Short-term Ship Motion Prediction Report MARIN

Maria Zafeiraki, Quinten Sandberg and Jolan Keijzer
Faculty of Science
Utrecht University

June 27, 2022

Abstract

This paper describes the use of three machine learning algorithms to predict the time series heave motion of a ship to an horizon of 10 and 20 seconds into the future in order for flying vehicles to land on the deck with safety. This research aims to discover the feasibility of ship motion prediction by comparing a multi-output Support Vector Regression (SVR), a Long Short-Term Memory (LSTM) neural network and a CNN-LSTM neural network to an ARIMA baseline model. The comparison of the models revealed that the LSTM and CNN-LSTM models outperform ARIMA using a rather small training input, yet, SVR is also a promising method when trained on a large enough dataset.

1 Introduction

In off-shore multi-body operations, it is important to accurately determine the movements of the vessels to be able to safely merge or split vessels (e.g. connect two vessels, side-by-side transfer, helicopter landing, etc) [Zhao et al., 2004]. During these operations, the success of the operation is highly contingent on ship movements like positioning, speed and heading, and the six ship motions: surge, sway, heave, roll, pitch and yaw [Cheng et al., 2019]. These ship motions are visualized in Figure 1. In general, matching up speed and heading between separate vessels is quite straightforward, whereas, matching the ship motions is more challenging. This issue is especially evident in multi-body operations that involve helicopters or drones, due to the relative motion between the landing deck and the helicopter [Yang et al., 2008].

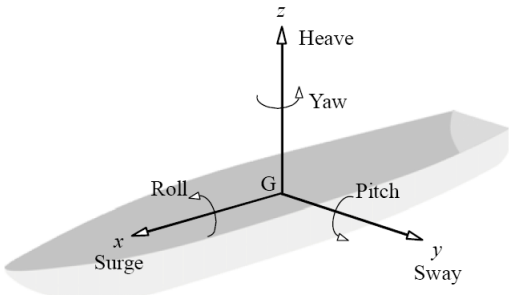


Figure 1: The six ship motions visualized as degrees of freedom [Tanaka, 2018]

As of yet, landing operations with human pilots are guided by the human judgement of landing deck superintendents whom indicate safe landing windows based upon intuition about the ship's motions. Therefore, accurate and reliable predictions of these ship movements can help to replace or augment the human judgement in these operations [Khan et al., 2005]. In this paper only the heave motion is addressed, because this has a large effect on helicopter/off-shore vessel operations. To be able to predict the heave motion, several machine learning models are examined and compared to an ARIMA baseline model. The different models and the (dis)advantages of these models are summarized in Table 1.

	Advantages	Disadvantages
ARIMA	• Suitable for non-stationary time series • Only needs endogenous variables (past values determine the future values)	• Dealing with outliers • Does not react to nonlinear relationship
SVR	• Less over fitting and robust to noise • No local minimal • Good in generalisation	• Computationally expensive • Lack of transparency of results (low interpretability) • Selection of kernel function
LSTM	• Handles non-linearity well • Strong ability with sequences • Learns long- and short-term	• High computational cost • Propensity of overfitting
CNN	• Understanding of connections between various observations • Dilations between observations	• Overfitting • Exploding gradient

Table 1: Advantages and Disadvantages of selected methods

2 Data and Methods

2.1 Data

FREDYN Simulations: The data are simulations generated by the MARIN research department by means of time domain simulations using FREDYN, a computer software that mimics the dynamic behavior of a steered ship under the influence of waves and wind. Here it was used to simulate the 5415M vessel in wave conditions and all six degrees of freedom are calculated in the time domain. Rigid-body dynamics with large angles and fluid flow effects produce non-linearity. The hydrodynamic forces operating on the hull are calculated using a nonlinear strip theory technique.

