Utrecht University

Faculty of Science

A Comparison of ARIMA model and Support Vector Regression in Short-Term Ship Motion Prediction

Maria Zafeiraki
1981226

July 1, 2022

A Comparison of ARIMA model and Support Vector Regression in Short-Term Ship Motion Prediction

Maria Zafeiraki
Faculty of Science
Utrecht University

July 1, 2022

Abstract

This paper describes the use of time series analysis methods to predict the heave motion of a ship to an horizon of 10 and 20 seconds into the future in order for flying objects to land on the deck with safety. The methods used are a statistical approach, the Auto-Regressive Moving Average (ARIMA) model, where past values contain information to predict the future values by incorporating the statistical properties of the data and a supervised learning algorithm, a multi-step Support Vector Regression (SVR) that aims to fit a plane to feature space minimizing the prediction errors. The dataset used is resampled at higher timesteps and a sensitivity analysis is conducted about how data resampling affects the performance of ARIMA and SVR model. The ultimate goal of this paper is to compare the performance of an SVR to the ARIMA model as a standalone study which will also serve as baseline model for a Convolutional Neural Network (CNN) and an Long Short-Term Memory (LSTM) neural network modelled by our group members. It is concluded that data resampling affects ARIMA's performance for predictions shorter than 10 seconds but not the performance of the SVR for any forecasting horizon. The comparison of the models revealed that SVR outperforms ARIMA when trained on a large enough dataset.

1 Introduction

The safety and effectiveness of maritime activities are directly affected by the random motions of a ship caused by sea waves. Short-term ship attitude forecast that is accurate for the future seconds plays a critical role in operational decision-making [Liu et al., 2020]. Specifically, the deployment and landing of air vehicles from a ship can be challenging and even dangerous due to the unpredictable nature of the ship's movements in an open ocean environment. Therefore, improving the accuracy of ship attitude prediction aids decision-making for such motion-sensitive maritime operations, and getting trustworthy prediction uncertainty information helps to minimize possible navigation issues. It can be argued that forecasting models are vulnerable to the error accumulation problem since they employ forecasted values from the past, yet a method to accurately estimate motion would enhance safety on board ships and allow for more precise vehicle deployment from ships [Khan et al.,]. For the deployment of such operation it is crucial to accurately predict the ship motion for the next few seconds. The motion of the ship in an open sea state is influenced by navigational status such as the load and the speed but also by environmental external factors such as wave, wind and current that are too complex to predict. This causes the ship to move in six degrees of motion: heave, sway, surge, roll, pitch, and yaw (Figure 1) with the most important for landing procedures to be the heave motion. Time series problems like this are traditionally approached by statistical methods including Auto-regressive model (AR), Moving Average model (MA), Auto-regressive Moving Average as well as their variations with the ARIMA model to have shown superior precision and accuracy in forecasting future time series lags. Other combined methods with ARIMA such ARIMA-ANN [Wang et al., 2013] or Ensemble Empirical Mode Decomposition - ARIMA, a self-adaptive decomposition technique suggest improved accuracy [chuan Wang et al., 2015]. Additionally, while it is studied that an LSTM neural network may outperform ARIMA [Siami-Namini et al., 2018], for small amount of data ARIMA may give comparable result or perform better [Yamak et al., 2019].

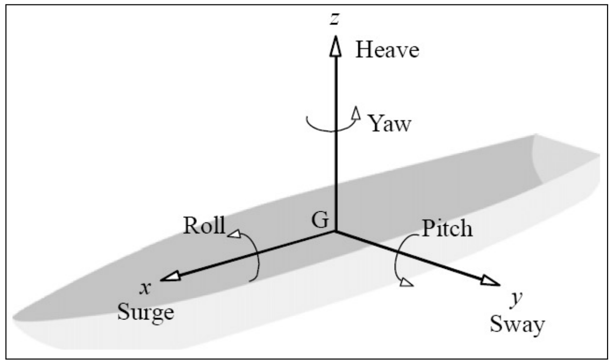


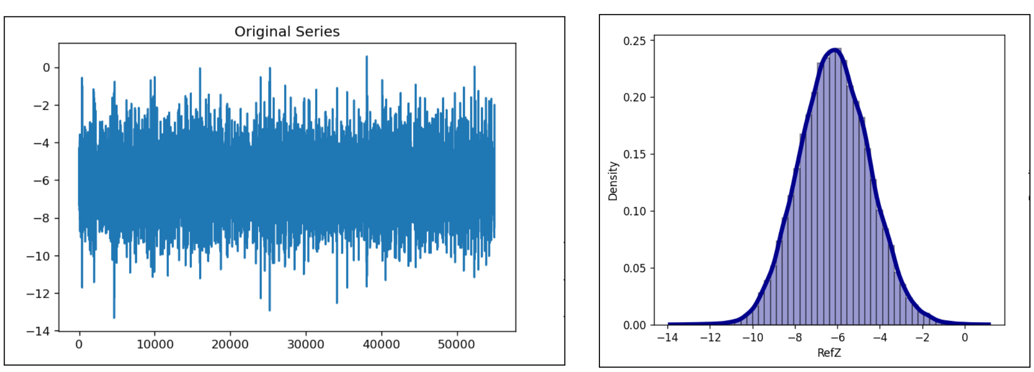
Figure 1: Ship motions

To deal with the substantial non-linearity of ship motion, different artificial intelligence approaches have been developed to forecast ship motion attitude [Zhang, 2003]. Past research suggest that machine learning models predict accurately the short-term ship motion attitude for launch and recovery of air vehicles by discovering patterns within the data, including a neural-network-based modelling such a nonlinear autoregressive exogenous network (NARX) [Li et al., 2017] analysing the impact of three different learning strategies, i.e. offline, online and hybrid learning, where the results show the feasibility of generating a data-driven model through modelling and analysis of the neural network for ship motion prediction, an LSTM-GPR network [Sun et al., 2022] for time series under both motion and static conditions and a feed-forward neural network (FNN) [Khan et al.,] utilizing a combination of the singular value decomposition and genetic algorithm.

