff, for being available each week to spar with me about my challenges. After our brainstorming sessions, I always had a clearer idea about where I wanted to go. I also want to thank the chair, Mark van Koningsveld, for his enthusiasm about the subject and helping me place the research better in the context of surrounding studies. Lastly, I want to thank my second supervisor, Riccardo Taormina. Your knowledge and critical mind helped me improve my work.

Finally, I would like to express gratitude to my friends and family for their support in the past months.

D.F.G. Tijdeman
Delft, August 2024

Contents

List of Figures

List of Tables

List of abbreviations

AE Auxiliary Engine

AIS Automatic Identification System

Bi-LSTM Bidirectional Long Short-Term Memory Network

Bi-RNN Bidirectional Recurrent Neural Network

CCUS Carbon Capture, Utilization and Storage

CO2 Carbon Dioxide

CNN Convolutional Neural Network

DWT Deadweight Tonnage

FOC Fuel Oil Consumption

FNN Feed-forward Neural Network

GEBCO General Bathymetric Chart of the Oceans

GHG Greenhouse Gasses

GPS Global Positioning System

GRU Gated Recurrent Unit

GT Gross Tonnage

HM Holtrop Mennen

HP Hyperparameter

IMO International Maritime Organization

IWT Inland Waterway Transport

kW kiloWatt

kWh kiloWatt hour

LNG Liquefied Natural Gas

LSTM Long Short-Term Memory Network

MAE Mean Absolute Error

MAPE Mean Average Percentage Error

MCR Maximum Continuous Rated Engine Power

ME Main Engine

ML Machine Learning

MLP Multi Layer Perceptron

MMSI Maritime Mobile Service Identity Number

MoSES modular ship emission modeling system

MSE Mean Squared Error

NN Neural Network

NOx Nitrogen Oxides

PM10 Particulate Matter (10 micrometers or smaller)

ReLU Rectified Linear Unit

RNN Recurrent Neural Network

RMSE Root Mean Squared Error

RPM Revolutions Per Minute

SGD Stochastic Gradient Descent

SGM Shaft Generator Motor

SFOC Specific Fuel Oil Consumption

SO2 Sulfur Dioxide

TEU Twenty-foot Equivalent Unit

TPE Tree-structured Parzen Estimator

WAM WAve Model

Abstract

As the frequency and intensity of natural disasters increase, there is growing recognition of the need to address climate change and limit the increase in global average temperature. The shipping industry, which contributes 2.9% to global anthropogenic greenhouse gas emissions, plays a significant role by releasing substantial amounts of CO_2, other harmful gases, and fine particles, negatively impacting climate change and the health of those near ports or waterways. As part of a wider initiative to reduce those effects, the International Maritime Organization has set the goal for the shipping industry to achieve net zero emissions by 2050. In order to accomplish this, accurate and comprehensive information on emissions is crucial.

Various methods have been developed to estimate emissions in the shipping industry. Top-down methods are applied using large-scale data to estimate emissions over a wide area. This approach provides a comprehensive overview but lacks the specific details required for local interventions. In contrast, bottom-up methods are applied and start with detailed data at the source and aggregate information to estimate the total emissions. Bottom-up methods offer more precise insights, however, more extensive data and complex modeling is required. To obtain more precise understandings, the modern bottom-up models use Automatic Identification System as input, a globally used system for tracking vessels. AIS data consists, among others, of time-dependent variables such as speed, location, and ship identification number. The data is used to assign an operational mode to the ship (sailing, maneuvering, anchoring, berthing). Based on this mode, it is determined if the main engines of the ship are on. If they are, the engine power is calculated from the resistance force acting on the moving ship. It is possible to convert engine power to fuel consumption and later to emissions with Specific Fuel Oil Consumption and emission factors.

Currently used emission models calculate engine power with the use of empirical formulas. So far, these models have mostly been validated by comparing them either to a noon report, or to another model. As a result, it gives no insight into the capability of a model to predict with a high spatial resolution, which is important in port areas and inland waterways.

There is little room for improvement in current semi-empirical bottom-up methods. However, in recent years, machine learning has shown the capability of replacing and often outperforming empirical models in other fields of research. This is due to the ability to model nonlinear behavior, find relations that humans can not, and work in higher dimensions. As a ship's engine power is reliant on a multitude of factors, machine learning might be the solution towards more accurate predictions.

This research aims to assess the capability of machine learning models to predict a ship's engine power with a high spatial resolution, with a focus on LSTM models. To do so, onboard sensor data from two ships was used; one sea-going container vessel and one inland tanker. The measured data was used to validate the semi-empirical method. Afterwards, the machine learning model was trained against the same sensor data. The predictions of the semi-empirical and machine learning models were then compared to one another.

Comparisons showed that the error for total energy used for the container vessel went from +62.87% to -0.82% when using a Bi-LSTM model with speed, acceleration, draught and depth as input. For the assessment of the spatiotemporal predictions, the MAE and RMSE were used. Based on these performance indicators, a normal LSTM network with speed, acceleration, draught and depth as input performed best. The MAE compared to the reference went down from 9542 to 2863, or about 7% of the maximum engine power.

The second case study highlighted the challenges that come with engine power prediction in inland waterways. Firstly, the fact that the speed over ground is used has a bigger influence in this case, as the ship will always sail with or against the current. Secondly, the influence of the blockage factor can not be ignored without having higher losses. In addition to these two shortcomings, the model was trained on a diesel-electric ship. This ship has a much more constant power profile than ships with a more classical propulsion run on fossil fuels. This likely made the model more robust to changes. The combination of these three aspects

caused less accurate predictions. Although there was more data available, errors were higher than for the case study. The best performing models showed a +3.41% error on total energy use and an MAE of 205.65, or about 12.8% of the maximum engine power.

This research has contributed new insights to the field of maritime emission modeling. The potential of machine learning models has been shown in comparison to an existing semi-empirical model. In addition, this model has now been validated using measurement data for a seagoing container vessel. Finally, shortcomings of the method using machine learning were exposed and a solution is proposed for a more complete, generalizable model.

1 Introduction

1.1. Research context

In recent years, awareness of climate change has significantly increased. People are realizing the urgent need for action to prevent the global average temperature from increasing by more than 1.5°C, as outlined in the Paris Agreement United Nations Framework Convention on Climate Change, 2015. The shipping sector contributes over 2.9% to global Greenhouse Gasses (GHG) emissions and is critical in reducing fuel consumption (International Energy Agency, 2022). Large ships emit substantial amounts of CO2, other harmful gases, and fine particles as they enter and leave ports, impacting climate change and the people living and working near ports. Consequently, the International Maritime Organization (IMO) member states have set the goal for the maritime shipping industry to achieve net zero emissions by 2050 (International Maritime Organization, 2023). In order to meet this goal, E-fuels and alternative fuels are currently being developed. However, significant time and effort will be required for these changes to be implemented on a large scale. Therefore, there is a pressing need to minimize the use of currently employed fuels.

To reduce the emission of harmful gases, it is necessary to have a better understanding of their causes. Accurate emission data can be used to implement countermeasures like Carbon Capture, Utilization and Storage (CCUS), which involves capturing and storing Carbon Dioxide (CO2), often underground. This approach can help achieve net-zero emissions without requiring zero emissions. A clearer understanding of emissions can also help track emission changes over time, assess the effectiveness of emission-reduction strategies, and determine the optimal timing and location for using alternative energy sources. Additionally, it can identify areas where the most significant emission reductions can be achieved, such as shore power or electric tugs.

1.2. Research problem

Several emission calculation methods have been developed over the years, which can be divided into two main groups: top-down and bottom-up. In the top-down approach, total emissions are calculated directly without considering vessel characteristics. Emissions are calculated from total worldwide fuel sales (International Maritime Organization, 2020b). Because vessel characteristics are not considered, the emissions per type of vessel are hard to distinguish. Furthermore, the method calculates no spatial information from total fuel sales numbers. For CO2, this might not be of high importance, but for more locally harmful emissions, such as Nitrogen Oxides (NOx) or Particulate Matter (10 micrometers or smaller) (PM10), such information can be invaluable to the health of citizens (International Maritime Organization, 2023).

In the bottom-up approach, the energy use per vessel is calculated first, which makes it possible to calculate fuel use and emissions. Holtrop and Mennen, 1982, introduced formulas to calculate energy use from friction acting on a ship, which is still used nowadays to calculate engine power. The method used is based on the principle that, at constant speeds, the ME power is proportionate to the resistance. In recent years, the mandatory Automatic Identification System (AIS) on vessels has been used to implement the method to estimate emissions. Since 2004, AIS transponders are mandatory for all ships of 300 gross tonnage and upwards engaged on international voyages, cargo ships of 500 gross tonnage and upwards not engaged on international voyages, and all passenger ships irrespective of size (International Maritime Organization, 2020a). The

fact that AIS data are readily available makes it possible to obtain a complete overview of all larger vessels in international waters and approximate their engine loads over time. Van Den Berg, 2022, proposed a model based on AIS that calculates emissions specifically in the port area. The model uses a distinction between multiple operational modes, as suggested by International Maritime Organization, 2020b. When a ship is assumed to be sailing or maneuvering, friction formulas are used to calculate the Main Engine (ME) power. When anchoring or berthing, the ME is assumed to be off. In addition, this model calculates Auxiliary Engine (AE) power based on the operational mode, the type of ship, and the size of the ship, which leads to a complete and detailed overview of the amount and cause of emissions. While the model shows promise in this regard, it has so far only been tested in a port environment and was validated against other models, but not against measurements. As a result, the accuracy of the model is not known. Lastly, the distinction between operational modes, as given by the IMO, is meant as a general guide to determine the operational mode for all ship types. As various ship types have their own characteristic behavior, this distinction might be improved for some ship types. For example, a ship is berthing when it is on- or offloading. For a container vessel this happens only in the port, but a tanker could also load at a jetty outside the port. Since the sizes of ships differs, the speed at which they are maneuvering is hard to generalize.

As bottom-up methods were developed and improved, Machine Learning (ML) gained in popularity due to advancements in computer science and increased computing power. Pedersen, 2009, showed that it is possible to predict the mean propulsion power during a period of 10 minutes with a relative error of less than 2.7%. This is done in the research using a single hidden layer Neural Network (NN) and uses ship speed, relative wind speed and direction, air temperature, and seawater temperature. Ahlgren and Thern, 2018, proposed a new method to use automated ML to predict a cruise ship's Fuel Oil Consumption (FOC), showing a significant decrease in error from linear regression. The ML models use sensor data as input, so to calculate total emissions with the bottom-up method, sensor data is needed from all vessels considered. As AIS is more widely available than the sensor data used in the studies above, a model that uses AIS as the input could be used for more ships. This has not been done yet, and would give a better idea of the ship's location when it emits GHGs. In Chen et al., 2022, a shallow neural network was shown to have a mean average percentage error (MAPE) of over 23%. Nevertheless, more potential was expected with the right training and configuration. ML in combination with AIS was considered before in Murray and Perera, 2021, where AIS is used to train a ship behavior prediction model for collision avoidance. Additionally, ML can speed up calculations. For example, mapping ships' emissions worldwide can require significant computing power, but using an LSTM can be much quicker than numerical simulation. When replacing a traditional numerical oil reservoir simulation with an LSTM model, CPU time goes down by 89.57% (Huang et al., 2022).

To summarize, there is a growing need for better FOC prediction or emission estimates. While methods have been developed to do so, they have trouble capturing the nonlinear relations between speed, vessel characteristics, and engine power. ML has proved useful in this regard, although it has its own limitations. A lack of data would make it hard to train an accurate model. Additionally, ML has only been used to model ME power so far, neglecting AE altogether. A combination of an operational mode based bottom-up model and ML for the sailing mode could tackle all shortcomings mentioned above. Additionally, developed bottom-up models have often been compared to other models or to noon reports, leaving a gap in knowledge about their spatial accuracy.

1.3. Research objectives

The goal of this research is to investigate if it is useful to use a combination of an operational mode-based bottom-up model and ML, while finding the optimal input variables. The focus is on nonlinear parts that are hard to model with empirical methods. This should lead to more accurate predictions of engine power, and therefore emissions, during sailing. Figure 1.1 shows that the most time is spent sailing, making it the most important mode to improve accuracy for. By combining the ML with the operational mode-based bottom-up model, no new method has to be devised to predict other modes. Additionally, this keeps the possibility of modeling AE power and emissions. Since the focus is only on the sailing part of each trip, it is important to assign the right operational modes. Therefore, the current method developed by the IMO will be assessed. Lastly, since high frequency sensor data is available, a current state-of-the-art emission model can be validated in its spatiotemporal accuracy.

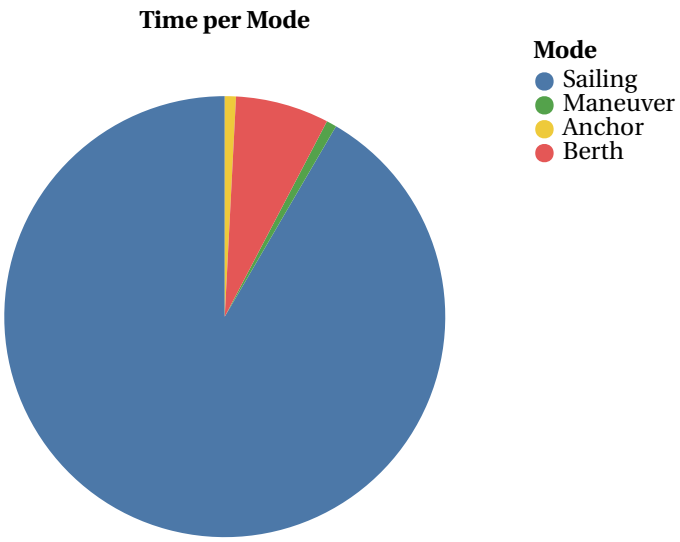


Figure 1.1: Time spent in each mode

1.4. Research scope

In this research, the focus will be on the energy use of sea-going container vessels and inland liquid bulk vessels. It is possible to apply the same theory to more types of vessels, such as passenger vessels (cruise ships). However, because these types of ships might have significantly different fuel usage patterns, other sorts of vessels than the seagoing container and inland tanker ship will not be part of this study. As a direct consequence, the goal is to provide a more accurate estimate of the energy use of these ships. Since the available data are, for now, limited to two ships, variables such as size and weight are not taken into account. However, the models will be set up in such a way that these static variables can be easily added in future. The variables that could be used to make a generalizable model are addressed as well. In addition to the measured data, AIS, and bathymetry, no other data sources are used. Lastly, while decreased computation times are a compelling reason to investigate ML models, the decrease in computational effort is not assessed in this research.

1.5. Research questions

From the research gap found, the following research question can be formulated:

How can supervised machine learning models, trained on on-board sensor data and applied to AIS data, enhance the prediction of a ship's main engine power?

To answer this research question, the following subquestions should first be asked:

1. What are current state-of-the art emission models that could be compared to a new proposed ML method?
2. How are operational modes assigned, and how could this be improved?
3. Which machine learning algorithms could be suitable for predicting a ship's ME power using AIS as input?
4. How do ML models, trained on measurement data, perform against an existing state-of-the-art emission model?
5. Which variables are important when estimating ME power from AIS using ML models?
6. What is needed to make an ML model that predicts ME power using AIS generalizable?

1.6. Outline

This research will address the subquestions before delving into the main research question. In Chapter 2, a background of emission modeling will be provided, identifying the strengths and weaknesses of existing models. This will help in choosing a state-of-the-art emission model to compare with the ML models. Chapter 3 will utilize literature to find one or more suitable ML algorithms. This will involve identifying the type of problem that needs to be solved and finding model requirements along the way. The discussed models will then be compared with respect to the set requirements and suitable models will be chosen. Chapter 4 will outline the steps needed to compare the IMO/HM model with the ML models. An overview of the cases will be given and specific choices for each case will be explained. Subsequently, Chapter 5 will discuss the results for both cases. This will be followed by Chapter 6 which will include a discussion of limitations and recommendations for future research. Finally, Chapter 7 will summarize the findings by answering each subquestion, after which the main research question will be addressed.

2

Maritime emission modeling

Several methods were developed to estimate shipping emissions, each with specific goals, strengths, and weaknesses. This chapter will provide background information on some of these models, answering the sub-question: **What are current state-of-the art emission models that could be compared to a new proposed ML method?** By presenting the strengths and weaknesses of the considered models, a suitable one can be chosen to be compared to the ML models. While emission models are considered, onboard sensors do not measure emissions. They do however measure engine power. As there is a clear consensus about the calculation from engine power to emissions, all models will be explained up to the engine power, which is what will eventually be compared to the sensor data. The focus will be on bottom-up models only. Top-down do have a place in emission modeling, however, they do not provide spatiotemporal information about emissions. This makes them unsuitable for local emission studies. Additionally, bottom-up models predict more emissions in general, suggesting that the top-down approach might give an incomplete image. A possible cause for this is lacking fuel sales data International Maritime Organization, 2020b. As AIS is mandatory in all ships with a Gross Tonnage (GT) of over 300 tons, it is quite complete.

2.1. Research history

2.1.1. STEAM

Using the a simplification of the principles for ship resistance, Jalkanen et al., 2009, introduced the Ship Traffic Emission Assessment Model (STEAM), which utilizes AIS data to estimate emissions in the Baltic Sea. The model's core lies in the ability to estimate a ship's instantaneous power, P, as a function of its velocity, V. To achieve this, they simplify the complex relationship between power and velocity, given by:

$$P_{transient} = (C_F + C_R + C_A + C_{AA})(\frac{1}{2} * V^3 * S)\frac{1}{\epsilon_0} \tag{2.1}$$

where:

- C_F is the frictional resistance
- C_R is the residual resistance
- C_A is the appendage resistance
- C_{AA} is the air resistance
- V Is the speed of the ship
- S is the wetted surface of the ship
- ϵ_0 is the propulsive coefficient

The authors simplify this equation by assuming that the resistance coefficients (C_F, C_R, C_A, C_{AA}), wetted surface (S), and propulsive coefficient (ϵ_0) are constant for a given ship. This simplification allows them to express the instantaneous power as:

$$P_{transient} = \frac{k * V_{transient}^3}{0.514^3} \tag{2.2}$$

Where k is a ship-specific constant calculated as:

$$k = 0.514^3 * \frac{\epsilon_p * P_{installed}}{(V_{design} + V_{safety})^3} \tag{2.3}$$

Where:

- ϵ_p is the main engine load at maximum continuous rating (assumed to be 0.8)
- $P_{installed}$ is the total installed power of the main engines (kW)
- V_{design} is the design speed of the vessel (m/s)
- V_{safety} is a safety margin added to the design speed (m/s)

This simplified equation allows for a more practical implementation of the model using readily available data. However, it is important to note that the assumption of constant resistance coefficients may introduce some inaccuracies, especially since frictional resistance is known to vary with velocity. According to their research, however, the difference is likely to be limited.

In addition to the main engine power, STEAM also estimates AE power based on ship type and operational mode (e.g., cruising, maneuvering, hotelling). Passenger ships are assumed to use 4000 kW of auxiliary engine power in any operational mode. For other ship types, 750 kW of AE power is used during sailing, whereas 1250 kW is used during maneuvering and 1000 kW when berthed. Note that no distinction is made in sizes, only per ship type. A maximum 20% of the installed ME power is assumed for AE output. Furthermore, the model incorporates the effect of waves on fuel consumption and emissions, using data from the WAve Model (WAM) and empirical relationships to adjust the power requirements accordingly.

The STEAM model is validated by comparing its predicted fuel consumption with actual values reported by shipowners. The results of the model were in line with the actual reported usage, particularly for passenger vessels, with an accuracy of 6%. This validation process emphasizes the model's capacity to provide reliable emissions estimates, which are crucial for formulating effective emission reduction strategies in the maritime industry.

While STEAM represents a significant advancement in emissions modeling, it has some limitations. The model does not account for particulate matter (PM) emissions due to insufficient data on fuel types and ash content. Additionally, the model's accuracy can be affected by outdated or inaccurate technical data for some ships, as well as gaps in AIS data. Furthermore, all vessels are handled as single-propeller ships. Despite these limitations, STEAM provides a valuable framework for estimating ship emissions and has paved the way for further research.

2.1.2. MoSES

The modular ship emission modeling system (MoSES) was created to calculate ship emissions with high spatial and temporal resolution, as proposed in Schwarzkopf et al., 2021. First, it gathers AIS signals for a specific area using a bounding box to capture the ship locations. Each signal contains a unique Maritime Mobile Service Identity Number (MMSI) number, which is used to gather trip data for each ship and extract ship characteristics from the IHS Markit database, where important characteristics for almost all large vessels are stored. Examples are Deadweight Tonnage (DWT), GT, length, width, ship type, fuel type, engine age, and engine type, among others. The AIS and ship data are combined using the MMSI, which is included in the AIS signal. When the GT is unknown, a method is suggested to estimate it using the ship type and the length or draught (Equation 2.4). The GT can then be used to estimate the main engine's maximum power, again

relying on the ship type (Equation 2.5). The service speed S_{ser} is a function of the main power. Vessel speed obtained from AIS finally determines the engine load (Equation 2.8), which defines energy consumption. Finally, the energy consumption is converted to emissions using emission factors.

$$GT(x) = a \cdot x^b \text{ with } x = L \text{ or } D \tag{2.4}$$

$$P_{main}(GT) = a \cdot \sqrt{GT} \tag{2.5}$$

$$RPM_{main}(P_{main}) \text{ or } S_{ser}(P_{main}) = a + (b-a) \cdot e^{-cP_{main}} \tag{2.6}$$

$$P_{aux}(P_{main}) = a \cdot P_{main} \tag{2.7}$$

$$EL = \left(\frac{S}{S_{ser}}\right)^3 \tag{2.8}$$

$$E = \Delta T \cdot (EL_{main} \cdot P_{main} + EL_{aux} \cdot P_{aux}) \tag{2.9}$$

Where:

- $GT(x)$: Gross tonnage of the vessel
- x: Either the length ((L)) or the deadweight tonnage ((D)) of the vessel
- a: An empirically found coefficient
- b: An empirically found coefficient
- $P_{main}(GT)$: Power of the main engine
- $RPM_{main}(P_{main})$ or $(S_{ser}(P_{main}))$: Revolutions per minute of the main engine or service speed of the vessel
- a, (b): Minimum and maximum RPM or speed
- c: A coefficient
- P_{main}: Power of main engines
- P_{aux}: Power of the auxiliary engines
- EL: Engine load
- S: Current speed of the vessel
- S_{ser}: Service speed of the vessel
- E: Total energy consumed
- ΔT: Time interval

Several steps in the process above introduce uncertainties that may compound one another. Characteristics like GT, ME power, and service speed are all estimates based on empirical functions. ME power is a result of the ratio between the actual speed and the service speed. No distinction seems to be made between engine age or hull shape. One of these assumptions on its own is not likely to lead to large errors, but together, they might. In addition, the relation between vessel speed and engine power is assumed to be of the third power, which is a simplification of reality.

2.1.3. IMO/HM Model

The two previously mentioned models assume a correlation between power and velocity through a third order power, in contrast to the comprehensive resistance calculation by Holtrop & Mennen. Based on this method, Segers, 2021, proposed a model to calculate emissions in inland waterways, in which shallow water effects are considered. This was later expanded on in Van Den Berg, 2022, by including a distinction in operational modes, as suggested by the IMO in International Maritime Organization, 2020b. Lastly, whereas the MoSES model uses formulas obtained from regressions to fill in unknown ship variables, this model uses a probability distribution for the ship parameters, based on the ship type. This should lead to more reliable estimates, when looking at a large enough data set. From now on, this model will be named the IMO/HM model.

Holtrop and Mennen, 1982, presented a statistical method for predicting the power requirements of a ship during the initial design stages. The method is based on regression analysis of model experiments and full-scale trial data. The authors provide a set of empirical formulas to estimate the various components of ship resistance, including frictional resistance, form factor, appendage resistance, wave resistance, and additional resistance due to a bulbous bow or transom stern.
The total resistance of a ship is given by:

$$R_T = R_F(1+k) + R_{APP} + R_W + R_B + R_T R + R_A \tag{2.10}$$

Where:

- R_T is the total resistance
- R_F is the frictional resistance (calculated using the ITTC-1957 formula)
- (1+k) is the form factor
- R_{APP} is the appendage resistance
- R_W is the wave making resistance
- R_B is the additional pressure resistance of the bulbous bow
- R_{TR} is the additional pressure resistance due to the transom stern
- R_A is the model-ship correlation resistance (accounts for roughness and air resistance)

In addition to the resistance formulas, the authors provide formulas for estimating the propulsion factors (wake fraction, thrust deduction, relative rotative efficiency) and propeller efficiency. These factors are essential for calculating the required propulsive power of the ship.

The authors acknowledge that the method may not be accurate for unconventional hull forms or extreme combinations of parameters. However, it provides a useful tool for estimating the power requirements of a ship during the initial design stages, when detailed information about the hull form and appendages may not be available.

The proposed method is based on a combination of the resistance method and IMO guidelines from recurring IMO studies aimed towards emission modeling. These studies aim to increase understanding of worldwide emissions in the shipping sector and the progress of these emissions throughout the years. According to the IMO, the ME and AE fuel consumption profiles differ based on the operational modes. The IMO/HM method uses AIS to identify the right operational mode. If the ship is considered to be in anchoring or berthing mode, the ME is considered to be switched off, and ME emissions should, therefore, be zero. If the assigned mode is maneuvering or sailing, the method of Holtrop and Mennen, 1982, is used.

When deciding on an operational mode, the distance to the destination or starting point, along with the speed, plays a significant role. If a ship is not in motion it is either berthed in a port or anchored at sea. Additionally, ships are considered to be maneuvering when moving near a port, as they need to slow down when entering and speed up when leaving. Tugs are often used to assist with these maneuvers. When at sea, and above a certain speed, ships are assumed to be sailing. The decision matrix, as given in International Maritime Organization, 2020b, is presented in Table 2.1. In Chapter 5, the performance of this matrix will be

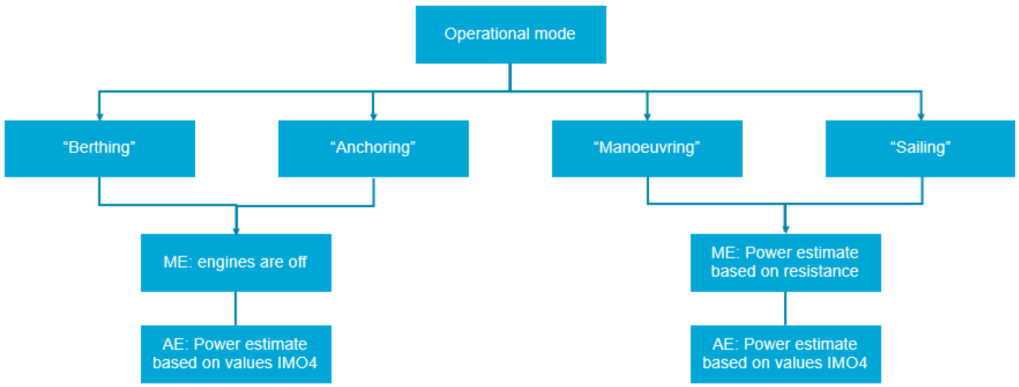


Figure 2.1: Flowchart of method developed in Van Den Berg, 2022

evaluated. If needed, adjustments can be made.

	Distance to destination		
Speed over ground	**< 1 nm**	**1 – 5 nm**	**> 5 nm**
< 0.5knots	Berthing	Anchoring / Berthing*	Anchoring
0.5 – 3knots	Anchoring	Anchoring	Anchoring
3 – 5knots	maneuvering	Maneuvering	Sailing
> 5knots	maneuvering	Sailing	Sailing

Table 2.1: Decision matrix to assign operational mode (International Maritime Organization, 2020b)

When the assigned mode is 'berthing' or 'anchoring,' the ME is assumed to be off. The engine power is estimated based on the resistance for' sailing' or' maneuvering.' Resistance acting on the ship is calculated as in Equation 2.10. The mode selection is based on speed and was expanded on by incorporating the geodesic distance to the destination of the vessel. The formulae are provided below:

$$\partial y = 12430 \frac{|lat_1 - lat_2|}{180} \tag{2.11}$$

$$\partial x = 24901 \frac{|long_1 - long_2|}{360} \cos\left(\frac{lat_1 + lat_2}{2}\right) \tag{2.12}$$

$$d = \sqrt{\partial x^2 + \partial y^2} \tag{2.13}$$

where:

- ∂_y= distance in y-direction [km]
- ∂_x = distance in x-direction [km]
- lat = latitude of points [degrees]
- lon = longitude of points [degrees]
- d = total distance between points [km]

The ITTC57 method for determining the dimensionless friction coefficient is commonly used but neglects the impact of restricted waterways, particularly shallow water effects in ports. To address this, Zeng et al., 2018, proposed a modification that accounts for shallow water by adjusting the frictional resistance coefficient. The original ITTC57 correlation line for friction resistance is:

$$C_f 0 = 0.075(logRe - 2)^2 \tag{2.14}$$

Where Re is the Reynolds number, calculated as:

$$Re = V_0 * L/\nu \tag{2.15}$$

With L being the ship length and ν the kinematic viscosity (assumed to be 10^-6 m^2/s).

Zeng et al. (2018) introduced separate coefficients for deep water (D/L > 1) and shallow water (D/L < 1) based on regression analysis of Computational Fluid Dynamics calculations. The deep water coefficient is:

$$C_f, deep = 0.08169(logRe - 1.717)^2 \tag{2.16}$$

The final modified coefficient for shallow water incorporates the proposed coefficient, the Katsui line coefficient for a flat plate in unrestricted conditions, and factors accounting for the non-horizontal wetted surface area and corrected velocity under the ship's bottom:

$$C_{f,shallow} = C_{f0} + (C_{f,proposed} - C_{f,Katsui}) * SB/S * (V_1/V_0)^2 \tag{2.17}$$

Where:

$$C_{f,proposed} = 0.08169(logRe - 1.717)^2 * [1 + 0.003998(logRe - 4.393) * (D/L)^{-}1.083] \tag{2.18}$$

$$C_{f,Katsui} = 0.0066577(logRe - 4.3762)^a \tag{2.19}$$

$$a = 0.042612 logRe + 0.56725 \tag{2.20}$$

- SB = L * B = area of the flat bottom
- V_1 = corrected velocity under the ship's bottom
- D = h - T = water depth minus draft

The corrected velocity V1 is determined by:

$$V_1 = 0.4277 * V_0 * exp[(h/T) - 0.07634] \text{ for } h/T \leq 4 \tag{2.21}$$

$$V_1 = V_0 \text{ for h/T} > 4 \tag{2.22}$$

This modification by Zeng et al., 2018, allows for a more accurate estimation of frictional resistance in shallow water conditions, which is crucial for understanding vessel behavior and emissions in port areas.

With the resistance calculated, this can be converted to the power delivered to the screw. At a constant speed, the delivered power equals the velocity times the resistance force.

$$P_e = V_0 \cdot R_{tot} \tag{2.23}$$

where:

- P_e = effective power [kW]
- V_0 = vessel speed [m/s]
- R_{tot} = ship's total resistance [kN]

Efficiency losses have to be accounted for to calculate the delivered power from screw power. Losses that must be considered are open-water, relative rotational, and hull efficiency. The open water efficiency can be explained as the ratio of thrust power to the power absorbed by the propeller operating without a hull attached, and the hull efficiency as the ratio of thrust power to effective power (Wartsila, 2024b, Wartsila, 2024a). The relative rotational efficiency accounts for the differences in the torque absorption characteristics of a propeller when operating in mixed wake and open water flows (Carlton, 2007).

$$P_d = \frac{P_e}{\eta_o \eta_r \eta_h} \tag{3.6}$$

With:

- P_d = delivered power [kW]
- P_e = effective power [kW]
- η_o = open water efficiency of propeller [-]
- η_r = relative rotative efficiency [-]
- η_h = hull efficiency [-]

Lastly, the engine power is computed. This will eventually be evaluated against the sensor data to validate the model's accuracy. The engine power is always larger than the delivered power, as there are some losses between the engine and the propeller, which occur mainly due to gearing.

$$P_b = \frac{P_d}{\eta_g} \tag{3.7}$$

With:

- P_b = brake power [kW]
- P_d = delivered power [kW]
- η_g = gearing efficiency [-]

In this research, the brake power will be used as the final output. The HM/IMO model is also capable of determining AE energy use, calculating FOC and emissions, and visualizing the spatial distribution of these emissions. As the available sensors do not measure the emissions, this can not be validated. However, the sensor data can be used to validate the model's performance regarding power output. When the output is accurate, there is a good chance that the emission estimate will also be predicted reasonably well.

2.2. Model choice

This chapter aims to answer the subquestion: **What are current state-of-the art emission models that could be compared to a new proposed ML method?** An overview of some relevant methods and models for maritime emission estimation is provided. The models STEAM and MoSES calculate ME power as a function of the speed, using a third power relationship. Both models assume resistance coefficients to be static. This is a simplification of reality. The method Holtrop Mennen does not simplify the friction coefficients. It calculates six different types of friction separately, adjusting for shallow water conditions, hull shape, and speed. All formulas are based on extensive empirical testing over 40 years. Factors like draught and changing environmental conditions can be taken into account. Lastly, missing ship variables are assigned according to their probabilities, leading to more reliable outcomes in large-scale emission estimates. For those reasons, the IMO/HM model is chosen to be used as a reference in this research.

3

Machine Learning Models

In Chapter 2, the models were described that can be used to predict ME power. In this chapter the topic regarding suitable ML models will be further investigated. An answer will be provided to the research sub-question: **"Which machine learning algorithms could be suitable for predicting a ship's ME power using AIS as input?"** To answer this question, the requirements are found that make a model suitable. From these requirements, a choice is made for which models to asses.

This research addresses a multivariate time series issue, with as goal to forecast a ship's energy usage based on various inputs that change over time and may affect ME power. These variables can have a linear or non-linear relationship with ME power and can influence one another. Since the study involves time series data, employing a model that accounts for its sequential nature, a characteristic known as sequential bias is advantageous. Since ME power at a particular time step depends on both the current values of the input variables and their previous values, it is beneficial to remember previous inputs.
The most suitable models for this type of problem are RNN, and more specifically, LSTM networks and Bidirectional Long Short-Term Memory Network (Bi-LSTM) networks. These models are designed to handle sequential data and can capture long-term dependencies between the input variables and the output variable (Tardini and Suharjito, 2024).