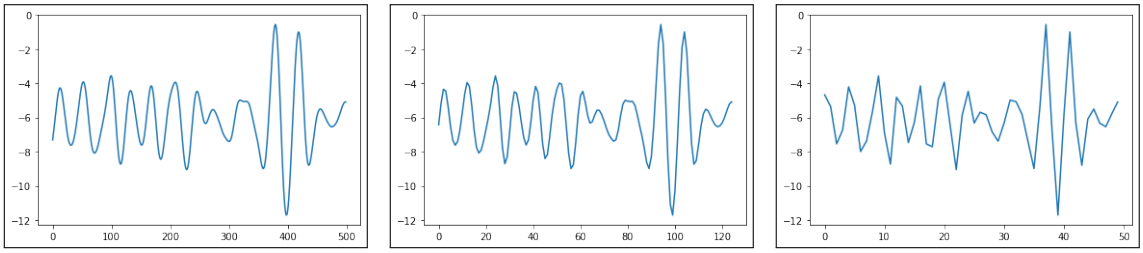
Ship Motion Dataset: For this analysis 100 simulations of the six motions of the ship were obtained for a single sea state (5) and for time traces each 3 hours long, that is 55.000 initial observations for timesteps of 0.2 seconds. Each time series dataset features the six motions of the ship: roll, pitch, yaw in radians and also the heave, sway and surge motion in meters, measured from a reference point at the stern and also from the center of gravity of the ship. The heave motion (RefZ) variable lies on a bell curve distribution with a mean value of -6.13 and a standard deviation of 1.65, thus standardising this feature is meaningful. Additionally, the data contains features on the wave height, heading of the ship and sea state. Importantly, this information should not be used in making predictions of the ship motions because this paper attempts to predict ship motions solely based upon patterns in the ship motions. To make the prediction problem more feasible, this paper solely focuses on predicting the heave motion of the reference point located at the stern of the vessel, which is denoted as RefZ.

2.2 Data preparation - All models

Data Resampling: To reduce computation time and noise in the data, the dataset is resampled from a granularity of 0.2 seconds per timestep to 0.4, 0.8 and 2.0 seconds. To do so, 2, 4 and 10 timesteps are combined respectively. This is achieved by looping over the data in non-overlapping windows and locally fitting trend lines. Afterwards, the minimum value is selected in descending trends and the maximum value is selected in ascending trends. The result of this is visualized in Figure 2, which shows that too much resampling results in significant information loss. This loss of the peak and valley details and change of the temporal relationships of the data occurs from 1 second granularity onwards and thus the data is resampled at 0.8 second granularity.

2.3 Data preparation - ARIMA and SVR

Z-score normalization: As Support Vector Regression models are not scale invariant, the data should take small values and all the features should have values at approximately the same range. To do this, Z-score normalization is trained on the train set from the first simulation and the remaining



(a) Original data (b) Resampled data 0.8 seconds (c) Resampled data 2.0 seconds

Figure 2: Data Resampling

data. In doing so, all values are centered around a mean of 0 with standard deviation 1 according to Equation 1.

$$Z = \frac{x - \mu}{\sigma} \tag{1}$$

where Z represents the normalized value, x represents the observed values, μ the mean value of x and σ the standard deviation of x.

Train-test split: Finally, the first simulation data set is split into train and test sets with an 80/20 split. For the above data set no missing data and no outliers are found.

2.4 Data preparation - Neural Networks

Sampling examples: Noting that the data is sourced from 100 separate simulations, the sequences can not simply be merged. This is due to the absence of temporal relationships between the end of one simulation and the start of the next. To deal with this, examples are sourced from every separate simulation to create distinct train, validation and test sets. This split is done by taking simulations 1 to 70 as training set, 71 to 85 as validation set and 86 to 100 as test sets. Due to the large amount of data and overlap between adjacent training examples, sub-sampling is applied using strides. In short, this stride determines how many time-steps are skipped before a new example is created.