This paper compares ARIMA and SVR model with respect to their performance in reducing error rates. ARIMA is used as a representation of classical forecast modeling for the ability to deal with non-stationary data, whereas SVR by not being limited to the statistical properties of the data, flexibly describes it by deploying a linear or non-linear kernel function. For the reason that the wave sensors are expensive systems and are installed only on larger ships, in this case study we attempt to predict the heave motion of a ship using univariate time series data without including any information about wave behaviour or weather conditions, so that a flying vehicle such a helicopter can land on the ship with safety. For computational reasons and for noise reduction, the data is resampled at higher timesteps, and a sensitivity analysis is performed to determine how data resampling impacts the ARIMA and SVR model's performance.

2 Data

2.1 Selected data exploration results



(a) RefZ time series simulation

(b) Distribution of RefZ

Figure 2: Heave motion of ship

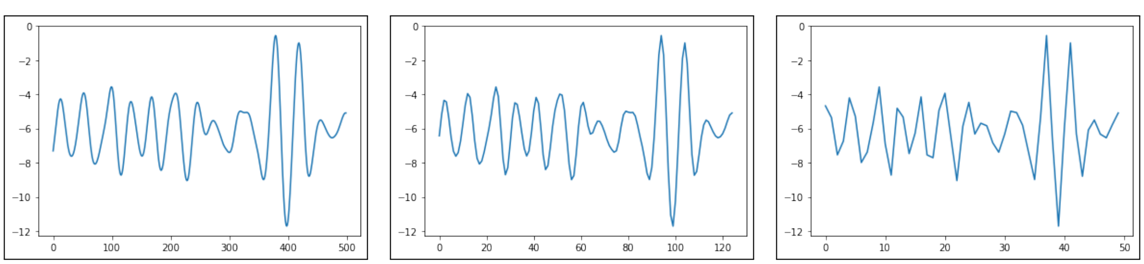
Frequency	Parameter	Description	Unit
5 Hz (0.2 s)	time	time of observation	seconds
	zeta	wave elevation from Cog (center of gravity)	meters
	RefX	X position of Reference point (surge)	meters
	RefY	Y position of Reference point (sway)	meters
	RefZ	Z position of Reference point (heave)	meters
	CogX	X position of center of gravity	meters
	CogY	Y position of center of gravity	meters
	CogZ	Z position of center of gravity	meters
	Roll	The tilting rotation of a vessel about its transverse/Y axis	radians
	Pitch	The up/down rotation of a vessel about its longitudinal/X	radians
	Yaw	The turning rotation of a vessel about its vertical/Z axis	radians

Table 1: Table of Features

FREDYN Simulations : The data are simulations generated by the MARIN Research department by means of time domain simulations and 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. The movements can be large up to the point of capsize, and all six degrees of freedom are calculated in the time domain. Rigid-body dynamics with large angles and fluid flow effects produce here non-linearities. The hydrodynamic forces operating on the hull are calculated using a nonlinear strip theory technique. When model test data is unavailable, the software is meant to be utilized in the early stages of design. It is possible to cope with both a fully intact and a partially damaged spacecraft. The input among others consists of: the geometry of the ship, resistance curve and propeller characteristics, wind data, wave spectrum type, significant wave height and average period, initial position and speed, type of manoeuvre and characteristics of the automatic pilot or human helmsman. The output at each time step consists of: the ship's position, the velocities and rotation rates for six degrees of freedom, the rudder angles, the force and moment contributions of the different components, statistics of all quantities at the end of each run [Source: MARIN]. The above simulations are exclusively used for academic purposes and for providing knowledge about ship motion prediction. The original dataset is not to be published in any form.

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 (Table 1). 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 (Figure 2). Furthermore, vessel speed, wave spectrum and direction are constant and no information will be included about these variables. The author chose the heave (RefZ) variable as the only feature of time series to be fed into the ARIMA and SVR models.

2.2 Data preparation



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

Figure 3: Data Resampling

Data Resampling: For reducing computation time, as well as reducing noise in the data before training, the dataset is resampled from a granularity of 0.2 seconds per timestep to 0.4, 0.8 and 2.0 seconds that is 2, 4 and 10 timesteps resampling respectively. To do so, we look into sequential non overlapping windows within the dataset, then locally fit the best fit lines to capture the trend and assign for this interval the minimum value for a descending or the maximum value for an ascending trend line. In other words, instead of calculating the mean value of every window of observations and obtain a highly smoothed version of the data, we conditionally keep the minimum or maximum value, so the original shape is preserved with only little loss. In Figure 3 where some peaks are smoothed, is showed that a 2.0 seconds resampling results in significant information loss, so we may examine lower resampling timesteps and only compare with such high resampling. The above resampling helped reducing the autocorrelation that data exhibited in Figure 6, which is a key handling before applying ARIMA model.

Z-score normalization: As SVR model is not scale invariant, the data should take small values and all the features should have values at approximately the same range. The process of normalizing every value in a dataset such that the mean of all of the values is 0 and the standard deviation is 1 is called Z-score normalization. To do this, we import StandardScaler from sklearn library, train it on the train set from the first simulation and then apply it on the test set and on the next ones. In this way, no information is leaked from train to test set and/or between the simulations. In other case, that would introduced bias to our models. The above scaling was also applied for ARIMA method, so that results would be comparable. The equation for the Z-score normalization is as follows:

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

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, each time series data set was split into two subsets: the training and test datasets, that is 80% of each dataset was used for training and the remaining 20% of each dataset was used for testing the accuracy of models. Table 5 lists the number of time series observations for each dataset. For the above data set no missing data and no outliers were reported.

2.3 Assessment metrics

The Root Mean Square error (RMSE) is a commonly used metric for evaluating a model's prediction accuracy. It calculates the disparities between actual and predicted values, often known as residuals. The measure compares the prediction errors of different models for a specific dataset rather than across datasets.The formula for computing RMSE is as follows:

$$RMSE = \sqrt{\Sigma_{i=1}^{n}\Big(\frac{(y_i predicted - y_i actual)^2}{N}\Big)}$$

where N is the total number of observations, $y_i actual$ is the actual value; and $y_i predicted$ is the predicted value. The main benefit of using RMSE is that it penalizes large errors more than smaller ones. Another useful property is that it also scales the scores in the same units as the forecast values (i.e., meters per timestep for this study).