3.1. Multi-Layer perceptron

The MLP is a fundamental type of artificial neural network that consists of multiple layers of interconnected nodes, or neurons (Karande, 2019). A schematic of such a model can be seen in Figure 3.1. It is a Feed-forward Neural Network (FNN), meaning that information flows from the input layer through one or more hidden layers to the output layer. MLPs are widely used for various tasks, including classification, regression, and pattern recognition.

An MLP consists of perceptrons. The perceptron is a simple computation unit that takes a weighted sum of its inputs, adds a bias term, and passes this value through an activation function. Activation functions introduce non-linearity, which allows the network to model complex relationships in the data. Common activation functions include the sigmoid, hyperbolic tangent, and Rectified Linear Unit (ReLU) functions.

The input layer of the MLP receives the raw data, which is then processed by the hidden layers. Each hidden layer consists of multiple perceptrons that transform the input data into a higher-level representation. The final hidden layer's output is then passed to the output layer, which produces the final prediction or classification. The model can be represented mathematically as:

$$y_t = f(W_x x_t + b) \tag{3.1}$$

where:

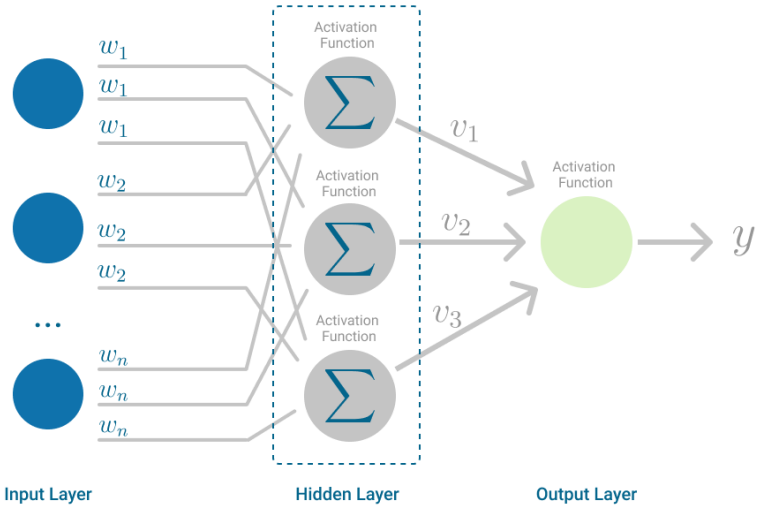


Figure 3.1: Schematic of an MLP , source: https://towardsdatascience.com

- y_t is the output at time step t
- x_t is the input at time step t
- W_x is the weight matrix
- b is the bias vector
- f is the activation function

Training an MLP involves an iterative process where the model learns to adjust its internal parameters (weights and biases) to minimize the error between its predictions and the actual target values. This process involves the following key concepts:

- **Epoch:** An epoch represents one complete pass of the entire training dataset through the network. During each epoch, the model updates its parameters based on the error it observes.
- **Stochastic Gradient Descent (SGD):** SGD is a widely used optimization algorithm for training neural networks. It updates the model parameters in the direction of steepest descent of the loss function, which is a measure of the model's error. This is done by computing the gradient of the loss function with respect to the parameters, and then taking a small step in the opposite direction of the gradient. The stochastic part refers to the fact that the gradient is estimated using a random subset (mini-batch) of the training data at each iteration, rather than the entire dataset.
- **Learning Rate:** The learning rate is a hyperparameter that controls the step size taken during each update of the model parameters by SGD. A larger learning rate may lead to faster convergence but also risks overshooting the optimal solution, while a smaller learning rate may lead to slower convergence but a more stable and accurate solution.

Various MLP models have been successfully applied to diverse time-series prediction issues, such as forecasting stock prices, predicting energy consumption, and modeling weather patterns. The ability to learn complex non-linear relationships in the data makes them a valuable tool for capturing the underlying dynamics of time-series data. However, MLP models have limitations when dealing with sequential data, where the order of the data points is important. As a result, the MLP will not model the temporal dependencies in the data, which can lead to sub-optimal performance in tasks where the past values of the input variables are crucial for predicting the future values of the output variable. Another significant disadvantage in this use-case is that they do not allow for inputs of variable lengths.

3.2. Recurrent neural networks

RNNs are a class of artificial neural networks designed to process sequential data, where the order of the data points is crucial for understanding the underlying patterns. Unlike traditional feed-forward neural networks, RNNs have connections that loop back on themselves, allowing information to persist across time steps. This enables them to model temporal dependencies and capture the dynamics of sequential data, which makes them well-suited for tasks such as natural language processing, speech recognition, and time series forecasting. In sequences that are known beforehand, these models can also be implemented in two directions, so every time step is influenced by the previous (and future) time steps.

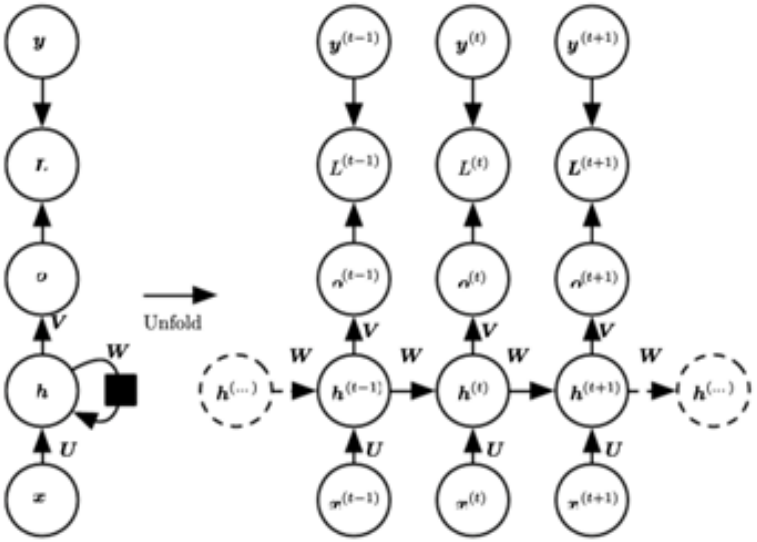


Figure 3.2: Schematic of an RNN, source: https://medium.com

The basic structure of an RNN consists of a hidden state, which acts as a memory that stores information from previous time steps, and an input layer, which receives the current input. A combination of the previous hidden state and the current input is used to input the hidden state, after which the output for the current time step is generated. This process can be represented as follows:

$$h_t = f(W_{xh}x_t + W_{hh}h_{t-1} + b_h) \tag{3.2}$$

$$y_t = g(W_{hy}h_t + b_y) \tag{3.3}$$

where:

- h_t is the hidden state at time step t
- x_t is the input at time step t
- y_t is the output at time step t
- W_{xh}, W_{hh}, and W_{hy} are weight matrices
- b_h and b_y are bias vectors
- f and g are activation functions

The activation functions f and g are typically non-linear functions, such as the sigmoid or hyperbolic tangent function, which allow the RNN to model complex relationships in the data.

One of the key challenges in training the RNN model is the vanishing gradient problem, where the gradients of the loss function with respect to the weights become very small as they are propagated back through time.

As such, it is difficult for the RNN to learn long-range dependencies in the data. To address the issue, various modifications to the basic RNN architecture have been proposed, such as the LSTM and the GRU.

LSTM and GRU models introduce additional gates that control the flow of information through the network, allowing them to capture long-term dependencies better. These gates include an input gate, a forget gate, and an output gate. The input gate determines how much new information should be added to the memory, the forget gate determines how much old information should be forgotten, and the output gate determines how much of the memory should be used to generate the output.

In conclusion, RNN models are a powerful tool for modeling sequential data, and their ability to capture temporal dependencies makes them well-suited for a wide range of applications. However, they can be challenging to train due to the vanishing or exploding gradient problem. This problem especially arises with longer sequences. Therefore, RNN models seem to be unsuitable for the requirements set in the beginning of this chapter.

3.3. Long Short-term Memory network

LSTM networks are an RNN type designed to address traditional limitations in handling long-term dependencies. While RNNs excel at processing sequential data, they often struggle to retain information from earlier samples due to the vanishing or exploding gradient problem. LSTMs overcome this issue by incorporating a memory cell and gating mechanisms that regulate the flow of information.

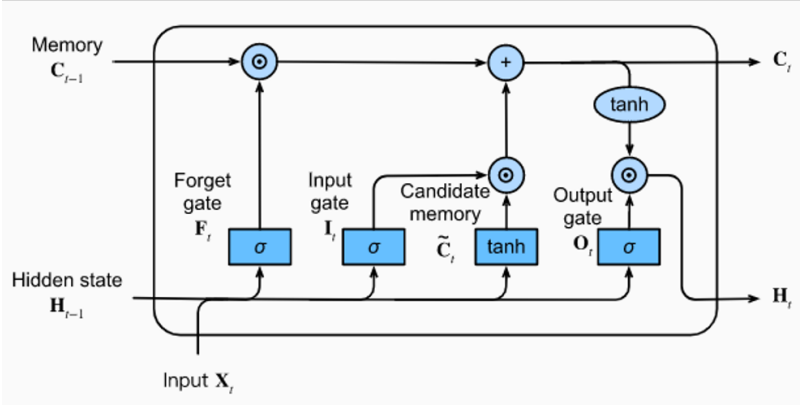


Figure 3.3: Schematic of an LSTM unit (source: https://medium.com)

The core components of an LSTM unit, as seen in Figure 3.3, include:

Cell State: The cell state acts as a conveyor belt, carrying information across time steps with only minor linear interactions, which allows the LSTM to preserve long-term dependencies.

Forget Gate: The Forget gate decides what information to discard from the cell state. To do so, it takes the previous hidden state and current input and outputs a number between zero and one for each element in the cell state. A one indicates all information should be kept, while a zero indicates that everything prior can be forgotten.

Input Gate: The Input Gate determines what new information to store in the cell state and consists of two parts: a sigmoid layer that decides which values to update and a tanh layer that creates a vector of new candidate values that could be added to the state.

Output Gate: The output Gate decides what to output based on the cell state and current input. It uses a sigmoid layer to filter the cell state and a tanh layer to create a scaled version, which is then multiplied by the sigmoid layer's output to produce the final output.

The mathematical representation of these operations is as follows:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{3.4}$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{3.5}$$

$$\tilde{C}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \tag{3.6}$$

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \tag{3.7}$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{3.8}$$

$$h_t = o_t * \tanh(C_t) \tag{3.9}$$

Where:

- f_t is the forget gate's activation vector
- i_t is the input gate's activation vector
- $\tilde{C}_t$ is the candidate values vector
- C_t is the cell state
- o_t is the output gate's activation vector
- h_t is the hidden state
- x_t is the input vector at time (t)
- W_f, W_i, W_C, and W_o are weight matrices
- b_f, b_i, b_C, and b_o are bias vectors
- σ is the sigmoid function
- $tanh$ is the hyperbolic tangent function

LSTM models have been used successfully in rainfall-runoff modeling (Xiang et al., 2020), oil well modeling (Huang et al., 2022), and solar radiation prediction (Salkuti et al., 2022). Especially when creating a rainfall-runoff model, LSTMs proved useful. The problem is comparable to ship engine modeling, where you have a complex nonlinear time series problem. Runoff is determined by the rainfall in an area, as well as topography, basin size, and other rivers in the vicinity. In addition, there is a delay between the rainfall and the runoff, just like there is a delay between a rise in engine power and a higher vessel speed. In this context, LSTM models are particularly useful for modeling the energy consumption of vessels, as they can account for long-term dependencies in AIS data, such as the impact of past speed and acceleration on current ME power. By incorporating LSTMs, the model can learn more complex patterns and relationships in the data, leading to more accurate predictions of vessel energy consumption. As the goal is to work with historical AIS data, the total sequences are already known, unlike in, for example, financial modeling where the stock price for tomorrow is predicted. Knowing the entire sequence opens the door for using Bi-LSTMs. It is called bi-directional because it looks at the sequence from the start and end. The same principle is used in language models to increase textual understanding. Bi-LSTMs and LSTMs with multiple layers and sizes of layers can be considered to find the best-performing configuration.

3.4. Gated recurrent units

Another type of RNN designed to address the vanishing gradient problem is the GRU, as proposed by Cho et al., 2014. GRU models simplify the LSTM architecture by combining the forget and input gates into a single update gate and merging the cell and hidden states. The aforementioned reduction in complexity makes GRUs faster to train and computationally less expensive than LSTMs while still maintaining the ability to capture long-term dependencies.

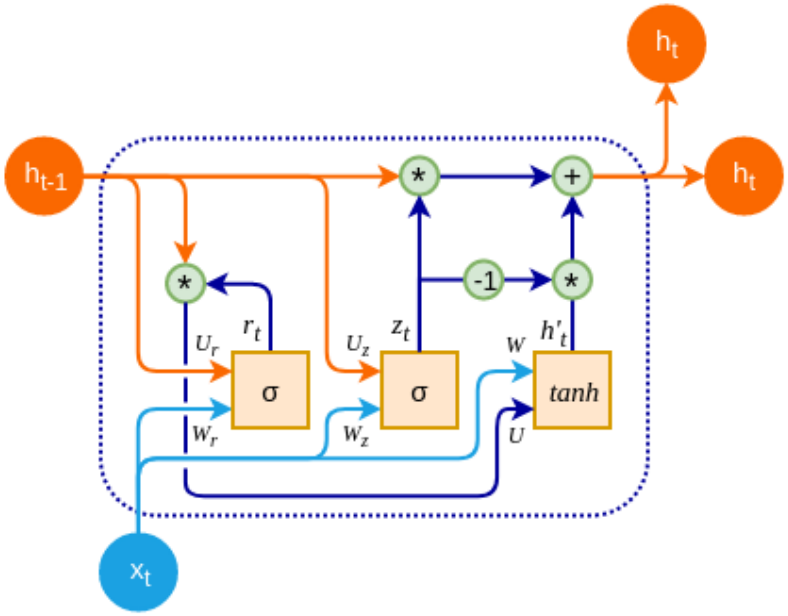


Figure 3.4: Schematic of a GRU unit, source: https://medium.com

The critical components of a GRU unit, as seen in Figure 3.4, include:

Update Gate: Determines how much of the past information to keep and how much new information to add.

Reset Gate: Controls how much past information to forget before incorporating new information.

Hidden State: The internal representation of the sequence at each time step, updated based on the current input and the gates. The mathematical representation of these operations is as follows:

$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t] + b_z) \tag{3.10}$$

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t] + b_r) \tag{3.11}$$

$$\tilde{h}_t = tanh(W \cdot [r_t * h_{t-1}, x_t] + b) \tag{3.12}$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t \tag{3.13}$$

where:

- z_t is the update gate's activation vector
- r_t is the reset gate's activation vector
- $\tilde{h}_t$ is the candidate hidden state
- h_t is the hidden state

- x_t is the input vector at time t
- W_z, W_r, and W are weight matrices
- b_z, b_r, and b are bias vectors
- σ is the sigmoid function
- $tanh$ is the hyperbolic tangent function

GRUs have been successfully applied to various sequence modeling tasks, including machine translation, speech recognition, and time series prediction. The simplified architecture and computational efficiency of the model make it an attractive alternative to LSTM models, especially when dealing with large datasets or limited computational resources. When compared to LSTM, there has been no conclusive answer to which one performs better in general (Chung et al., 2014).

3.5. Choice for Machine Learning algorithms

In the context of multivariate time series characterized by non-linear relationships, traditional linear regression methods are not optimally applicable. The variable length of these time series renders them unsuitable for MLPs. RNNs present a suitable alternative, albeit encountering challenges when managing sequences exceeding 1000 time steps due to the potential for vanishing or exploding gradients. Consequently, the remaining viable options are limited to LSTM or GRU models. Generally, LSTMs perform better with complex sequences, while GRUs provide faster training. The current project will focus on using (Bi-)LSTMs to examine the ability of machine learning in power prediction.

4

Materials and Methods

A state-of-the-art emission model has been found and potentially suitable ML models to replace this method have been identified. This chapter outlines the methodology for answering the research question: **"How do ML models, trained on measurement data, perform against an existing state-of-the-art emission model?"** Two case studies are conducted: one involving a sea-going container vessel in the North Sea and another following an inland tanker vessel, of which the important information and differences are enumerated. The importance of data pre-processing is underlined, including outlier removal, speed and distance calculation, acceleration calculation, data synchronization, and trip creation. This chapter also discusses the implementation of the IMO/HM model for estimating engine power and the use of machine learning models, particularly (Bi-)LSTM models, for predicting engine power based on various input variables. Additionally, it addresses hyperparameter tuning using Hyperopt and the evaluation of model performance using metrics like Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and total percentage error.

4.1. Materials

The diverse datasets used in this research were processed and analyzed using the **Python** programming language. Python's flexibility, extensive libraries, and strong community support made it an ideal choice for handling the complexities of the data and implementing the machine learning models. The following key Python packages were used in this process:

- **Pandas**: This data analysis library was used extensively for handling and manipulating the AIS and sensor data, which was primarily in tabular format. Pandas facilitated tasks such as data cleaning, preprocessing, outlier removal, interpolation of missing values, and visualization, enabling efficient exploration and understanding of the data and its associated attributes. The ability to work with DataFrames proved valuable for organizing and transforming the data into a suitable format for further analysis and modeling.
- **MovingPandas**: As an extension of Pandas specifically designed for movement data, MovingPandas played a crucial role in analyzing the spatiotemporal aspects of the AIS data. It enabled the creation of vessel trajectories, detection of stops and other significant events, and generation of both static and interactive visualizations, providing valuable insights into vessel behavior and movement patterns. The `StopSplitter` function was particularly useful for segmenting the AIS trajectories into distinct trips based on vessel inactivity, facilitating the separation of different operational modes and their associated energy consumption patterns.
- **Keras** and **TensorFlow**: The machine learning models, particularly the LSTM and Bi-LSTM networks, were implemented and trained using Keras, a high-level neural networks API. Keras's user-friendly interface simplified the process of building and experimenting with different model architectures, including defining layers, specifying activation functions, and compiling models with appropriate loss functions and optimizers. The underlying computations were powered by TensorFlow, a widely-used open-source machine learning framework known for its flexibility and efficiency, especially when working

with large datasets and complex models. TensorFlow's automatic differentiation capabilities streamlined the training process by efficiently computing gradients for backpropagation.

- **Hyperopt**: This Python library was employed for hyperparameter optimization, automating the process of finding the best combination of hyperparameters for the machine learning models. Hyperopt's Bayesian optimization approach, specifically the Tree-structured Parzen Estimator (TPE) algorithm, efficiently explored the hyperparameter space, leading to improved model performance and generalization.

- **Matplotlib** and **Seaborn**: These plotting libraries were used to create various visualizations throughout the research, including correlation matrices, violin plots, and time-series plots of engine power predictions. Matplotlib provided the basic plotting functionalities, while Seaborn offered enhanced aesthetic appeal and specialized plot types for statistical data visualization.

In addition to these software tools, the research relied on several key data sources:

- **AIS Data**: The AIS data provided real-time information on vessel locations, speeds, and other navigational attributes. This data served as the primary input for both the IMO/HM model and the machine learning models, enabling the estimation and prediction of engine power and emissions.

- **IHS Markit Database**: This comprehensive maritime database provided essential vessel characteristics, such as vessel dimensions, engine specifications, and fuel types. This information was crucial for implementing the IMO/HM model and understanding the factors influencing vessel emissions.

- **Onboard Sensor Data**: For the case studies, onboard sensor data from the selected vessels were used. This data included high-frequency measurements (1-minute intervals for the sea-going vessel and 1-second intervals for the inland vessel) of engine power, fuel consumption, and other operational parameters, providing a ground truth for validating the models and assessing their accuracy, in addition to providing a target for model training.

- **GEBCO Bathymetry Data**: The General Bathymetric Chart of the Oceans (GEBCO) dataset provided depth information for the areas where the vessels operated. This data was incorporated into the models to account for the influence of water depth on vessel resistance and engine power, particularly in shallow water conditions where these effects are more pronounced.

The combination of these software tools and data sources allowed for a comprehensive analysis of vessel emissions and the development and evaluation of machine learning models for predicting engine power. The selection of appropriate tools and data was critical for the success of the research, enabling the author to address the research questions effectively and contribute to the broader understanding of maritime emissions and their environmental impact.

4.2. Introduction to case studies

4.2.1. Case 1: sea-going container vessel

Length [m]	Width [m]	Twenty-foot Equivalent Unit (TEU)	DWT [tonnes]	ME power [kW]
368	51	14,000	150,000	37,000

Table 4.1: Vessel characteristics

The initial case study focuses on the transportation of container vessels. To gather data, a vessel was monitored in the North Sea for three months. The approximate specifications of the vessel are presented in Table 4.1. During the time it was tracked, nineteen port calls were made. It can carry around 14,000 TEU when fully loaded, including a reefer capacity of approximately 1,000 TEU. Additionally, the DWT is roughly 150,000 tonnes and is equipped with an ME that generates over 37,000 kW to power a single non-pitching propeller with a diameter of ten meters. Four diesel generators, a boiler, and a Shaft Generator Motor (SGM) account for onboard power and heat demand.

Multiple sources of data are used. AIS is used to obtain the speed, location, and MMSI. Using the MMSI, vessel characteristics are loaded from the IHS ship database. Data from onboard sensors are used to validate the IMO/HM model and training ML models. It contains speed over ground, location, ME power, AE power, fuel use, and more. Not all sensors operate at the same frequency; some values are shown every ten seconds, while others are only updated once every minute. Furthermore, open-source depth data is added from GEBCO (GEBCO Compilation Group, 2023). In Figure 4.2, the bathymetry map is shown. All mentioned data originate from different sources. This poses the challenge of combining everything and making sure the data are synchronized well.

The bathymetric data is sampled in an interval of 15 arc-seconds, a distance unit that varies in length based on longitude. It is largest near the equator and decreases in size the further away from it. In the areas of interest, between latitudes of roughly 50 and 54 degrees, this comes down to about 270 to 300 meters (Esri, 2000). This map also contains surface elevation on land. Around the ports, depth data are unreliable since the resolution is approximately 300 meters, and the ship is close to land in those cases. For simplicity, it is assumed that the ship never enters a place that is too shallow, so all depths are set to a maximum of -15 meters. As this research focuses on the mode of 'sailing,' little to no port entries are expected to show up in the training data, so this inaccuracy near the port is not of great importance.

The ME power signal has a high variability. In contrast to electric engines, combustion engines do not provide steady power, but rather change power over time. This can result from temperature changes affecting oil pump and engine efficiency. Currents, waves, and wind might also affect the resistance of the propeller through the water. Additionally, the engine could increase load when more auxiliary power is needed, but AEs are not on or not delivering enough power to provide onboard energy demand. Lastly, the SGM could cut in or out at moments, resulting in a sudden change in delivered power.

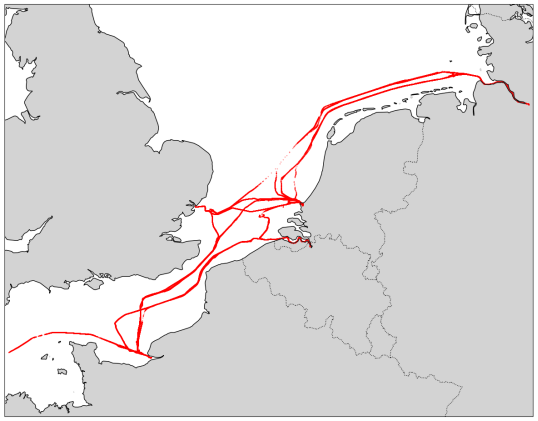


Figure 4.1: Overview of all trips

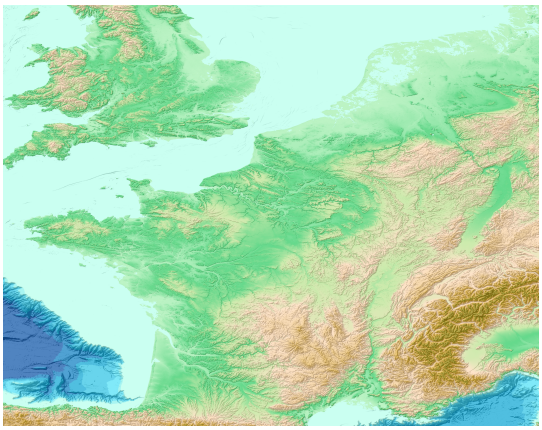


Figure 4.2: Bathymetry map (GEBCO Compilation Group, 2023)

4.2.2. Case 2: Inland waterway transport

The second case study follows an inland tanker. This is reflected in the size. With 110 meters in length and 14 in width, it is significantly smaller than the previously mentioned container ship. Contrary to most ships, this one is propelled by electric engines. Each of the three engines power one propeller, for a total maximum power of 1600 kiloWatt (kW). Energy is provided by three diesel generators that charge the onboard batteries, which can be used for the electric engines, pumps, and heating. Diesel is still needed to power the generators, but the main advantage is that this system can be retrofitted. If, in the future, bio-fuels, hydrogen, or another cleaner alternative becomes financially viable, this can replace diesel engines that are already there. Furthermore, the diesel generators can run at their optimum RPM (VistiKhetmaar, 2020). A schematic of the propulsion is shown in Figure 4.3. For liquid bulk storage, the tanker comprises 14 separate tanks with a total capacity of 5000 m^3. Each tank has its own pump, with an approximate maximum capacity of 200 m^3/h. Most trips were between Antwerp and Rotterdam, but it also visits the port of Amsterdam and locations deeper in the Belgian mainland, see Figure 4.5. Important approximate vessel characteristics are found in Table 4.2.

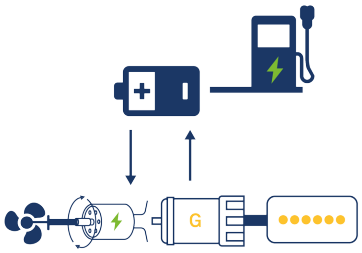


Figure 4.3: Example of diesel-electric propulsion system (Marine Industrial Transmissions Ltd. (MIT), 2024)

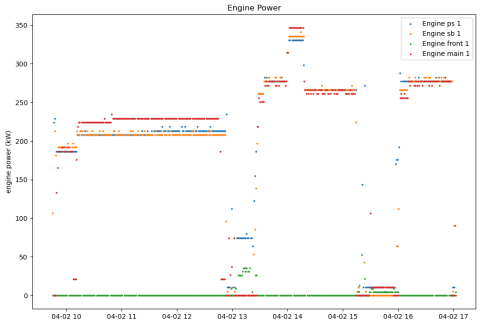


Figure 4.4: Engine power for diesel-electric vessel over a single trip

Length [m]	Width [m]	DWT [tonnes]	Engine power [kW]	Pump capacity [m^3/h]
110	14	5,000	3*533 = 1600	14 * 200 = 2800

Table 4.2: Vessel characteristics

In contrast to the first case study, the location and sensor data are now all stored in one data set. This has the advantage that the AIS information does not have to be synchronized with the sensor data. It contains measurements with a 1-second interval. Because of that, the data is much denser. Depth information is included in the data as well. It should be noted, however, that the ship's depth sensors are unreliable. Steps will be undertaken later to handle this.

As can be seen in Figure 4.6, the ship slows down at some locations. At these places, there are locks. During a locking operation, a vessel often waits before entering and while in the lock. Extra attention should be placed on preventing the trips from being stopped during such a waiting time. This might lead to a slightly different method for the creation of trips.

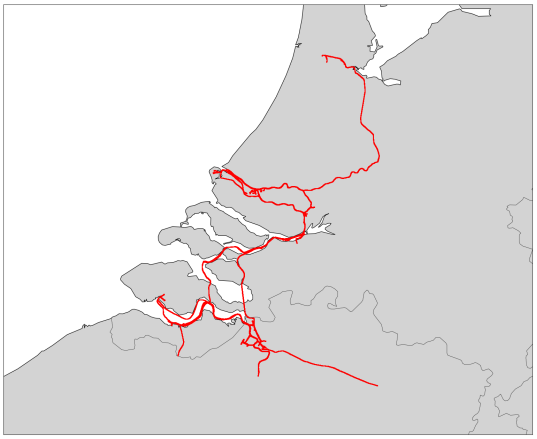


Figure 4.5: Area of inland trips for case study

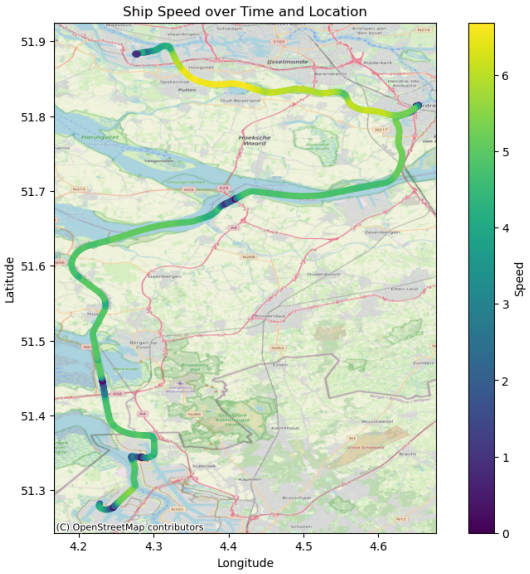


Figure 4.6: Speed over ground for a single trip

A missing variable in the previous case was the headwind acting on the vessel. Especially for container vessels, a relatively large part of the area is above water. Together with currents, this can have a significant influence on energy consumption. In this case, wind sensors have been placed on the vessel, making it possible to see the direction and magnitude of the wind speed relative to the ship. By combining these two components, the headwind can be calculated, as seen in Equation 4.1.

$$headwind = v_{\text{wind}} \cdot \cos(\text{radians}(\theta_{\text{wind}})) \tag{4.1}$$

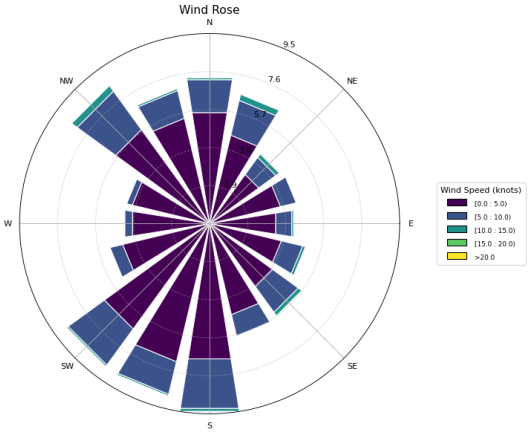


Figure 4.7: Windrose

4.2.3. Differences between cases

The two cases mentioned do share similarities; however, there are some differences as well. Firstly, the sizes differ significantly. Where the first case study concerns a 368-meter-long container ship with high stacks of containers on board, the second one follows a 110-meter-long tanker, which has less surface above water. Another difference lies in the environment. Where a sea-going vessel might encounter currents from the side, in an inland waterway, the ship always moves with or against the current. Therefore, it is always moving in the most or the least efficient way when considering the speed over ground. The available data per case are shown in Table 4.3, as well as where to find them.

Other differences are related to the waterways. Ships in restricted waterways are limited in their maximum speed, which is a function of the blockage factor. The blockage factor is the frontal area of the ship below the water surface, divided by the channel cross-section area. The higher the blockage factor, the higher the return current becomes. With this higher return current, the speed over water increases for the same speed over ground. A detailed map of the waterway, with depth and with, would make blockage factor calculations possible for each part of a trip, which can be used as input for the (Bi-)LSTM, to improve accuracy. On the other hand, it will also likely increase computation time and effort, since the fairway information would be needed for each vessel at each time step and location.

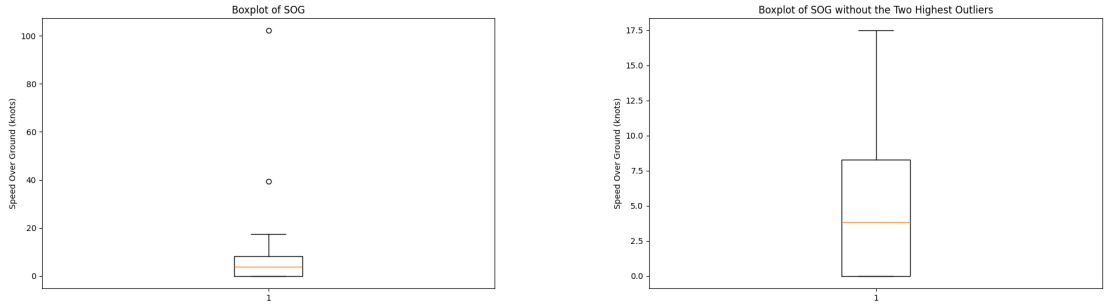
During overtaking operations on a waterway, return currents and water level depressions from the two ships will reinforce each other. The resistance on the ships will in turn increase, as the speed over water effectively goes up (van Koningsveld et al., 2021). Overtaking actions can, therefore, have a large impact on engine power, making it harder to predict.

Variable	Case study 1	Case study 2
Speed over ground	AIS	Measurements
Location	AIS	Measurements
Draught	AIS	-
Depth	Bathymetry	Measurements
MMSI	AIS	-
ME power	Measurements	Measurements
AE power	Measurements	Measurements
Wind speed	-	Measurements
Wind direction	-	Measurements

Table 4.3: Available data for each case

4.3. Data pre-processing

Firstly, the available data should be cleaned. The necessary steps are, in order of execution: outlier removal, speed and distance calculation, acceleration calculation, data synchronization, and the creation of trips. All of these actions, and why they are needed, will be explained below.
Firstly, unrealistic outliers must be removed. This can be done for speed and the location of the vessel. If a vessel moves a large distance in a short amount of time, it is probably due to a Global Positioning System (GPS) error. These outliers must be removed to obtain a more reliable and usable data set. In Figure 4.8, two boxplots are shown of the speeds of the container vessel. For this particular vessel, all speeds are between 0 and 20 knots, except for two measurements of around 40 and 100 knots. These speeds can be considered unrealistic and are therefore filtered from the data. We are left with the boxplot on the right, which indicates more realistic vessel speeds.



(a) Before filtering

(b) After filtering

Figure 4.8: Box plots of speed over ground, before and after filtering outliers

Secondly, additional data may be incorporated as necessary. For instance, the distance between two points is determined by capturing the location for each sample and calculating the geodesic distance from the current point to the previous point. Moreover, the distances to the starting and ending points are computed. The smallest of these distances is saved for the operational mode assignment using Table 2.1. To achieve more realistic and consistent speeds, the vessel speed at each time step was calculated by determining the time difference between two data points, computing the distance, and dividing it by the time difference. Where the original speed provided by the AIS is in knots, the speed is now given in meters per second. This conversion is achieved by multiplying the velocity by 0.51.

