

Predicting short-term heave motion

Short-term Heave Motion Time Series Prediction Using CNN-LSTM Neural Networks

Quinten Sandberg 9351981



Master Applied Data Science
Utrecht University

UU Supervisor : Deb Panja
MARIN Supervisors: Eelco Frickel and Bulent Duz
Date : 1 July 2022
Programme : MSc Applied Data Science

Abstract

This report explains how to forecast first-order heave motions caused by a wave field of a helicopter deck with forwarding speed using a neural network. A CNN-LSTM model using 100 simulations with sea state level 5 was used to create these models. The decision is made to resample the data in order to reduce the computational cost. It is discovered that when data is resampled at 4, no information would be lost. Two CNN-LSTM models were created, one with one CNN layer and the other with two CNN layers. The latter called outperforms the others, according to the results. The results demonstrate that the scores for forecasting the heave motion are satisfactory. The validation loss of training was 0.00524 and the RMSE score was 1.176 for the model. It is indicated that the proposed model can explore effectively the features of univariate time series data. This could even be expanded to multivariate time series knowing the possibilities of neural networks.

Acknowledgements

I would like to thank my supervisor Deb Panja from Utrecht University, for his guidance throughout this project. This research would not have been possible without the support of my supervisors from MARIN, Eelco Frickel and Bulent Duz. Major thanks for your support and it was a pleasure to have a look at your office at MARIN in Wageningen. I am also thankful to my fellow students Jolan Keijzer and Maria Zafeiraki, with whom I have had a great time and collaboration during this project. Lastly, I would be remiss not mentioning my family. Their belief in me gave me always high motivation during this process.

Contents

1	Introduction	1
2	Data	4
2.1	Data exploration	4
2.2	Data preparation	5
3	Methods	7
3.1	CNN-LSTM	8
3.2	Hyperparameters	9
3.3	Evaluate model	10
4	Results	11
4.1	Tuning	11
4.2	Training	12
4.3	Prediction	13
5	Conclusion and Discussion	15
	Bibliography	18
	Appendices	a
A	Formulas	a
B	Figures and tables	b
C	Scripts	d

List of Tables

2.1	Descriptive summary	4
3.1	Hyperparameter values	9
4.1	Train and validation loss of tuning model	11
4.2	Optimal hyperparameter setting	12
4.3	Train and validation loss of training model	12
4.4	Average RMSE of models	14
B.1	Description of hyperparameters	b

List of Figures

1.1	Six-degrees-of-freedom ship motion. [Tanaka, 2018]	2
2.1	Left: Histogram distribution with density curve, Right: Boxplot	5
2.2	Part of a single simulation.	5
2.3	Resampled simulation with sample of 4.	6
3.1	Workflow	8
3.2	CNN-LSTM architecture	9
4.1	Validation and train loss tuning with two CNN layers	12
4.2	Validation and train loss of trained model.	13
4.3	RMSE per timestep	13
B.1	Validation and train loss tuning	c
B.2	Validation and train loss of training model	c

Chapter 1

Introduction

It is not uncommon that two bodies are merged or split during an offshore multi-body operation. Take for example, helicopter/drone landings and take-offs. The exact timing when to commence the final phase of the operation is crucial for a safe executing of the established procedure. In general a superintendent on-board is in control of the operation. This person is aware of the ship motions and indicates when the landing should take place. Due to random and disordered waves the safety and efficiency of marine operations is in danger. Improving the accuracy of ship motion prediction is conducive to decision-making for performing motion sensitive maritime activities. Obtaining reliable prediction uncertainty information is also beneficial to avoid potential navigation risks [Sun et al., 2022]. Therefore, the primary goal of our study was to predict upcoming heave waves, by using a machine learning model. The simulation data is provided by the Maritime Research Institute Netherlands (MARIN).

The motion of a ship is exhibited in figure 1.1 that depicts the six-degrees-of-freedom (6-DOF). Ship motions are defined by the 6-DOF that a ship or any other craft can experience. A body that does not face restrictions in motion can have 6-DOF. The ability of the body to move freely in that motion is referred to as a degree of freedom. There are three spatial axes, and the movements around these three axes are the roll, the pitch, and the yaw. Pitch refers to an up/down rotation around the lateral axis; roll refers to a tilting rotation around the longitudinal axis; and yaw refers to a turning rotation around the vertical axis. Heave is a vertical linear motion, sway is a lateral linear motion, and surge is a longitudinal linear motion [Tanaka, 2018]. In this paper only the heave motion will be addressed. This causes the ship to rise at particular points and has a large effect on helicopter/off-shore vessel operations. [Menon, 2021]