Akaike information criterion (AIC) for a model is defined as

$$\mathrm{AIC} = -2\log\big(\hat{L}\big) + 2k$$

where $\big(\hat{L}\big)$ is the likelihood of the model. The first term $-2\log\big(\hat{L}\big)$ in the AIC is two times the negative log likelihood, which corresponds to the residual sum of squares for the linear regression model with a Gaussian likelihood. As a result, 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. Models with smaller AIC values indicate a better balance between these two key characteristics [Narisetty, 2020]. Here, the AIC is used for determining the best set of parameters for the ARIMA model.

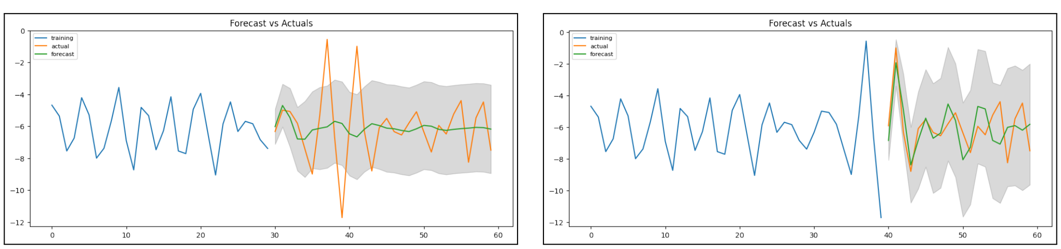
3 Methods

The ARIMA and SVR algorithms that were developed for forecasting the time series are based on an approach of a training window and a forecast horizon. The models estimate a set of training data lying into the training window, then multi-step forecasts are computed for the forecast horizon and then the training window is moved and the model is re-fitted before the next forecasting. By comparing the value of the Root Mean Square Error, the model is chosen using out-of-sample criteria. Python was used for implementing the algorithms, an open source machine learning library statsmodels for tuning and fit of the ARIMA model, sklearn for preprocessing, numpy, pandas and matplotlib for handling and visualizing the data. The multi-step SVR is performed according to Yukun Bao et al. proposed workflow [Bao et al., 2014]

	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 relationships
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

Table 2: Advantages and Disadvantages of selected methods

ARIMA is a suitable method because it performs on non-stationary time series, where the statistical properties of the data are changing through time and the autocorrelation pattern is not constant and also only needs endogenous variables, that is past values determine the future values. On the other hand, the model is not efficient dealing with outliers and does not react to nonlinear relationships. (Table 2). Another drawback of ARIMA model is that long-term forecasting value inclines to the mean (Figure 5) which makes it inappropriate for long-term predictions. Additionally, forecasting accuracy strongly depends on which events are included in the training input (Figure 4) which makes the output of the model very sensitive to the input vector. Finally, the model does not let us to understand what lies behind this data development. Forecasting approaches based on univariate techniques have strong historical coherence but low conceptual coherence, that means these models are often good at reproducing historical trends, but they give limited insight into their reasons.



(a) (b)

Figure 4: ARIMA short-term forecasting

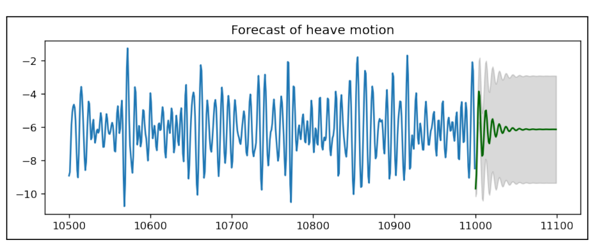


Figure 5: ARIMA long-term forecasting

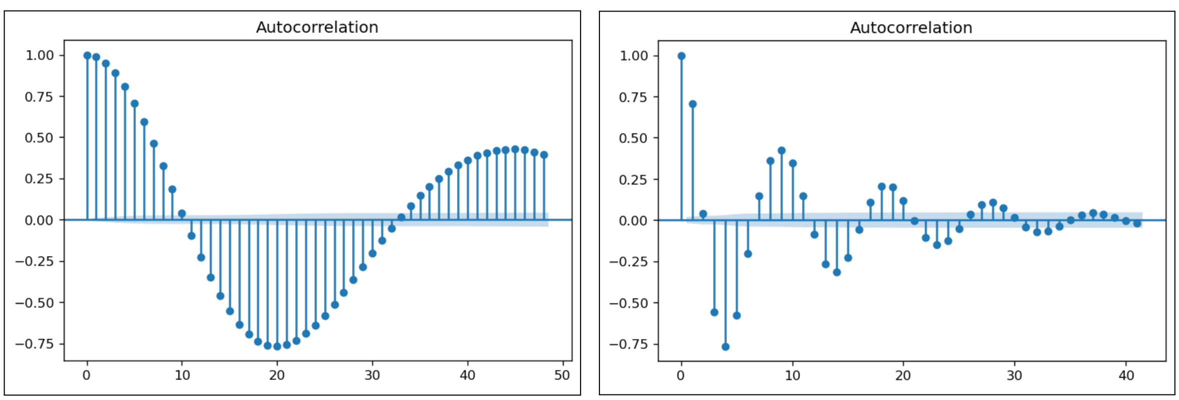
SVR addresses these issues at the cost of transparency of results, computational time and optimal selection of kernel function, epsilon-tube and gamma hyperparameters [Li et al., 2017]. One of the key advantages of SVR is that its computational complexity is independent of the input space's dimensions, as well as it also has a high prediction accuracy and great generalization capabilities. [Awad and Khanna, 2015].

3.1 Auto Regressive Integrated Moving Average (ARIMA)

The Auto Regressive Integrated Moving Average (ARIMA) is a powerful linear regression-based approach to forecasting, with application to financial time series, health management, signal processing and more. When the assumption of uncorrelated errors is violated, classical regression is not advised. [Din, 2015]. In this model, any observed value of time t is defined as the linear combination of the previous time t-1 observed values and error terms. [Newbold, 1983] The model attempts to determine future values by combining three key elements: 1) the autoregressive (AR) part where a linear regression is incorporated to relate past values of series to future values, 2) the integrated (I) part where time series is differenced to become stationary, and 3) the moving average (MA) part that relates past forecast residual errors to future values of time series. So an ARIMA model defined as ARIMA(p, d, q) model can be written in the form:

$$y_t = c + \sum_{i=1}^{p} \varphi_i y_{t-i} + \varepsilon_t + \sum_{i=0}^{q} \vartheta_i \varepsilon_{t-i}$$

where y_t is the stationary variable after differencing, c is constant, φ_i are the autocorrelation coefficients, ε_t are the white noise distributed residuals and ϑ_i are the weights applied on the stochastic terms.