MinMax normalization: The raw RefZ data is normalized to allow for better model convergence and performance [Jayalakshmi and Santhakumaran, 2011]. In this paper, MinMax normalization in the interval of [0, 1] is applied. This method normalizes the raw RefZ values to values in this interval.

$$Y_{minmax} = \frac{Y - minY}{maxY - minY} \tag{2}$$

2.5 Model tuning

ARIMA: In order to estimate the model the values p for the order of the autoregressive model, d for the degree of differencing for non-stationary time series and q for the order of the moving-average model, are identified using the *auto_arima* function of *pmdarima* package according to the Akaike Information Criterion (AIC).

$$\text{AIC} = -2\log\left(\hat{L}\right) + 2k$$

where $\left(\hat{L}\right)$ is the maximum value of the likelihood function for the model and k is number of estimated parameters in the model. The first term serves as a measure of the lack of fit of data for statistical models, with smaller values being preferable. The second term serves as a penalty term for models with large dimensions. [Narisetty, 2020]

SVR: For tuning the hyperparameters of the SVR model, a grid search cross validation algorithm is utilized which attempts to determine the best parameters gamma, epsilon and kernel function via an exhaustive generation of candidates from a grid of parameter values where the optimal combination is retained. Key elements of this tuning algorithm are the estimator (SVR), the parameter space, a

method for searching or sampling candidates, a cross-validation scheme and a score function that is by default the score function of the estimator.

LSTM & CNN-LSTM: This paper utilizes a tuning algorithm to determine the optimal hyper parameters in a more effective and efficient manner. In this paper, the Keras-tuner is applied. The reason for this decision is three-fold. First of all, the Keras-tuner is built specifically for tuning of hyper parameters in the widely used library Keras, which helps to improve the replicability of this study. Second, the wide usage of Keras has resulted in clear documentation and community support for the Keras-tuner. Third, the Keras-tuner library is especially flexible in tuning options and allows for straightforward sub-classing to expand tuning options even further. During this process the LSTM model is tuned in terms of number of layers, units per layer, learning rate, batch size, epoch amount, optimizer and dropout. Whereas, the CNN-LSTM model is tuned on number of CNN layers, units per LSTM layer, learning rate, batch size, epoch amount, dropout and kernel size.

2.6 Model evaluation

Root Mean Squared Error (RMSE) To evaluate the performance of the proposed model, the RMSE metric is used. Moreover, this paper calculates the RMSE per timestep to determine the RMSE for every position that is predicted in the future. This is done to gain a better understanding of the strengths and weaknesses of the model. The implementation of $RMSE_{timestep}$ is shown in Equation 4. In this metric, instead of calculating the RMSE over all the prediction errors, the RMSE is calculated separately on the errors at timestep t.

$$RMSE = \sqrt{\frac{1}{n}\Sigma_{i=1}^{n}\Big(d_i - f_i\Big)^2} \tag{3}$$

$$RMSE_{timestep} = \sqrt{\frac{1}{n^t}\Sigma_{i=1}^{n^t}\Big(d_i^t - f_i^t\Big)^2} \tag{4}$$

2.7 Workflow

ANN: To summarize, the workflow of this research is visualized in Figure 3. Even though this figure visualizes this research as a linear process, it is important to note that the actual process is iterative. Especially determining the used stride in sub-sampling, the input length, and the training and tuning are iterative processes.

SVR: As described at Figure 4, after the initial data preparation on 0.8 seconds resampled dataset, train and test data is converted to 2D tensors as inputs for the SVR and perform grid search cross validation for tuning the hyperparameters gamma, epsilon and the kernel via an exhaustive generation of candidates from a grid of parameter values where the best combination is retained. Then, the SVR model is estimated using the above best parameters, trained on train data and then make predictions on test data. Afterwards, the predictions and the original values are unscaled and the Root Mean Square Error is calculated as a performance metric. Finally, the model is retrained on the full data set and predictions are made on the rest 99 simulation data sets. The consistency of the model performance is validated by looking at the box plot of the prediction errors across simulations.

3 Results

The results, conclusion and discussion section of this paper are summaries and combinations of the chapters of the three separate papers of the authors. In these chapters, merely the best performing models are described and compared to each other. For more details about the separate models and their intricacies check out these separate papers. Links to these papers are found in the Github repository (here).