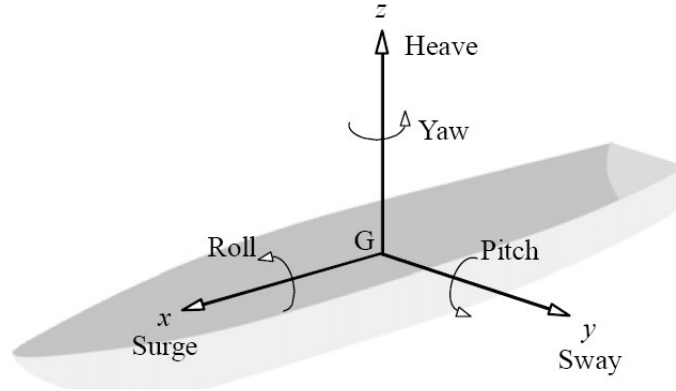


Figure 1.1: Six-degrees-of-freedom ship motion. [Tanaka, 2018]

Time series data is data collected for a single entity over time. It allows estimation of the effect on y of a change in x over time. Time series regression and forecasting are methods for predicting future values in time series data. Time series regression's main notion is that it forecasts the time series of interest y assuming that it has a linear relationship with other time series x [2021a]. Forecasting or predicting the future value over a period of time is known as time series forecasting. It comprises creating models based on historical data and using them to make observations and make strategic decisions in the future. [2021b]

The choice of model depends on the goal for the analysis and the properties of the data. The main distinction is that regression is interpolation whereas forecasting is extrapolation. In order to achieve the project's goal, a time series forecasting approach was chosen. Because the superintendent will have more information to make judgments with. This implies there will be more information on how many probable heaves are coming, as well as when they will arrive and whether they will be large or little.

The neural network model has been expanded in recent years with the emergence of deep learning to provide greater choice space for time series prediction [Li et al., 2022]. Kuremoto et al. [Kuremoto et al., 2014] employed a Deep Belief Network (DBN) to forecast time series and showed that DBN outperforms the classic Artificial Neural Network (ANN) and autoregressive integrated moving average (ARIMA) models. The Long Short-Term Memory (LSTM) network was utilized by Peng et al. [Peng et al., 2019] to forecast ship attitude, proving the viability of utilizing a recurrent neural network to predict ship motions. Zhang et al. [Zhang et al., 2019] used Convolutional Neural Network (CNN) and LSTM to create a rolling motion prediction model for unmanned surface vehicles (USV), and the experimental findings

demonstrate that the extraction of time series features of ship motions by CNN helps to improve the model's prediction accuracy. A single model's prediction accuracy is generally poorer than a hybrid model's when dealing with dynamically evolving non-linear ship motion time series [Duan et al., 2015]. Although the deep learning model has a high learning ability and can imitate the ship motions mechanism, the hyperparameter selection has a significant impact on prediction accuracy [Liu et al., 2020]. As a result, it is worthwhile to investigate an acceptable way of hyperparameter optimization. In general, the deep learning model's hyperparameters must be adjusted by a person, and this procedure necessitates a great deal of practical expertise [Li et al., 2022]. The goal is to predict the upcoming heave waves within 10 to 20 seconds using a hybrid model of LSTM and CNN.

The rest of this report is arranged as follows. Section 2 introduces the data preparation and manipulation. The model and analysis are presented in section 3. In section 4, are the results of ship motion prediction and the analysis of prediction results are presented. Section 5 summarizes the results of this research.

Chapter 2

Data

The data set are simulated time traces, which is based on a mass-spring-damper system with (Non) linear response where the vessel speed, wave spectrum and direction are constant. A frigate ship (5415M vessel) has been used in a simulation program called Fredyn, a computer software that mimics the dynamic behavior of a steered ship under ship conditions like waves and wind. The simulations are run for a single sea state and ship speed. There are in total 100 simulations each 3 hours long with a sample time of 0.2 seconds. The simulations' attribute "RefZ" is called as the heave motion. A descriptive statistical summary of the essential aspects is presented in table 2.1, which shows that the mean of heave motion for all the simulations is -6.1341 with a standard deviation of 0.0004.

Table 2.1: Descriptive summary

Variable	N	Mean (sd)
RefZ	100	-6.1341 (0.0004)

2.1 Data exploration

Exploratory data analysis, often known as data exploration, provides a collection of simple techniques for gaining a fundamental understanding of a dataset. These data exploration results can help you understand the data's structure, value distribution, and the occurrence of extreme values. Figure 2.1 shows a histogram with a density curve on the left and a boxplot for both the heave motion and the means distribution of all the simulations on the right.