(a) Original data

(b) Resampled data

Figure 6: Data Stationarity

In order to estimate the model the values p, d and q are identified where p is the order of the autoregressive model, d is the degree of differencing for non-stationary time series and q is the order of the moving-average model. By looking at the autocorrelation function (ACF) it is clear that the autocorrelation the original data exhibited is reduced by resampling the series at a higher timestep (Figure 6), yet no differencing term will be used. This is verified by applying the Augmented Dickey Fuller (ADF) test under the null hypothesis that the data series is not stationary. In another case, the coefficients estimated would not be stationary and therefore not invertible. As concluded by Clark and Avery [Clark and Avery, 2010] when larger groups are formed, the effect of spatial autocorrelation decreases by increasing internal heterogeneity within the groups, and decreasing heterogeneity between the groups. The autoregressive term q is also identified by the autocorrelation function (ACF) and the lag p by the partial autocorrelation function (PACF).

Here, the motivated settings of the model are based on the AIC value, determining the best model. Therefore, for 0.4 seconds data resampling we use ARIMA(1,0,0) with AIC 10899.02 and for 0.8 and 2.0 seconds resampling we use ARIMA(4,0,4) with AIC 5154.21 and -27407.45 respectively. The above setting is validated by the *auto_arima* function of *pmdarima* package. With the above models we proceed to make predictions and compare the outcomes to the test set once the model has been fitted to the training dataset. We determine the Root Mean Square Error score (RMSE) after computing the residual error for each prediction made.

Additionally, a sensitivity analysis is conducted using different forecast horizons (2.0, 4.0, 10.0 seconds) and different input length (10.0, 20.0, 50.0 seconds) and comparing the prediction errors of these models (Table 5).

Resampling			Forecasting		Performance		Model	
Agg_data (timesteps)	Agg_data (seconds)	#obs	training window (seconds)	horizon (seconds)	RMSE (1-step)	RMSE (k-steps)	ARIMA (p,d,q)	AIC
2	0.4	27500	10	2	0.511	1.462	1,0,0	10899.02
10	2.0	5500	12	2	1.480	1.480	4,0,4	5154.21
2	0.4	27500	20	4	0.510	2.039	1,0,0	10899.02
10	2.0	5500	20	4	1.097	1.156	4,0,4	5154.21
2	0.4	27500	48	10	0.507	1.898	1,0,0	10899.02
10	2.0	5500	50	10	0.932	1.432	4,0,4	5154.21
4	0.8	13750	560	10	0.114	1.164	4,0,4	-27407.45
4	0.8	13750	1120	20	0.113	1.034	4,0,4	-27407.45

Table 3: ARIMA training inputs and performance

3.2 Support Vector Regression (SVR)

A Support Vector Machine (SVM) is a supervised learning method that creates a hyperplane or group of hyperplanes in a high or infinite-dimensional space that may be used for classification, regression, and other tasks. The introduction of an e-insensitive zone around the function, known as the e-tube, allows SVM generalization to **Support Vector Regression (SVR)**. Unlike linear regression models that use the line of best fit to reduce the error between the actual and projected values, SVR is able to fit the best line within a threshold of values. Hence, the optimization problem is reformulated in discovering the tube that best approximates the continuous-valued function while balancing model complexity and prediction error. The model uses a symmetrical loss function to train as a supervised-learning strategy, penalizing both high and low misestimates equally. Specifically, using Vapnik's e-insensitive technique, a symmetrically created flexible tube of minimal radius is built around the estimated function, with absolute values of errors less than a given threshold e discarded both above and below the estimate. Points outside the tube are punished in this way, while points inside the tube, whether above or below the function, are unaffected. As in Support Vector Machines, SVR is an optimization problem where the most influential parameters that determine the form of the tube are the support vectors, and the train and test data are considered to be identically distributed and independent (iid), that means obtained from the same fixed but unknown probability distribution function in a supervised-learning setting. A brief introduction to the mathematical context of the problem is provided below.

Given a time series $\varphi_1, \varphi_2, \ldots, \varphi_N$, we define multi-step ahead forecasting as determining the relation between the present and the previous observation $x = [\varphi_t, \varphi_{t-1}, \ldots, \varphi_{t-d+1}]$ and future value $y = [\varphi_{t+1}, \varphi_{t+2}, \ldots, \varphi_{t+H}]$ from the train sample. The multi-step SVR solves this problem by determining the regressor w^j and b^j, $(_{j=1,\ldots,H})$ that minimizes for every output:

$$L_p(W, b) = \frac{1}{2}\sum_{j=1}^{H} ||w^j||^2 + C\sum_{i=1}^{n} L(u_i)$$

where :

$$u_i = ||e_i|| = \sqrt{(e_i^T e_i)}$$
$$e_i^T = y_i^T - \varphi(x_i)W - b^T$$
$$W = [w^1, \ldots, w^H]$$
$$b = [b^1, \ldots, b^H]^T$$

C is a hyperparameter that sets the trade-off between the regularization and the error reduction term, and $\varphi(.)$ is a nonlinear transformation to the feature space, which is normally a higher dimensional space. [Bao et al., 2014]

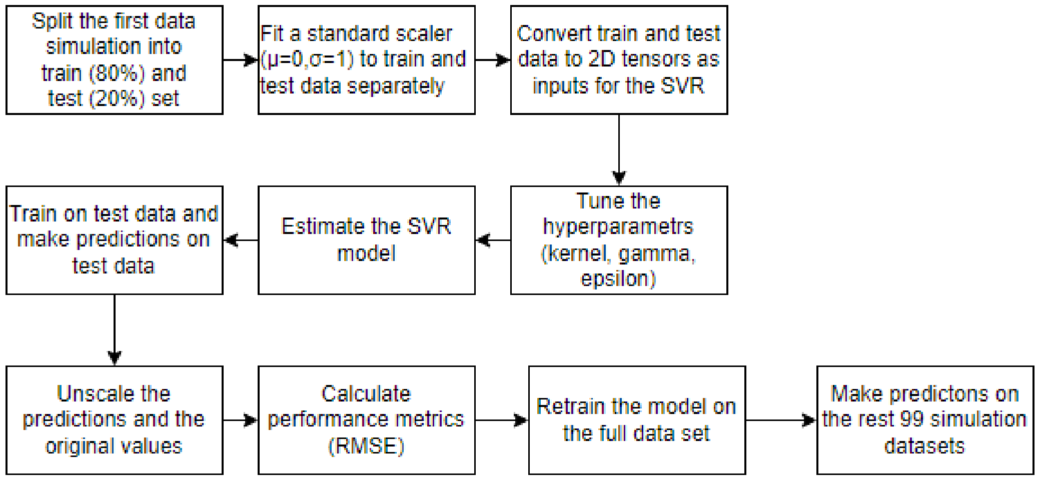


Figure 7: SVR model workflow

As described at Figure 7 with the workflow followed, after the initial data preparation on 0.8 seconds resampled dataset, we convert train and test data 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 make predictions 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.