To be able to make accurate predictions, a finely tuned model is required. In this paper, the **LSTM** model has 3 layers with 368, 368 and 240 neuron units respectively. Furthermore, the Adam optimizer is used and no dropout is used. Finally, the model uses a learning rate of 0.0005 and batch size of 64. With regards to the **CNN-LSTM**, has two CNN layers and one LSTM layer with 64 units. The ReLu

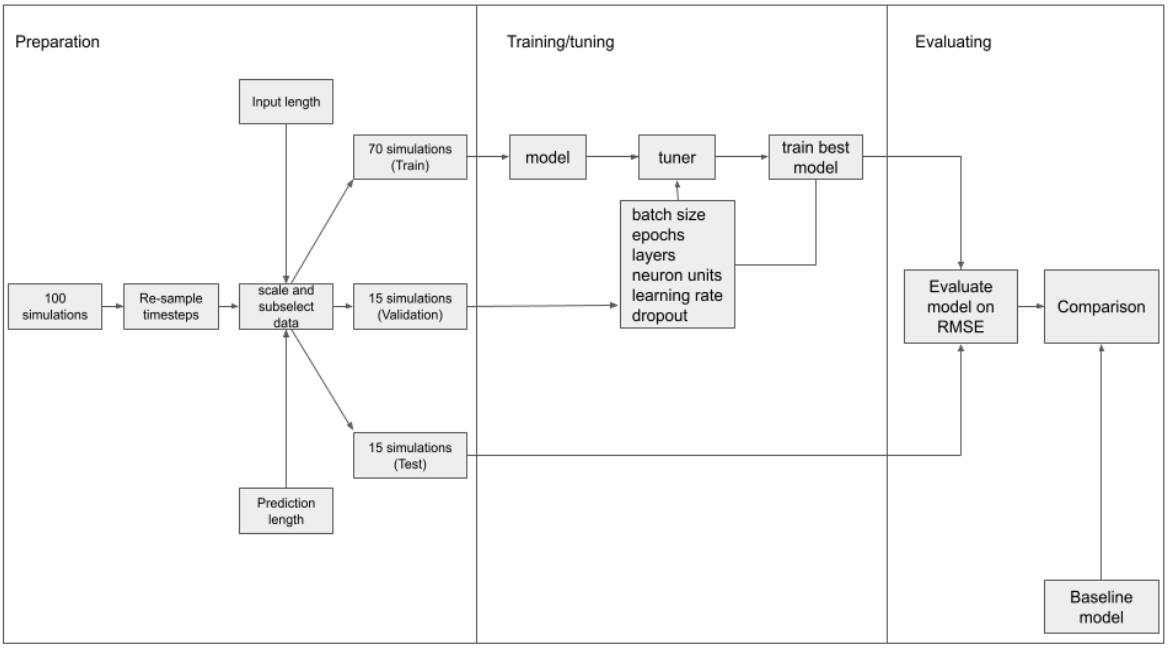


Figure 3: ANN Workflow

Model (input and stride)	RMSE unscaled (10.4 sec / 13 steps)	RMSE unscaled (20 sec / 25 steps)
LSTM (200, 1)	0.5976	1.0474
2CNN-1LSTM (100, 5)	0.8017	1.1755
SVR (700, 1)	0.8200	-
SVR (1400, 1)	-	0.8400
ARIMA (700, 1)	1.0340	-
ARIMA (1400, 1)	-	1.3208

Table 2: RMSE per model

optimizer is used with a dropout of 0.1. The model performs with a learning rate of 0.0005, batch size of 128 and kernel size of 3.

The RMSE performance of the models is summarized in Table 2. In this table the input length represents the number of observations that are used in a training examples. Whereas, the stride represents the number of examples skipped before sampling a new example.