In contrast to empirical models, which provide specific formulas, ML models are driven by data. Therefore, it's beneficial to supply ample data to help the algorithm comprehend essential underlying relationships and reduce the required training datasets. This practice is referred to as data augmentation (Hallaj, 2023). Most current emission models are designed to work with sailing resistance and do not operate as well during the acceleration stages of a trip. This is because they often do not consider slip, higher during acceleration (Lamers, 2022). In addition, a decelerating ship uses less energy than an accelerating one for the same speed. Acceleration is computed by dividing the difference in speed by the time delta for each time step.

As mentioned, the depth sensors on the inland tanker are not reliable, according to the data provider. This is also found when looking at the output of the depth sensor. The data provider recommended smoothing to reduce outliers. However, even after doing this, some data points still show depths of over 60 meters, even though the depth on the waterways rarely exceeds 20 meters. From a visual search on online depth charts, the deepest part is 24 meters. The data is smoothed, and all values are limited to 25 meters to make them more reliable. The depths before and after smoothing and limiting are compared in Figure 4.9.

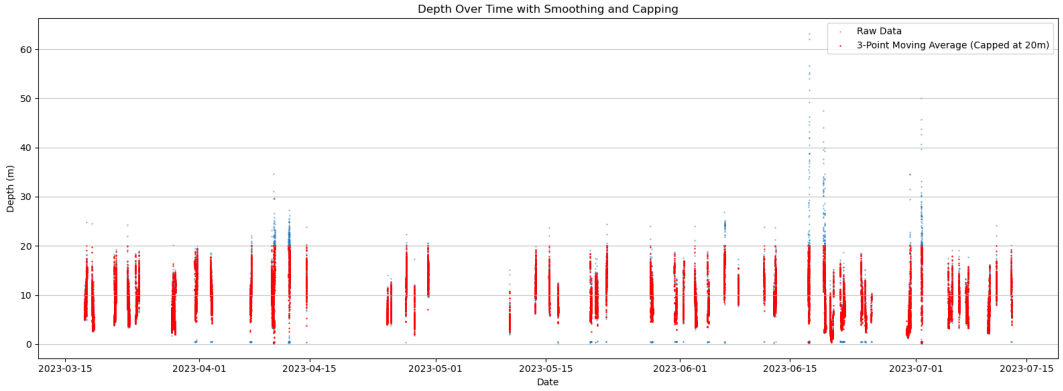


Figure 4.9: Depth before and after smoothing and limiting

It is important to ensure the time steps of the AIS and sensor data line up. Without this, some relevant performance indicators, like the MSE of RMSE, can not be calculated. For the sea-going case study, the sensor and AIS data sets are synchronized to accomplish this. The time interval is set to one minute, the numerical values will be interpolated between the two closest values to that particular minute. The new entry for non-numerical values is the last entry before the full minute. As the inland case study involves a single data set that stores location, speed over ground, measurement data, and environmental conditions, synchronizing is not necessary. However, as not all sensors update at the same frequency, empty entries are interpolated to eliminate missing variables, which could interfere with predictions later.

As stated earlier, the time interval for the inland case is set to one second, meaning there is 60 times more data for the same time frame. However, as seen in Figure 4.4, the engine power is quite static. Therefore it is deemed that a 30-second-interval will be enough to extract relevant data from the set. As a result, the data set is changed so that one sample is taken every 30 seconds, which helps decrease computational load and save time.

Based on the MMSI, all AIS signals are assigned to one trajectory, essentially all movement of one vessel. Next, the trajectory is split up into trips. This study sees a trip as the voyage between one port and the next. Splitting into trips can be done easily using the movingpandas function StopSplitter, which takes the trajectory dataframe and cuts it up when the vessel does not leave an area of a certain diameter for a certain time. This is useful, as the trips need to be cut up when the vessel is at berth, where it normally does not leave for a few hours. To ensure the trips are not stopped while anchoring at sea, the diameter should be small enough so that the vessel's swinging movement around the anchor keeps the function from cutting up the trajectory.

Since a large container ship is considered, loading and offloading times are high, often taking 24 hours or

more. For a generalizable method, values should be chosen that prevent an early cut-off while still correctly identifying the end of a trip. Finding the right values is done by trial and error. From trying multiple combinations of diameter and time interval, it is found that 40 meters is a large enough area to not accidentally end trips. For the interval, 5 hours consistently delineates trips commencing at one port and concluding at another. In total, 20 unique trips are found in the data.

In inland waterways, locking operations often take place. The Volkeraksluizen and Kreekraksluizen are visible as the two places where the ship stops at Figure 4.6. For trip creation, the StopSplitter function is used again. As stopping at a lock takes less time than loading or unloading, this should only stop a trip when it is really in a port. Loading and offloading likely takes shorter than for the container vessel, so using the same stop time of five hours is not recommended. Table 4.2 shows that the ship has a capacity of 5000 m^3 and a maximum onboard pump capacity of approximately 2800 m^3/h, meaning that, in theory, the vessel could be emptied within two hours. However, trips could be stopped during long waiting times at a lock when the interval is set to two hours. For a three-hour interval and a 20-meter radius, all trips start and end at a berth. A three-hour window is chosen with the assumption that ships might not only unload but also load at most ports. Secondly, the ship does not directly start pumping, because it has to be coupled to the shore first. Lastly, the pumps are not likely to function at their maximum capacity at all times. The 2800 m^3/h capacity is theoretical and depends on the type of liquid. Visual checks for the trip creation can be found in **??**.

The process of assigning operational modes is a critical step in understanding vessel behavior and optimizing performance, and is the last step in the data pre-processing. The decision matrix detailed in Table 2.1 is employed. This matrix, developed through extensive research and domain expertise, is used to determine the most probable operational mode for each distinct phase of a voyage. The assigned operational modes are visualized alongside the corresponding trips using multiple techniques. This makes it possible to find potential shortcomings and increase understanding in where to improve the mode assignment.
By dividing trips into operational modes, a distinction can be made in AE power. When berthing or anchoring, the ME power is assumed to be zero. By comparing the measured to the predicted power in these modes, it can be shown if this is the case. In Chapter 5, the error of the IMO/HM model per operational mode will be shown. Consequently, the model's capacity can be assessed to predict each mode accurately.

4.4. IMO/HM model

While the objective is to evaluate the effectiveness of various machine learning models, it's important to note that this assessment has less meaning in isolation. It is important to compare the performance of these models with existing ones in order to determine their added value. One way to directly compare their predictive accuracy is by applying the IMO/HM model to the same test set that was used to evaluate the LSTM models. As the model was developed primarily for sea-going vessels around a port area, this model will only be evaluated for the first case, involving the container vessel.

AIS is used to obtain vessel characteristics and time-dependent variables, like speed and depth. The IHS Markit database is used to retrieve all relevant vessel characteristics. These include the DWT, the number of screws, the number of main engines, the build year, the installed power of main engines, and fuel type. Vessels are found by IMO number and stored in the local vessel database for later calculations. Moreover, the vessel's location is found through AIS, together with a time stamp and the vessel's speed. With the trips defined and the necessary data ready, the IMO/HM model is applied to estimate engine power in kW for each trip sailing segment. The resulting predictions will serve as a benchmark against the performance of the (Bi-)LSTM models, providing a comprehensive assessment of their potential value in enhancing maritime operational efficiency and decision-making.

The correct operational modes must be assigned to implement the IMO/HM model. As this case study focuses on only one vessel, the assignment of modes might be less general and more specialized towards this one ship. The previously shown Table 2.1 is used as a starting point. Trips have already been created, and the assigned operational modes can be checked. In reviewing the results with the original decision matrix, it is evident that the assigned modes do not represent reality well around the port. As seen in Figure 4.10 and Figure 4.11, anchoring is often assumed in this area. A vessel is not expected to anchor in a port basin, blocking the entrance for other vessels. Additionally, if the ship were anchored, it would not slowly move out of the

port. In this case, it would be more logical if the mode were labeled 'maneuvering'. The problem seems to be in the second row of the decision matrix, where, independent of the location, the vessel is always assumed to be anchored when the speed is between 0.5 and 3 knots. Also, between maneuvering and sailing, the mode often switches. As there are probably no significant maneuvers at open sea, a more likely scenario is that the ship is slow steaming, decreasing its speed to save fuel to prevent needing to anchor before the scheduled port call.

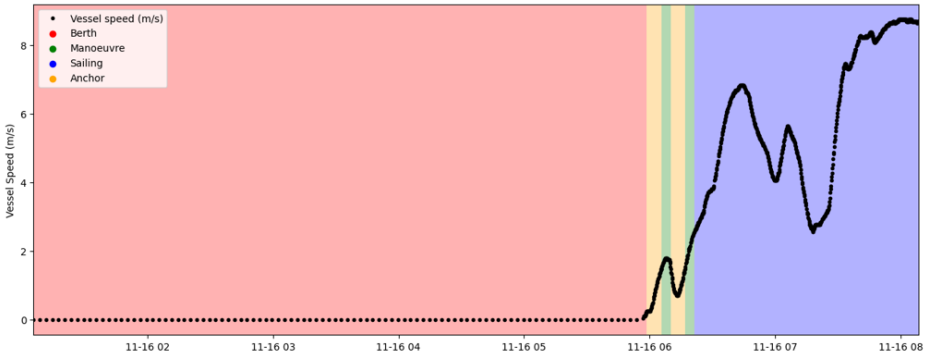


Figure 4.10: Modes assigned to the trip, when leaving the port

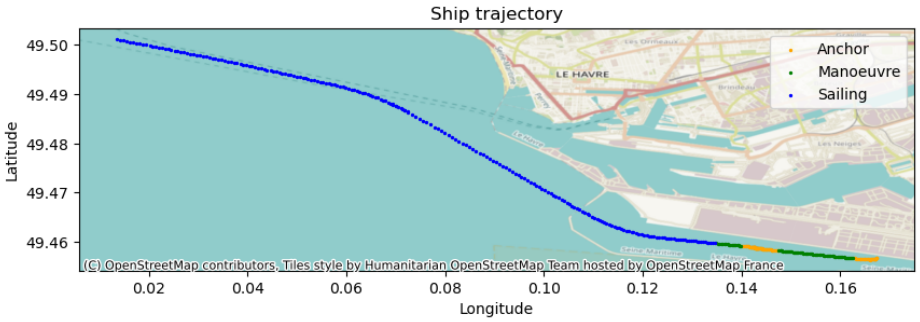


Figure 4.11: Port departure with original mode assignment

When the ship is close to the start or end point of the trip, the likelihood of maneuvering or berthing is higher. The vessel is considered anchoring only when the speed is below 0.5 knots. When the speed is above 5 knots, sailing is the mode that is assumed unless the vessel is already less than one NM from the destination. In that case, it is way more likely that the ship is already maneuvering, as it is within the braking zone before the port entrance. With the same logic, it can be seen as maneuvering when leaving the port, as it still has to change course and is accelerating. The main difference between sailing and maneuvering is mainly the amount of energy used to change speed or course versus the energy needed to counter resistance and keep the same speed due to the help of tugs when maneuvering.

	Distance to destination		
Speed over ground	**< 1 nm**	**1 – 5 nm**	**> 5 nm**
< 1knot	Berthing	Anchoring	Anchoring
1 – 3knots	Maneuvering	Maneuvering	Maneuvering
3 – 5knots	Maneuvering	Maneuvering	Maneuvering
> 5knots	Maneuvering	Sailing	Sailing

Table 4.4: Altered decision matrix to determine operational mode

By changing the decision matrix to the one in Table 4.4, the misdiagnosing of modes is effectively reduced. Also, there is no rapid switching between sailing and maneuvering anymore. Now, the mode is deemed sailing only when the speed is over 5 knots and the distance to the destination is more than one nautical mile. In Figure 4.11, a departure of the container vessel from the port of Le Havre is shown. The same departure is shown in Figure 4.12, using the revised mode assignment matrix. This illustrates that the revised method performs better at assigning the right modes, ensuring that sailing ships are more often outside the port area, where the ship is not only propelled by its engines but also by tug boats. The same decision matrix was applied to the case with the inland tanker vessel. In Figure 4.13 an example is given of the assigned modes for this ship. The ship is assumed to be sailing for almost the entire trip, except for near locks, where it maneuvers just before and after the lock, and is anchored in it. Sometimes, when it has to wait for the lock to open, it is also anchored before the lock.

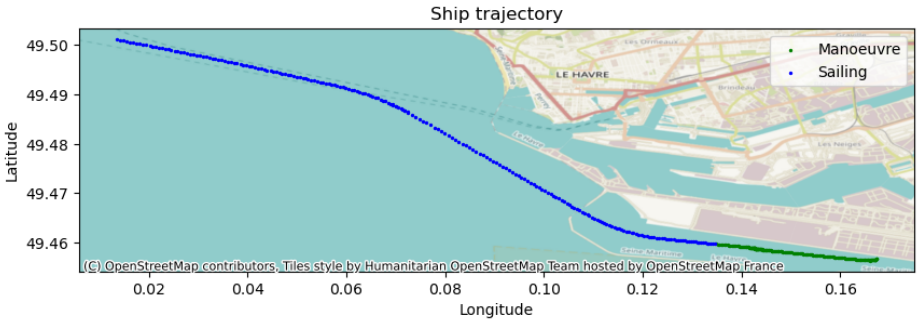


Figure 4.12: Port departure with revised mode assignment

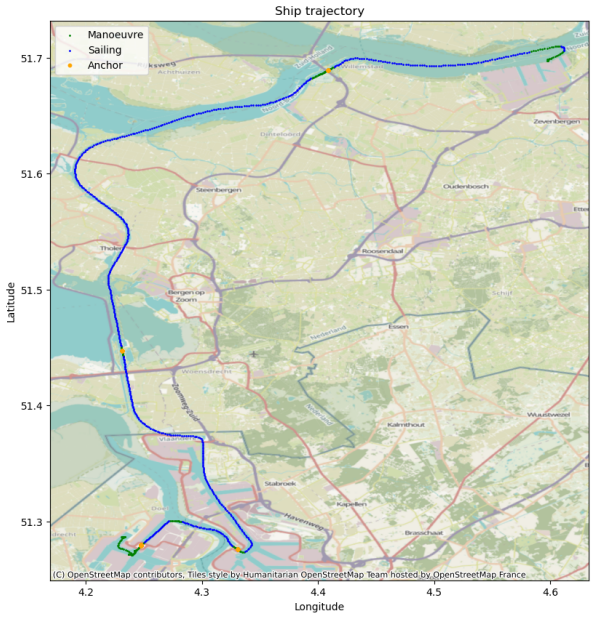


Figure 4.13: Operational modes for case study 2

With the operational modes established, the IMO/HM model will be implemented as explained in Chapter 2. This is done for all trips of the container ship. The exact approach to do so is laid out in Van Den Berg, 2022. Vessel-specific parameters are extracted from the IHS ship database. In Table 4.5 the input parameters are shown. The draught is variable, and found in the AIS signal. The table shows length, width, block coëfficient, bulbous bow shape factor, open water efficiency, relative rotative efficiency, and gearing efficiency. These parameters can be used to calculate all other relevant variables.

The ship length and width are extracted from the IHS Markit data base, together with the engine power. The database shows a maximum ME power of 37600 kW. This is used to calculate engine load. Block coefficient C_b is the fraction of volume of the ship underwater compared to a rectangular section with the same width and length of the underwater part of the ship, where the height of the rectangular section equals the draught. For a container vessel, this coefficient roughly equals 0.66 (Van Den Berg, 2022). η_0, η_r, and η_g represent the open water, relative rotative, and gearing efficiency. Open water efficiencies vary between 0.55 and 0.7. In this case, 0.6 is assumed. To remain consistent with previous studies, a relative rotative efficiency of 1 is assumed for the ship and a gearing efficiency of 0.96.

LOA [m]	Width [m]	Engine power [kW]	C_b [-]	C_{BB}	η_0 [-]	η_r [-]	η_g [-]	$\rho_w[kg/m^3]$
368	51	37600	0.66	0.19	0.6	1.0	0.96	1025

Table 4.5: IMO/HM model inputs

The prediction error is calculated per operational mode, for the old and the new mode assignment. By doing so, the improvement of the mode assessment can be quantified. Afterwards, the predictions of the model will be compared to the outcomes of the ML models, giving a clear view of their performance against a state-of-the-art emission model in Chapter 5.

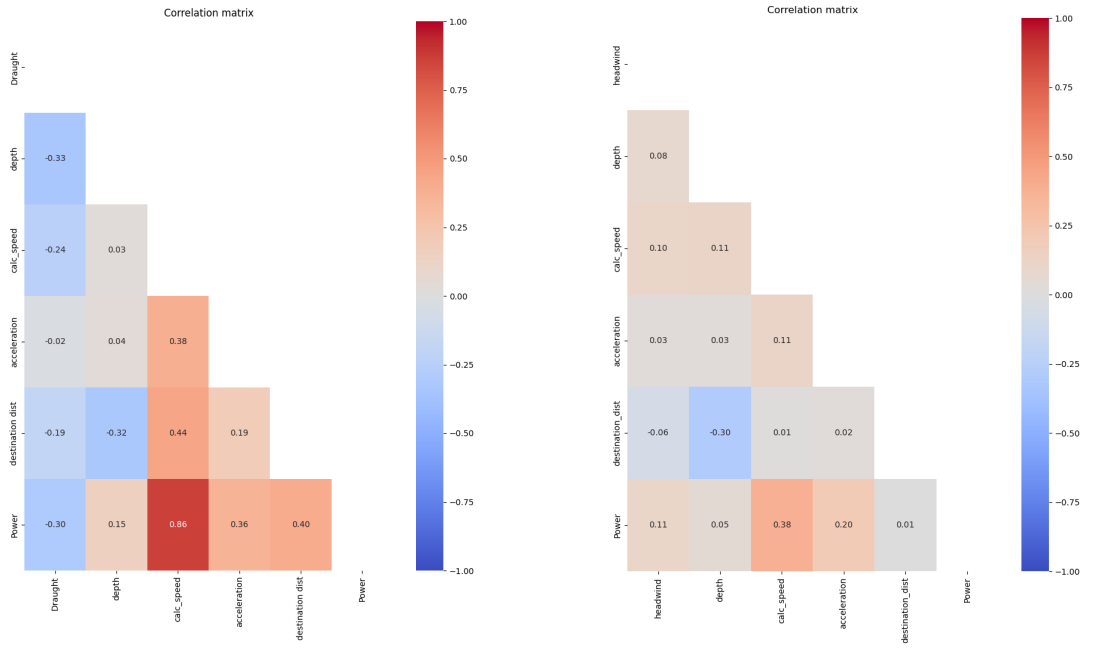
4.5. Machine Learning model

After analyzing the reference model, ML models can be implemented to assess their ability to estimate engine power using multivariate time series inputs obtained from the AIS signal. In Chapter 3, relevant models have been mentioned. Afterward, it discussed the particular characteristics of these models and showed that the focus would be on (Bi-)LSTM models. As mentioned earlier, the models will be compared against one another. For the sea-going vessel, these predictions are compared to the reference model as well. According to Van Den Berg, 2022, 55% of CO_2 emissions are the result of sailing when looking at emissions around a port. As this part is higher when looking at full trips at sea, this is the mode on which the Bi-LSTM models will be trained. The cases are too different to compare directly, so the results will be case-specific, and the models will be trained on only one ship. Because of that, the ship characteristics are non-variable and do not influence the outcome of the used models. It could still be used as proof of concept to show the potential of using ML in ME power prediction as a surrogate to the existing physical model.

Firstly, potentially viable input variables are explored. As both cases offer slightly different information, this will be done twice. A good indicator of the worth of an input variable is its correlation with the prediction target. For both cases, a correlation matrix is calculated for all numerical values in the data using the linear Pearson's correlation.These matrices are shown in Figure 4.14.

In both cases, the strongest correlation with the target variable is the speed of the vessel. After that, the acceleration seems to influence engine power the most. In the sea-going case, the draught negatively correlates with the ME power. The draught is the distance from the bottom of the ship to the waterline and is always a negative number. When this negative number is larger, the ship is loaded more heavily and has a larger wetted surface area, which increases resistance. The inland vessel shows a small correlation with the headwind. Headwind increases resistance as well. Both cases show a small correlation between power and depth. The more shallow the water, the higher the resistance due to return currents. The correlation is likely lower in the second case due to the low reliability of the depth sensor.

From the correlation matrices, the most important input variables are identified. For the sea-going vessel,



(a) Sea-going (b) Inland

Figure 4.14: Correlations between potential input variables and target variable

speed over ground and acceleration are likely the most important indicators of engine power. The draught seems to be most important after the first two. Lastly, the depth is of influence. If the resolution of the bathymetry data was higher, this influence could be larger. For those reasons, the Bi-LSTM models will be trained for the following combinations of input variables:

- Speed & Acceleration (SA)
- Speed & Acceleration & Draught (SAD)
- Speed & Acceleration & Draught & Depth (SADD)

Since other information is used in the inland case study, other combinations of input variables are employed. Speed and acceleration show the highest correlation to engine power again, although less than in the other case study. The third most important factor seems to be the headwind. Lastly, probably due to the low reliability of the sensor, shows a low correlation. It is still incorporated into the training to assess whether it improves prediction.

- Speed & Acceleration (SA)
- Speed & Acceleration & Headwind (SAH)
- Speed & Acceleration & Headwind & Depth (SAHD)

When setting up the LSTM models, the data are first split into a training set, a validation set, and a test set. The training is done on the training set, not the test set. The test set only assesses the model's performance after all training. This is done to ensure the model works with unseen data. Validation data will be used in training but only to validate the model between epochs, which should prevent over-fitting. All performance indicators in this research will describe the difference between the predictions and actual values based on this test set. 80/20 is a split that is often used in machine learning. This means that 80 percent of the data is used to train, and 20 percent is used to assess the performance against unseen data. The splitting is done over the set of trips, regardless of the length of trips. A random state of 42 is specified to ensure replicability.

There are 20 sailing trip segments for the sea-going case, and 94 for the inland case. As the trips need to be split into a training and test set, there is not much left in the sea-going case. To increase the available training

data, the time series are cut up into frames, where the frames overlap to artificially create more data, helping to provide more training information without replicating frames. As a result, the start and end are different each time, but patterns are kept. This can help train ML models in some cases, which was shown in Gao et al., 2019, and is considered a form of data augmentation. It is often used in speech recognition.The frame size is set to 60 minutes, with 45 minutes of overlap. In other words, each next frame loses 15 minutes at the start and gains 15 at the end. This way, there are more sequences to train on that keep the same structures, but have a different start and end.

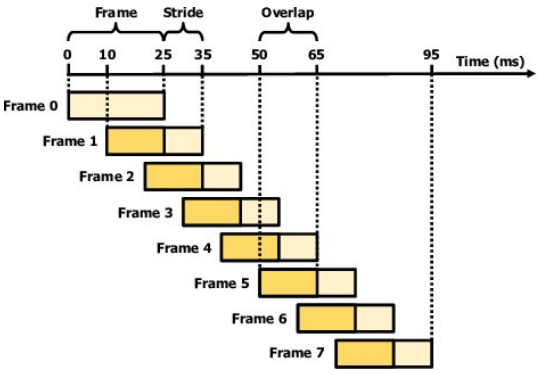


Figure 4.15: Example of overlapping frames (Gao et al., 2019)

Next, all data is scaled. It is a method to normalize the data within a particular range, usually between zero and one. Since features are on a similar scale, this can help convergence of stochastic gradient descent (Natarajan, 2023). It is crucial to perform this scaling after the data has been split. If this step is done incorrectly, the scaler will be partially fitted on the test set, inadvertently causing the test data to affect predictions. Doing so can result in the model performing unusually well on the test data compared to other unseen data, a situation known as data leakage.

4.6. Hyperparameter tuning

One of the most important parts of machine learning is the HP tuning process. HPs are parameters of the model that influence the architecture. They are not parameters that the model learns through the learning process. Possible HPs that can be altered are batch size, the optimizer type, learning rate, the number of layers, size of layers, activation functions, and the performance metric. As most of these HPs are discrete, gradient descent can not be used to optimize them (Prince, 2023). The goal is to identify the optimal configuration that minimizes the prediction error on the validation set, ensuring the model's ability to generalize to new, unseen data.

The brute-force approach, known as grid search, involves testing all possible combinations of hyperparameters, which is computationally expensive and time-consuming. A more efficient alternative is Hyperopt, a Python library for hyperparameter optimization. Hyperopt uses a Bayesian optimization approach, TPE, a probabilistic model-based method for finding the minimum of a function. In the context of hyperparameter tuning, the function to be minimized is the validation error of the machine learning model.

The process starts by randomly selecting a set of hyperparameters and evaluating the model's performance on the validation set. The prediction is classified as 'good' or 'bad', after which a probabilistic objective function model is updated for each evaluation. The probabilistic model is used to guide the selection of the next set of hyperparameters to find a set that will likely result in a lower validation error. The predicted likelihood of a point giving a 'good' result (l(x)), divided by the likelihood of a 'bad' one (g(x)), is the promisingness score. HP configurations with a higher promisingness score are expected to yield better results and thus will be tried. This iterative process continues, with the probabilistic model being updated after each evaluation until a stopping criterion is met, such as a maximum number of evaluations or a desired level of performance (Berk, 2020).

Hyperopt's Bayesian optimization approach is more efficient than grid search because it focuses on exploring

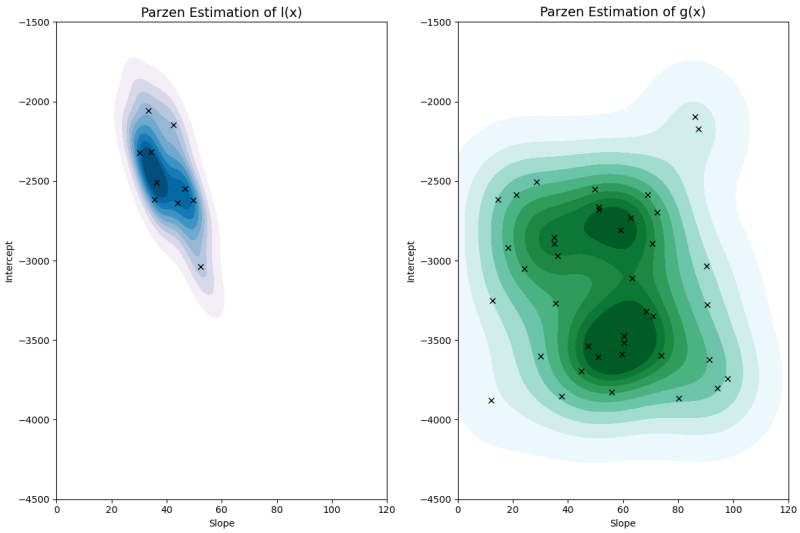


Figure 4.16: Example of Tree-structured Parzen Estimator with two variables (Yates, 2020)

areas of the hyperparameter space that are more likely to yield good results based on the information gathered from previous evaluations. This allows it to find a good set of hyperparameters with fewer evaluations, saving computational resources and time.

In both case studies, an LSTM and a Bi-LSTM model will be tuned and tested. This will be done for 150 epochs during 50 evaluations. The best-performing model over the epochs is saved, together with its loss. The best model is saved for all evaluations. This saved model can later be used for testing and will produce the same results each time for the same data, meaning its results are reproducible. The same options and ranges are used in both cases, but the specific HP configuration might vary based on the case-specific needs and provided data. These ranges are shown below, in Table 4.6. The output of the (Bi-)LSTM is the same size as the number of LSTM units. Therefore, the output is a vector of length (LSTM units). To interpret this output, there will always be one fully connected layer at the end to ensure each time step gets one value returned. The size of this fully connected or dense layer is also an HP.

Hyperparameter	Range
Batch size	2, 4, 8, 16, 32
LSTM layers	1 - 3
LSTM units	10 - 50
Dropout	0 - 0.3
Size of dense layer	8 - 32
Optimizer	Adam, RMSprop, SGD
Learning rate	0.0001 - 0.10

Table 4.6: Grid for hyperparameter tuning

The choice for these HPs and what they change in the architecture or training is explained below.

Batch Size: The batch size is a crucial hyperparameter that determines the number of samples processed before the model updates its internal parameters. Smaller batch sizes can lead to faster learning and better generalization. However, they can also introduce more noise during training. Larger batch sizes are computationally more efficient and can stabilize training by providing smoother gradient estimates. However, they might require more epochs to converge to a good solution. All options are a power of two because "The number of batch sizes should be a power of 2 to take full advantage of the GPU processing." (Kandel and Castelli, 2020). This means that the training should be more efficient when using a power of 2, which decreases training times. This research chooses a range of [2, 4, 8, 16, 32] to explore the trade-off between stability and generalization. Speed could also be a factor, but it is not used to measure performance in this research.

LSTM Layers: The number of LSTM layers controls the model's capacity to learn complex representations. Deeper networks with multiple LSTM layers can capture more intricate patterns and hierarchical relationships in the data. However, increasing the number of layers also increases the risk of over-fitting, especially when training data is limited. Therefore, a range of 1-3 LSTM layers is considered to balance model complexity and generalization.

LSTM Units: The number of units (or neurons) in each LSTM layer determines its representation power. More units can capture finer details and nuances in the data but also increase the model's complexity and over-fitting risk. A range of 10-50 units is chosen to explore the optimal number of units that can effectively capture the relevant patterns in the data without over-fitting.

Dropout: Dropout is a regularization technique that randomly deactivates a fraction of neurons during training, which helps prevent over-fitting by reducing the model's reliance on any single neuron and encouraging it to learn more robust representations. A range of 0-0.3 is considered to find a dropout rate that can effectively regularize the model without hindering its learning capacity.

Size of Dense Layer: The dense, fully connected layer is usually placed after the LSTM layers to produce the final output. The size of such a layer determines the dimensionality of the output and affects the model's ability to map the learned representations to the desired output. A range of 8-32 neurons is chosen for the dense layer to explore different output dimensionalities and their impact on the model's performance.

Learning Rate: This hyperparameter determines the step size at which the model's internal parameters (weights and biases) are updated during each iteration of the optimization algorithm. A higher learning rate allows for faster convergence but may risk overshooting the optimal solution, leading to fluctuations in the loss function. Conversely, a smaller learning rate results in slower convergence but offers a more stable and potentially more accurate solution. In this study, a range of 0.0001 to 0.10 is explored.

Optimizer: The optimizer is an algorithm that guides the learning process by determining how the model's parameters are adjusted based on the calculated gradients of the loss function. Different optimizers employ various strategies to navigate the parameter space and reach the optimal solution. Three common optimizers are considered in this research:

- **Adam**: Adam (Adaptive Moment estimation) is a popular optimizer that combines the benefits of two other optimizers, RMSprop and AdaGrad. It computes adaptive learning rates for each parameter, making it efficient for large datasets and complex models.
- **RMSprop**: RMSprop (Root Mean Square Propagation) is another adaptive learning rate optimizer that addresses the issue of rapidly diminishing learning rates in traditional stochastic gradient descent. It maintains a moving average of squared gradients to normalize the learning rate, leading to faster convergence.
- **SGD**: SGD (Stochastic Gradient Descent) is a simple yet widely used optimizer that updates the model parameters based on the gradients calculated from a mini-batch of training data. While it can be slower than adaptive optimizers like Adam and RMSprop, it often finds good solutions with proper tuning of the learning rate.

4.7. Generalizability

The research data is derived from two vessels: a sea-going container vessel and an inland tanker. The significant disparity between the two cases and the limited number of ships makes it challenging to consider ship-specific parameters. For instance, if the vessel size is used as an input, there are only two distinct values. This is akin to performing a regression with just two data points, resulting in a simple linear relationship between power and size. Therefore, it is not possible to assess generalizability of the ML models. However, some methods are suggested as to how to make a generalizable model in future research.

AIS-based emissions models are the subject of this research. The advantage of these models is that they can easily be scaled to work for a large range of vessels. For example, they can be used to map all emissions in a port area, like in Van Den Berg, 2022, or on a specific waterway, as seen in Segers, 2021. For this to work, ship-specific attributes must be taken into account. The resistance is based on the shape of the hull, but also on draught, width, and length of the ship. Wind conditions have different impacts based on the area of the vessel above water. Energy consumption is also influenced by the weight of the total ship, including cargo. Lastly, the engine age/type, type of propellers, number of propellers and fuel type influence the efficiency of the propulsion.

In other words, a universal model is needed that can generalize across various vessels. Although this ability can not be tested with the data that is currently available, a probable solution is included. In Rahman et al., 2020, the possibility of using static, as well as dynamic variables in LSTM sequence classification is explored. Three methods are proposed: separate training, direct input, and indirect input. Separate training is, as discussed, not suitable for the proposed goals of this research. Direct input is where the static variables are concatenated to the dynamic variables in each time step. In other words, static variables get stretched into a vector of length 'T'. This way, each LSTM cell takes in extra information about the ship. In the second method, indirect input, only the dynamic variables are passed through the LSTM. The static variables are then passed through an FNN, after which the two outputs are concatenated before being forwarded to the output layer. The indirect method performed slightly better than the direct method, for that classification case.

4.8. Assessing performance

In Chapter 5, the results of the HP tuning will be shown and discussed in addition to the results of the predictions. There are three performance indicators that tell something about the accuracy of the models. First of all, there is the MAE. The MAE is the average of the absolute difference of each singular prediction, meaning that the deviation between the prediction and the target is measured at each time step. This is especially important when looking at the spatio-temporal emission of NO_x or particulate matter. When looking at the large-scale emissions of GHGs, this metric is less descriptive. The total fuel consumption that is predicted could still be accurate with a high MAE, if the errors are balanced between over- and underestimation. The same holds for the RMSE. This is similar to the MAE in the fact that it looks at all predicted time steps, but now it takes the root of the mean of squared errors, which penalizes larger errors more. The errors are more constant if the RMSE is close to the MAE in magnitude. Lastly, the total percentage error is looked at. To do this, the area under each prediction curve is taken and compared to the one under the target curve. It signifies the difference in total energy used. If the total percentage error is zero, the energy consumption is exactly the same as the target. When looking at large-scale GHG emissions, this metric is most descriptive.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{4.2}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{4.3}$$

$$\text{Total Percentage Error} = \frac{\sum \hat{y}_i - \sum y_i}{\sum y_i} \times 100\% \tag{4.4}$$

4.9. Conclusions on Materials and Methods

This chapter was allocated to lay out the available materials and explain the methods that are used to answer the sub-question: **"How do ML models, trained on measurement data, perform against an existing state-of-the-art emission model?"**

To answer this question, two case studies are performed. In the first case study, a sea-going container vessel is followed. In the second one, the subject is an inland liquid bulk tanker. As the ships, the data, and the environments differ significantly, both case studies have their own challenges and opportunities. The first case study is characterized by a ship with a more common combustion engine, whereas the second case study follows a diesel-electric ship. In the first case study, all data has to be combined from different sources, with

challenges in synchronizing and completing partially empty data sets. The challenges in the second case study lie more in the fact that the ship sails on restricted inland waterways, where blockage factor and current play a bigger role. Opportunities lie in the fact that wind data is available, and no data synchronization or interpolation has to take place.