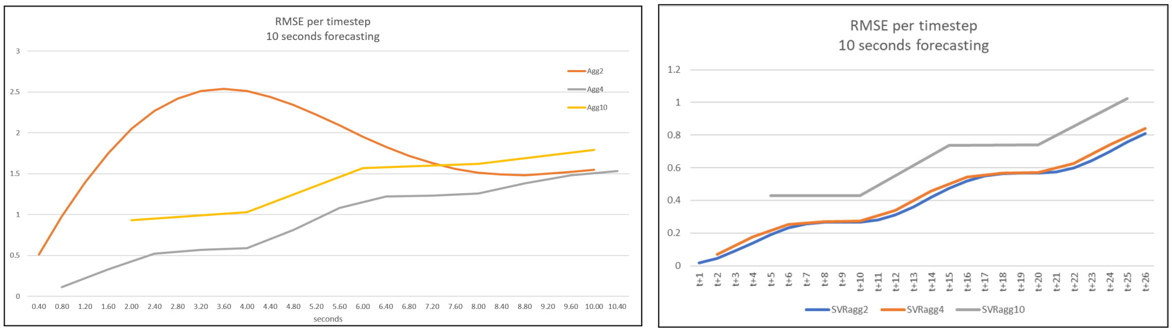
4 Results

	Forecasting		**Performance**	
	training window (seconds)	**horizon (seconds)**	**RMSE (1-step)**	**RMSE (k-steps)**
ARIMA	560	10	0.114	1.164
	1120	20	0.113	1.034
SVR	560	10	0.175	0.840
	1120	20	0.066	0.820

Table 4: Model performance comparison on 0.8 seconds resampled data

The model performance of this study is presented at Table **4** along with the training windows and the forecasting horizon in seconds. As seen in Table **4**, for 10 and 20 seconds forecasting SVR achieves an overall RMSE of **0.840** and **0.820** seconds whereas ARIMA achieves **1.164** and **1.034** respectively. Regarding the sensitivity analysis, RMSE of ARIMA model on different resampling timesteps increases for 0.4 seconds resampling where a different model (1,0,0) was used, from **0.50** to **2.50** until 4.0 seconds forecasting and beyond this point decreases to approximately **1.60**. For predictions 10.0 and 20.0 seconds ahead, the predictions errors of ARIMA are not affected by the resampling timestep used and they are all stabilized and incline to **1.60** (Figure 8a). In the contrary, RMSE of SVR is not significantly affected by the resampling timestep used (Figure 8b), so for a 10.0 seconds ahead prediction RMSE has a value of approximately **0.84** for 0.4 and 0.8 seconds resampling and **1.10** for 2.0 seconds resampling.

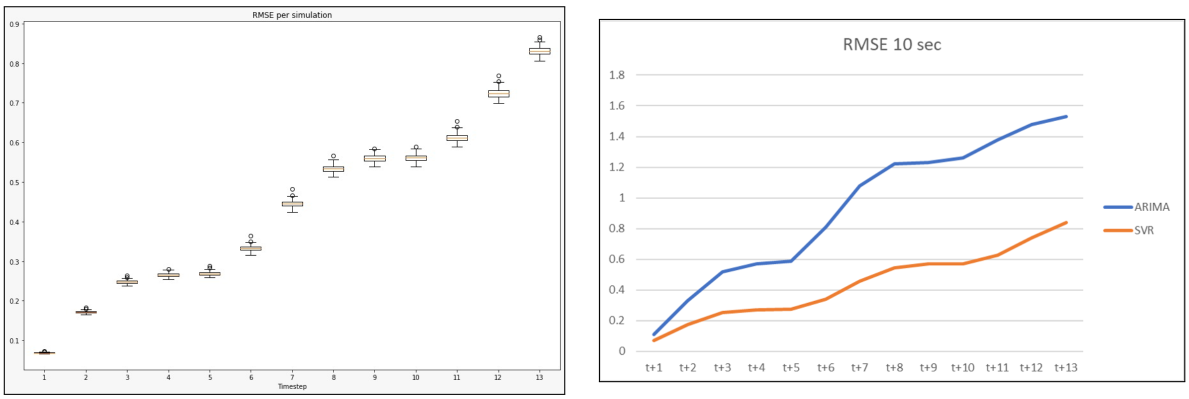
Moreover, the SVR model trained on the first simulation performs consistently to all the rest 99 simulations, no outliers are observed at the box plot (Figure 9a). Eventually, as presented in Figure 9b, SVR outperforms ARIMA model for every timestep.



(a) ARIMA RMSE per timestep

(b) SVR RMSE per timestep

Figure 8: Forecasting 10.0 seconds ahead at 0.4, 0.8, 2.0 seconds resampled data



(a) SVR RMSE by timestep across 99 simulations

(b) ARIMA vs SVR model Root Mean Square Error

Figure 9: Forecasting 10.0 seconds ahead at 0.8 seconds resampled data

5 Discussion & Conclusion

Over the last few decades, time series analysis and forecasting has been a major research topic. In maritime operations, many decision-making processes depend on the accuracy of time series forecasting, so research into ways to make forecasting models more effective has never ended. One of the most widely used techniques in forecasting study and application is the ARIMA model, while with their nonlinear modeling capabilities, Support Vector Regressions (SVR) have also demonstrated promise in time series forecasting applications. Although both ARIMA and SVR are capable of modeling a wide range of issues, neither one is the optimum model that can be used uniformly to all forecasting scenarios.