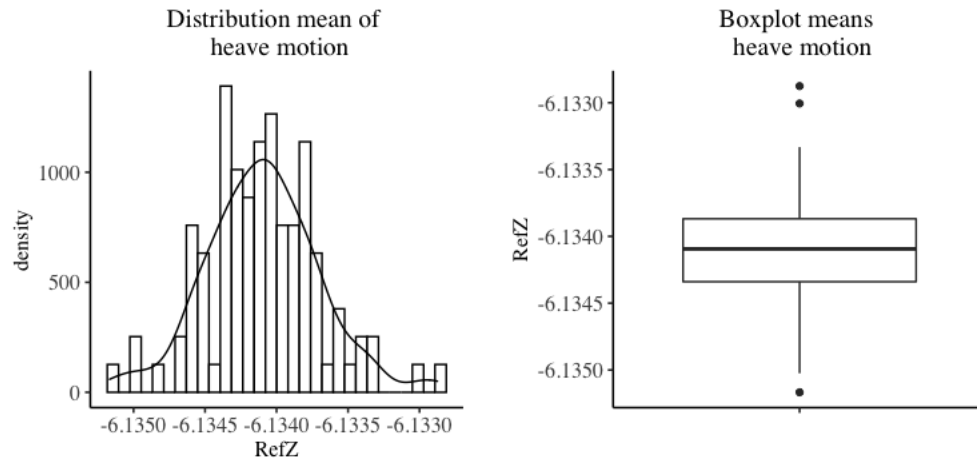


Figure 2.1: Left: Histogram distribution with density curve, Right: Boxplot

2.2 Data preparation

The data are simulations with a total of zero missing values. There are 55000 values in each simulation. The data will be resampled using a shift window, which will filter the data without losing too much information, to reduce the computational cost. Figure 2.2 illustrates a part of a single simulation without resampling.

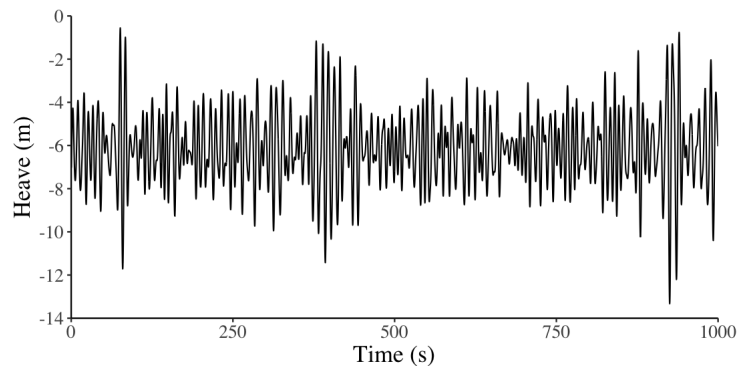


Figure 2.2: Part of a single simulation.

The resampled data with a sample time of 4 is displayed in figure 2.3.

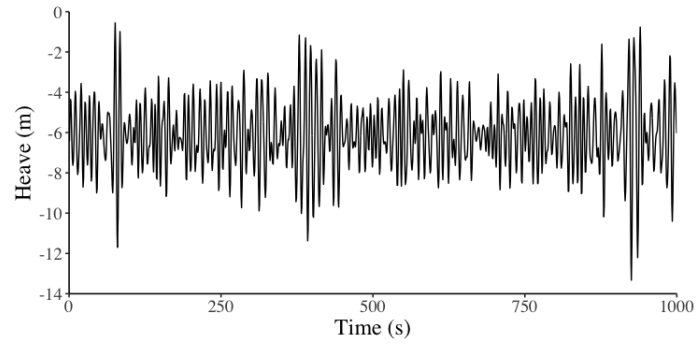


Figure 2.3: Resampled simulation with sample of 4.

As seen in Figures 2.2 and 2.3, resampling the data does not result in the loss of any critical information. It maintains the pattern and even retains the signal peaks.

Chapter 3

Methods

The purpose of this study is to forecast heave motions in the next 10 to 20 seconds. The question is, "Can a CNN-LSTM model with 100 simulations be used to anticipate these impending heave motions in such a way that it can help the superindendent on-board in general?"

The data will be prepared according to the instructions in section 2. The data will be divided into three splits: training (70), validation (15), and testing (15). The Min-Max scaler (IA.1) will be used to improve the algorithm's performance by bringing the results closer to the normal distribution. After that, it will loop through the datasets, creating samples with a specific input length and prediction length for each dataset. The input length is the number of samples that will be used to train the data, whereas the prediction length is the number of samples that will be validated later. The CNN LSTM model must then be trained and tuned. The Keras-tuner will be used to determine hyperparameters such as the number of units in LSTM layer, learning rate, dropout, epochs, kernel size and batch size. This is because it is specifically built for tuning hyperparameters from the library Keras, which has clear documentation. Two CNN-LSTM models will be built, with the first having only one CNN layer and the second having two CNN layers. The best model will be utilized to anticipate the heave motion based on the performance. The workflow of the entire process is shown in figure 3.1.