The used methods for data pre-processing, assignment of modes, selection of input variables, and HP tuning are all explained. The original mode decision matrix is used to assess how well this works, especially near a port, and a revised method is proposed. Visual checks are performed to validate this method. Correlation matrices help select features that might be important for training ML models in predicting ME power. Based on the correlations, different sets of input variables are introduced. Lastly, the process of HP tuning is explained and the metrics are determined that will be used to assess model performance.

In addition, this chapter aimed to answer the question: **What is needed to make an ML model that predicts ME power using AIS generalizable?** Firstly, the important (static) variables are mentioned. These include ship type, length, width, DWT or GT, waterway characteristics, engine age, year of build, and more. A short literature review signifies that LSTM models can work with static and dynamic variables. Three methods to do so are explored. Relevant literature mentions that two of these methods work well: direct and indirect input. In the tested classification case, the indirect method slightly outperforms the direct one.

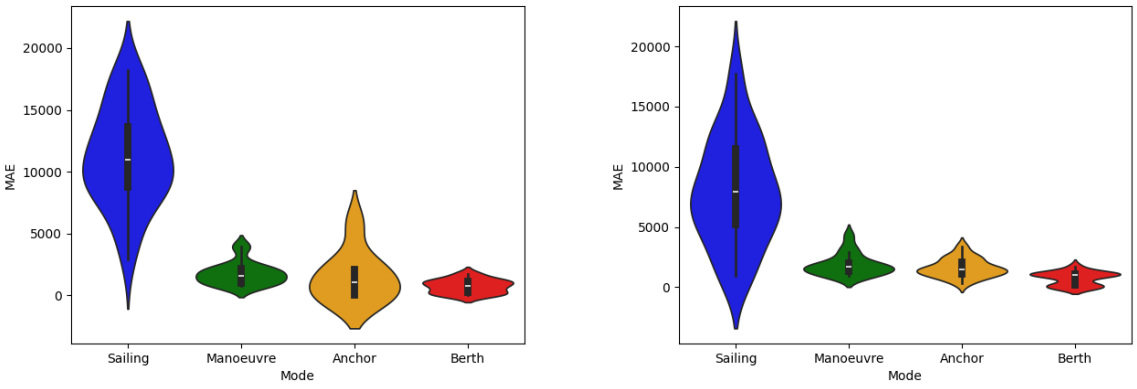
5
Results

From the materials and methods provided in Chapter 4, results will be discussed in this chapter, starting with the output of the IMO/HM model. The errors per operational mode will be quantified. Subsequently, the performance of the ML models will be compared to the target data. For the ML models, the relevant performance indicators mentioned in Chapter 4 will be calculated. The same is done for the IMO/HM model. Finally, combinations of HPs that worked best are shown.

5.1. Case study 1: sea-going container ship

5.1.1. Mode assignment

In Figure 5.1, violin plots illustrate the RMSE per mode for all trips of the sea-going vessel. The width of the violin at any given point indicates the likelihood of errors clustering around that value, with the white horizontal line representing the modal, while the vertical bars denote the interquartile ranges. It is evident from the plot that the sailing mode exhibits the highest prediction errors. Initially, anchoring shows higher errors compared to maneuvering, which can be attributed to the misclassification of the anchoring mode. As the ME power is assumed zero during anchoring, a relatively low engine power can already result in a high error. However, with the revised decision matrix, such misclassifications are significantly reduced. For sailing, the widest point of the violin moves from roughly 10,000 to 7,000. A similar trend is observed for anchoring and berthing modes, where attributing the low speeds to maneuvering results in fewer prediction errors. The plots underline the importance of improving predictions during the sailing stages of trips.



(a) With old decision matrix

(b) With new decision matrix

Figure 5.1: Comparison of MAE with old and new mode assignment

Next, the performance of the final model is evaluated against the actual power output using MAE, RMSE,

and the total percentage difference is compared to the real-world scenario. These three metrics are of importance for different reasons. On one side, shipping companies and governments place significant importance on total emissions and energy consumption to determine if climate objectives are achieved and to avoid excessive fueling. Conversely, examining the variation at each time step is crucial to verify the accuracy of the spatio-temporal emission predictions. If this condition is satisfied, the model can also be applied to simulate emissions such as CO_2 or NO_x in a port region.

5.1.2. HP tuning

The process of HP tuning was explained in Chapter 4. The results of this process are shown in Table 5.1.

	LSTM			Bi-LSTM		
HP || Inputs	SA	SAD	SADD	SA	SAD	SADD
activation functions	linear	linear	ReLU	linear	ReLU	ReLU
LSTM layers	1	3	1	3	1	1
LSTM units	40	10	50	10	10	30
dense units	32	32	8	24	8	8
dropout	0,07	0,09	0,13	0,05	0,06	0,05
batch size	2	8	2	8	32	2
optimizer	rmsprop	rmsprop	adam	sgd	rmsprop	sgd
learning rate	0,0003	0,0025	0,0002	0,06	0,0004	0,0006

Table 5.1: Results of HP tuning for the first case study

One might assume that the number of LSTM units or layers would increase with more input variables, as the predictions become more complex. However, as observed, this is not the case. One possible explanation is that the tuning process may not identify the best performing configuration. The TPE algorithm initializes with random guesses and then calculates a promisingness score. With eight hyperparameters to optimize, there are likely many local minima in the solution space. This does not necessarily mean that the current configurations are poor, but it is reasonable to assume that with more tuning, the results could be slightly improved.

5.1.3. Prediction results

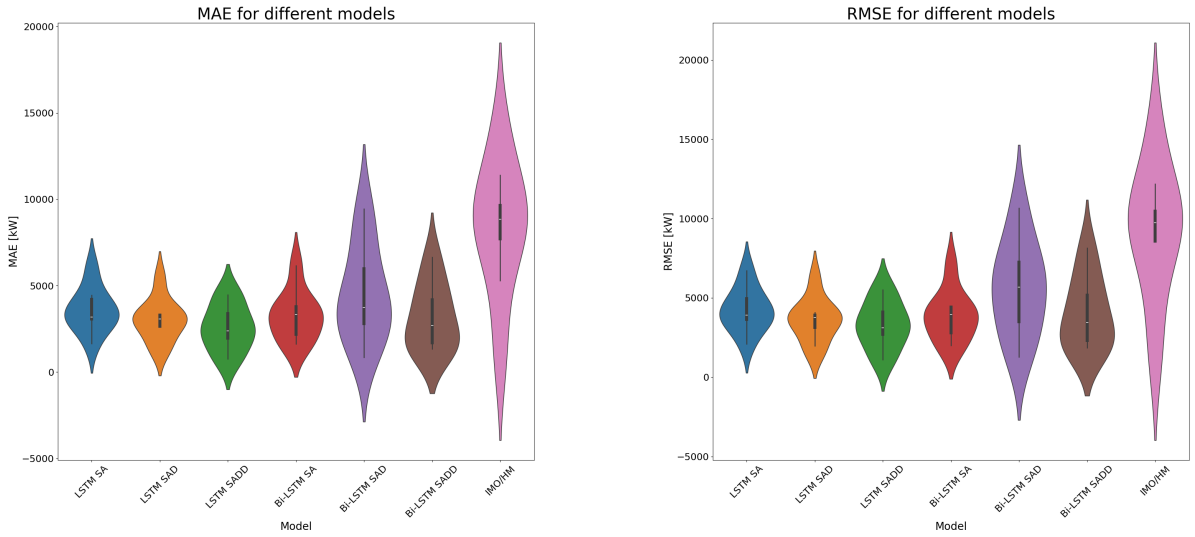
The results of predictions for the first case study are shown below, in Table 5.2. In this table, the MAE, RMSE, and total percentage error are shown, which will be used to identify the best performing model and compare it to the benchmark IMO/HM model.

In the initial case study, only limited data is available, comprising just eight trips in the test data. To augment the training set, it was cut into overlapping frames in order to artificially create more data. While additional test data would be beneficial, there is an argument to be made that eight trips may be sufficient, considering they each consist of roughly 340 to 1200 data points. Collectively, around 6500 time steps are predicted. These points should provide an adequate basis for evaluating the MAE and RMSE to assess the performance. However, on the other hand the total percentage error is calculated based on the total energy used per trip, resulting from only 8 data points. As a result, it is considered to be a less reliable indicator compared to the MAE and RMSE. The use of the limited dataset based on eight trips is however important in the determination whether the prediction predominantly over- or underestimates.

Model	Holtrop Mennen	LSTM model			Bi-LSTM model		
Input	Speed, Depth, Characteristics	SA	SAD	SADD	SA	SAD	SADD
MAE (kg/h)	9542	3067	2974	2863	2820	4190	4127
RMSE	10833	3760	3644	3570	3505	5008	5054
Total error (%)	+62.87	-4.05	+10.42	-3.15	-11.03	-1.32	-0.82

Table 5.2: Results for all models

The results revealed in Table 5.2 demonstrate that all models appear to surpass the benchmark. In each model-input combination, the RMSE exceeds the MAE, but only by a small margin, suggesting the absence of significant outliers. The LSTM model with SADD inputs exhibits nearly the lowest MAE and RMSE, trailing closely behind the Bi-LSTM with SA inputs. Nevertheless, the analysis of error distribution across the test sequences, as depicted in Figure 5.2, illustrates a narrower range of MAE and RMSE for this particular combination. The IMO/HM model displays the highest errors on average, as well as the largest spread, indicating it has the lowest accuracy and precision of all models.



(a) MAE

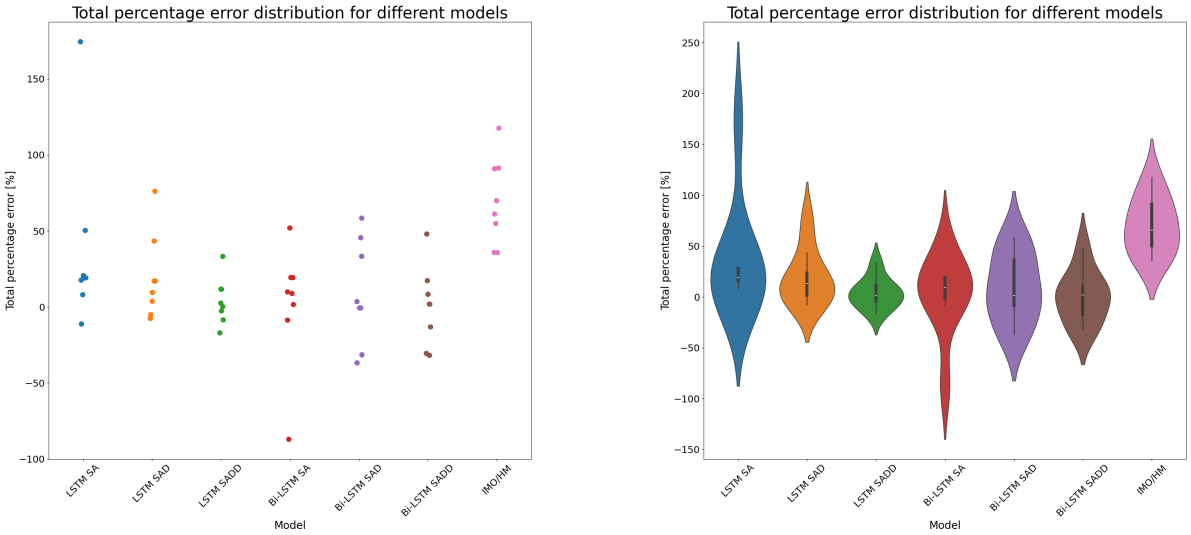
(b) RMSE

Figure 5.2: MAE and RMSE for all models - seagoing

In Chapter 4, correlation matrices were used to help choose the different combinations of input parameters. The speed over ground had the highest correlation with the engine power, followed by acceleration, draught, and depth. The results are in line with the correlations. Speed and acceleration alone already give a decent prediction of engine power. When draught and depth are added as input, the accuracy increases for the LSTM as well as the Bi-LSTM.

Although the Bi-LSTM model outperforms in terms of average percentage error, it exhibits higher MAE and RMSE. The complexity of the model may be the reason behind its robustness to changes, possibly leading to under-fitting. This would also explain why the errors increase with more input variables. LSTMs however do perform better with more input variables. This could be attributed to an increased contextual understanding of the situation, especially when considering physical implications in maritime operations. Understanding the long-term trends in a ship's speed and acceleration, but also depth and draught could potentially enhance predictive capabilities.

Figure 5.3 shows the distribution over the total percentage errors, using a strip plot and a violin plot. The violin plot expresses the density of a given result, while the strip plot helps identify the magnitude of outliers. Together, they provide a clearer picture of how each model would predict total emissions in a large-scale study. For this goal, the Bi-LSTM-SADD has the highest accuracy, with an average error of -0.82%. However, the LSTM-SADD has a lower spread of errors, making it more precise. The average total percentage error is -3.15%, which is still an improvement over what was seen in the STEAM model, that was explained in Chapter 2, with its average error of 6%. The Bi-LSTM-SADD has a spread between -35 and +50, while the LSTM-SADD stays within -20 to +35 percent.



(a) Strip plot

(b) Violin plot

Figure 5.3: Total percentage errors - seagoing

An example of a prediction for a sailing segment from the best performing LSTM model is shown in Figure 5.4. The predicted power closely mirrors the actual power, although the predicted power appears to be more consistent. This phenomenon may be attributed to varying currents. The speed considered in this scenario is the speed over ground. If this speed remains constant, the predictions are not likely to change significantly. Although not included in this research, changes in currents relative to the vessel may result in increased engine power for the same speed over ground. AIS data do not contain information regarding currents and consequently heights of currents do not manifest in the prediction.

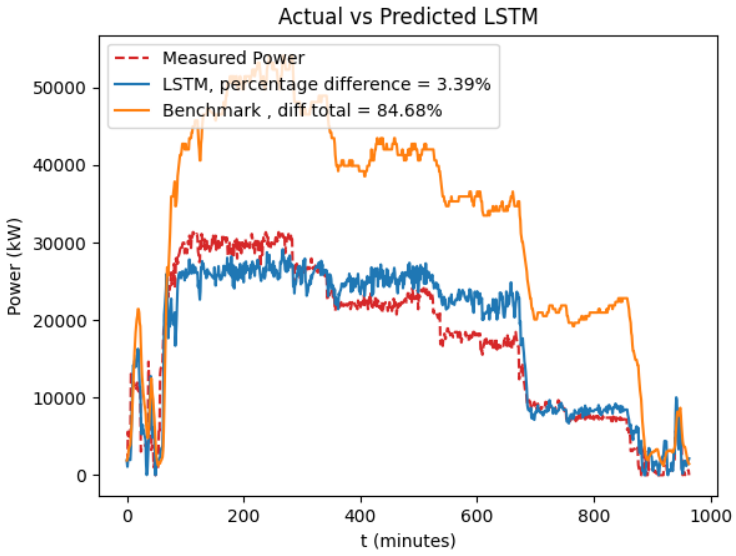


Figure 5.4: Prediction of ME power for one trip - seagoing

5.1.4. Reflection of the first case study

This chapter presented the initial case study, validating the reference IMO/HM model. The tests showed that the model performs adequately, although there is potential for improvement. The main area needing enhancement is the assignment of operational modes. Specifically, the anchoring mode was assigned too often when it should not have been. Modifying some of the anchoring modes in the decision matrix for maneuvering reduces the MAE for all modes. This modification is more realistic, as the MAE should be zero for anchoring, given that the model predicts that the ME is turned off.

The operational mode 'sailing' exhibited the highest RMSE. Consequently, and because the ME provides the most power during sailing, the models were exclusively trained on this mode. Since they output a sequence, it can be seamlessly integrated back into the IMO/HM model. The performance of the benchmark, the LSTM, and the Bi-LSTM was compared against the target sequences. Hyperparameter tuning for both ML models was conducted with TPE using Hyperopt. Parameters such as batch size, layer size, and the number of layers were adjusted to identify the best-performing model. Ultimately, the LSTM emerged as the superior option for the most precise time-step predictions. Compared to the baseline model, the MAE decreased from 9542 to 2863, and the total fuel consumption error reduced from +62.87% to -3.15%. If total energy consumption is a priority, Bi-LSTMs become appealing. Although the MAE is higher at 4127, the total percentage error is only -0.82%. For local emissions, such as NO_x, LSTMs may be preferable, whereas for global emissions estimates, Bi-LSTMs could be more valuable. For both models, the best results were achieved using speed, acceleration, depth, and draught as inputs.

It must be noted that the (BI-)LSTM models are trained for one ship in a particular area, mostly around the North Sea and the English Channel, whereas the IMO/HM model should be usable everywhere. Since trips are relatively short, the shipping line might plan them to align with tidal currents to save fuel. This could be the reason for the high overestimation of the benchmark. If the ship would sail against this current, the LSTM models would then be expected to underestimate actual engine power. The speed over water is a better indicator of engine power, but

5.2. Case study 2: inland tanker ship

The second case study followed a diesel-electric ship in inland waterways. This scenario poses its own challenges, as will be seen later in this section. As for the first case study, three different combinations of input variables are used. This section will show the results of HP tuning and prediction, to assess the models' capability in prediction ME power consumption for inland shipping.

5.2.1. HP tuning

Again, Hyperopt was used to find a suitable set of hyperparameters. The found HPs are shown in Table 5.3.

	LSTM			Bi-LSTM		
HP Inputs	SA	SAH	SAHD	SA	SAH	SAHD
activation functions	relu	linear	linear	ReLU	ReLU	linear
LSTM layers	1	3	1	3	2	3
LSTM units	50	30	10	30	40	50
dense units	8	24	16	8	32	24
dropout	0,12	0,28	0,16	0,08	0,08	0,05
batch size	32	16	4	8	2	2
optimizer	adam	sgd	sgd	adam	sgm	adam
learning rate	0,002	0,019	0,0007	0,005	0,005	0,0005

Table 5.3: Results of HP tuning for the second case study

In this case study, greater regularization in the form of dropout is needed compared to the first one. It is interesting to note that the dropout rate is lower for the Bi-LSTM than for the LSTM, indicating that Bi-LSTMs generally require less regularization overall. This could be due to the fact that predictions in Bi-LSTMs are influenced by both past and future data, inherently providing regularization. Similar to the previous case, there is no clear correlation between the number of LSTM layers and the input variables. Additionally, the learning rate decreases as the number of input variables increases. During hyperparameter tuning, there is no definitive consensus on whether ReLU or a linear activation function is preferable for the dense layer. However, considering that it's physically impossible to have negative engine power, using ReLU, which always produces positive outputs, may be the more reliable choice.

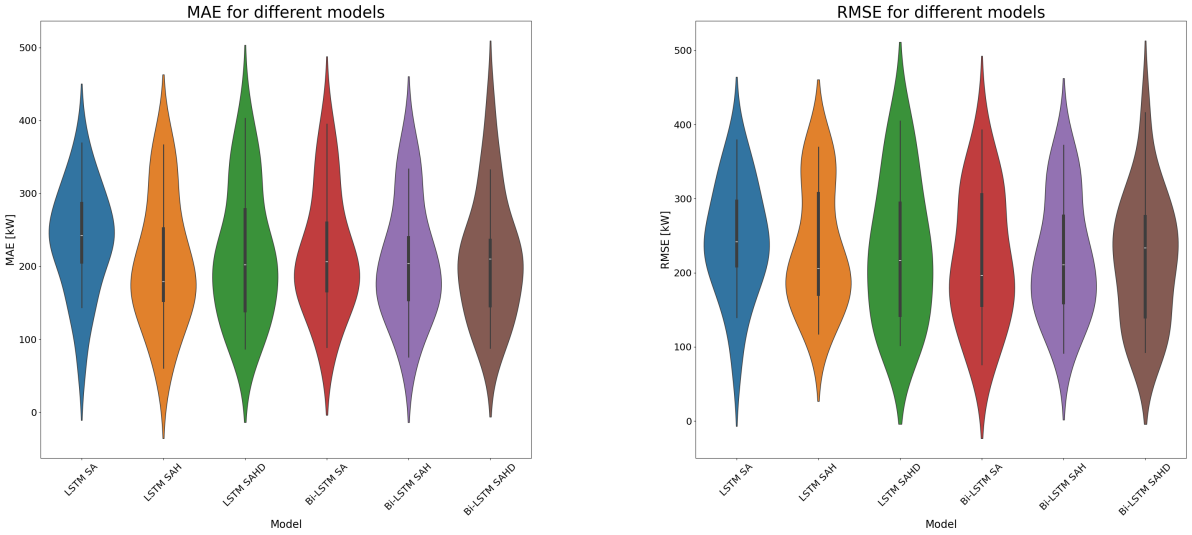
5.2.2. Prediction results

Model	LSTM model			Bi-LSTM model		
Input	SA	SAH	SAHD	SA	SAH	SAHD
MAE (kW)	236.29	217.28	217.20	201.25	182.45	213.52
RMSE	244.42	227.23	229.83	205.65	198.65	219.01
Total error (%)	-4.79	+3.41	+5.11	+19.16	+13.40	-4.15

Table 5.4: Results for all trained models after training

In this second case study, the results presented in Table 5.4 reveal that the Bi-LSTM-SAH model, using speed, acceleration, and headwind as inputs, demonstrates the lowest MAE and RMSE when predicting the main engine power of the inland tanker vessel. However, the MAE of 182.45 kW, which is more than 10% of the maximum combined engine power of 1600 kW, indicates that the model's predictions can still deviate significantly from the actual values. The LSTM-SAH model, incorporating speed, acceleration, and depth, shows the lowest average percentage error for predicting energy consumption over entire sailing segments. However, its MAE is higher than the Bi-LSTM-SAH model, suggesting that it might not be as consistent in its predictions. These results are in line with the correlations shown in Figure 4.14b, where a weak correlation between depth and engine power was found (0.05). The headwind, although only having a Pearson correlation of 0.11, does improve predictions for both models.

When comparing all the models in Figure 5.5, it is clear that there is a significant variation in both MAE and RMSE. Upon comparing the two metrics, it is evident that the violins for the RMSE are now thicker at the top, indicating the presence of more large outliers compared to the first case study. The ranges of MAE and RMSE are closely aligned over all models.

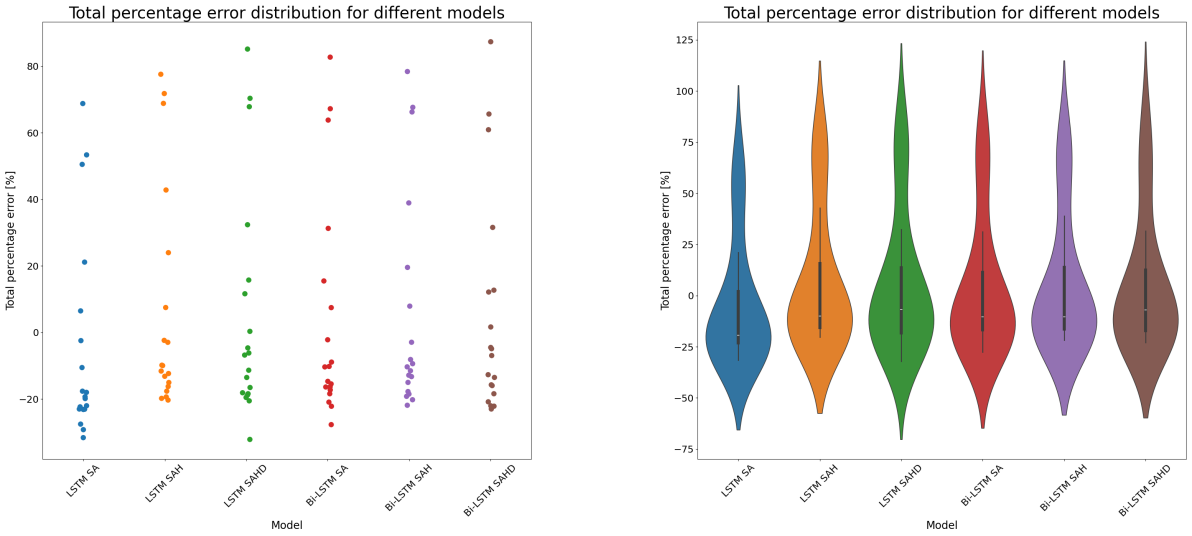


(a) MAE

(b) RMSE

Figure 5.5: MAE and RMSE for all models - inland

In reviewing Figure 5.6, it is apparent that the predictions deviate significantly from their intended values. The percentage errors for all models range from -20 to +70 percent. This discrepancy can be attributed to the absence of crucial information about the waterway. While the models perform reasonably well in open waters, various influential factors come into play in constrained waterways. Without accounting for variables such as current, blockage, and reliable depth data, achieving highly accurate predictions is likely unattainable.



(a) MAE

(b) RMSE

Figure 5.6: Total percentage errors - inland

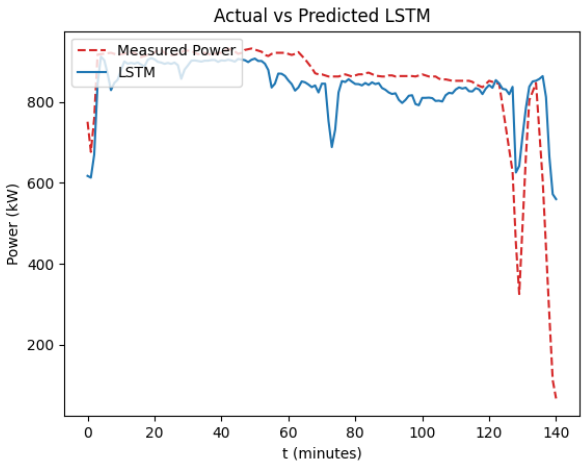


Figure 5.7: Prediction of ME power for one trip - inland

Significant errors are evident when analyzing individual trips, as depicted in Figure 5.7. It seems that LSTMs introduce noise in their predictions, even when the training data has minimal noise. While this discrepancy was less conspicuous in the first case study, this can likely be attributed to the noise present in both the training and test data. The model accurately anticipates the dip in engine power around minute 130; however, the prediction error escalates thereafter. This could be due to unaccounted conditions, such as waterway width or overtaking actions, that are not factored into the model's input. Additionally, the depth sensor's unreliability could lead to an unseen decrease in depth, thereby raising resistance and diminishing speed over ground. A lower speed over ground would yield a lower ME power estimate, without the model discerning the true cause.

5.2.3. Reflection of the second case study

In comparing the predictions to those of case study 1, it is evident that the predictions in the first case study are more precise. This discrepancy can be attributed to a variety of factors, many of which are associated with the limitations imposed by the restricted waterways.

The waterways that the ship navigates are typically rivers or channels. Consequently, ships are constantly navigating either with or against the flow. This presents potential challenges when training models. In this specific scenario, where flow data is not accessible, only the speed over ground is considered. Figure 5.8 shows a partial trip from Rotterdam to the Volkerak Locks. For this trip, the speed over ground of the vessel is shown. Next to this, the engine power is shown for comparison. The flow of water is directed from east to west, toward the North Sea. Whereas the engine power stay mostly constant, between 850 and 900 kW, the speed of the vessel increases when entering the Hollands Diep and moving with the flow, instead of against it. It is likely that the speed over water does not change significantly in this situation. However, the speed over ground changes by little over 1 m/s, or 20%.

In restricted waterways, ships are limited in their maximum speed, which is determined by the blockage factor. This factor is calculated by dividing the ship's frontal area underwater by the channel cross-section area. A higher blockage factor results in a stronger return current, causing the speed over water to be higher than the speed over ground. To accurately calculate the blockage factor for different parts of a trip, a detailed map of the waterway, including depth and width, would be essential. This information could be used as input for (Bi-)LSTM to improve accuracy, although it would also likely increase computation time and effort as fairway information would be needed for each vessel at each time step and location.

In addition, the depth data in this case study are likely unreliable. Previous findings have indicated that having information about the depth can enhance predictive accuracy. Considering that depth can significantly impact currents and blockages in restricted waterways, its importance is even more pronounced in inland

Figure 5.8: Speed and delivered power during a trip

waterways. It is anticipated that more accurate depth information will substantially improve predictive capabilities. As the GEBCO chart has a resolution of around 300 meters, it was also deemed not accurate enough to be used as a substitute.

As mentioned in subsection 4.2.3, when one ship overtakes another on a waterway, the return currents and changes in water level caused by the two ships will strengthen each other. This will increase the resistance on the ships, leading to a higher effective speed over water. As a result, overtaking maneuvers can significantly impact engine power, even if the ship being overtaken is not actively involved in the maneuver. The current model does not account for this effect.

In future research, it is recommended to incorporate current speeds as a factor, which is likely to improve the accuracy of predictions. While the speed over ground may be a reasonable input for sea-going vessels, it is essential to consider current speeds for inland waterways. Additionally, it is advised to gather data from a wider range of ships and include static variables such as deadweight tonnage, vessel dimensions, hull characteristics, engine specifications, and propeller details in the models to enhance their predictive capabilities and improve generalizability. In its current state, it is not advisable to use this model for spatiotemporal engine power estimations.

5.3. Reflection on all results

In both case studies, the LSTM and Bi-LSTM models were employed to predict the main engine power of vessels using different input combinations. The results varied significantly between the two cases.

In the first case study, focusing on a sea-going container vessel, both models outperformed the benchmark IMO/HM model in terms of accuracy. The LSTM model excelled in predicting power at individual time steps, making it suitable for estimating spatiotemporal emission distributions. On the other hand, the Bi-LSTM model demonstrated superior performance in predicting total energy consumption over a trip, making it valuable for estimating overall emissions.

In contrast to the first case study, the second case study, involving an inland tanker vessel, presented more challenges. Both models struggled to achieve the same level of accuracy as in the first case study. This discrepancy can be attributed to the unique characteristics of inland waterways, such as stronger currents, blockage factors, and varying water depths, which significantly influence engine power. The lack of reliable depth data and the absence of information on currents and blockage factors further hindered the models' predictive capabilities.

6

Discussion

This chapter delves into a detailed discussion of the results presented earlier. An analysis of the disparities between the predicted and actual ME power is provided, with a focus on the 'anchoring' operational mode. Potential reasons for these disparities are explored, including the constraints of the decision matrix used for determining operational modes. The impact of an alternative decision matrix on the accuracy of ME predictions, is assessed. Additionally, limitations associated with the use of limited data and machine learning models are examined.

Differences in case studies
Upon comparing the results of both case studies, it becomes apparent that the first case predicts with higher accuracy and precision. This is likely due to the influence of environmental factors and the underlying data structure. It is worth noting that a sea-going ship does not necessarily follow the same or opposite direction of the current. In a narrow inland waterway, this almost always holds true, indicating that the ship consistently navigates in the most or least efficient manner in terms of speed over ground. These observations might also elucidate the abrupt surges in engine power for the same speed in the second case. When a ship travels with the current and then changes direction, there is a significant increase in power required to maintain the same speed over ground. In addition, the unreliable depth sensor, blockage factor, and overtaking actions could contribute to the higher errors in the second case.

Case study 1: sea-going container vessel
In the first case study, the best performing Bi-LSTM model had an average error of -0.82% on the total energy consumption of a ship. While the model generally captured patterns accurately, there were some deviations. Potential reasons for these deviations will now be explored. Firstly, it's possible that unusual environmental conditions, such as storms, greatly differed from the norm. Storms can lead to increased engine power requirements, especially when sailing against the wind, and can also cause higher wave resistance. Furthermore, the ML models were trained based on speed over ground. Ships can optimize fuel consumption by timing their trips to travel with the current. The study primarily took place in the English Channel, an area renowned for its strong tidal currents that move alongshore. This makes it feasible for ships to harness these currents, particularly when navigating the Dutch, Belgian, English, or French coastlines, leading to a significant reduction in required engine power. Upon reviewing Figure 4.1, it is evident that this is the area where the ship spends the majority of its time sailing. The high over-estimations of the IMO/HM model regarding ME power might be a result of this as well, as it uses speed over ground.

Case study 2: diesel-electric inland tanker
The second case presents more challenging predictive scenarios, primarily due to the currents encountered in inland waterways. The alternating nature of ship movement up and down the current necessitates the model to strike a balance between the two states. Additional complexities arise from factors specific to inland waterways, particularly the phenomenon of blockage. When a vessel navigates through narrow waterways, it propels water forward, causing the surrounding water to decelerate or even reverse its flow, in accordance with the principle of mass conservation. These opposing currents effectively augment the vessel's speed over water and subsequently increase resistance. In shallower waters, the restricted flow beneath the

ship accelerates the return current, leading to a condition known as squat, which further deepens the ship's submersion and increases resistance. These combined effects significantly influence resistance but were not considered in the case study. Doubt is cast on the reliability of depth sensors, as evidenced by their inability to enhance the predictions of the standard LSTM model. The provision of a comprehensive fairway information map could potentially alleviate these challenges and substantially improve prediction accuracy. Another contributing factor to the high prediction errors could be the mismatch between the employed models and the operational characteristics of the engines. Unlike the diesel engines found in large container ships, the engine power in this case remains relatively stable. While LSTMs demonstrate efficacy in the first case and in other multivariate applications such as rainfall-runoff and stock market prediction, where noisy signals are commonplace, the target signal in this scenario exhibits high stability while the predictions retain a degree of noise, albeit to a lesser extent than in the first case.

Interpretability
The nonlinearity of ML models presents a dual nature. It enables the creation of powerful models adept at identifying relationships that traditional statistics-based approaches cannot. Additionally, these models are adaptable and data-driven, requiring only raw data to make accurate predictions. However, interpreting these models can be challenging. While it has been demonstrated that adding more input variables enhances accuracy, the precise impact remains unquantifiable due to the large number of trained parameters. Moreover, commonly used performance metrics like R-squared are not suitable for nonlinear predictors.

Generalizability The limited data available in this study, derived from only two vessels, poses a challenge in establishing a truly generalizable model. The significant differences between the sea-going container vessel and the inland tanker, coupled with the absence of data from other ship types, hinder the development of a universally applicable model. However, the research acknowledges the potential for generalizability and suggests that future studies incorporate a wider range of vessels and include static variables such as deadweight tonnage, vessel dimensions, hull characteristics, engine specifications, and propeller details. By training on a diverse dataset and considering these additional factors, it is anticipated that a more robust and adaptable model can be developed, capable of accurately predicting engine power and emissions across a broader spectrum of ships and operating conditions.