The best ARIMA validated model for forecasting future values of the time series was presented in this empirical study. The ARIMA modeling's success in predicting is one of the reasons for its popularity. However, even if the model allows us to see what it may be the heave motion of the ship in the coming seconds, it does not allow us to look behind this data progression, so the output highly depends on the input. When it comes to forecasting horizon, the lower prediction errors in this study imply that both models are more effective at short-term forecasting. Additionally, when predicting further into the future from a timestep t, then the very next timestep $t+1$ predictions are improved because a longer training input is used. In total, SVR performs significantly better than ARIMA for 10 and 20 seconds predictions on 0.8 seconds resampled data (Appendix: Figure 17). In practice, ARIMA is able to only capture the trend, while systematically underestimates the predicted values (Appendix: Figure 14) and most importantly it misses the actual volume of the peak, which is a critical point for this study. Moreover, because both models' performance drops for multi-step prediction, it is required to extend the training input long enough. Eventually, SVR performs well to unseen data that is the unused simulations, by using large enough training windows (here, the author incorporated a 9 minutes training window to predict the next 10 seconds) and by discovering patterns, whereas ARIMA performs analogically better by using already shorter training windows than SVR. In contrast with ARIMA, SVR is able to capture not only the trend but also the volume of the forthcoming change sufficiently (Appendix: Figure 15).

Another issue is that training models in high granularity data is indeed time consuming, hence, data resampling is more than a necessity. Following this handling, SVR performance drops for highly resampled data but not significantly, so it is preferred to train both models on 0.8 seconds resampled data, rather than 0.4 seconds for computational reasons. Finally, a blind spot using SVR and other machine learning models is the optimal hyperparameter tuning. For the reason that a brute force technique across every possible combination is considered infeasible, the model performance is tested for only a short list of potential best parameters.

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, the results presented reveal a lot of possibility if this performance holds up in practical implementation. In this paper, a uni-variate solution was approached. To improve human judgment in real-world applications, precise estimates of the other ship motions are essential and future study into multi-variate ship motion prediction is therefore necessary. Furthermore, this study uses simulation data, and the simulation only considers one sea condition. Therefore, it is necessary to investigate if 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. The work described in this paper was mainly focused on predicting a ship motion time series using different data resampling levels. Future research may include a combination of the present two models, ARIMA and SVR taking advantage of the useful properties of both models.

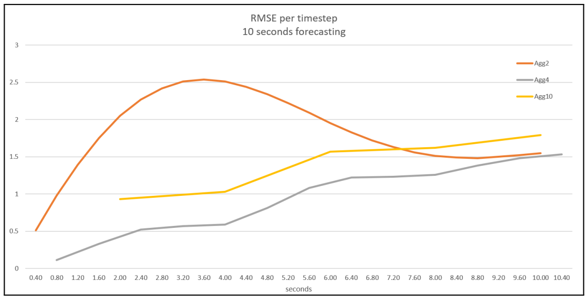
6 Acknowledgments

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

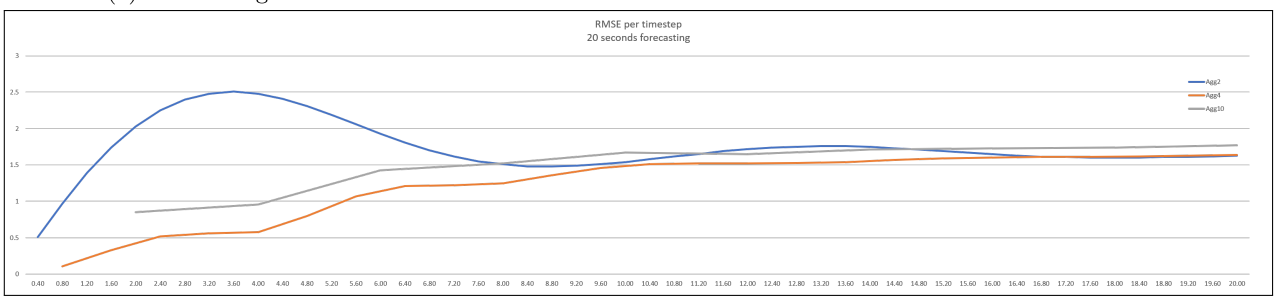
A Appendix

Links to the standalone studies for this case study may be found in the Github repository of this case study : https://github.com/QuintenSand/Thesis-Marin