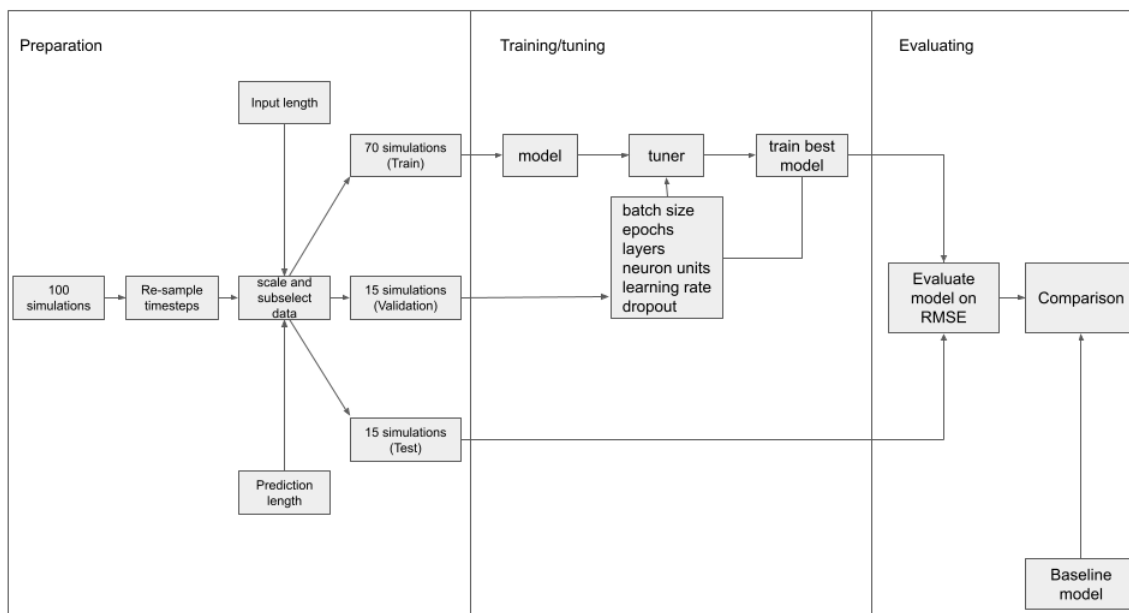


Figure 3.1: Workflow

3.1 CNN-LSTM

A combination of CNN and LSTM architecture is the CNN Long Short-Term Memory Network (CNN-LSTM). CNN layers for feature extraction on input data are paired with LSTMs to allow sequence prediction in the CNN-LSTM architecture. CNN layers on the front end are followed by LSTM layers with a Dense layer on the output to create a CNN-LSTM. The Dropout layer is between both layers, which helps minimize overfitting, sets input units to 0 at random with a rate frequency at each step during training time. The architecture is made up of two sub-models: the CNN Model for feature extraction and the LSTM Model for feature interpretation over time. The number of CNN layers that can be used varies. As a result, two models will be constructed. One CNN layer is used in the first model, while two CNN layers are used in the second model, both in conjunction with one LSTM layer. A ship motion model based on CNN-LSTM is constructed based on the properties of CNN and LSTM [Brownlee, 2019a]. Figure 3.2 presents the model structure.

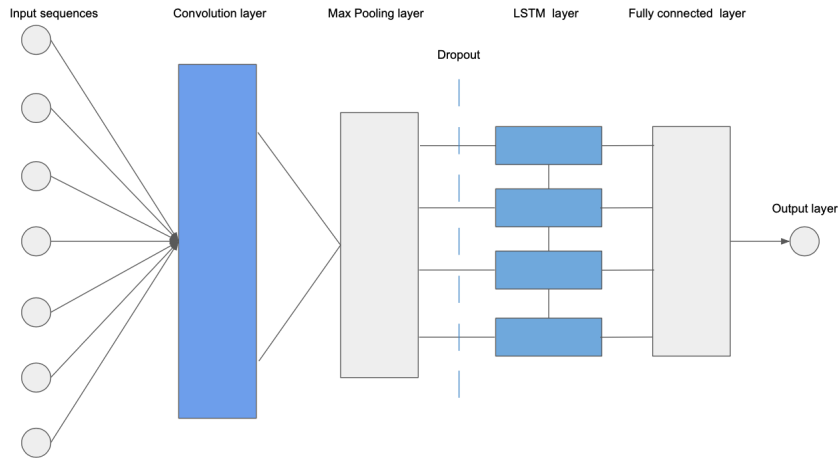


Figure 3.2: CNN-LSTM architecture

3.2 Hyperparameters

Hyperparameters are variables whose values influence the learning process and affect the model parameters that a learning algorithm learns. The trained model parameters are referred to as the model at the end of the learning phase [Nyuytiybiy, 2020]. There are numerous hyperparameters that can be used to improve the model's learning process. The parameters are tweaked in order to achieve the best model performance. The method for determining the optimal parameters is to tweak just the relevant parameters in order to achieve the best prediction performance while keeping the remaining parameters constant. The tuned hyperparameters with values are listed in table 3.1.