Evidently, the SVR model achieves the best performance with regards to both the 10 and 20 second ahead predictions. Additionally, Table 2 shows that all models are able to achieve better prediction performance than the ARIMA baseline model. With regards to the Neural Networks, the best performance is achieved by the LSTM and thus the additional CNN layers do not seem to allow for better predictions. However, it is important to note that the CNN-LSTM is trained on different input lengths. Therefore, increased input lengths might result in the CNN-LSTM performing better. This observation also holds for the ARIMA and SVR model, because both these models require very large amounts of data to achieve the provided prediction performance. The models use 700 and 1400 input observations to predict 10 and 20 seconds respectively.

For a more detailed analysis, the RMSE per timestep of the different models are visualized in Figure 5. This figure confirms the findings of Table 2 in terms of the ranking in model performance. Interestingly, the differences in model performance are most evident until around timestep 17 and later where the model performance becomes increasingly similar. However, it becomes apparent that the

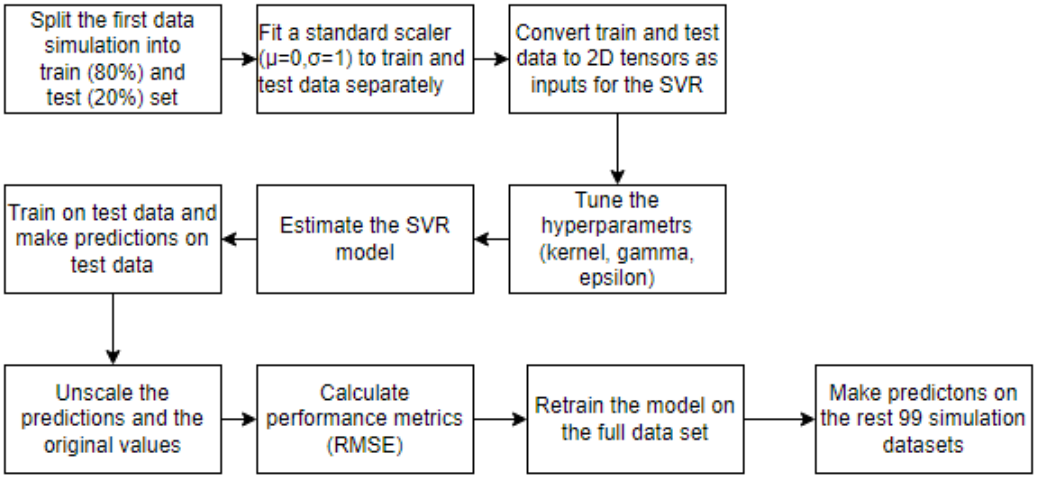


Figure 4: SVR workflow

model performance of the SVR model is considerably better than the other models. Again, a potential explanator for this substantial difference is the large difference in input data that is used, because the ARIMA and SVR model both use 1400 timesteps as input. Potentially, the performance of the LSTM and CNN-LSTM would improve upon increasing the input to these large amounts. However, the high computational load that this entails prevents the researchers from analyzing this.

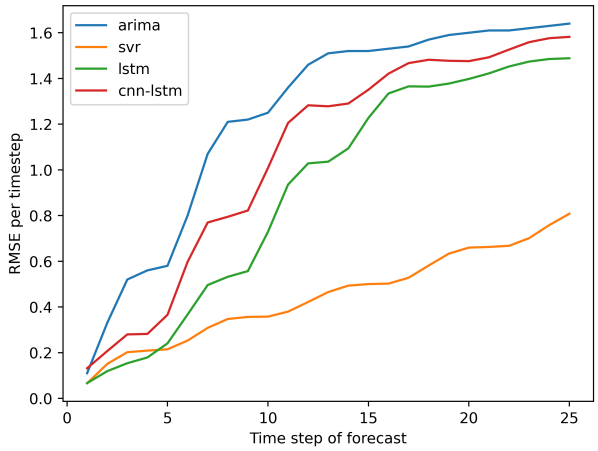


Figure 5: RMSE unscaled per timestep for trained models

4 Conclusion & Discussion

This research has set out to discover the feasibility of ship motion prediction without the usage of wave excitation information by comparing SVR, LSTM and CNN-LSTM to an ARIMA baseline model.