Tug boats
This research has primarily focused on sailing vessels. Sailing was chosen as the primary focus due to the significant amount of time spent and marine engine fuel consumed during sailing. Even a small decrease in percentage error can be highly valuable. Additionally, the decision not to focus on maneuvering is due to the common use of tugs in port areas to turn and navigate a ship. As the tugs provide most of the engine power in these instances, their usage does not register in the sensor data. Therefore, using this data could potentially distort predictions for the remainder of the voyage.

Computational effort
The literature referenced in Chapter 1 highlights that the integration of ML can result in a significant improvement in computational speed, often up to 90%. This acceleration could be particularly beneficial for emission estimates, enhancing the accessibility of these methods. It's worth noting that although the ML approach offers substantial advantages, the creation of trips and the operational mode assignment still require the current model. On a standard laptop, these tasks can still consume multiple hours when analyzing the trajectories of only one ship over a three-month period. Thus, streamlining this process through ML could prove valuable.

Despite the limitations mentioned above, this study has showcased the potential of ML models in predicting engine power for maritime operations, especially when sailing. In particular for a container vessel on the open sea, LSTM models are able to predict total energy used with an accuracy of within one percent, using only AIS and a depth chart. In the second case, although performance was worse, some useful insights were generated. For example, the influence of current and bloackage factor in restricted waterways. Additionally, the influence of headwind was shown in the comparison between input combinations. Both cases displayed the robustness of Bi-LSTM models compared to LSTMs. While useful for emission calculation, the models could also be used by shipping companies to conserve fuel: simulating a trip could provide insight in the amount of fuel needed to reach the destination, which makes it possible to load less fuel in the ship. This is

not expected to make an enormous change, but could contribute slightly to cleaner shipping in the future.

7

Conclusions

7.1. Research questions

In the following chapter, the sub-questions will be examined in order to provide a comprehensive response to the primary research question. The research question and sub-questions are presented as follows:

How can supervised machine learning models, trained on on-board sensor data and applied to AIS data, enhance the prediction of a ship's main engine power?

1. What are current state-of-the art emission models that could be compared to a new proposed ML method?
2. How are operational modes assigned, and how could this be improved?
3. Which machine learning algorithms could be suitable for predicting a ship's ME power using AIS as input?
4. How do ML models, trained on measurement data, perform against an existing state-of-the-art emission model?
5. Which variables are important when estimating ME power from AIS using ML models?
6. What is needed to make an ML model that predicts ME power using AIS generalizable?

What are current state-of-the art emission models that could be compared to a new proposed ML method? To answer this question, a study of various literature has been performed. Three models were explained: STEAM, MoSES, and IMO/HM. STEAM and MoSES are categorized as bottom-up models, both utilizing AIS data to estimate ship emissions. STEAM simplifies the relationship between a ship's power and velocity by assuming constant resistance coefficients, wetted surface, and propulsive coefficient. Through this simplification, the model enables the easier calculation of instantaneous power using readily available data. Furthermore, STEAM estimates auxiliary engine power based on ship type and operational mode, and it integrates the effect of waves on fuel consumption and emissions.

Conversely, MoSES places emphasis on calculating ship emissions with high spatial and temporal resolution. It leverages AIS signals and ship characteristics sourced from the IHS Markit database to estimate parameters such as gross tonnage, main engine power, and service speed. These parameters are then utilized to calculate engine load and energy consumption, which are subsequently converted to emissions using emission factors. However, MoSES introduces uncertainties due to the estimation of various parameters based on empirical functions.

In contrast to STEAM and MoSES, the IMO/HM model employs the full Holtrop & Mennen method to simplify the resistance calculation. This approach enables a more accurate estimation of resistance, considering factors such as variable draught, environmental conditions, and shallow water effects. Furthermore,

the IMO/HM model incorporates a distinction between different operational modes (sailing, maneuvering, anchoring, berthing) as per the guidelines of the IMO. The model's ME and AE power calculations are dependent on the size of the ship. Instead of relying on regression functions to estimate unknown ship variables, the IMO/HM model employs a probability distribution for ship parameters based on vessel type. Lastly, the model accounts for shallow water effects on frictional resistance, which is pivotal for comprehending vessel behaviour and emissions in port areas. Due to these characteristics, the IMO/HM model is considered the most suitable for comparison with the proposed ML models.

How are operational modes assigned, and how could this be improved?
The role of operational modes in bottom-up emission modeling has been explained in Chapter 2. The engine power profile for main and auxiliary engines depends on the mode the ship is in. When berthing or anchoring, main engines are usually not providing power. When this is the case, the auxiliary engines and boilers take over to provide the necessary heat and electricity on board. To assess whether the modes were assigned correctly, the IMO/HM model was applied to all trips of the sea-going container vessel, predicting the ME power. Subsequently, the MAE was calculated. With the decision matrix from the IMO, the highest errors occurred during sailing. This is to be expected, as that mode is the one with the highest engine powers overall. However, when looking at the other modes, the anchoring mode is the one with the highest errors. Since no engine power is expected when anchoring, a low speed could already result in a large MAE. To prevent this from happening, the threshold for berthing and maneuvering was set to 1 knot. All speeds between 1 and 3 knots are now seen as maneuvering, independent of the distance to the port. These changes significantly decreased the errors for each mode, suggesting that the mode assignment has improved, at least for container vessels. To support this, visual checks were performed, which showed less anchoring within ports or during slow steaming. The same visual checks were performed for the inland tanker vessel, where the instances of unexplained anchorings was reduced. The ship is only anchoring close to locks, which can be explained by the fact that it often has to wait before entering a lock, or in a lock while waiting for the water level to rise or drop.

Which machine learning algorithms could be suitable for predicting a ship's ME power using AIS as input?
When choosing a suitable model, it's essential to have the right prerequisites in place. Ideally, the model should be able to process multivariate input and generate a one-dimensional output. It should be capable of identifying (nonlinear) connections between input variables and the target. Given that the inputs are time series, the model must be equipped to handle this, ideally with an understanding of temporal relationships. Since the sequences are quite lengthy and vary in length, it would be advantageous for the model to have a memory function and the ability to manage sequences of variable length.

As discussed in Chapter 3, the focus has been on Artificial Neural Networks, with the primary models being MLPs, LSTMs, and Bi-LSTMs. MLPs are favored for their simplicity of implementation and long-standing presence. They have found extensive use in time-series predictions, capitalizing on their ability to learn nonlinear relationships. However, a significant limitation in this application is their inability to handle inputs of variable lengths. Additionally, they do not incorporate previous time steps as input; instead, they predict the ME power for each time step based on the (variable) inputs for that specific time step only.

In response to these challenges, LSTM networks were developed. These algorithms excel at processing inputs of varying lengths and possess a memory function, allowing them to capture overall patterns in data. An LSTM is composed of an input gate, a forget gate, and an output gate, which collectively determine which information is retained for the next time step and what the current output will be. Additionally, LSTMs are adept at handling multi-variate time series predictions. They have demonstrated their utility in prior studies focused on time series forecasting, particularly in the domains of rainfall run-off and stock price prediction.

Since the time series of the trips is already available, there is no need to predict future values. This presents an opportunity to employ a Bi-LSTM. Bi-LSTMs function similarly to standard LSTMs, but unlike LSTMs, they operate bidirectionally. This provides the added advantage of better understanding the ship's behaviour. With this approach, the model can anticipate whether the ship will slow down or speed up in the next few minutes, and leverage this knowledge to predict engine power more accurately.

In this study, (Bi-)LSTMs have been chosen for the prediction of ME power. This selection is based on their

capacity to capture nonlinear relationships, ease of implementation, and memory function, addressing the issue of vanishing or exploding gradients.

How do ML models, trained on measurement data, perform against an existing state-of-the-art emission model?
In the first scenario, the MAE and RMSE are lower for the LSTM model, whereas the Bi-LSTM exhibits lower errors when predicting the total energy used during a trip. As previously noted, ML models can be challenging to interpret due to their complexity and numerous parameters. However, the LSTM likely achieves a lower MAE and RMSE due to its sensitivity to changes, as it is influenced solely by past time steps, while the Bi-LSTM is influenced from both directions. Because of their distinctions, it is conceivable that both methods could be employed, albeit for different objectives. The LSTM appears to perform better when analyzing emissions in a smaller area, due to its better spatiotemporal accuracy. This could be used for protecting people in the vicinity of a port from harmful $PM_{2.5}$, or PM_{10} emissions, or for complying with European NO_X limits when estimating emissions in a port area. Conversely, when examining global emissions, such as the IMO does to establish new regulations for CO_2 reductions, Bi-LSTMs might offer a better solution by closely predicting the total energy used. It should be noted that this analysis was conducted on a relatively small test set of 8 trips. A more comprehensive understanding of model performance could be obtained by using more data or retraining both models with a larger number of seeds and averaging the results.

Which variables are of importance when estimating ME power from AIS using ML models?
In the two case studies, various input variable combinations were utilized to forecast engine power. Pearson's correlation was used to analyze the relationship between potential input variables and the ME power. Based on this analysis, the inputs were selected. Upon training the models and comparing them with measured data, it was determined that a reasonably accurate estimation can be achieved using only the ship's speed and acceleration. The first case study demonstrated that utilizing a standard LSTM, the ship's draught and depth contributed to better predictions. In the second case study, it was observed that headwind had a positive impact on prediction accuracy. The absence of improvement in predictions due to depth in this case may be ascribed to the unreliable depth sensor. Furthermore, other potential inputs for enhancing prediction accuracy include vessel size, waterway characteristics, wave height, and currents.

What is needed to make an ML model that predicts ME power using AIS generalizable?
The ultimate objective of this research is to contribute to developing an AIS-based model that can predict engine power and emissions for ships worldwide. The challenge lies in the diverse characteristics and operating conditions of ships globally, making it difficult to establish rules for predicting engine power. The IMO provides estimations for different types of ships, such as tankers, container ships, and cruise ships. However, variations in weight classes, engine types, and propulsion systems influence emissions as well. Instead of relying on predefined rules for each variable, the idea is to train an ML model using data from a diverse set of ships. Theoretically, this model could recognize different ships and predict engine power without relying on predefined rules. The potential of using LSTM networks, which can handle both static and dynamic variables, was explored in section 4.7. In previous research, it was shown that there are two possible ways to add static variables to a time series for better predictions. The first option, or direct input, is to add the static variables to the input vector at each time step. The second option, indirect input, involves passing the static variables through an FNN and concatenating this with the output of the LSTM, before passing through the final output layer. Indirect input outperformed the direct input. Therefore, it is recommended to follow this approach during future research into a more complete engine power prediction model. A prerequisite for this however is that there is enough data available from a wide array of different vessels.

Having answered the sub-questions above, the main research question can now be answered:

How can supervised machine learning models, trained on on-board sensor data and applied to AIS data, enhance the prediction of a ships main engine power?
The IMO/HM model provides a solid foundation for analyzing operational modes and their impact on engine power profiles during trips. By refining these operational modes and incorporating machine learning techniques, more accurate predictions can be made. A specific focus was placed on the sailing mode, where ships consume the most fuel. Leveraging the multivariate time series nature of AIS signals proved to be well-suited for LSTM and Bi-LSTM models. After training these models with various inputs, a Bi-LSTM utilizing

speed, acceleration, draught, and depth as inputs could predict engine power during the sailing phase of a trip within an average deviation of 0.82%. For improved accuracy per time step and more reliable spatio-temporal emission distributions, an LSTM was the preferable choice. With the potential addition of more measurement data in the future, the aim is to further enhance the performance and reliability of the predictions.

Estimating power for inland waterways presents its own set of challenges. Factors such as current and blockage factor play a more crucial role in the prediction process. Additionally, the type of propulsion may be less compatible with the LSTM approach due to the stable signals. In summary, there was insufficient data available to make LSTMs as effective for inland vessels as they are for sea-going vessels. However, in comparison to the current key figures provided by the Dutch Rijkswaterstaat, which underestimated fuel consumption by 2-3 times, it is still a considerable improvement.

The use of ML has demonstrated great potential in improving ME power predictions. The IMO/HM model is surpassed for sea-going vessels, and depending on specific requirements, an LSTM or a Bi-LSTM may produce superior results. In the case of inland waterways, environmental factors have a significant impact on energy consumption, making prediction more challenging. However, by integrating these environmental conditions into the model input, this obstacle may be overcome. The same holds true for the model's ability to be applied across different vessels. It is expected that generalizability can be achieved, but only through training on a sufficient number of vessels.

8

Recommendations

In order to further expand the research on LSTMs for engine power prediction in future studies, several avenues for exploration can be pursued. This chapter suggests potential future research directions and provides recommendations on their execution or potential improvement.

Universal model
It is believed that a universal model can be developed given a sufficient amount of data. The concept of generalizability delves into this possibility and discusses how it can be achieved. It is suggested that, for future research, efforts are made to gather data from additional ships. If successful, this endeavor could lead to the creation of a model capable of replacing the commonly used resistance calculation based on Holtrop & Mennen. Such a model would have the capability to directly estimate the Main Engine power for a wide range of ships. In addition, by keeping the general structure of the HM/IMO model, a distinction can still be made in operational modes.

Integration in existing model
An interesting opportunity with these models is that they can be integrated in the semi-empirical IMO/HM module. By doing so, the high MAE and total percentage error for the sailing mode can be decreased, while keeping all benefits from the previous model. For example, the spatial distribution visualisations are still being used. Hereby, the ability to model auxiliary engine power is preserved as well. Another advantage is that the models modularity provides the possibility to change parts of it over time. The AE energy or ME energy during another operational mode can be replaced by the same or other ML models, while being able to use the old methods over time.

Environmental conditions
Future model improvements could be made by incorporating waterway characteristics. Doing so would benefit the predictions, especially for inland cases. To start, the depth and width of a channel/river can be used to calculate the blockage factor of the ship in the waterway. This is likely to improve accuracy. Secondly, current has a significant impact on engine power. Without knowledge of the current, an estimation can still be made. However, since the speed over ground is used in stead of speed over water, accuracy is decreased. An example of this was shown in Chapter 5. The same holds for wind. In the results, it was shown that knowledge of the headwind improves the engine power predictions. While in this case the headwind was known from sensor data, this research focuses on the potential of ML models that use AIS or other widely available data as input. It is therefore advised to incorporate open-source wind data, for example from Copernicus.

LSTM for diesel-electric ships
As previously noted, the LSTM appears to struggle with estimating engine power in the inland waterway scenario. This may be partly due to the nature of electric engines; their power output is very consistent, with minimal noise. It is unclear whether all minor variations in power are captured in the data or if the data have already been smoothed. Furthermore, engine power does not directly correlate to emissions, as generators do not need to supply the same power as the electric engines simultaneously, given that energy is temporarily stored in batteries. A potential solution is to adopt the method used in LSTMs for rainfall-

runoff models. In these models, weather data are utilized to forecast river runoff. One could liken the rain to input, the catchment area to a battery, and the runoff to output. LSTMs are especially effective in this context due to their memory capabilities. This approach has already demonstrated its effectiveness in predicting rainfall runoff. Applying the same method to diesel-electric ships could be intriguing. Here, environmental conditions, speed, acceleration, and characteristics would serve as input. The battery would represent the catchment area, and the generators would correspond to the runoff. This approach could potentially enhance the overall accuracy of predictions while also providing a comprehensive view of emissions.

Bibliography

Ahlgren, F., & Thern, M. (2018). Auto Machine Learning for predicting Ship Fuel Consumption. ECOS 2018 - The 31st International Conference on Efficiency, Cost, Optimization, Simulation and Environmental Impact of Energy Systems.

Berk, M. (2020). Hyperopt demystified. https://towardsdatascience.com/hyperopt-demystified-3e14006eb6fa

Carlton, J. S. (2007). Marine propellers and propulsion (Second edition). Butterworth-Heinemann.

Chen, W., Tai, K., Lau, M., Abdelhakim, A., Chan, R. R., Adnanes, A. K., & Tjahjowidodo, T. (2022). Data-Driven Propulsion Load Profile Prediction for All-Electric Ships. 2022 International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME), 1–9. https://doi.org/10.1109/ICECCME55909.2022.9988368

Cho, K., van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using rnn encoder-decoder for statistical machine translation. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 1724–1734. https://doi.org/10.3115/v1/D14-1179

Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. (2014, December 11). Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. arXiv: 1412.3555 [cs]. Retrieved July 10, 2024, from http://arxiv.org/abs/1412.3555

Esri. (2000, April). Measuring in arc-seconds. https://www.esri.com/news/arcuser/0400/wdside.html

Gao, C., Braun, S., Kiselev, I., Anumula, J., Delbruck, T., & Liu, S.-C. (2019). Real-time speech recognition for iot purpose using a delta recurrent neural network accelerator. 2019 IEEE International Symposium on Circuits and Systems (ISCAS). https://doi.org/10.1109/ISCAS.2019.8702290

GEBCO Compilation Group. (2023). Gebco$_2$023$grid$. https://doi.org/10.5285/f98b053b-0cbc-6c23-e053-6c86abc0af7b

Hallaj, P. (2023). Data augmentation: Benefits and disadvantages. Retrieved March 24, 2024, from https://medium.com/@pouyahallaj/data-augmentation-benefits-and-disadvantages-38d8201aead

Holtrop, J., & Mennen, G. (1982). An approximate power prediction method. International Shipbuilding Progress, 29(335), 166–170. https://doi.org/10.3233/ISP-1982-2933501

Huang, R., Wei, C., Wang, B., Yang, J., Xu, X., Wu, S., & Huang, S. (2022). Well performance prediction based on long short-term memory (lstm) neural network. Journal of Petroleum Science and Engineering, 208, 109686.

International Energy Agency. (2022). International shipping. International Energy Agency. Retrieved March 14, 2024, from https://www.iea.org/energy-system/transport/international-shipping

International Maritime Organization. (2020a). Ais transponders. International Maritime Organization. Retrieved July 29, 2024, from https://www.imo.org/en/OurWork/Safety/Pages/AIS.aspx

International Maritime Organization. (2023). Cutting ghg emissions. International Maritime Organization. https://www.imo.org/en/MediaCentre/HotTopics/Pages/Cutting-GHG-emissions.aspx

International Maritime Organization. (2020b). Third imo ghg study 2020: Executive summary and final report (tech. rep.). International Maritime Organization. https://www.imo.org/en/OurWork/Environment/Pages/Greenhouse-Gas-Studies-2020.aspx

Jalkanen, J.-P., Brink, A., Kalli, J., Pettersson, H., Kukkonen, J., & Stipa, T. (2009). A modelling system for the exhaust emissions of marine traffic and its application in the Baltic Sea area. Atmospheric Chemistry and Physics, 9(23), 9209–9223. https://doi.org/10.5194/acp-9-9209-2009

Kandel, I., & Castelli, M. (2020). The effect of batch size on the generalizability of the convolutional neural networks on a histopathology dataset. ICT Express, 6, 312–315. https://doi.org/10.1016/j.icte.2020.04.010

Karande, S. (2019, December). Tag prediction of competitive programming problem using neural network [Accessed on 29 April 2024]. Retrieved April 29, 2024, from https://medium.com/@siddheshkarande17/tag-prediction-of-competitive-programming-problem-using-neural-network-45a72eff7511

Lamers, S. A. (2022). Improved estimations of energy consumption for dredging activities based on actual data [Master's thesis, Delft University of Technology].

Marine Industrial Transmissions Ltd. (MIT). (2024). Electric boat motor propulsion. Marine Industrial Transmissions Ltd. (MIT). Retrieved June 13, 2024, from https://www.mitgroup.co.uk/products/marine/electric-propulsion/

Murray, B., & Perera, L. P. (2021). An AIS-based deep learning framework for regional ship behavior prediction. Reliability Engineering & System Safety, 215, 107819. https://doi.org/10.1016/J.RESS.2021.107819

Natarajan, M. (2023). Scaling data: Before or after train-test split? https://medium.com/@megha.natarajan/scaling-data-before-or-after-train-test-split-35e9a9a7453f

Pedersen, B. P. (2009). Modeling of ship propulsion performance [Doctoral dissertation, Technical University of Denmark].

Prince, S. J. (2023). Understanding deep learning. The MIT Press. https://mitpress.mit.edu/9780262048644/understanding-deep-learning/

Rahman, M. H., Yuan, S., Xie, C., & Sha, Z. (2020). Predicting human design decisions with deep recurrent neural network combining static and dynamic data. Design Science, 6, e15. https://doi.org/10.1017/dsj.2020.12

Salkuti, S. R., Ghimire, S., Deo, R. C., Wang, H., Al-Musaylh, M. S., Casillas-Pérez, D., & Salcedo-Sanz, S. (2022). Stacked lstm sequence-to-sequence autoencoder with feature selection for daily solar radiation prediction: A review and new modeling results. Energies, 15(3), 1061. https://doi.org/10.3390/en15031061

Schwarzkopf, D. A., Petrik, R., Matthias, V., Quante, M., Majamäki, E., & Jalkanen, J.-P. (2021). A ship emission modeling system with scenario capabilities. Atmospheric Environment: X, 12, 100132. https://doi.org/10.1016/j.aeaoa.2021.100132

Segers, L. J. M. (2021). Mapping inland shipping emissions in time and space for the benefit of emission policy development: A case study on the rotterdam-antwerp corridor [Master's thesis, Delft University of Technology]. https://repository.tudelft.nl/islandora/object/uuid:a260bc48-c6ce-4f7c-b14a-e681d2e528e3

Tardini, G. A., & Suharjito, . (2024). Selection of Modelling for Forecasting Crude Palm Oil Prices Using Deep Learning (GRU & LSTM). Emerging Science Journal, 8(3), 875–898. https://doi.org/10.28991/ESJ-2024-08-03-05

United Nations Framework Convention on Climate Change. (2015, December 12). Paris agreement. https://unfccc.int/process-and-meetings/the-paris-agreement

Van Den Berg, R. (2022). Estimate vessel emissions in ports using AIS data [Master's thesis, Delft University of Technology].

van Koningsveld, M., Verheij, H., Taneja, P., & de Vriend, H. (2021). Ports and waterways: Navigating the changing world. Delft University of Technology, Faculty of Civil Engineering; Geosciences. https://doi.org/10.5074/T.2021.004

VistiKhetmaar. (2020). Factsheet dieselelektrisch varen (tech. rep.). VistiKhetmaar. Retrieved June 3, 2024, from https://vistikhetmaar.nl/app/uploads/2020/06/Factsheet-Dieselelektrisch-varen-3.pdf

Wartsila. (2024a). Hull efficiency [Accessed: 2024-06-09].

Wartsila. (2024b). Open water efficiency [Accessed: 2024-06-09].

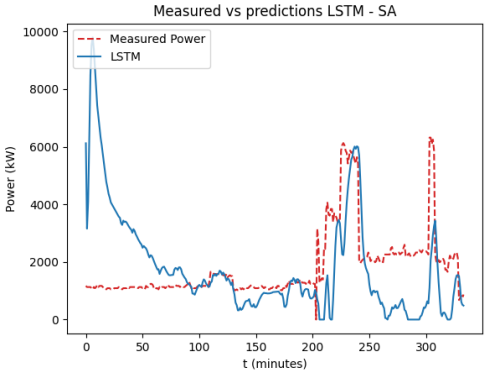
Xiang, Z., Yan, J., & Demir, I. (2020). A rainfall-runoff model with lstm-based sequence-to-sequence learning. Water Resources Research, 56(1). https://doi.org/10.1029/2019WR025326

Yates, A. (2020). Building a tree-structured parzen estimator from scratch (kind of) [Accessed: 2024-05-28]. Towards Data Science. https://towardsdatascience.com/building-a-tree-structured-parzen-estimator-from-scratch-kind-of-20ed31770478

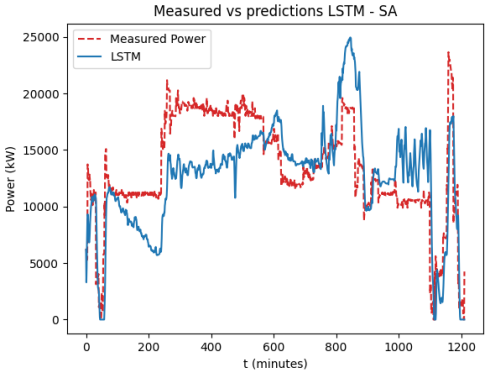
Zeng, Q., Thill, C., Hekkenberg, R., & Rotteveel, E. (2018). A modification of the ITTC57 correlation line for shallow water. Journal of Marine Science and Technology, 24(2). https://doi.org/10.1007/s00773-018-0578-7

A

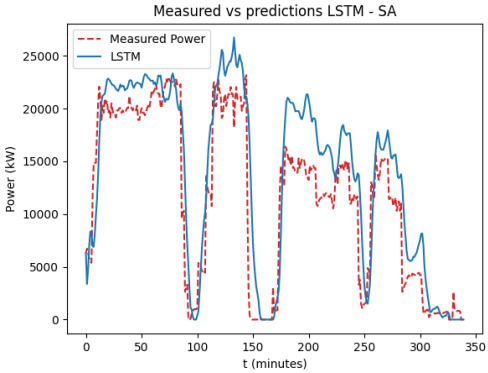
Results of LSTM for sea-going vessel



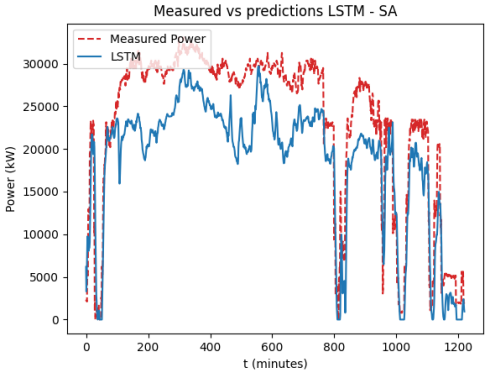
(a) LSTM vs. Measurements (Prediction 1)



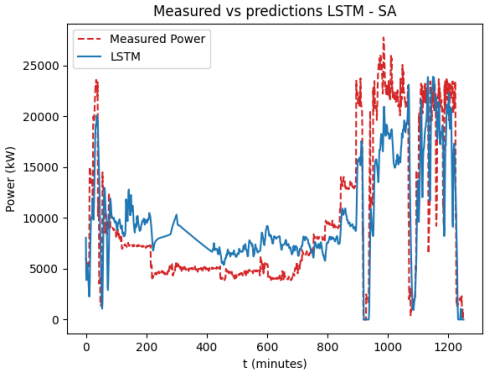
(b) LSTM vs. Measurements (Prediction 2)



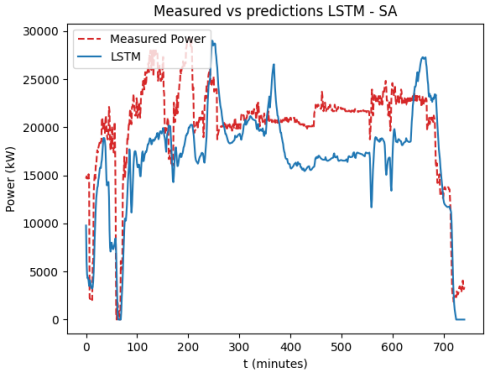
(c) LSTM vs. Measurements (Prediction 3)



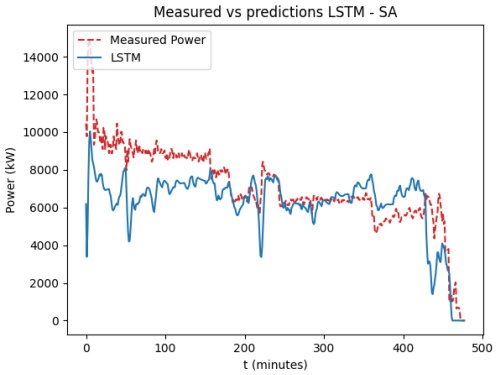
(d) LSTM vs. Measurements (Prediction 4)



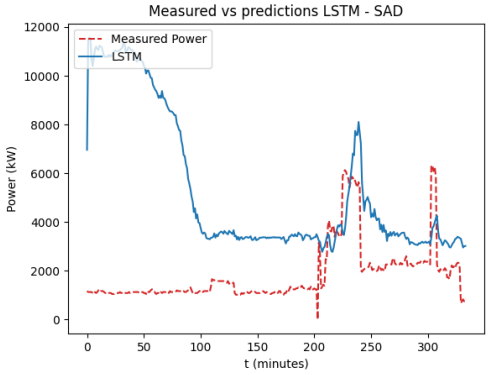
(e) LSTM vs. Measurements (Prediction 5)



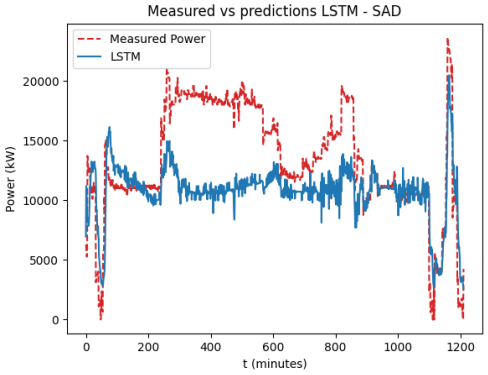
(f) LSTM vs. Measurements (Prediction 6)



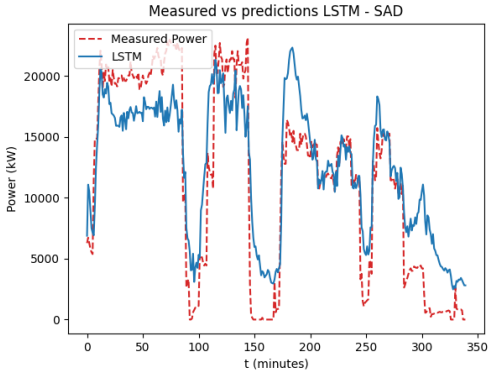
(g) LSTM vs. Measurements (Prediction 7)



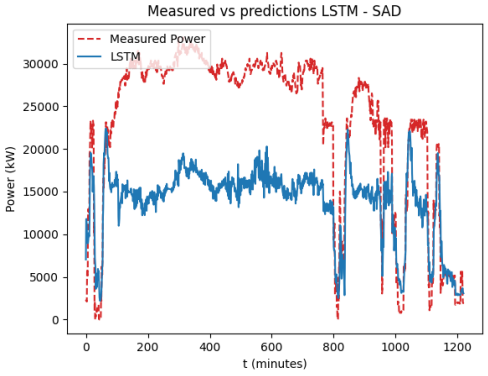
(a) LSTM vs. Measurements (Prediction 1)



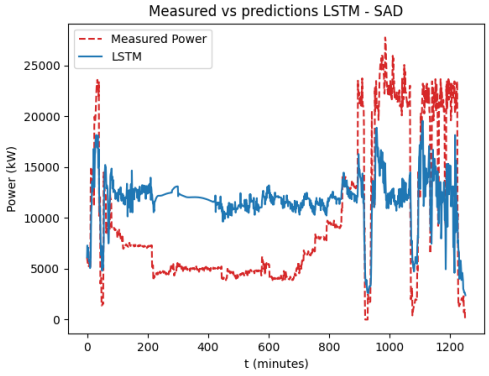
(b) LSTM vs. Measurements (Prediction 2)



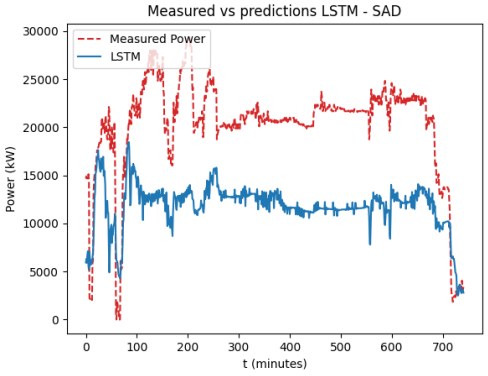
(c) LSTM vs. Measurements (Prediction 3)



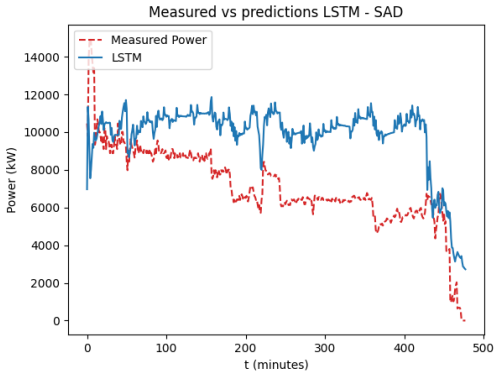
(d) LSTM vs. Measurements (Prediction 4)



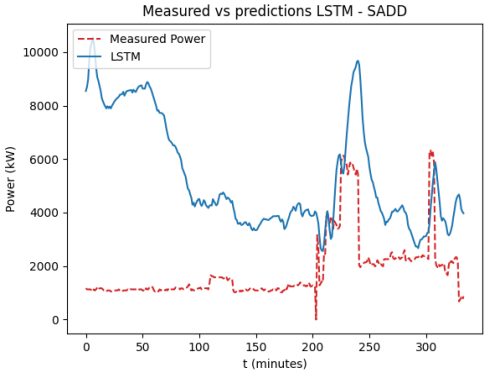
(e) LSTM vs. Measurements (Prediction 5)



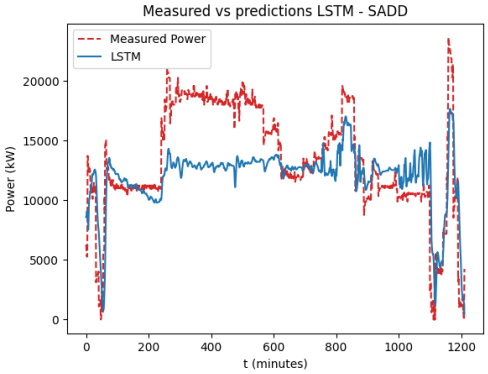
(f) LSTM vs. Measurements (Prediction 6)



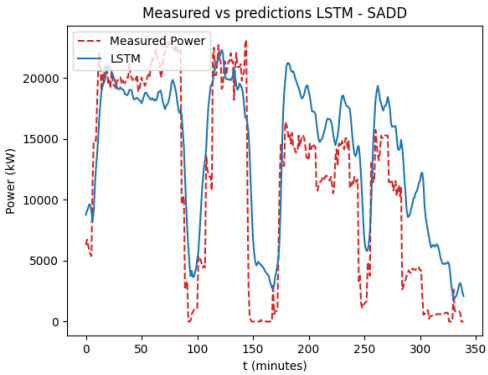
(g) LSTM vs. Measurements (Prediction 7)