Table 3.1: Hyperparameter values

Hyperparameters	Values
Batch size	32, 64, 128
Epochs	100
Activation	ReLU
Units	16, 32, 48, 64, 80, 96, 112, 128
Drop regularization	0.1, 0.2, 0.3, 0.4, 0.5
Learning rate	0.01, 0.001, 0.0005
Kernel size	3, 5, 7

A Bayesian technique will be used to optimize these parameters. In appendix B table B.1, Is given a summary of the descriptions of the hyperparameters that are applied.

3.3 Evaluate model

The data is divided into three sets: training, testing, and validation, with 70 percent training, 15 percent validation, and 15 percent test data. The training set is used to construct the model, the validation set is used to test the model, and the test set is used to generate predictions. As a result, training data is used to fit the model, while test data is utilized to test it. The machine learning method will predict continuous values, i.e., predicted values will be returned. The study's evaluation parameters are the root mean square error (RMSE), formula IA.2 in Appendix A. The RMSE is a measure of how far off these expected values differ from the actual values. The RMSE indicates how well a model fits a dataset. The lower the RMSE, the better. This will be assessed for both the total output length and 10 seconds prediction with the scaled and unscaled RMSE score.

Chapter 4

Results

The tuning and training part of the models, as well as predictions utilizing the tuned model, will be discussed in this section.

4.1 Tuning

It is important to diagnose faults and improve prediction performance by examining the train and validation loss of the model’s learning process. A summary of the train and validation loss for both tuned models can be found in table 4.1. The model with two CNN layers (validation loss: 0.00532) appears to perform slightly better than the model with one CNN layer (validation loss: 0.00553).

Table 4.1: Train and validation loss of tuning model

Model	CNN layers	Train loss	Validation loss
CNN-LSTM	1	0.00558	0.00553
CNN-LSTM	2	0.00538	0.00532

It suggests that using two CNN layers will improve performance. The train loss and validation loss are nearly identical over the epochs, as illustrated in figure 4.1. This indicates that the model is neither overfitting nor underfitting, which is useful information to have while training a model. Figure 4.1 illustrates the validation (blue) and train (red) loss of tuning the model with two CNN layers. In Appendix B are the results shown of the model with one CNN layer.

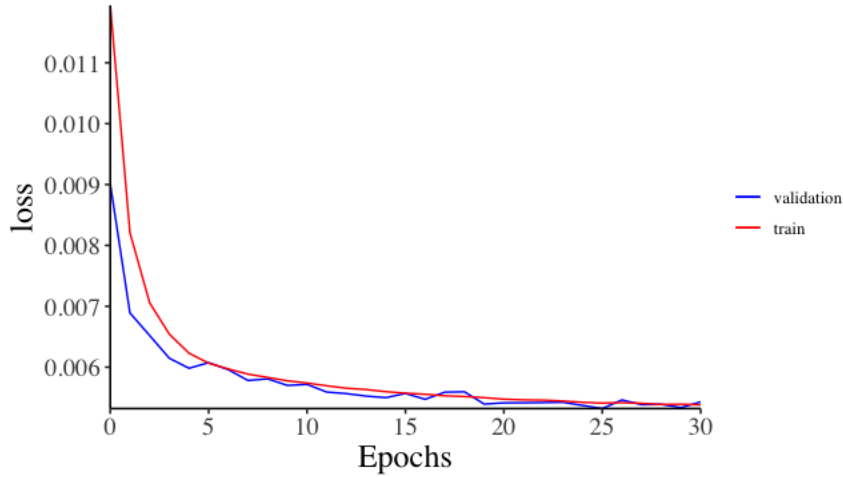


Figure 4.1: Validation and train loss tuning with two CNN layers

The following optimal values for the hyperparameters are shown in table 4.2 for both models. These will be used to train the model.

Table 4.2: Optimal hyperparameter setting

Model	Batch size	Epochs	Activation	Units	Drop	Kernel size	Learning rate
1CNN-1LSTM	64	45	ReLu	48	0	3	0.0005
2CNN-1LSTM	128	30	ReLu	64	0.1	3	0.0005

4.2 Training

The best hyperparameters will now be used to train the models. Table 4.2 lists the values for both models. A summary of the train and validation loss of the training part for both tuned models can be found in table 4.3. The model with two CNN layers (validation loss: 0.00524) appears to perform significantly better than the model with one CNN layer (validation loss: 0.00545).