A.1 ARIMA Sensitivity Analysis

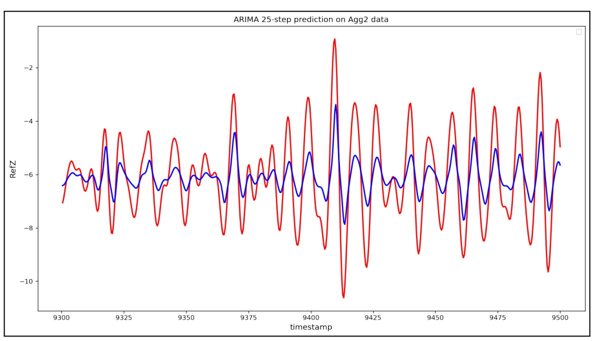


(a) Forecasting 10.0 sec ahead

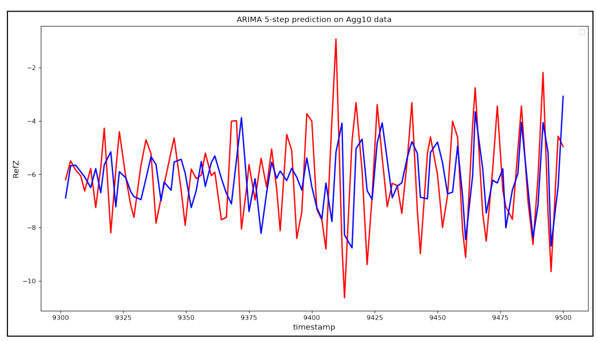


(b) Forecasting 20.0 sec ahead

Figure 10: ARIMA RMSE for 0.4, 0.8, 2.0 seconds resampled data

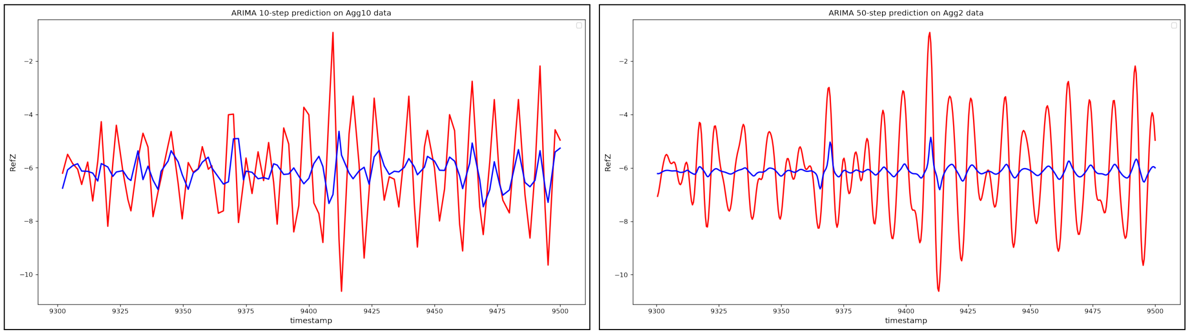


(a) Forecasting on 0.8 sec resampled data



(b) Forecasting on 2.0 sec resampled data

Figure 11: ARIMA model forecasting 10-seconds ahead



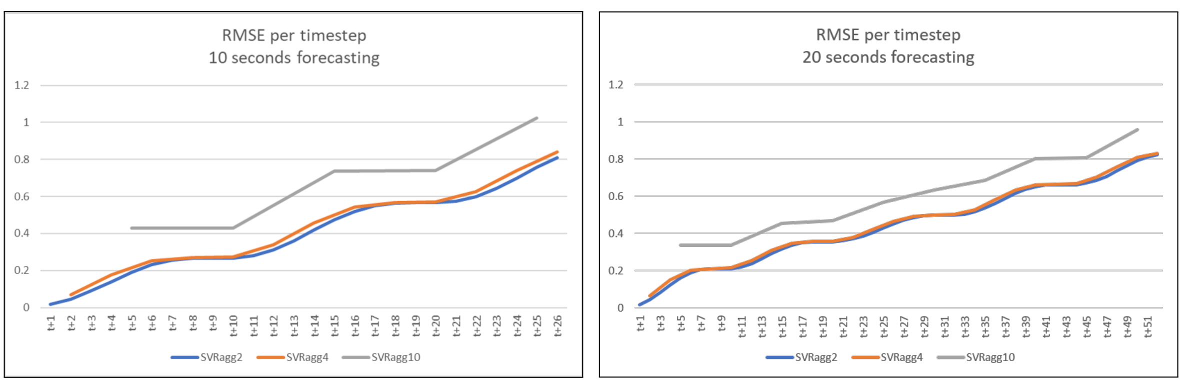
(a) Forecasting on 0.8 sec resampled data (b) Forecasting on 2.0 sec resampled data

Figure 12: ARIMA model forecasting 20-seconds ahead

Resampling			Forecasting		Performance		Model	
Agg_data (timesteps)	Agg_data (seconds)	#obs	training window (seconds)	horizon (seconds)	RMSE (1-step)	RMSE (k-steps)	ARIMA (p,d,q)	AIC
2	0.4	27500	10	2	0.511	1.462	1,0,0	10899.02
10	2.0	5500	12	2	1.480	1.480	4,0,4	5154.21
2	0.4	27500	20	4	0.510	2.039	1,0,0	10899.02
10	2.0	5500	20	4	1.097	1.156	4,0,4	5154.21
2	0.4	27500	48	10	0.507	1.898	1,0,0	10899.02
10	2.0	5500	50	10	0.932	1.432	4,0,4	5154.21
4	0.8	13750	560	10	0.114	1.164	4,0,4	-27407.45
4	0.8	13750	1120	20	0.113	1.034	4,0,4	-27407.45