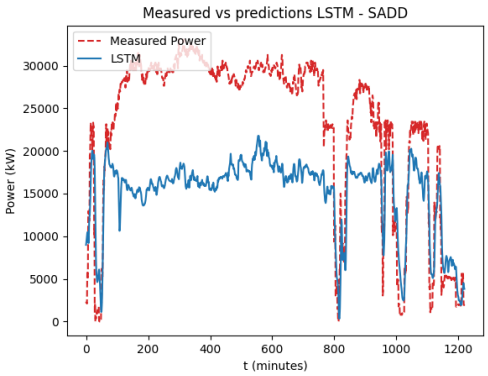
(a) LSTM vs. Measurements (Prediction 1)



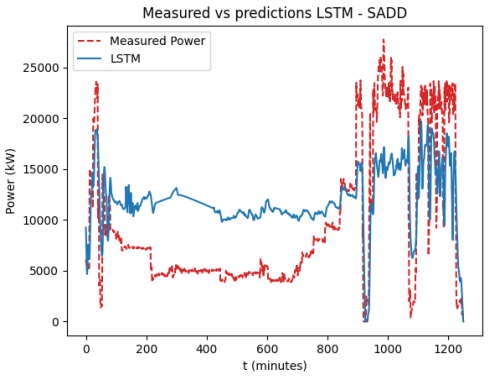
(b) LSTM vs. Measurements (Prediction 2)



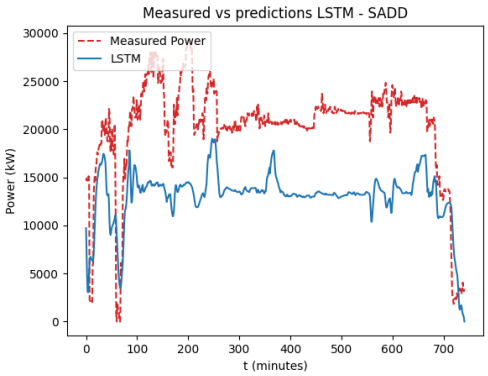
(c) LSTM vs. Measurements (Prediction 3)



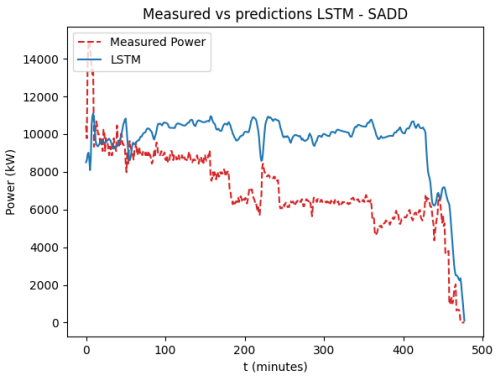
(d) LSTM vs. Measurements (Prediction 4)



(e) LSTM vs. Measurements (Prediction 5)



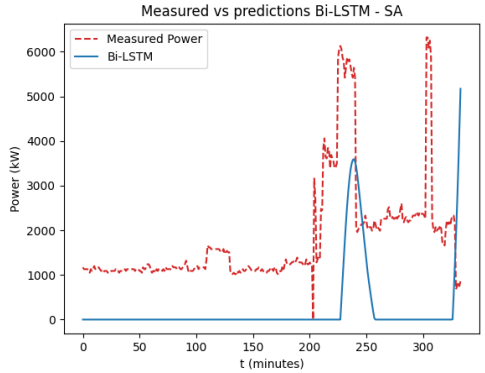
(f) LSTM vs. Measurements (Prediction 6)



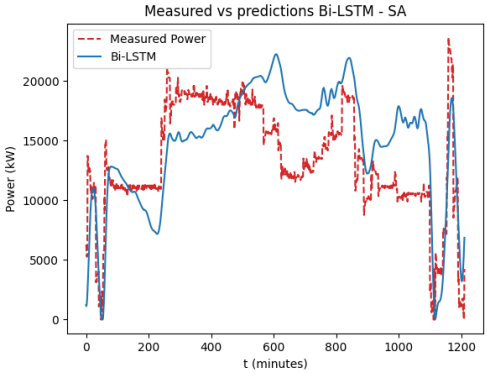
(g) LSTM vs. Measurements (Prediction 7)

B

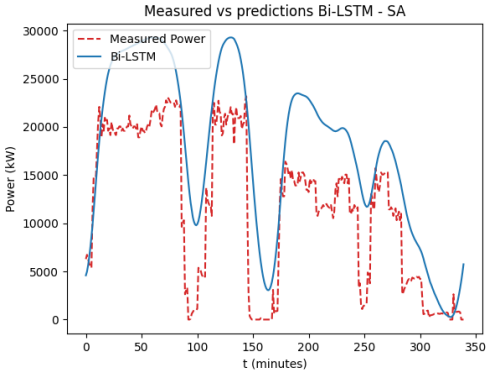
Results of Bi-LSTM for sea-going vessel



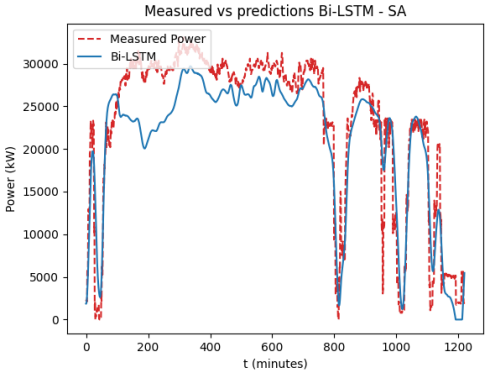
(a) Bi-LSTM vs. Measurements (Prediction 1)



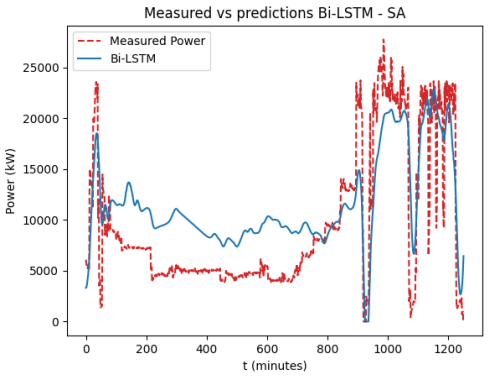
(b) Bi-LSTM vs. Measurements (Prediction 2)



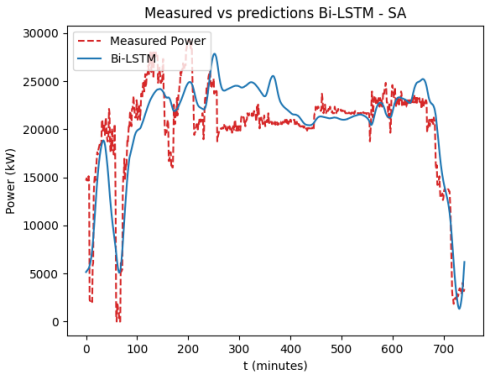
(c) Bi-LSTM vs. Measurements (Prediction 3)



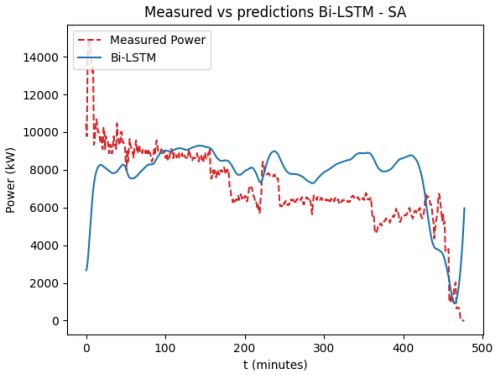
(d) Bi-LSTM vs. Measurements (Prediction 4)



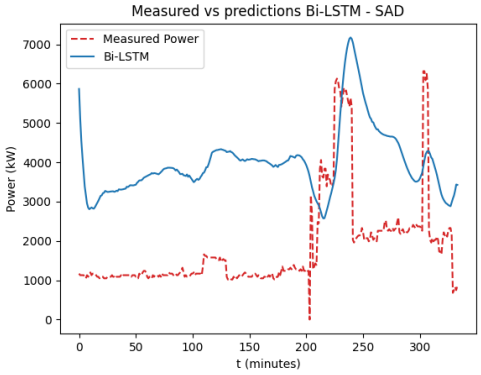
(e) Bi-LSTM vs. Measurements (Prediction 5)



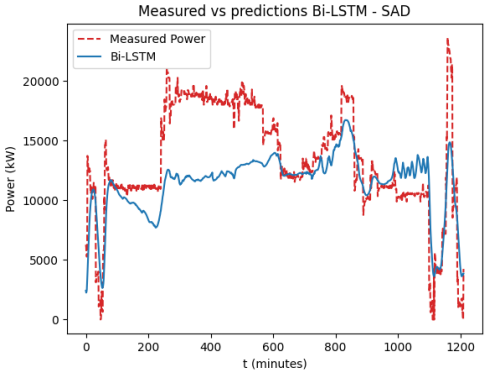
(f) Bi-LSTM vs. Measurements (Prediction 6)



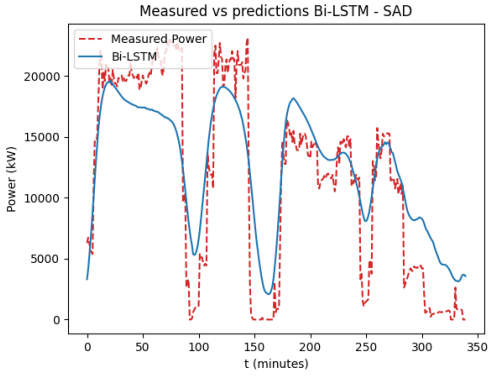
(g) Bi-LSTM vs. Measurements (Prediction 7)



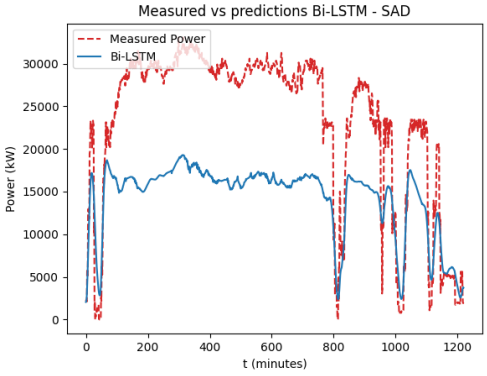
(a) Bi-LSTM vs. Measurements (Prediction 1)



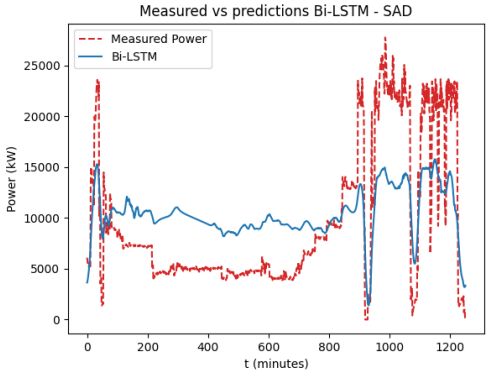
(b) Bi-LSTM vs. Measurements (Prediction 2)



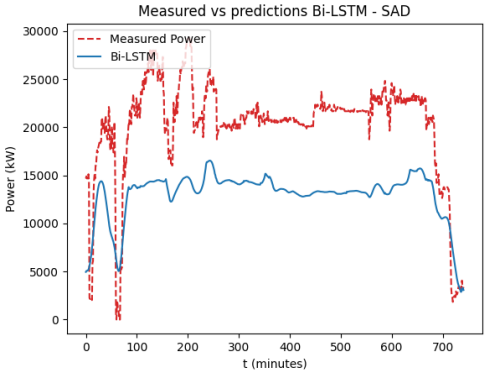
(c) Bi-LSTM vs. Measurements (Prediction 3)



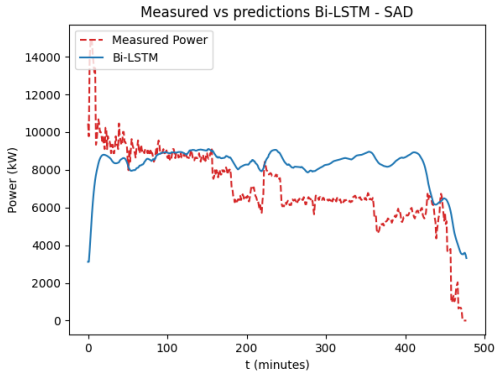
(d) Bi-LSTM vs. Measurements (Prediction 4)



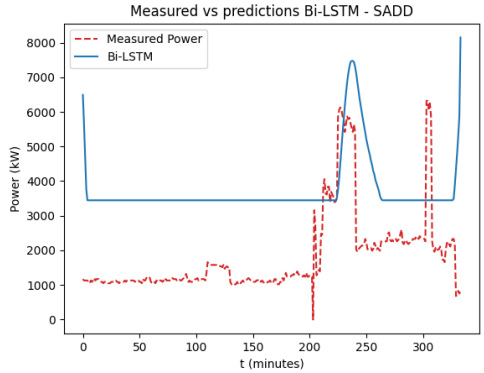
(e) Bi-LSTM vs. Measurements (Prediction 5)



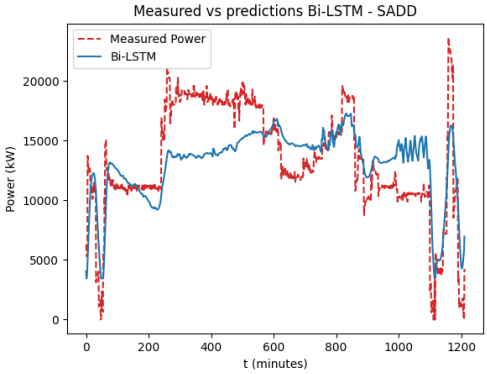
(f) Bi-LSTM vs. Measurements (Prediction 6)



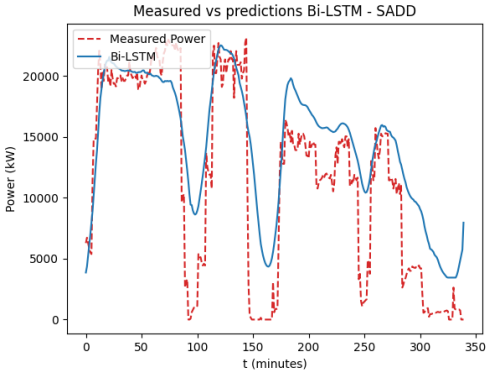
(g) Bi-LSTM vs. Measurements (Prediction 7)



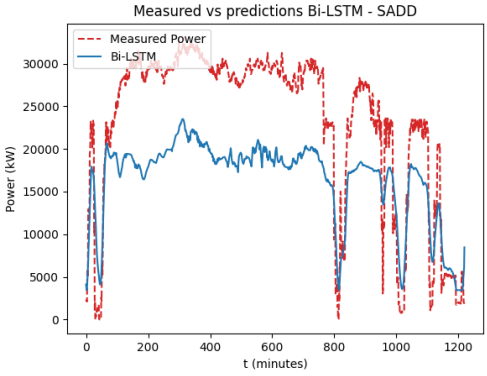
(a) Bi-LSTM vs. Measurements (Prediction 1)



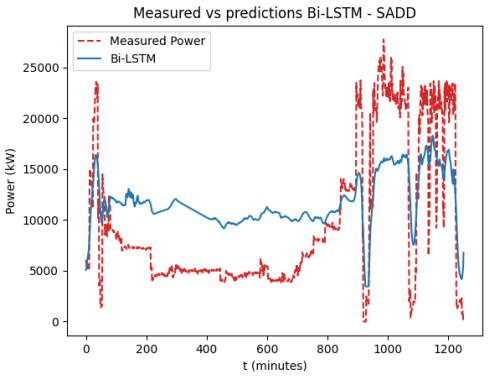
(b) Bi-LSTM vs. Measurements (Prediction 2)



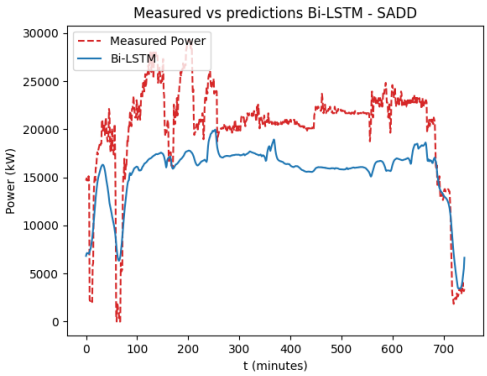
(c) Bi-LSTM vs. Measurements (Prediction 3)



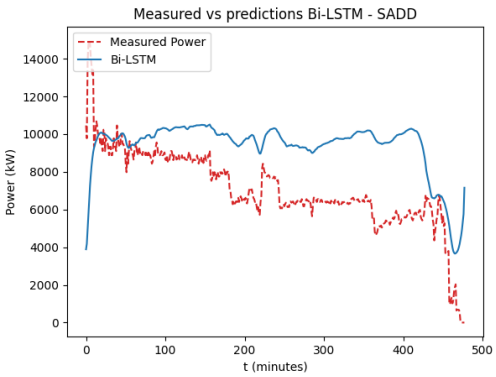
(d) Bi-LSTM vs. Measurements (Prediction 4)



(e) Bi-LSTM vs. Measurements (Prediction 5)



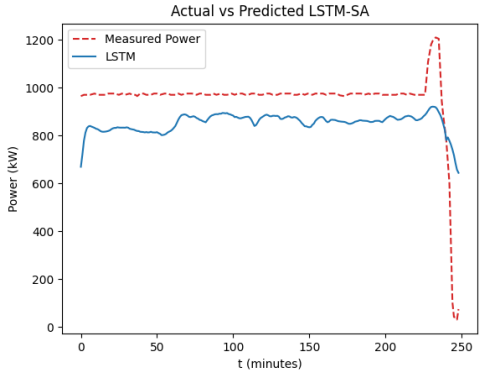
(f) Bi-LSTM vs. Measurements (Prediction 6)



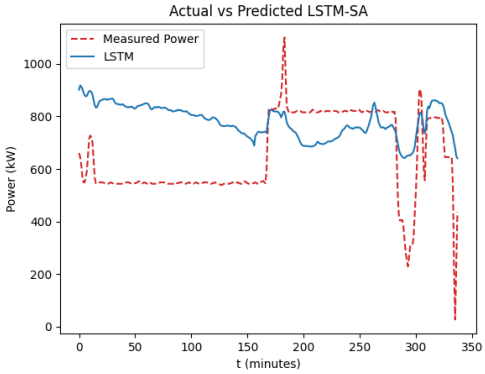
(g) Bi-LSTM vs. Measurements (Prediction 7)

C

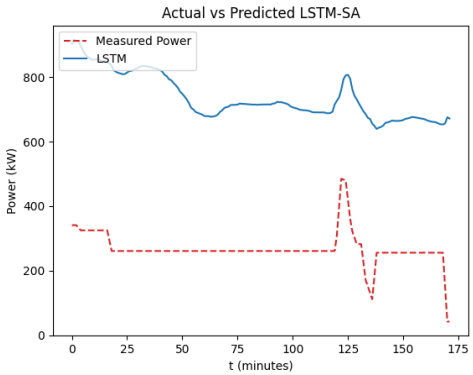
Results of LSTM for inland vessel



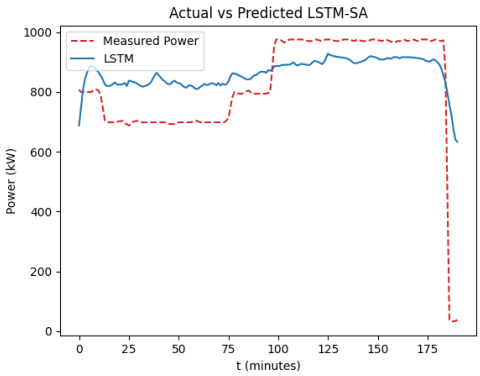
(a) LSTM vs. Measurements (Prediction 1)



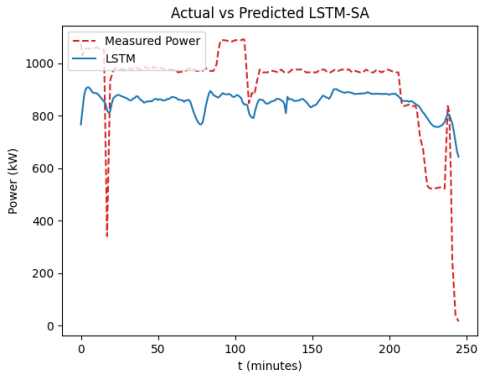
(b) LSTM vs. Measurements (Prediction 2)



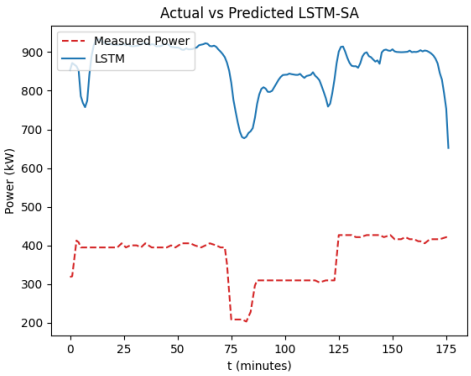
(c) LSTM vs. Measurements (Prediction 3)



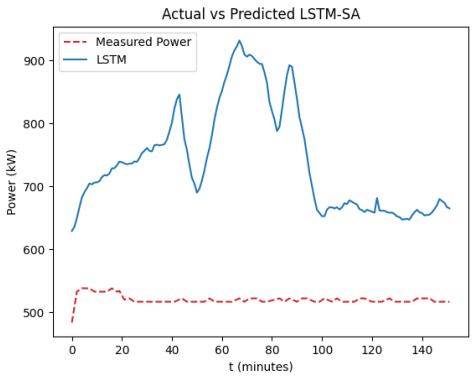
(d) LSTM vs. Measurements (Prediction 4)



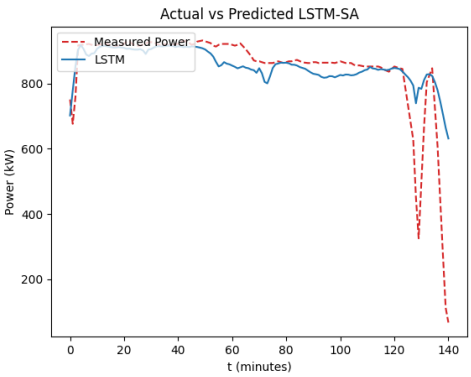
(e) LSTM vs. Measurements (Prediction 5)



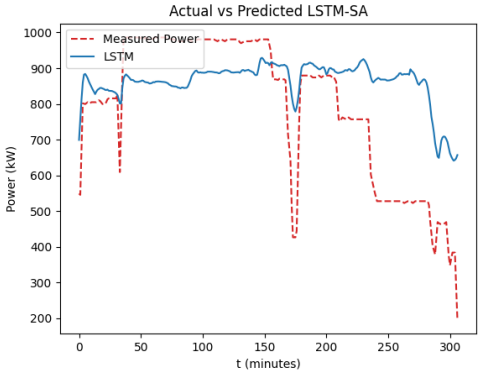
(f) LSTM vs. Measurements (Prediction 6)



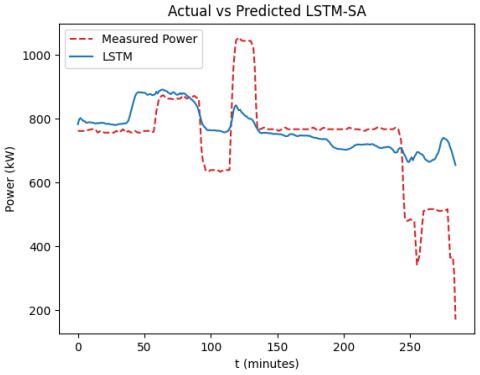
(g) LSTM vs. Measurements (Prediction 7)



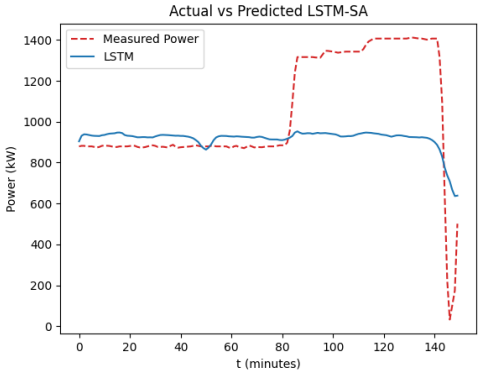
(h) LSTM vs. Measurements (Prediction 8)



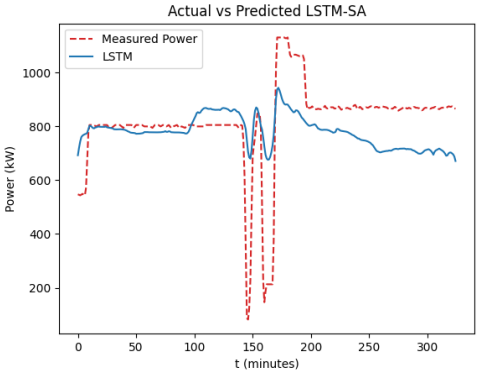
(a) LSTM vs. Measurements (Prediction 9)



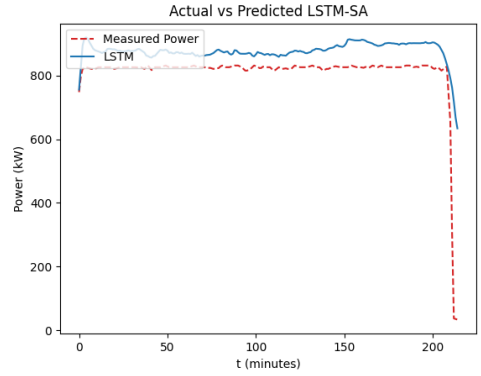
(b) LSTM vs. Measurements (Prediction 10)



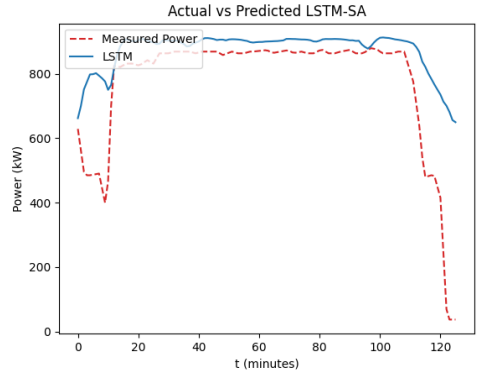
(c) LSTM vs. Measurements (Prediction 11)



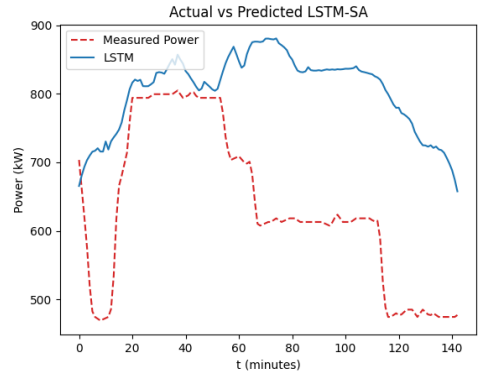
(d) LSTM vs. Measurements (Prediction 12)



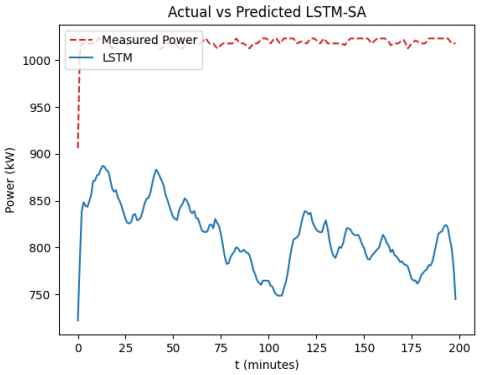
(e) LSTM vs. Measurements (Prediction 13)



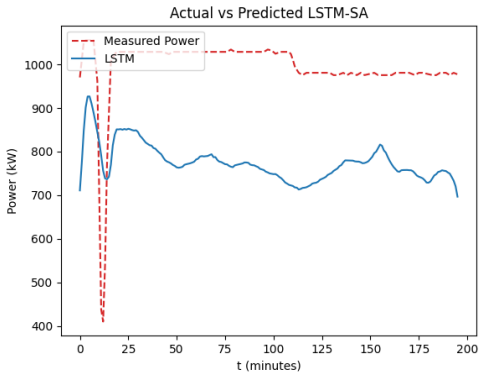
(f) LSTM vs. Measurements (Prediction 14)



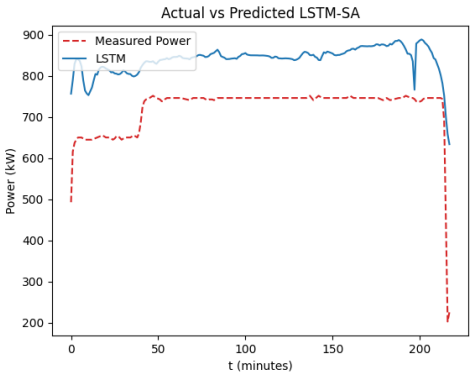
(g) LSTM vs. Measurements (Prediction 15)



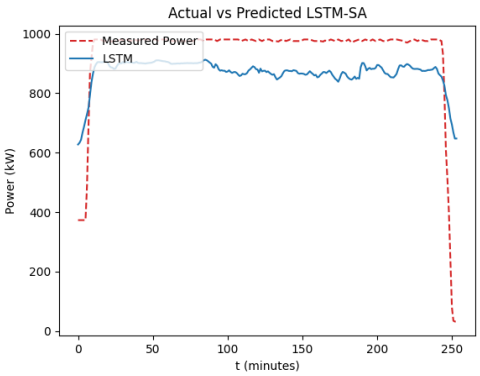
(h) LSTM vs. Measurements (Prediction 16)



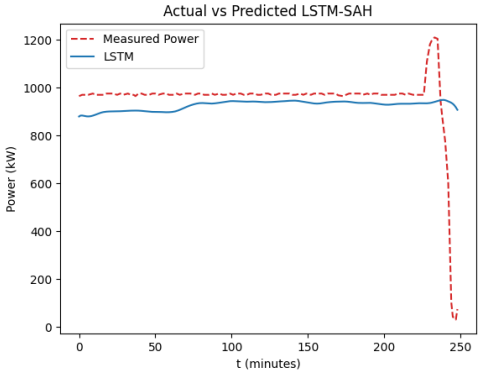
(a) LSTM vs. Measurements (Prediction 17)



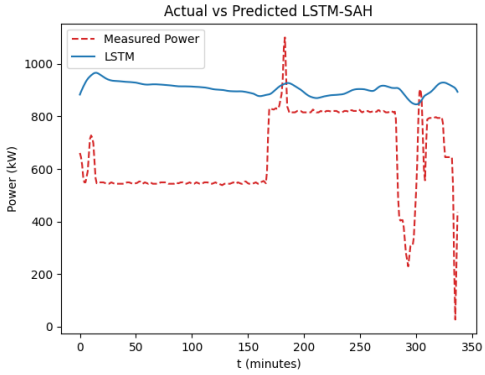
(b) LSTM vs. Measurements (Prediction 18)



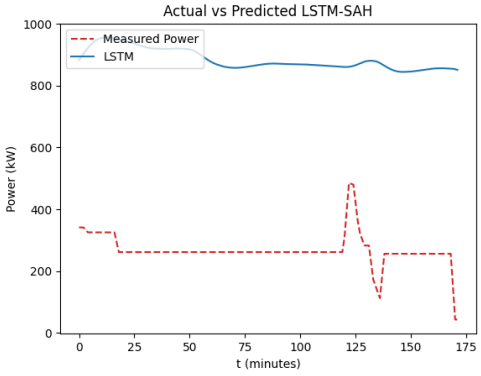
(c) LSTM vs. Measurements (Prediction 19)



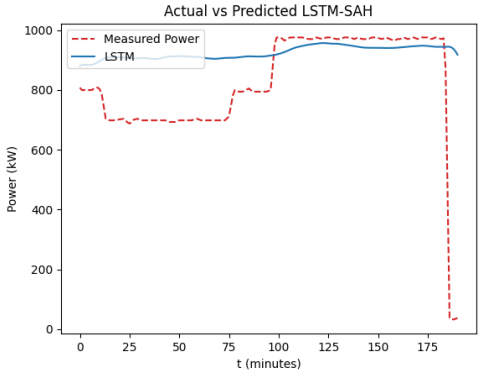
(a) LSTM vs. Measurements (Prediction 1)



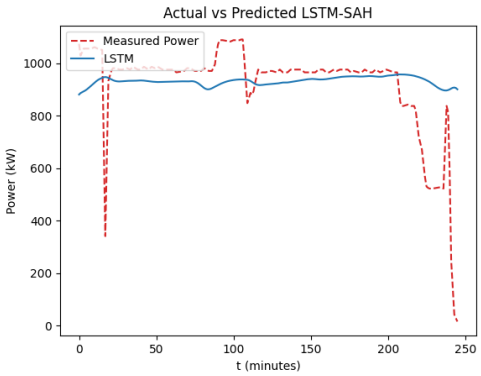
(b) LSTM vs. Measurements (Prediction 2)



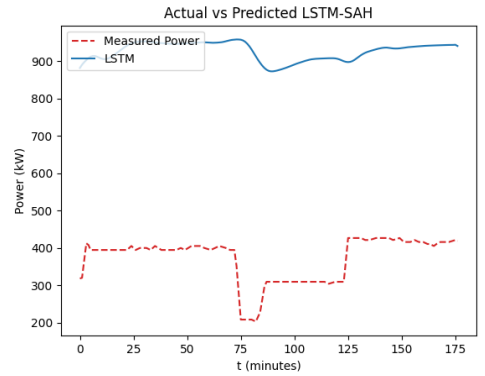
(c) LSTM vs. Measurements (Prediction 3)



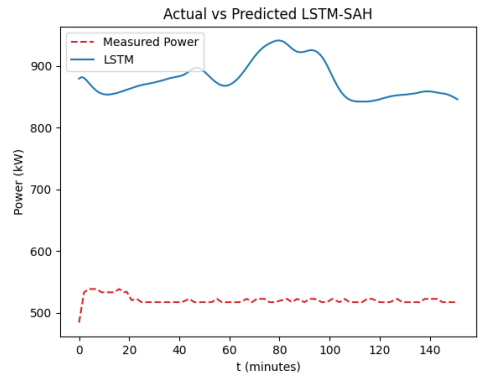
(d) LSTM vs. Measurements (Prediction 4)