Table 4.3: Train and validation loss of training model

Model	CNN layers	Train loss	Validation loss
CNN-LSTM	1	0.00549	0.00545
CNN-LSTM	2	0.00528	0.00524

Figure B.2 in Appendix B shows the train and validation losses of training the CNN-LSTM model with one CNN layer. Figure 4.2 shows the CNN-LSTM model's train and validation loss with two CNN layers.

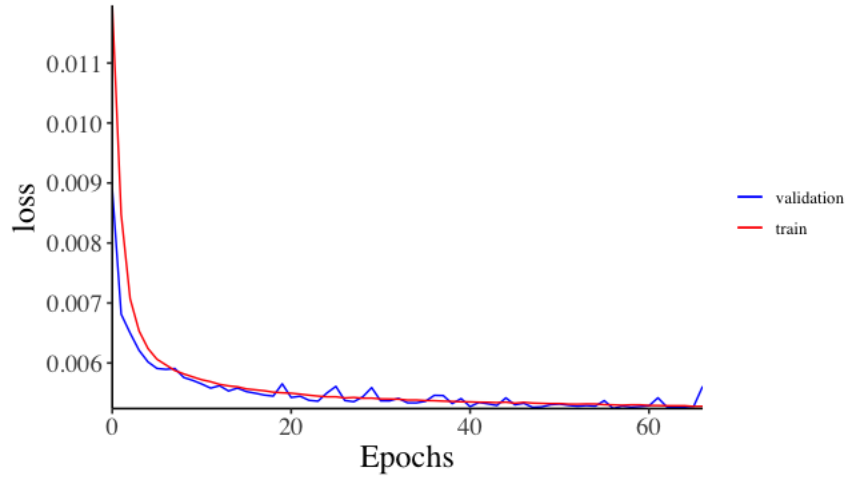


Figure 4.2: Validation and train loss of trained model.

4.3 Prediction

Finally, the trained model will be applied to the test data to create predictions. The RMSE per timestep will be evaluated to see how well the model performs over time. Figure 4.3 presents the results for both models. As shown in the results, the models appears to be getting worse over time. The first five timesteps are good, but there is a significant rise after that. Surprisingly, both models experience this exact same pattern.

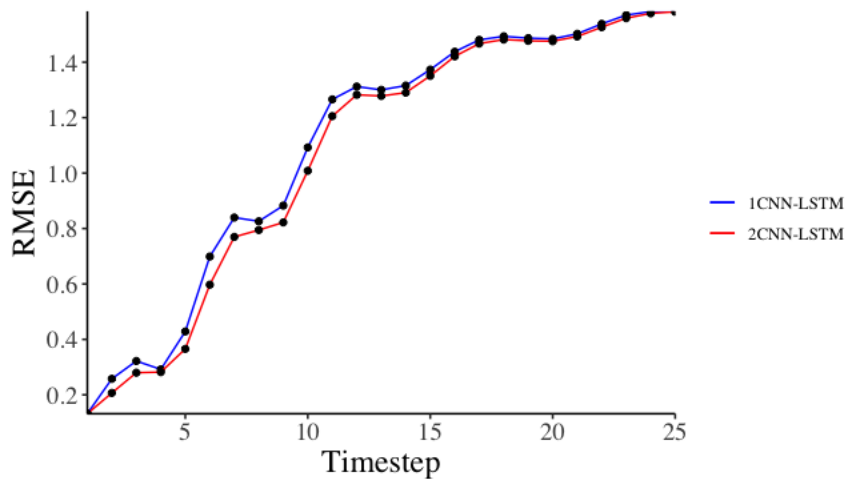


Figure 4.3: RMSE per timestep

An overview of the average RMSE scores for both models is presented in table 4.4.

It was found that the second model performed slightly better than the first model based on the average RMSE score (1.176). With an RMSE score of 0.802, the second model also performs marginally better in the first 10 seconds.

Table 4.4: Average RMSE of models

Model	CNN layers	Input length	Output length	RMSE scaled 10s	RMSE 10s	RMSE scaled	RMSE
CNN-LSTM 1	1	100	25	0.0521	0.847	0.0738	1.199
CNN-LSTM 2	2	100	25	0.0494	0.802	0.0724	1.176

Potentially, the performance of the models would improve when increasing the input length to larger amounts. However, the analysis was currently impossible due to the enormous processing time.