This paper shows that all models that are analyzed are able to outperform the ARIMA model. This is in line with *a priori* expectations on the model performance. Oppositely, the greatly superior prediction performance of the SVR model is quite surprising. Especially noting that the SVR makes use of a linear kernel, because it was expected that non-linear models would perform better. This is surprising because of the strong non-linear nature of ship motion prediction [Sun et al., 2022] and time-series prediction in general [Dumitru and Maria, 2013]. However, it should be noted that both the ARIMA and SVR model use a lot of input and therefore it becomes more difficult compare these models one-on-one with the neural network models. The high training costs involved with training both LSTM and CNN-lSTM models make this comparison even more difficult, because this prevents

the researchers from trying all models. Therefore, even though this paper provides initial learnings about heave motion prediction, further inquiry is beneficial. Moreover, for all models except the SVR the prediction performance decreases upon predicting further into the future. Thus, future research into ship motion predictions should be wary about attempting to predict too far into the future.

The high computational costs have prevented this research from trying out all different model configurations and training the best performing NN models with very lengthy inputs. Therefore, future research should look into using longer inputs for the NN models to check whether this is beneficial for the heave motion prediction performance. Even though the SVR performance is very promising, it should be considered that the input length that is currently used is quite long. Therefore, further inquiry is required to determine whether this is feasible in real-world applications. However, if this performance holds up in real-world application the provided results show a lot of opportunity. This paper solely focused on predicting the heave motion in a uni-variate prediction problem. However, in real-world applications the motions, accurate predictions of the other ship motions are crucial for augmenting human judgement. Therefore, future research into multi-variate ship motion prediction is warranted. In this research, simulation data is used and this simulation only takes one sea state into account. Therefore, for future applications it is required to research whether the promising results can also be achieved on simulation data with multiple sea states and eventually on prototyping and real-world data. Luckily, MARIN is very well equipped to handle these subsequent steps as well.

5 Acknowledgments

This work was supported by Maritime Reasearch Institute Netherlands (MARIN) and the Utrecht University (UU) Netherlands. The authors would like to thank Dr Bulent Duz and Mr Eelco Frickel for the valuable discussion of the problem and the data.

A Appendix

Links to the standalone studies combined at this report are found in the Github repository of this report : https://github.com/QuintenSand/Thesis-Marin

References

[Cheng et al., 2019] Cheng, X., Li, G., Skulstad, R., Major, P., Chen, S., Hildre, H. P., and Zhang, H. (2019). Data-driven uncertainty and sensitivity analysis for ship motion modeling in offshore operations. *Ocean Engineering*, 179:261–272.

[Dumitru and Maria, 2013] Dumitru, C. and Maria, V. (2013). Advantages and disadvantages of using neural networks for predictions. *Ovidius University Annals, Series Economic Sciences*, 13(1).

[Jayalakshmi and Santhakumaran, 2011] Jayalakshmi, T. and Santhakumaran, A. (2011). Statistical normalization and back propagation for classification. *International Journal of Computer Theory and Engineering*, 3(1):1793–8201.

[Khan et al., 2005] Khan, A., Bil, C., and Marion, K. E. (2005). Ship motion prediction for launch and recovery of air vehicles. In *Proceedings of OCEANS 2005 MTS/IEEE*, pages 2795–2801. IEEE.

[Narisetty, 2020] Narisetty, N. N. (2020). Chapter 4 - bayesian model selection for high-dimensional data. In Srinivasa Rao, A. S. and Rao, C., editors, *Principles and Methods for Data Science*, volume 43 of *Handbook of Statistics*, pages 207–248. Elsevier.

[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.

[Yang et al., 2008] Yang, X., Pota, H., Garratt, M., and Ugrinovskii, V. (2008). Ship motion prediction for maritime flight operations. *IFAC Proceedings Volumes*, 41(2):12407–12412.

[Zhao et al., 2004] Zhao, X., Xu, R., and Kwan, C. (2004). Ship-motion prediction: algorithms and simulation results. In *2004 IEEE international conference on acoustics, speech, and signal processing*, volume 5, pages V–125. IEEE.