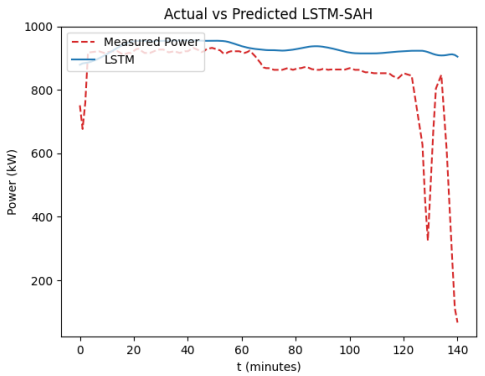
(e) LSTM vs. Measurements (Prediction 5)



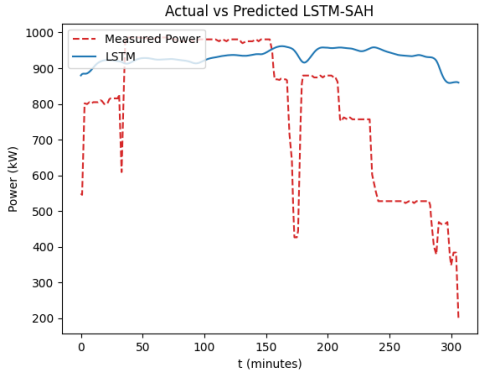
(f) LSTM vs. Measurements (Prediction 6)



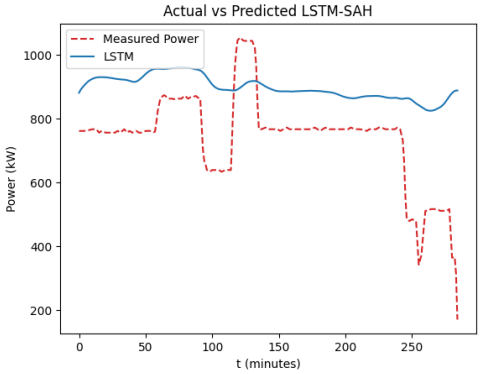
(g) LSTM vs. Measurements (Prediction 7)



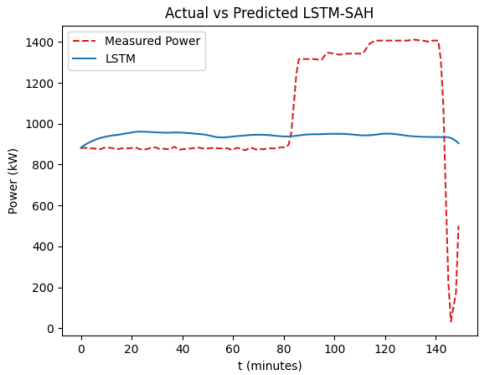
(h) LSTM vs. Measurements (Prediction 8)



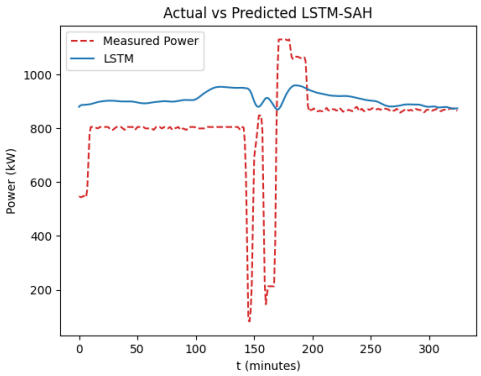
(a) LSTM vs. Measurements (Prediction 9)



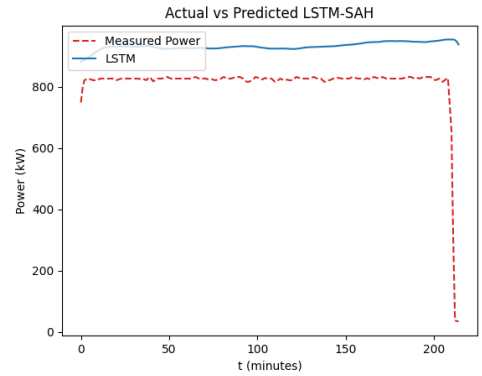
(b) LSTM vs. Measurements (Prediction 10)



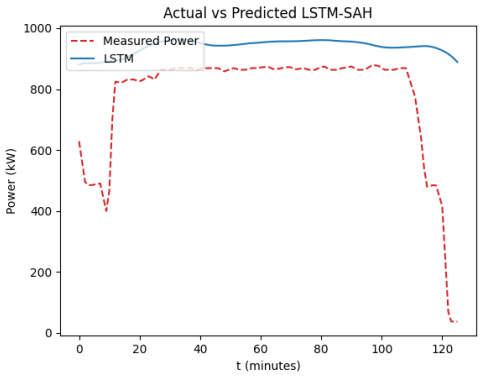
(c) LSTM vs. Measurements (Prediction 11)



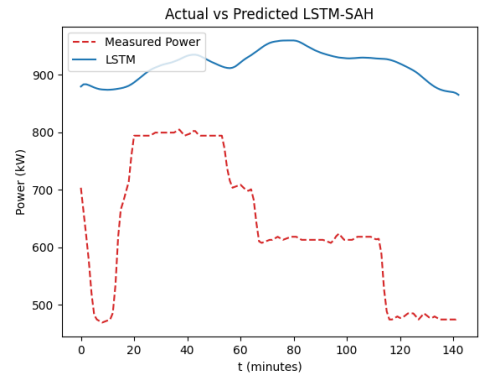
(d) LSTM vs. Measurements (Prediction 12)



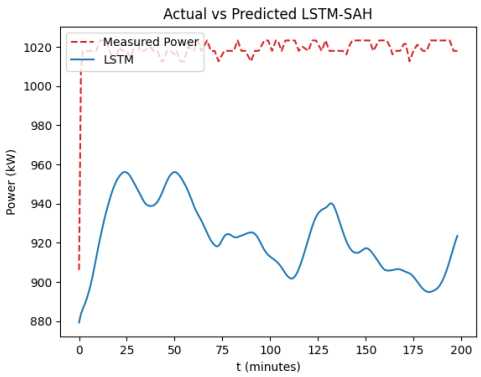
(e) LSTM vs. Measurements (Prediction 13)



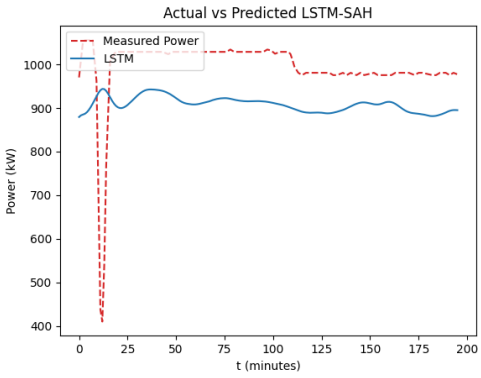
(f) LSTM vs. Measurements (Prediction 14)



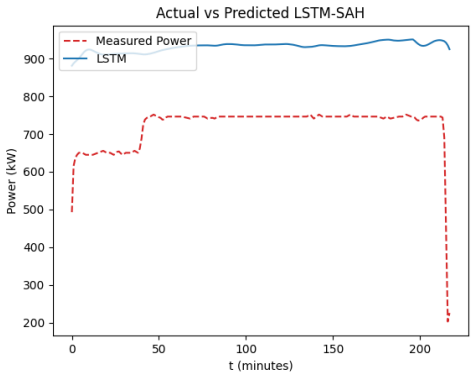
(g) LSTM vs. Measurements (Prediction 15)



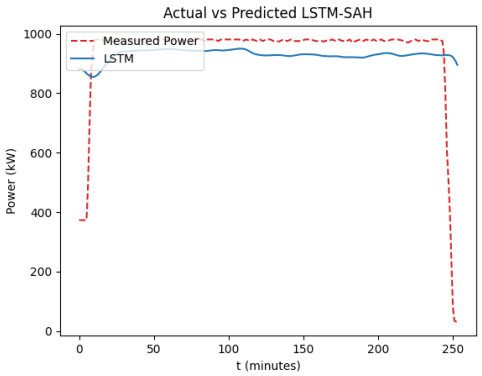
(h) LSTM vs. Measurements (Prediction 16)



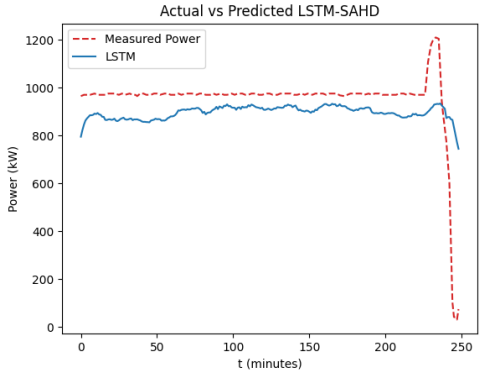
(a) LSTM vs. Measurements (Prediction 17)



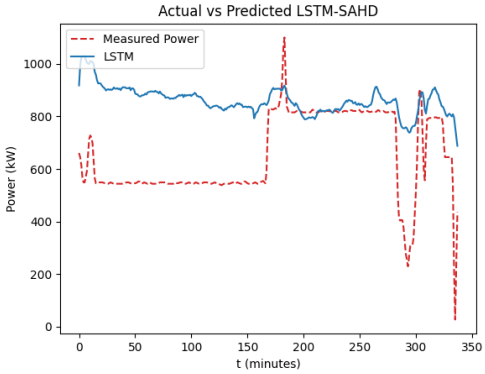
(b) LSTM vs. Measurements (Prediction 18)



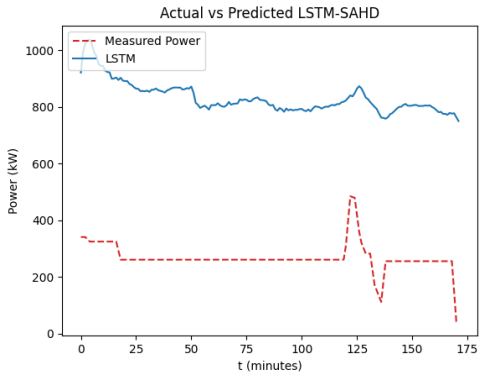
(c) LSTM vs. Measurements (Prediction 19)



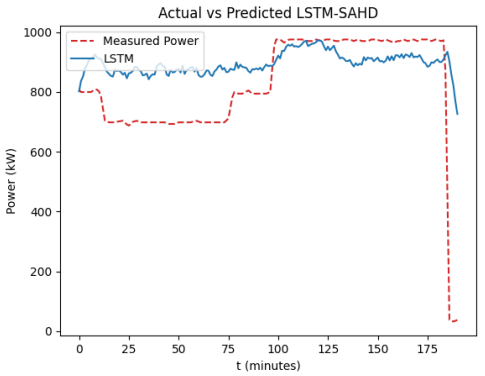
(a) LSTM vs. Measurements (Prediction 1)



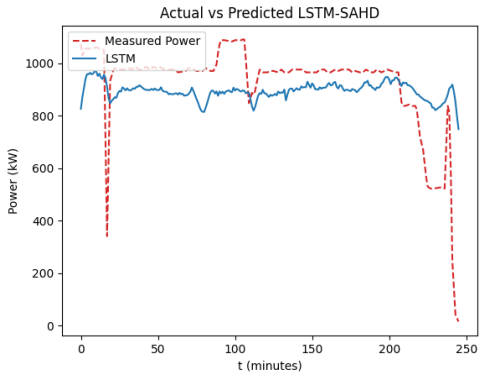
(b) LSTM vs. Measurements (Prediction 2)



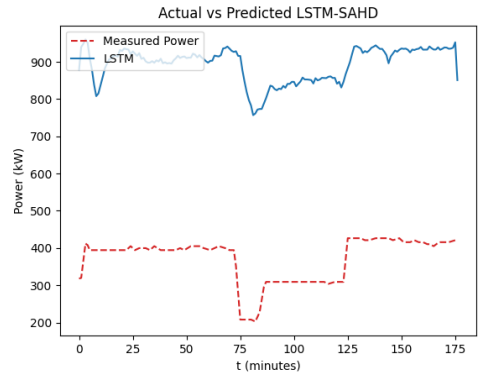
(c) LSTM vs. Measurements (Prediction 3)



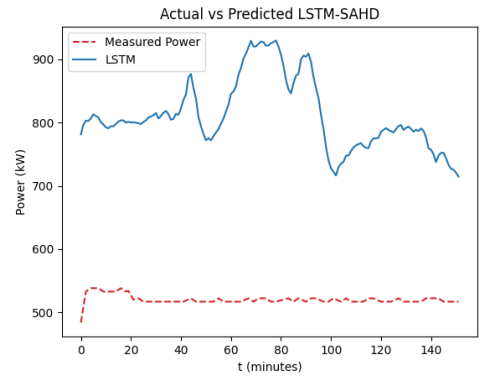
(d) LSTM vs. Measurements (Prediction 4)



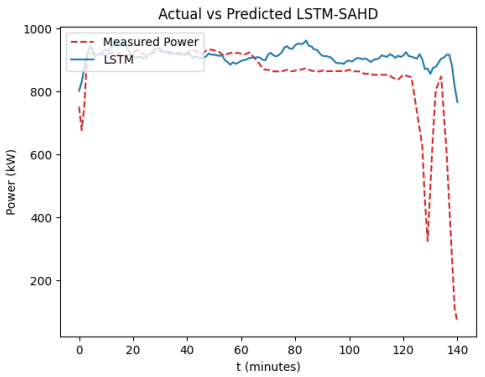
(e) LSTM vs. Measurements (Prediction 5)



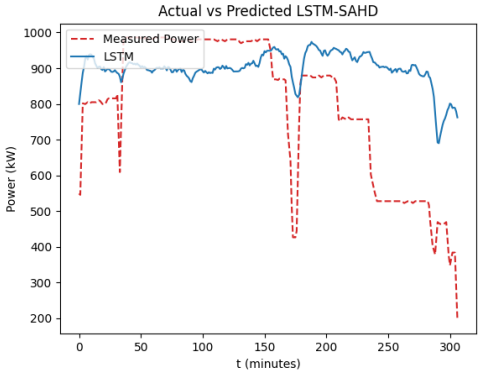
(f) LSTM vs. Measurements (Prediction 6)



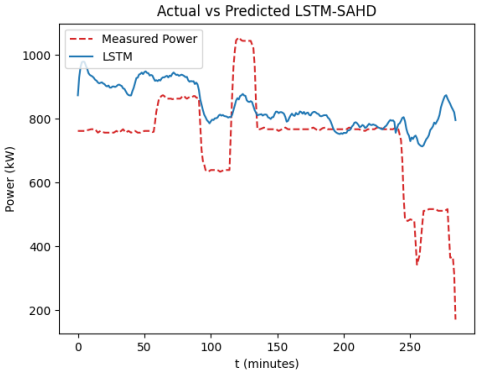
(g) LSTM vs. Measurements (Prediction 7)



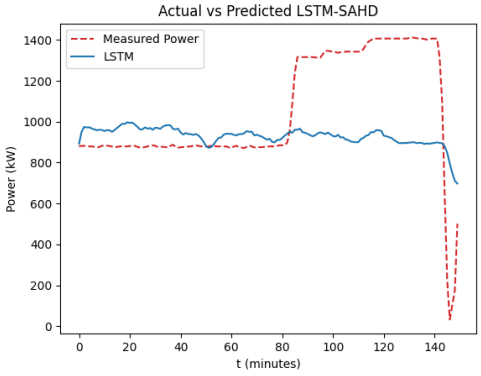
(h) LSTM vs. Measurements (Prediction 8)



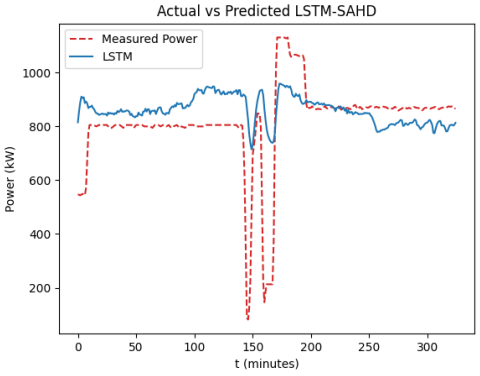
(a) LSTM vs. Measurements (Prediction 9)



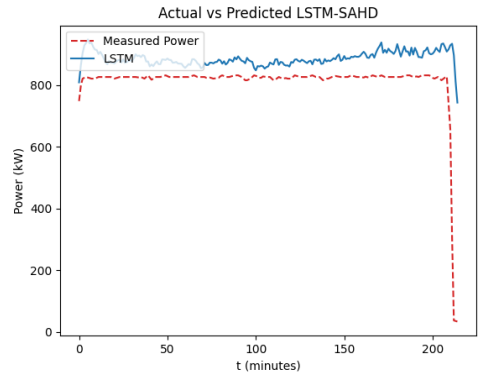
(b) LSTM vs. Measurements (Prediction 10)



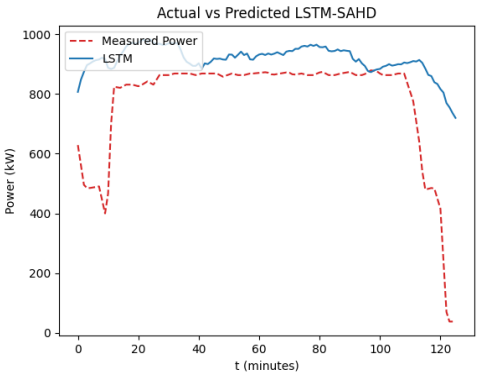
(c) LSTM vs. Measurements (Prediction 11)



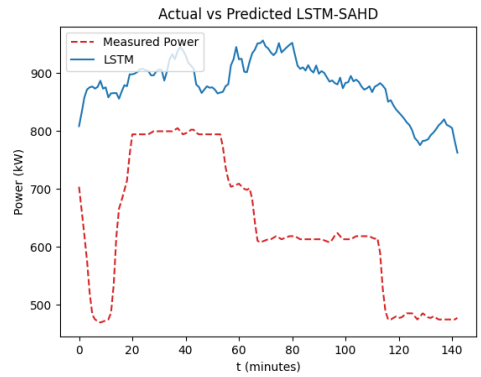
(d) LSTM vs. Measurements (Prediction 12)



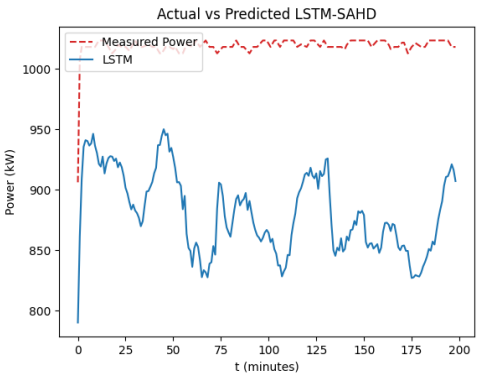
(e) LSTM vs. Measurements (Prediction 13)



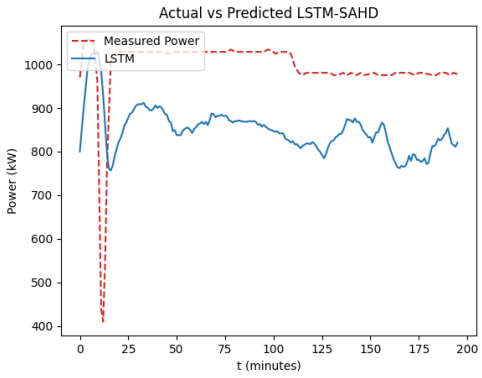
(f) LSTM vs. Measurements (Prediction 14)



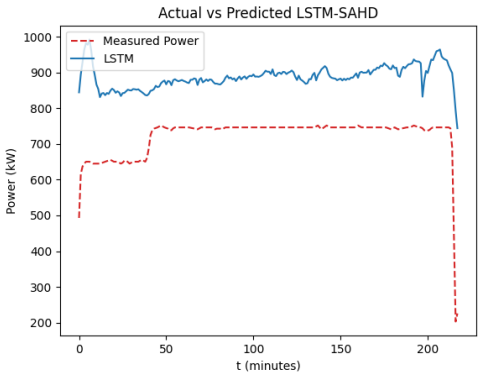
(g) LSTM vs. Measurements (Prediction 15)



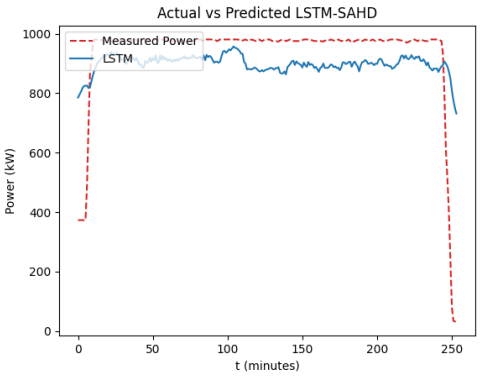
(h) LSTM vs. Measurements (Prediction 16)



(a) LSTM vs. Measurements (Prediction 17)



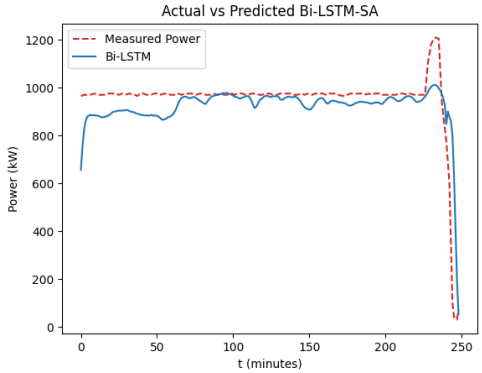
(b) LSTM vs. Measurements (Prediction 18)



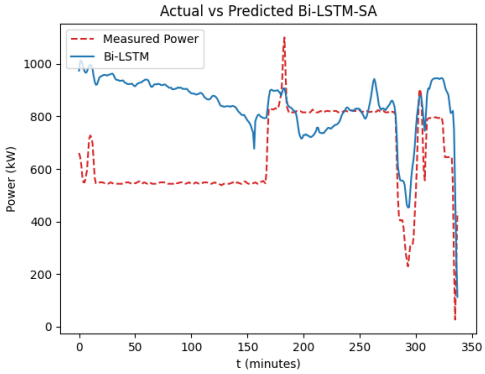
(c) LSTM vs. Measurements (Prediction 19)

D

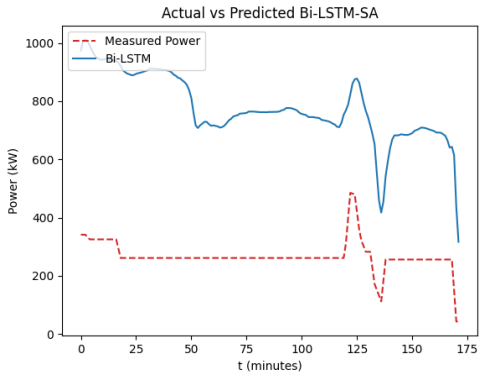
Results of Bi-LSTM for inland vessel



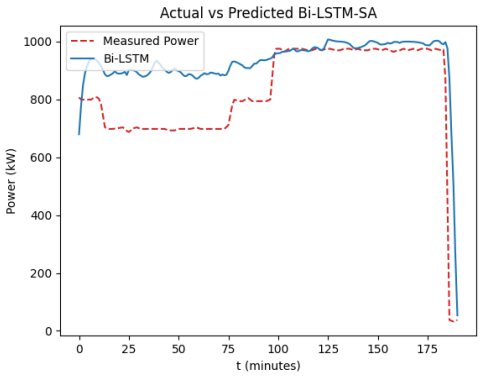
(a) Bi-LSTM vs. Measurements (Prediction 1)



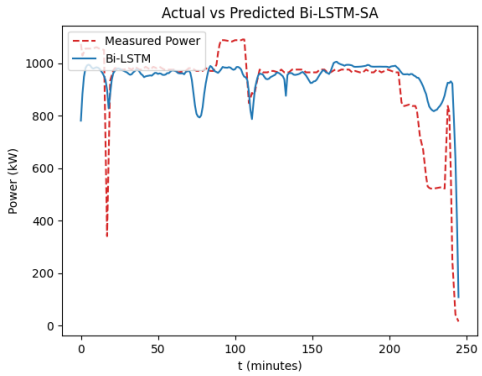
(b) Bi-LSTM vs. Measurements (Prediction 2)



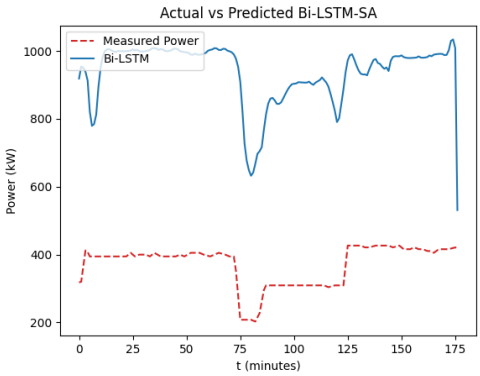
(c) Bi-LSTM vs. Measurements (Prediction 3)



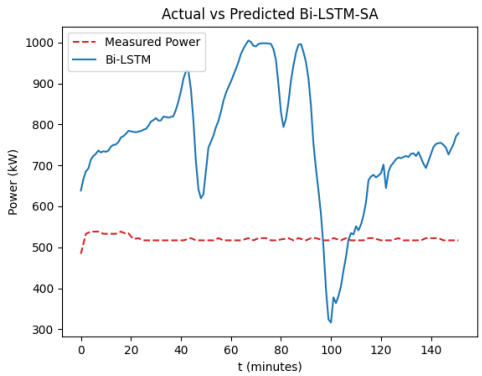
(d) Bi-LSTM vs. Measurements (Prediction 4)



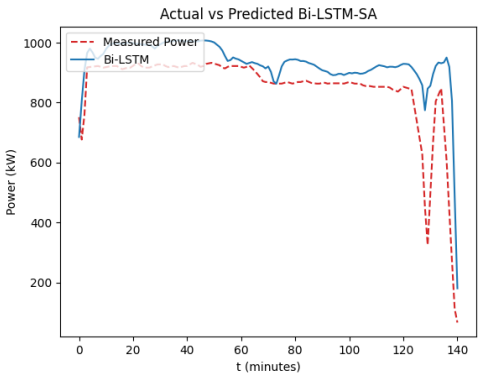
(e) Bi-LSTM vs. Measurements (Prediction 5)



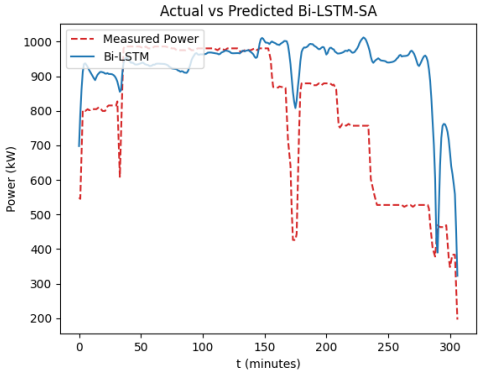
(f) Bi-LSTM vs. Measurements (Prediction 6)



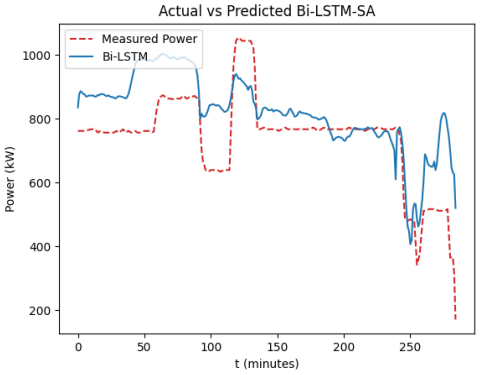
(g) Bi-LSTM vs. Measurements (Prediction 7)



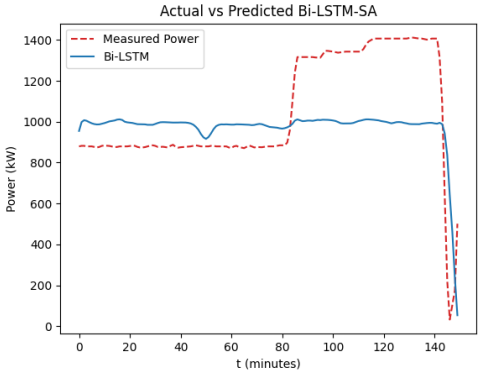
(h) Bi-LSTM vs. Measurements (Prediction 8)



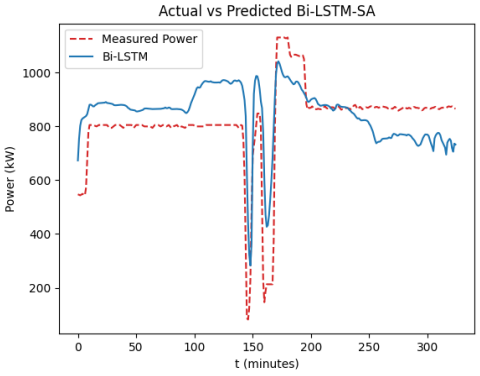
(a) Bi-LSTM vs. Measurements (Prediction 9)



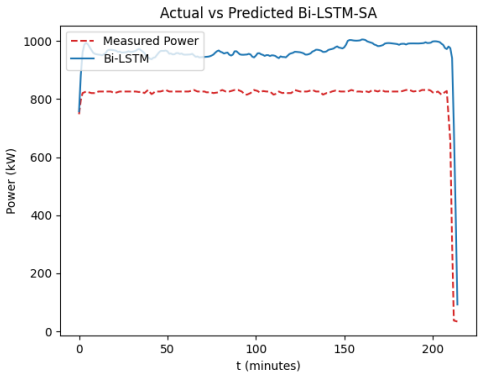
(b) Bi-LSTM vs. Measurements (Prediction 10)



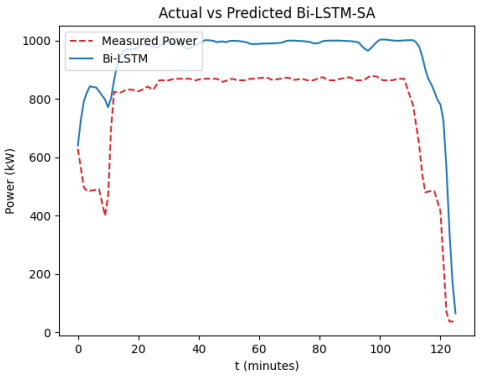
(c) Bi-LSTM vs. Measurements (Prediction 11)



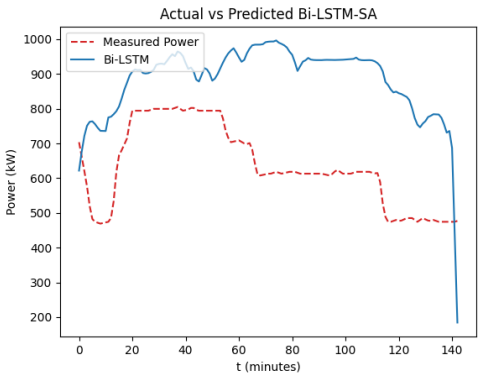
(d) Bi-LSTM vs. Measurements (Prediction 12)



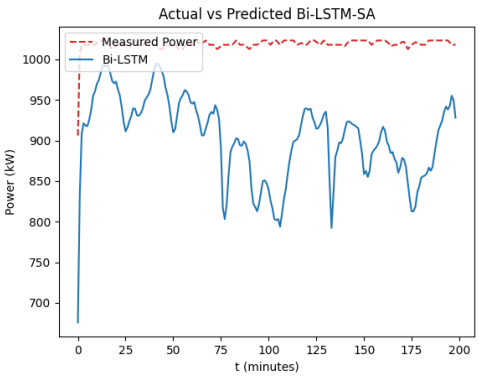
(e) Bi-LSTM vs. Measurements (Prediction 13)



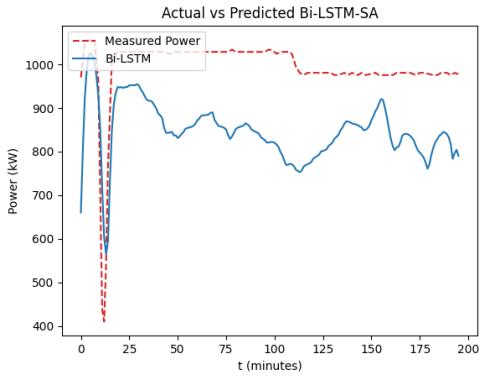
(f) Bi-LSTM vs. Measurements (Prediction 14)



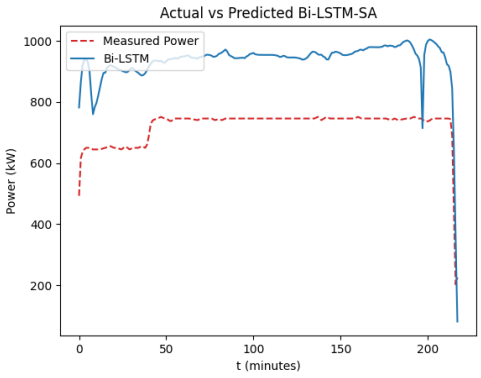
(g) Bi-LSTM vs. Measurements (Prediction 15)



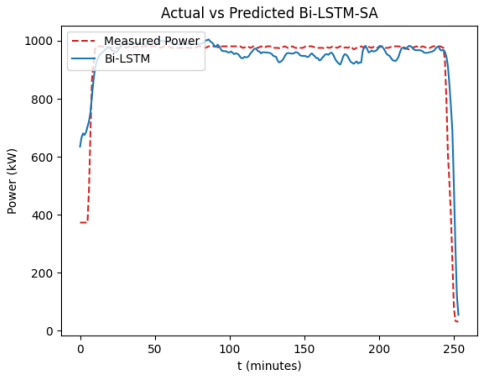
(h) Bi-LSTM vs. Measurements (Prediction 16)



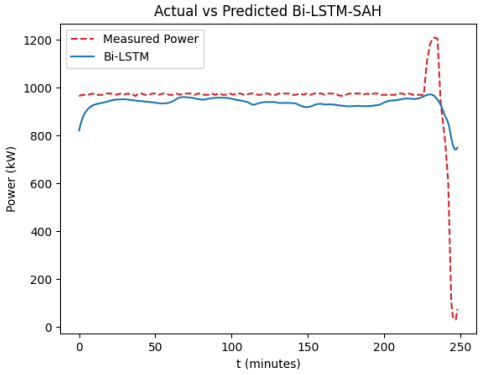
(a) Bi-LSTM vs. Measurements (Prediction 17)



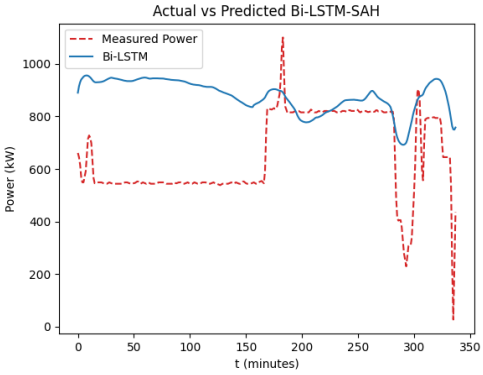
(b) Bi-LSTM vs. Measurements (Prediction 18)