Chapter 5

Conclusion and Discussion

In this research, a prediction model combining CNN and LSTM is suggested and applied to heave motion prediction of a ship. Therefore, the goal is to predict the upcoming heave waves within 10 to 20 seconds using a hybrid model of LSTM and CNN. The method uses heave motions (RefZ) data of 100 simulations. The CNN layer extracts spatially relevant and local time-series features of the data. The LSTM layer reflects the long-term movement process and predicts heave motion for the next moment. The fully connected layer decodes the LSTM output and obtains the final forecasting results. The results show that a CNN-LSTM model with two CNN layers performance slightly better than a CNN-LSTM model with one CNN layer when looking at the RMSE score in the timesteps ahead. CNN-LSTM is suitable for forecasting heave motions in the coming seconds and are able to give a useful reference for the ship's superintendent in ensuring a safe and efficient helicopter landing.

However, the model still has some shortcomings. The computational cost was quite high due to the large amount of data (tuning and training one model took days to offer some perspective) and this research had some limitations. It has prevented this research from trying out all different kind of configurations and training the best performing model with very long input data. Future research should look into using longer input data to check whether this is beneficial for the performance. As a result, it has been decided to resample the data. Although some information was lost, the peaks were not lost. Finally, given the sensitivity of many hyperparameters in the proposed CNN-LSTM and its great complexity, it is feasible that additional optimal setup and, in particular, feature engineering could boost forecasting ability further. This report focuses on applying the models to a univariate problem, future work will focus on modeling all six degrees of freedom motion which is called a multivariate problem. In this research the simulation data is based on only one sea state. Therefore, it

is necessary to investigate whether the positive outcomes can also be obtained on simulation data with various sea conditions, and finally on prototype and real-world data for use in future applications.

Bibliography

- [, 2021a] (2021a). Introduction to time series regression and forecasting.
- [, 2021b] (2021b). Time series regression models.
- [, 2022] (2022). Learning rate.
- [Brownlee, 2019a] Brownlee, J. (2019a). Cnn long short-term memory networks.
- [Brownlee, 2019b] Brownlee, J. (2019b). Difference between a batch and an epoch in a neural network.
- [Brownlee, 2021] Brownlee, J. (2021). How to choose an activation function for deep learning.
- [Duan et al., 2015] Duan, W.-y., Huang, L.-m., Han, Y., Zhang, Y.-h., and Huang, S. (2015). A hybrid ar-emd-svr model for the short-term prediction of nonlinear and non-stationary ship motion. *Journal of Zhejiang University-SCIENCE A*, 16(7):562–576.
- [Kuremoto et al., 2014] Kuremoto, T., Kimura, S., Kobayashi, K., and Obayashi, M. (2014). Time series forecasting using a deep belief network with restricted boltzmann machines. *Neurocomputing*, 137:47–56.
- [Li et al., 2022] Li, M.-W., Xu, D.-Y., Geng, J., and Hong, W.-C. (2022). A ship motion forecasting approach based on empirical mode decomposition method hybrid deep learning network and quantum butterfly optimization algorithm. *Nonlinear Dynamics*, pages 1–21.
- [Liu et al., 2020] Liu, Y., Duan, W., Huang, L., Duan, S., and Ma, X. (2020). The input vector space optimization for lstm deep learning model in real-time prediction of ship motions. *Ocean Engineering*, 213:107681.
- [Maklin, 2019] Maklin, C. (2019). Dropout neural network layer in keras explained.

- [Menon, 2021] Menon, A. (2021). Ship motions - the ultimate guide.
- [Nyuytiybiy, 2020] Nyuytiybiy, K. (2020). Parameters and hyperparameters in machine learning and deep learning.
- [Peng et al., 2019] Peng, X., Zhang, B., and Zhou, H. (2019). An improved particle swarm optimization algorithm applied to long short-term memory neural network for ship motion attitude prediction. *Transactions of the Institute of Measurement and Control*, 41(15):4462–4471.
- [Sun et al., 2022] Sun, Q., Tang, Z., Gao, J., and Zhang, G. (2022). Short-term ship motion attitude prediction based on lstm and gpr. *Applied Ocean Research*, 118:102927.
- [Tanaka, 2018] Tanaka, Y. (2018). Active vibration compensator on moving vessel by hydraulic parallel mechanism. *International Journal of Hydromechatronics*, 1(3):350–359.
- [ujjwalkarn, 2016] ujjwalkarn (2016). A quick introduction to neural networks.
- [Zhang et al., 2019] Zhang, W., Wu, P., Peng, Y., and Liu, D. (2019). Roll motion prediction of unmanned surface vehicle based on coupled cnn and lstm. *Future Internet*, 11(11):243.