Table 5: ARIMA training inputs and performance

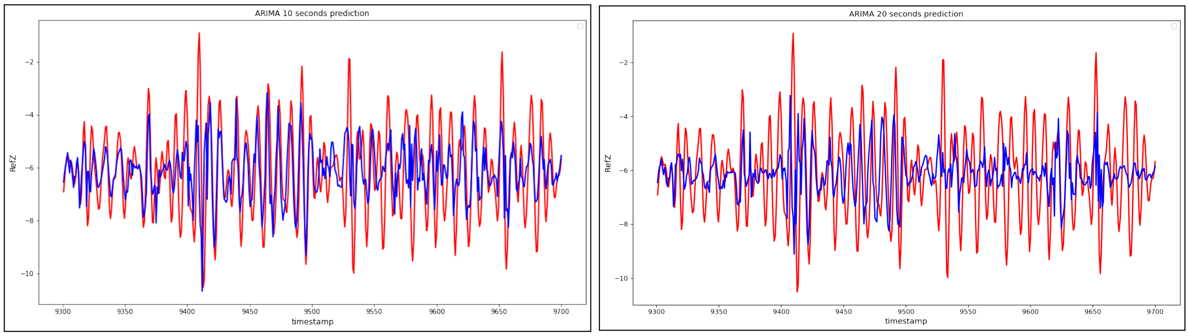
A.2 SVR Sensitivity Analysis



(a) Forecasting 10-seconds ahead (b) Forecasting 20-seconds ahead

Figure 13: SVR RMSE for 0.4, 0.8, 2.0 seconds resampled data

A.3 ARIMA Predictions

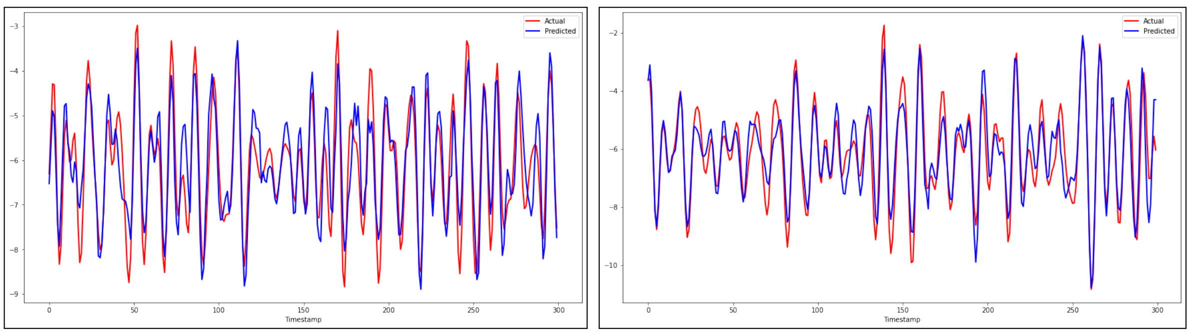


(a) 10-seconds forecasting

(b) 20-seconds forecasting

Figure 14: ARIMA model forecasting on 0.8 seconds resampled data

A.4 SVR Predictions

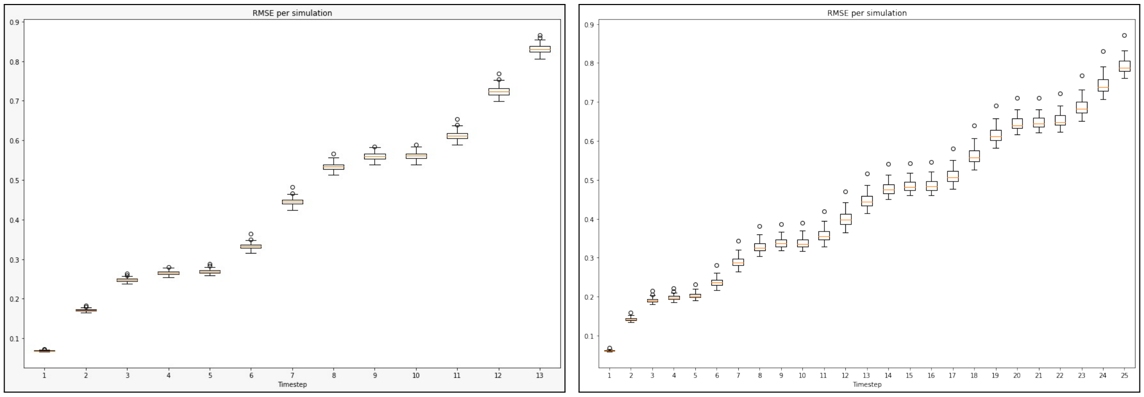


(a) 10-seconds forecasting

(b) 20-seconds forecasting

Figure 15: SVR model forecasting on 0.8 seconds resampled data

A.5 SVR performance



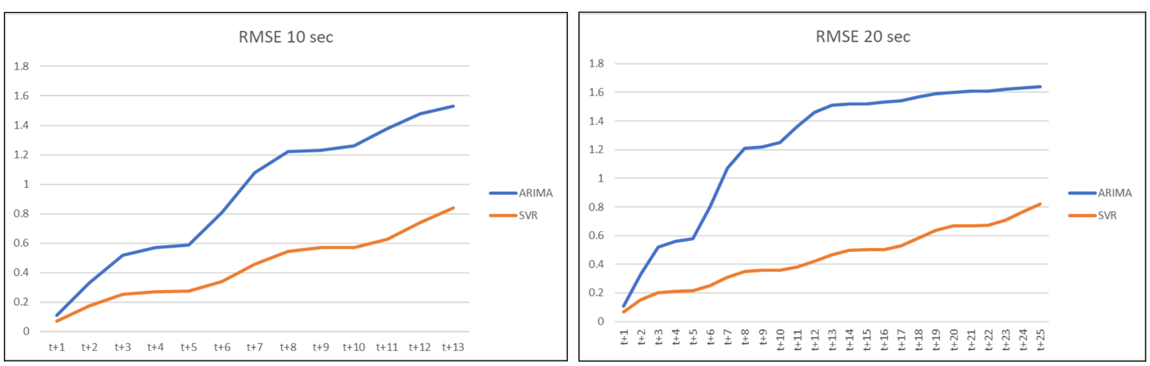
(a) Forecasting 10.0 sec ahead (b) Forecasting 20.0 sec ahead

Figure 16: SVR RMSE box-plot by timestep across 99 simulations

A.6 Model Comparison

	Forecasting		Performance	
	training window (seconds)	horizon (seconds)	RMSE (1-step)	RMSE (k-steps)
ARIMA	560	10	0.114	1.164
	1120	20	0.113	1.034
SVR	560	10	0.175	0.840
	1120	20	0.066	0.820

Table 6: Model performance comparison on 0.8 seconds resampled data



(a) Forecasting 10.0 seconds ahead (b) Forecasting 20.0 seconds ahead

Figure 17: ARIMA vs SVR model Root Mean Square Error

References

[Awad and Khanna, 2015] Awad, M. and Khanna, R. (2015). Support vector regression. In *Efficient Learning Machines*, pages 67–80. Apress.

[Bao et al., 2014] Bao, Y., Xiong, T., and Hu, Z. (2014). Multi-step-ahead time series prediction using multiple-output support vector regression. *Neurocomputing*, 129:482–493.

[chuan Wang et al., 2015] chuan Wang, W., wing Chau, K., mei Xu, D., and Chen, X.-Y. (2015). Improving forecasting accuracy of annual runoff time series using ARIMA based on EEMD decomposition. *Water Resources Management*, 29(8):2655–2675.

[Clark and Avery, 2010] Clark, W. A. V. and Avery, K. L. (2010). The effects of data aggregation in statistical analysis. *Geographical Analysis*, 8(4):428–438.

[Din, 2015] Din, M. A. (2015). Arima by box jenkins methodology for estimation and forecasting models in higher education.

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

[Li et al., 2017] Li, G., Kawan, B., Wang, H., and Zhang, H. (2017). Neural-network-based modelling and analysis for time series prediction of ship motion. *Ship Technology Research*, 64(1):30–39.

[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.

[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.

[Newbold, 1983] Newbold, P. (1983). Arima model building and the time series analysis approach to forecasting. *Journal of Forecasting*, 2(1):23–35.

[Siami-Namini et al., 2018] Siami-Namini, S., Tavakoli, N., and Namin, A. S. (2018). A comparison of ARIMA and LSTM in forecasting time series.

[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.

[Wang et al., 2013] Wang, L., Zou, H., Su, J., Li, L., and Chaudhry, S. (2013). An arima-ann hybrid model for time series forecasting. *Systems Research and Behavioral Science*, 30(3):244–259.

[Yamak et al., 2019] Yamak, P. T., Yujian, L., and Gadosey, P. K. (2019). A comparison between arima, lstm, and gru for time series forecasting. page 49–55.

[Zhang, 2003] Zhang, G. (2003). Time series forecasting using a hybrid ARIMA and neural network model. *Neurocomputing*, 50:159–175.