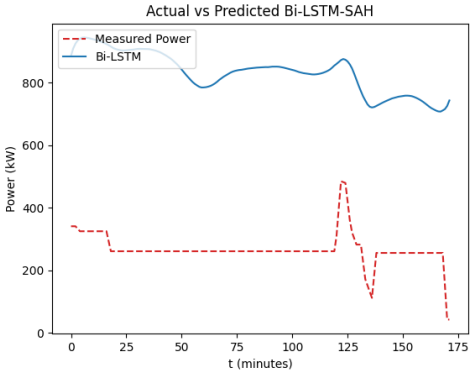
(c) Bi-LSTM vs. Measurements (Prediction 19)



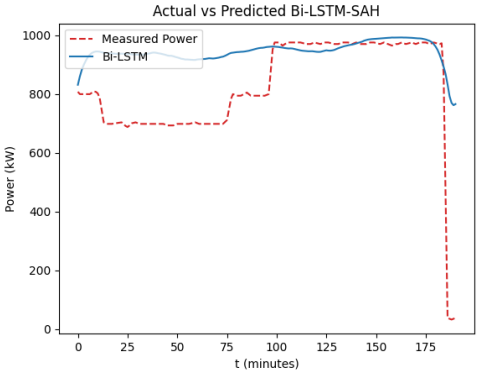
(a) Bi-LSTM vs. Measurements (Prediction 1)



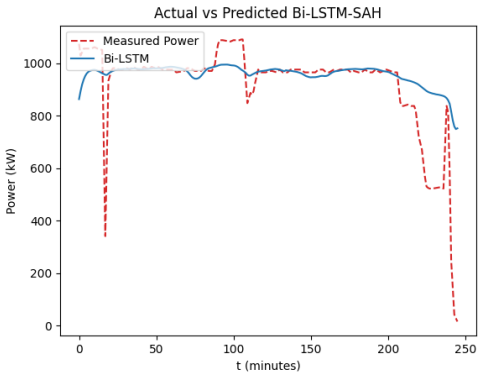
(b) Bi-LSTM vs. Measurements (Prediction 2)



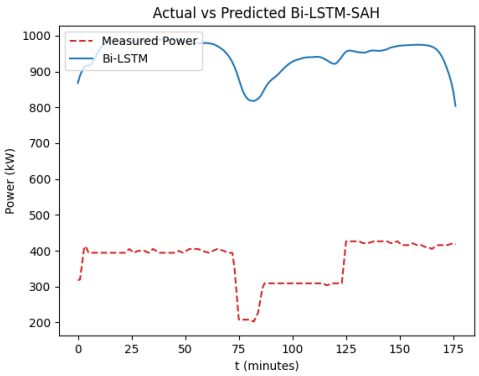
(c) Bi-LSTM vs. Measurements (Prediction 3)



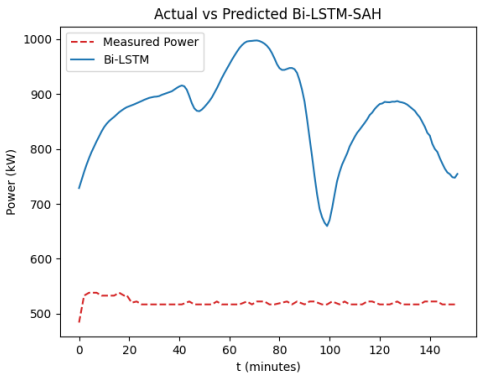
(d) Bi-LSTM vs. Measurements (Prediction 4)



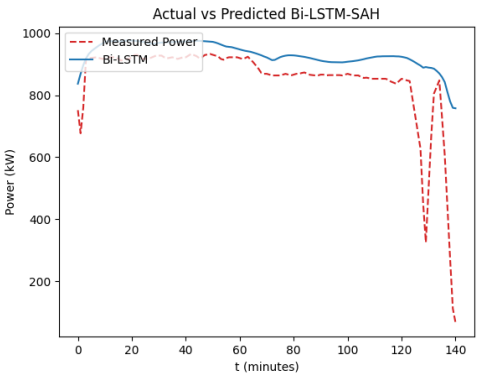
(e) Bi-LSTM vs. Measurements (Prediction 5)



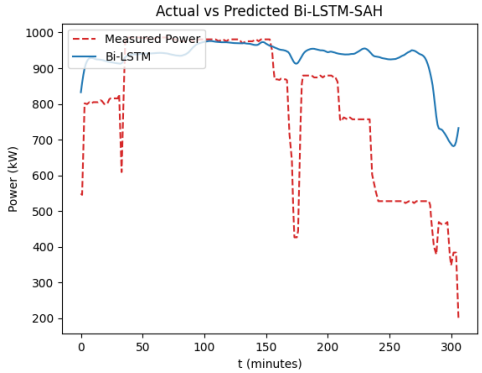
(f) Bi-LSTM vs. Measurements (Prediction 6)



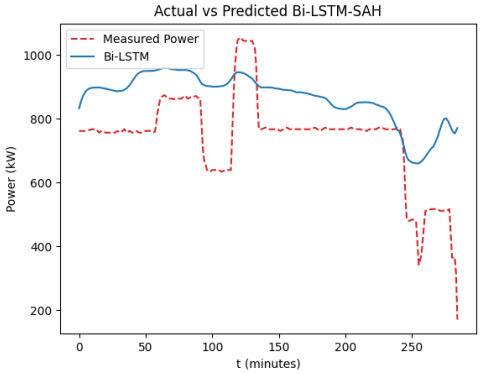
(g) Bi-LSTM vs. Measurements (Prediction 7)



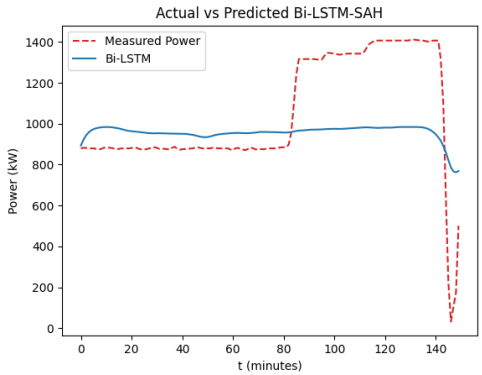
(h) Bi-LSTM vs. Measurements (Prediction 8)



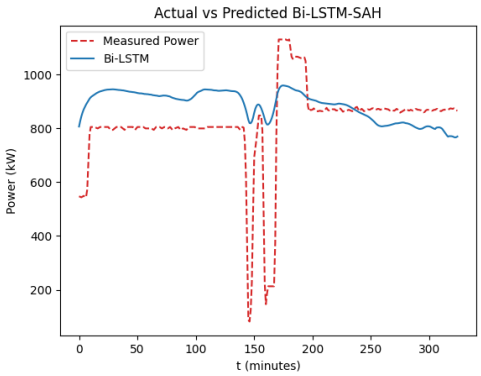
(a) Bi-LSTM vs. Measurements (Prediction 9)



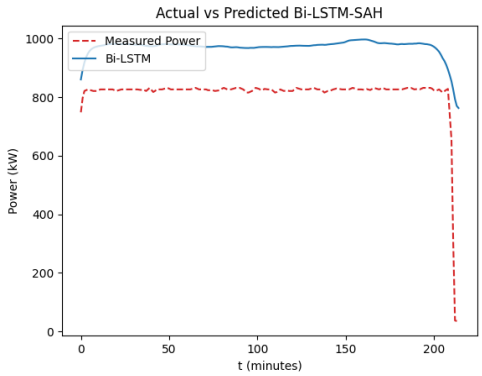
(b) Bi-LSTM vs. Measurements (Prediction 10)



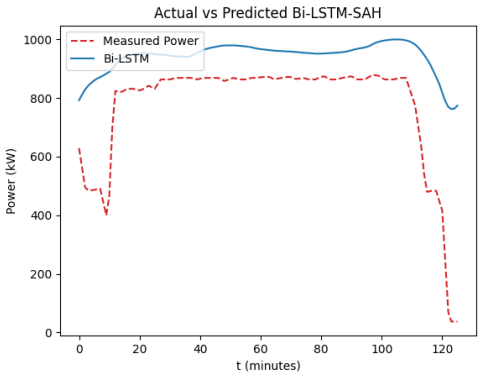
(c) Bi-LSTM vs. Measurements (Prediction 11)



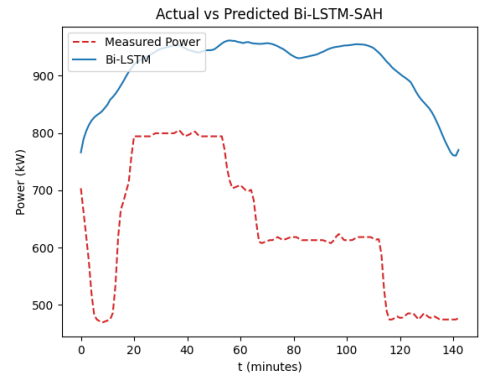
(d) Bi-LSTM vs. Measurements (Prediction 12)



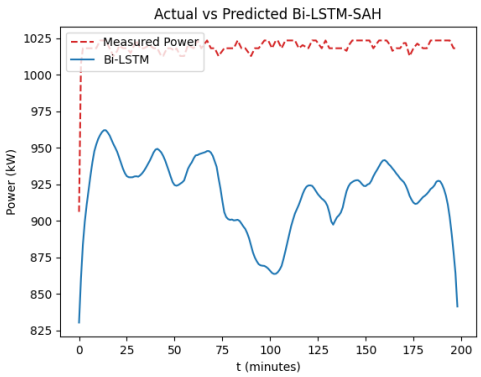
(e) Bi-LSTM vs. Measurements (Prediction 13)



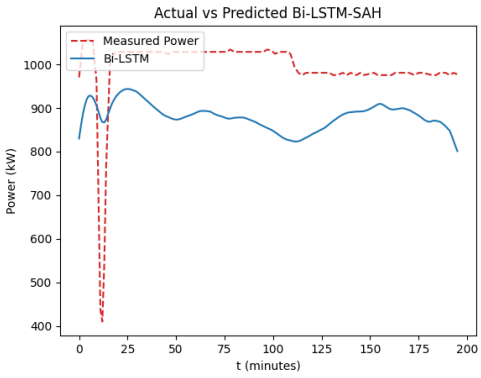
(f) Bi-LSTM vs. Measurements (Prediction 14)



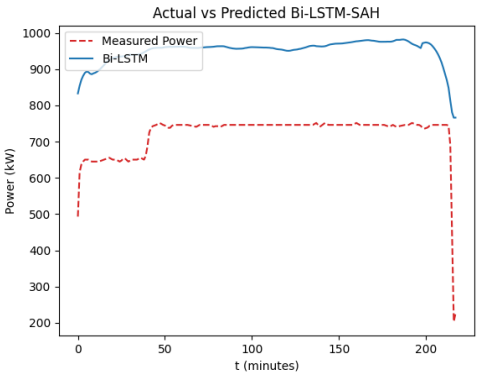
(g) Bi-LSTM vs. Measurements (Prediction 15)



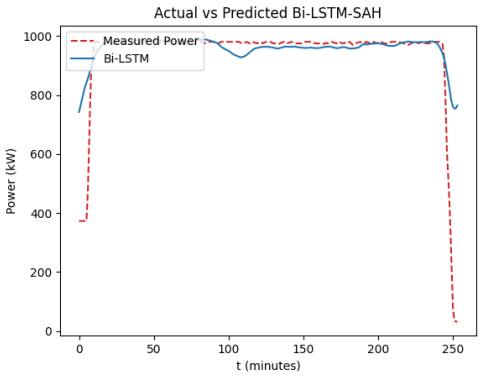
(h) Bi-LSTM vs. Measurements (Prediction 16)



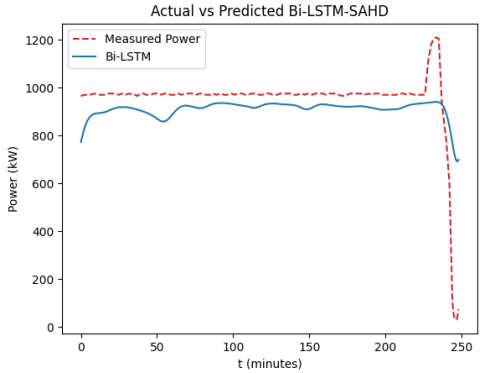
(a) Bi-LSTM vs. Measurements (Prediction 17)



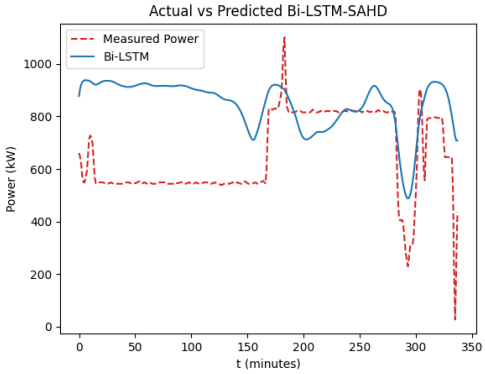
(b) Bi-LSTM vs. Measurements (Prediction 18)



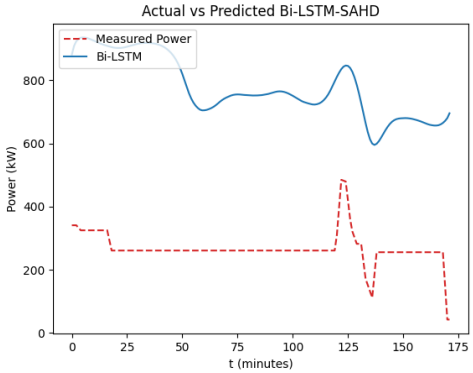
(c) Bi-LSTM vs. Measurements (Prediction 19)



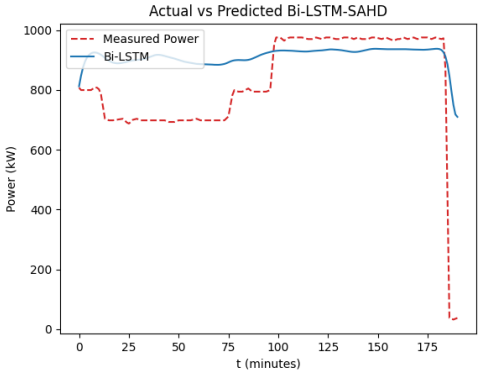
(a) Bi-LSTM vs. Measurements (Prediction 1)



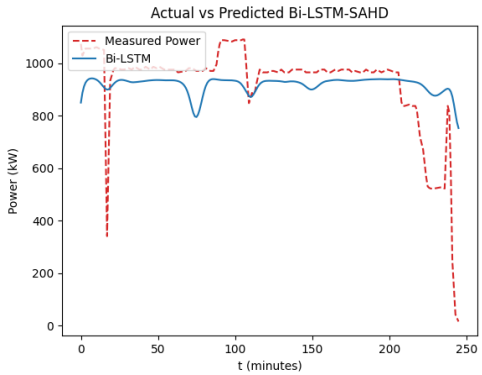
(b) Bi-LSTM vs. Measurements (Prediction 2)



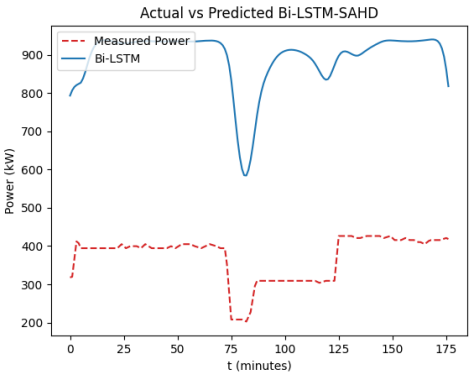
(c) Bi-LSTM vs. Measurements (Prediction 3)



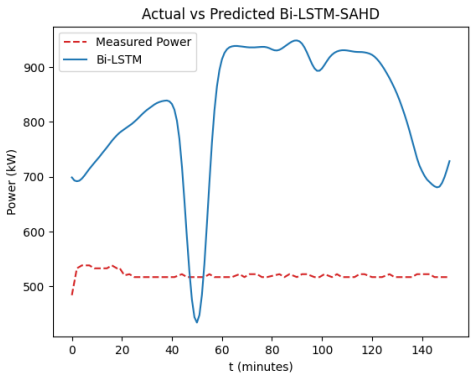
(d) Bi-LSTM vs. Measurements (Prediction 4)



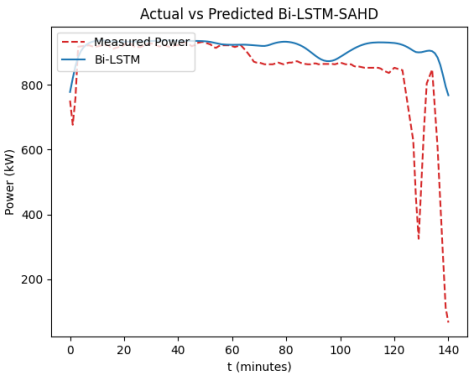
(e) Bi-LSTM vs. Measurements (Prediction 5)



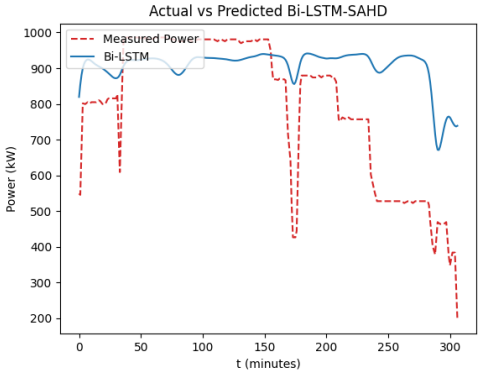
(f) Bi-LSTM vs. Measurements (Prediction 6)



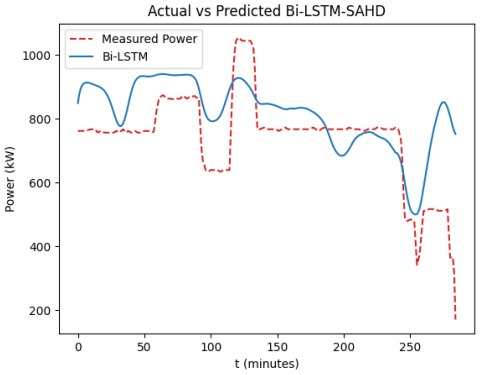
(g) Bi-LSTM vs. Measurements (Prediction 7)



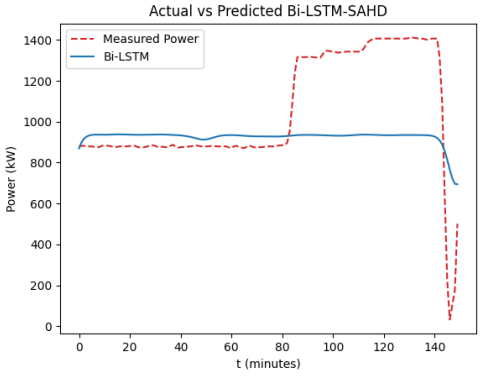
(h) Bi-LSTM vs. Measurements (Prediction 8)



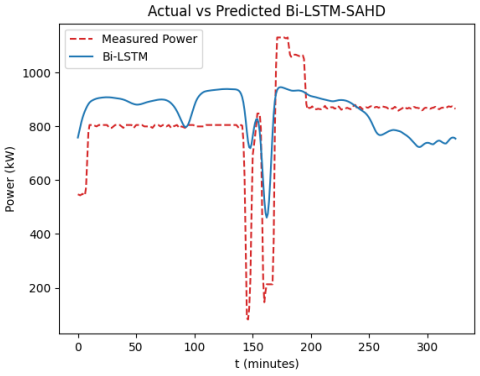
(a) Bi-LSTM vs. Measurements (Prediction 9)



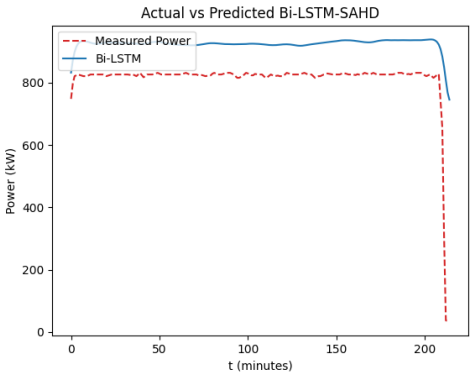
(b) Bi-LSTM vs. Measurements (Prediction 10)



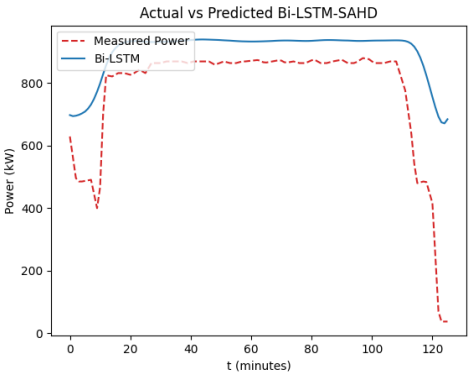
(c) Bi-LSTM vs. Measurements (Prediction 11)



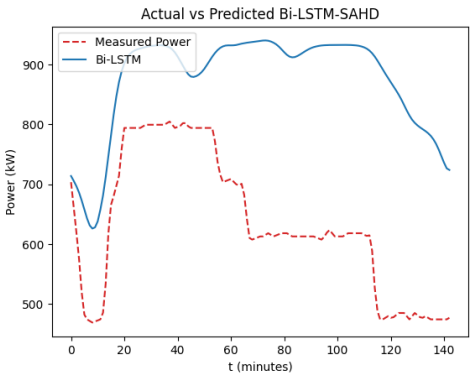
(d) Bi-LSTM vs. Measurements (Prediction 12)



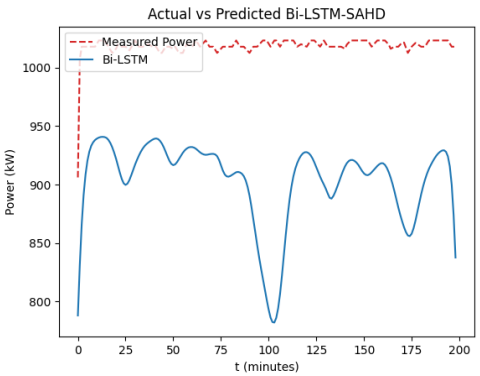
(e) Bi-LSTM vs. Measurements (Prediction 13)



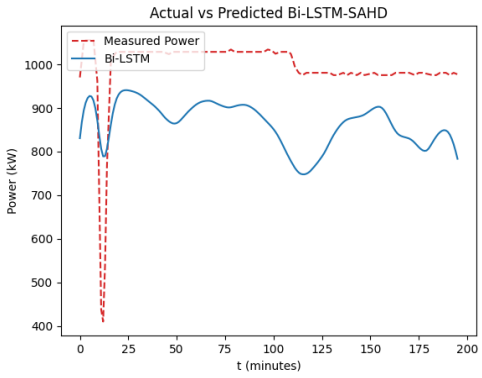
(f) Bi-LSTM vs. Measurements (Prediction 14)



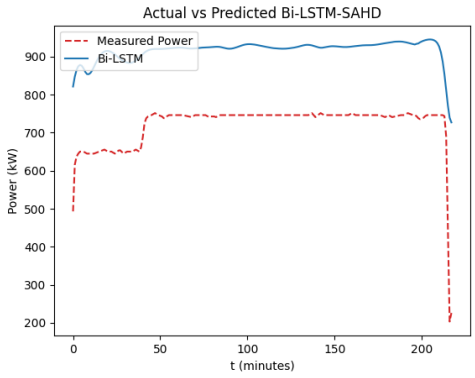
(g) Bi-LSTM vs. Measurements (Prediction 15)



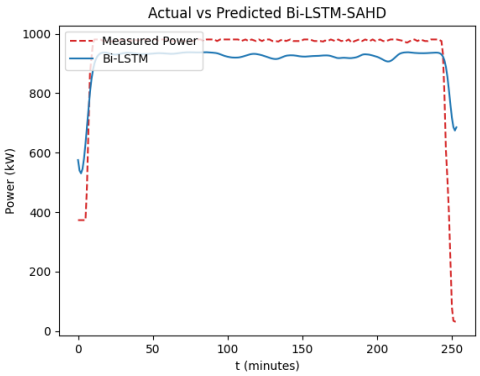
(h) Bi-LSTM vs. Measurements (Prediction 16)



(a) Bi-LSTM vs. Measurements (Prediction 17)



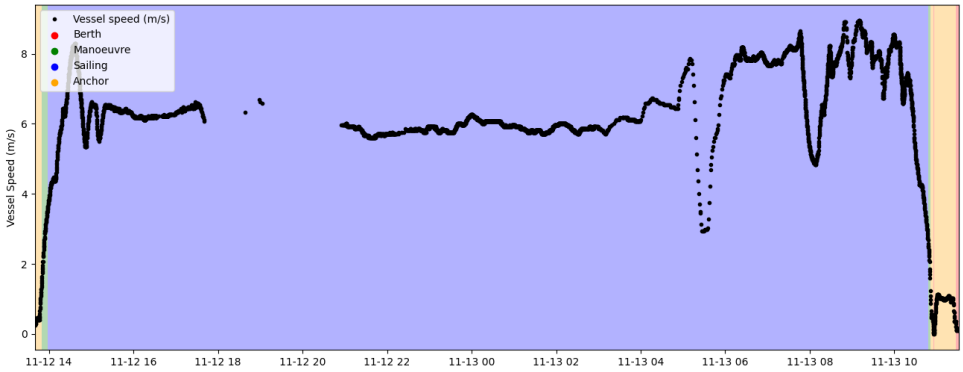
(b) Bi-LSTM vs. Measurements (Prediction 18)



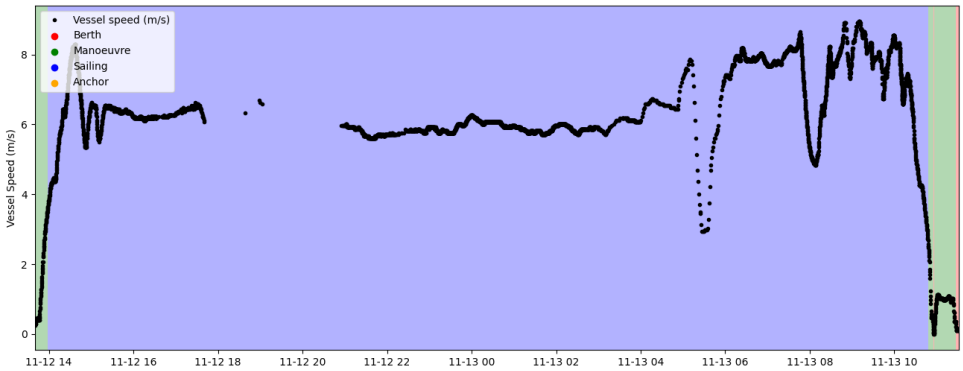
(c) Bi-LSTM vs. Measurements (Prediction 19)

E

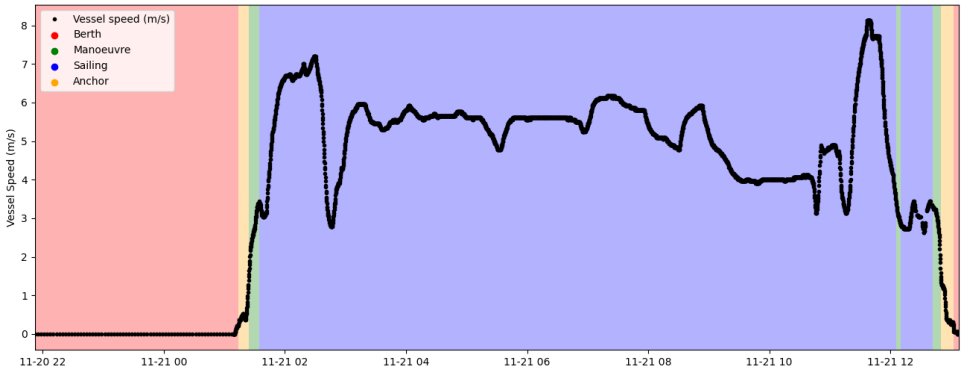
Operational mode visualizations



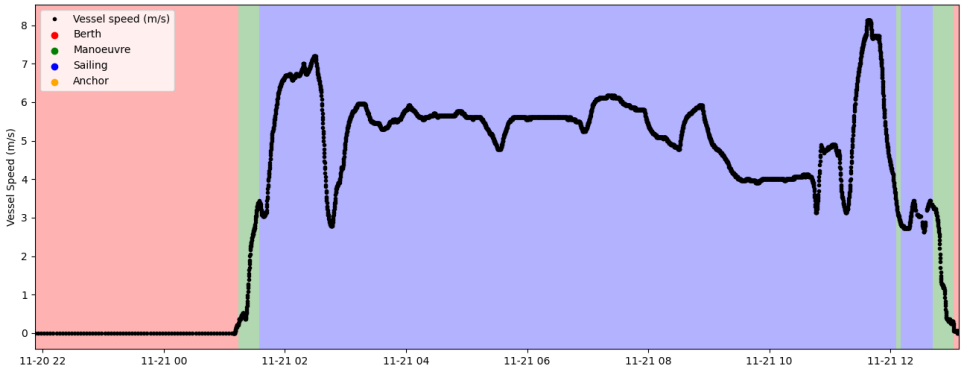
(a) Old mode assignment



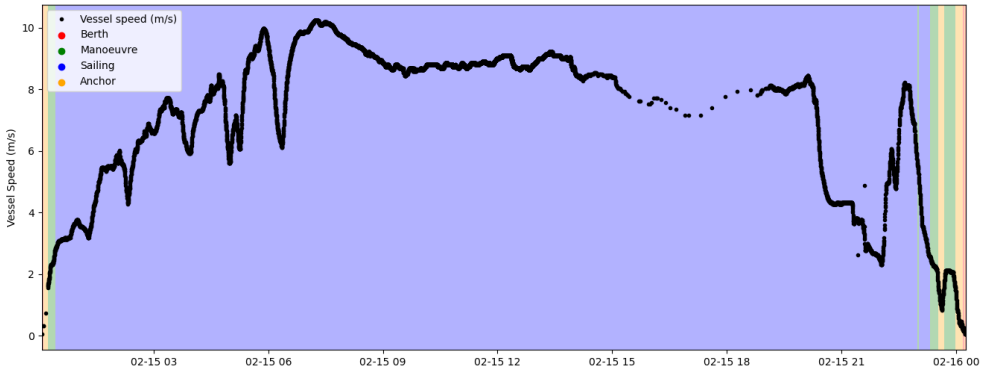
(b) New mode assignment



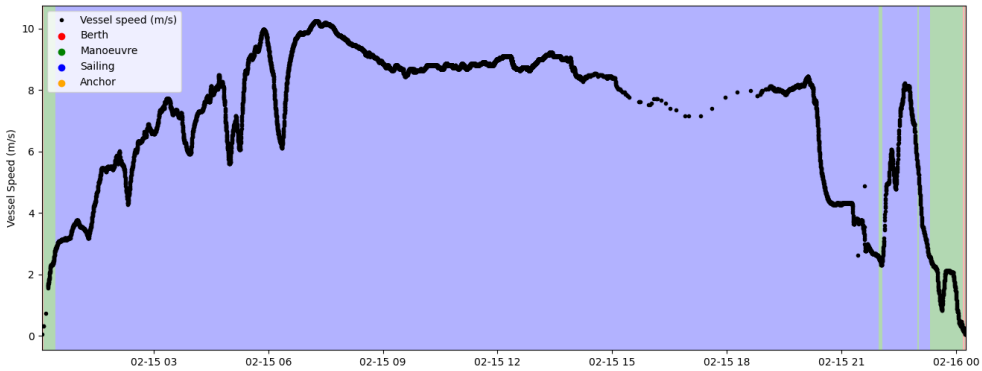
(c) Old mode assignment



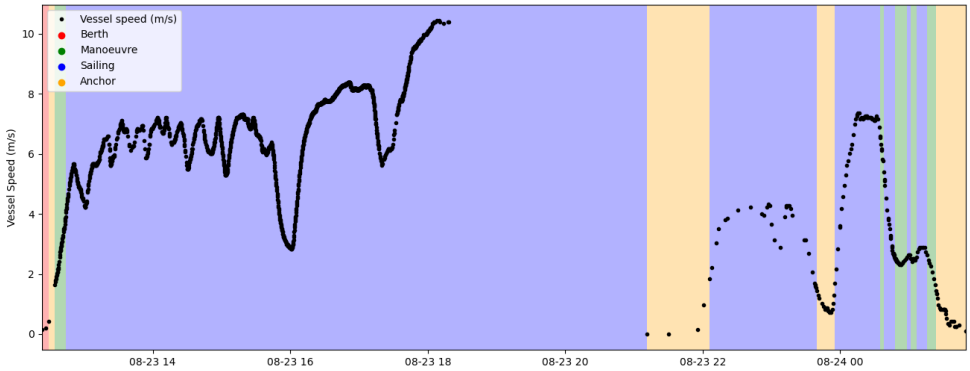
(d) New mode assignment



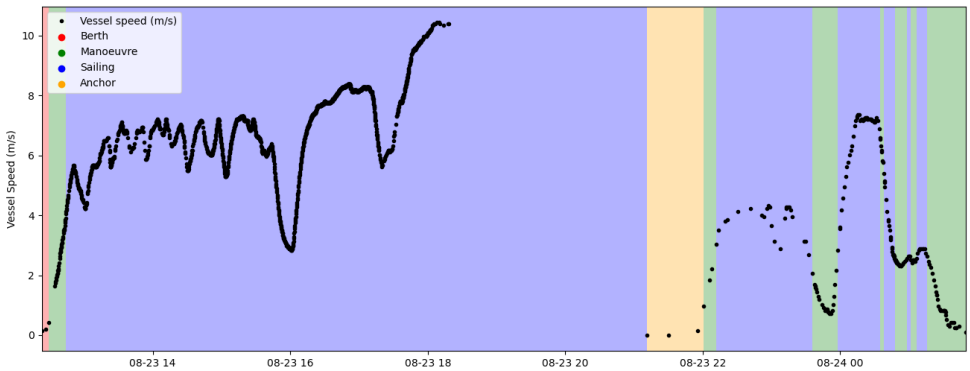
(a) Old mode assignment



(b) New mode assignment



(c) Old mode assignment



(d) New mode assignment