Appendix A

Formulas

Some formulas that were employed during the process are shown in Appendix A. The Min-Max scaler is given by formula IA.1:

$$x_{scaled} = \frac{x - x_{min}}{x_{max} - x_{min}} \quad (\text{IA.1})$$

The Root Mean Square Error (RMSE) is described by formula IA.2:

$$RMSE = \sqrt{\frac{\sum_{i=1}^n (\hat{y}_i - y_i)^2}{n}} \quad (\text{IA.2})$$

Appendix B

Figures and tables

Table B.1 lists the hyperparameters that were employed with some definitions. [Brownlee, 2021], [Brownlee, 2019b], [, 2022], [Maklin, 2019], [ujjwalkarn, 2016]

Table B.1: Description of hyperparameters

Hyperparameters	Definition
Batch size	The batch size is a hyperparameter that defines the number of samples to work through before updating the internal model parameters
Epochs	The number of epochs is a hyperparameter that defines the number of times that the learning algorithm will work through the entire training dataset
Activation	An activation function defines how the weighted sum of the input is transformed into an output from a node or nodes in a layer of the network
Units	The neuron, often called unit, receives an input from some other nodes, or from an external source and computes an output. Each input has an associated weight (w), which is assigned on the basis of its relative importance to other inputs.
Drop regularization	Drop regularization is used to prevent a model from overfitting. It randomly sets the outgoing edges of hidden units to 0 at each update of the training phase.
Learning rate	The learning rate determines the step size at each iteration while moving toward a minimum of a loss function.

Figure B.1 shows the results of the validation and train loss during the tuning process of the CNN-LSTM model with one CNN layer.

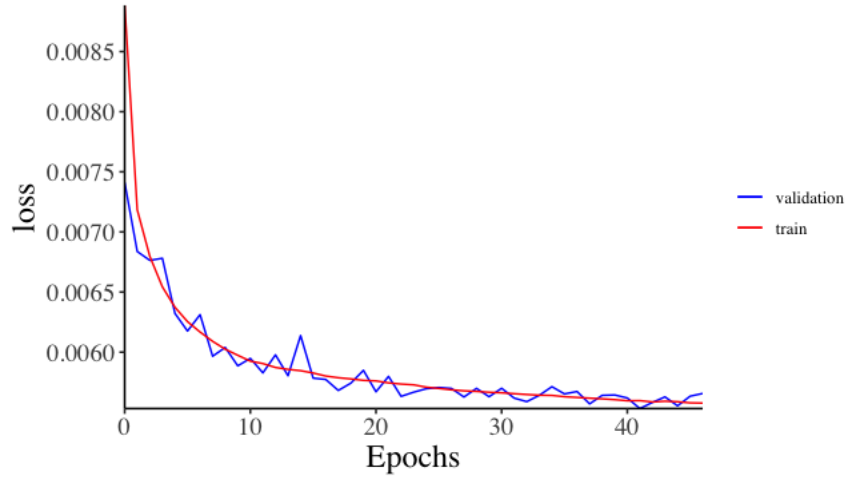


Figure B.1: Validation and train loss tuning

Figure B.2 shows the results of the validation and train loss during the training process of the CNN-LSTM model with one CNN layer.

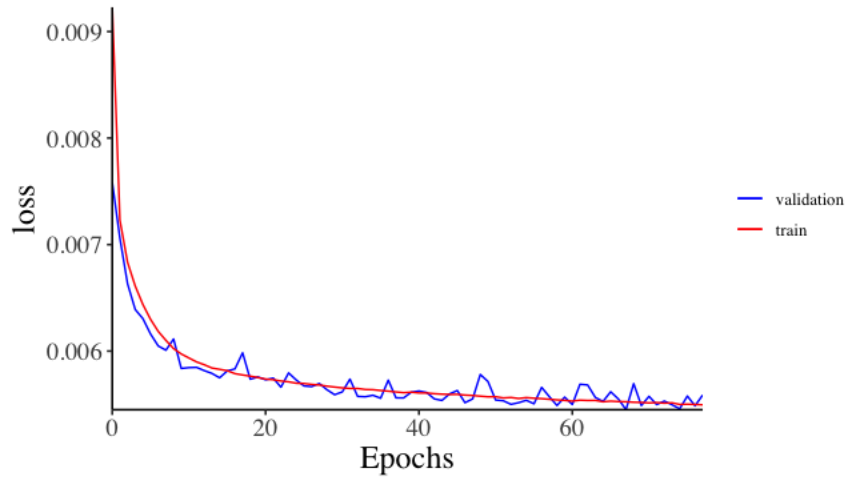


Figure B.2: Validation and train loss of training model

Appendix C

Scripts

All of the collaboration I had with my fellow students on our projects were recorded on GitHub. The link to the repository can be found here: [link to GitHub](#). The GitHub page has been designed to be user-friendly. The "CNN-LSTM tutorial" Jupyter Notebook provides instructions for building and utilizing a CNN-LSTM similar to the one used in the report. The user will receive some additional information about this's practical application. Additionally, some scripts have reusable functions which have been used during the project. If there are some issues, they can easily be suggested in the issues menu of GitHub.