Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnLine 55-31 (2022) 313–319

Quay Contact Detection for Ships using Motion Sensors and Machine Learning ⋆

Håkon Hagen Helgesen * Kjetil Sekse Kristiansen * Bjørnar Vik ** Tor Arne Johansen *

* Centre for Autonomous Marine Operations and Systems. Department of Engineering Cybernetics, Norwegian University of Science and Technology, Trondheim, Norway (e-mail: hakon.helgesen@ ntnu.no)
** Kongsberg Maritime AS, Aalesund, Norway

Abstract: Detecting when and where contact with the quay appears is an important ability for ships undergoing a docking maneuver. The motion of the quay, environmental forces, and hydrodynamic effects caused by the interaction between the ship and the quay are all important factors that affect if a docking procedure succeeds or not. This paper studies how contact with the quay can be detected using conventional motion sensors. This includes sensor data from inertial measurement units, global navigation satellite systems, and estimates from an inertial navigation system. The contact detection task can be interpreted as a binary classification problem deciding if contact has occurred or not. Three conventional supervised machine learning methods are studied using experimental data captured in full-scale experiments. Logistic regression, a support vector machine and a long-term short memory network have been trained and investigated. The results are promising and a proof of concept illustrating that supervised machine learning is a viable strategy for quay contact detection using motion sensors.

Keywords: Quay contact detection, Supervised learning, Motion sensors, Ship docking

1. INTRODUCTION

Automatic docking systems need the ability to detect when and where contact with the quay appears during a docking maneuver. This is a critical ability for a ferry undergoing an automatic docking procedure and for the mooring system that should be activated immediately after contact. A docking control system typically requires specific changes in the thruster input when contact is obtained to push the ferry towards and stay connected with the quay. Abrupt changes in the desired control action and lingering integral action effects are potential consequences if contact is not detected due to unexpected contact forces and hydrodynamic effects close to the dock. These effects must be considered in the control system after impact.

Detecting when a ferry is in contact with the quay share many similarities with the more general topic of collision detection Kockara et al. (2007); Lee et al. (2020). Collision detection has been a relevant research topic in the automobile industry for several decades as researched in Amin et al. (2014); Kumar et al. (2021); Hannan et al. (2011). Moreover, contact detection has also been important for research on robotic manipulators that interact with other bodies, e.g. Bajo and Simaan (2010). However, contact detection for a ferry is another form of collision detection due to the low accelerations and force impact that are experienced near the end of a docking maneuver. Passenger ferries typically aim for low speed and a negligible impact for passenger comfort and to avoid damage. In addition, there are usually significant vibrations caused by rotating machinery, and motions caused by waves. Methods that look for significant changes in the force or acceleration are therefore not necessarily applicable for ferries. Consequently, this paper addresses how contact with quay can be detected for passenger ferries where the speed and contact forces are small.

Several strategies are reasonable for detecting contact with the quay. One option is to use a positioning system that estimates the relative position to a specific location on the quay. However, such a strategy requires a navigation system with precision of a few centimeters. Moreover, longer ships that aim for contact on one of the sides need precise heading estimates to calculate when and where contact occurs. This typically means that it is necessary with local infrastructure on the quay or additional sensors on the ferry measuring the relative position to the quay along the ship side. Infrastructure on the quay will restrict the applicability of the method to specific locations. Another option is to use a mathematical dynamic ferry model to compare the predicted motion with the actual ferry motion measured by sensors on-board the ferry. This approach is not transferable since a unique model must be identified for each vessel. Moreover, environmental forces must be included in the model and particularly wind gusts and waves from other vessels are troublesome to estimate.

A simple way to detect collisions with high impact is to use a filter with a threshold on acceleration, velocity and/or

⋆ This work was supported by Kongsberg Maritime through the University Technology Centre (UTC) at NTNU. It was also in part supported by the Research Council of Norway through the Centres of Excellence funding scheme, AMOS project number 223254.

10.1016/j.ifacol.2022.10.448

angular rate. It is also possible to use a threshold for the expected rate of change. However, it is challenging to find limits that are applicable in different locations and weather conditions for collisions with a small impact. Wind, waves, and ocean currents are factors that affect the magnitude of these signals, and these forces might dominate the contact forces. Thresholding strategies have been tested on experimental data, but the findings indicate that it is not a robust strategy that can be generalized to several datasets. Using exteroceptive sensors (cameras or LIDARs) might be a viable approach but adds cost and complexity. Moreover, careful placement is crucial, and the contact point is normally much lower than the deck. This can be troublesome for larger ships with restricted areas for mounting since the sensor range and field of view are limited and especially if first contact can happen anywhere along the ship side. Sensitivity to weather conditions is also an obvious concern.

Because of the aforementioned considerations, an attractive option is to find a strategy that avoids local infrastructure on the quay, additional sensors on-board the ferry, and a ferry model. A flexible solution that can be used in several locations is desirable. One interesting option that fits within this mindset is supervised machine learning methods where motion sensor data are used to train a classifier for contact detection. Detecting contact with the quay can be interpreted as a binary classification problem classifying if contact has occurred or not. This paper studies three different classifiers. Two classifiers are based on traditional methods such as logistic regression (LR) and support vector machines (SVM), both described in Dreiseitl and Ohno-Machado (2002). The third classifier is a deep learning approach based on the long short-term memory (LSTM) neural network (Hochreiter and Schmidhuber (1997)). A recurring pattern must be present in the data if supervised learning is a viable approach for contact detection. Therefore, the main motivation behind this work is to investigate whether such a repeatable pattern exist when the contact forces are minor and without extrema of a significant magnitude compared to the general motion.

The classifiers are based on time-series data from typical motion sensors on ships. This includes acceleration and angular rate measured in an inertial measurement unit (IMU), and speed and heading measurements from a dual-antenna global navigation satellite system (GNSS). Data from these sensors are expected to provide the most useful information when typical ship sensors are considered. Similar measurements were utilized in Heyn and Skjetne (2016, 2015) for detection of contact with sea ice. However, they focused on analysis in the frequency domain and used multiple IMUs. Nevertheless, there are obvious similarities between detection of ice-induced motions and quay contact.

The aim of this paper is not to develop the best or optimal classifier for contact detection, but to investigate whether sensor data from an IMU and GNSS can be used to detect contact with the quay with reasonable accuracy. Therefore, this paper should be interpreted as a feasibility study of supervised machine learning methods for quay contact detection. The chosen sensor suite is a low-cost alternative without sensors solely used for contact detection. The rest of the paper is dived into four sections. Section 2 briefly describes the theoretical foundation for the supervised learning methods used in this paper. The methods are well known in the literature and this part is kept short intentionally. Implementation aspects and preparation of data are described in Section 3. The full scale experimental results are presented in Section 4 before the paper is concluded in Section 5.

2. CONTACT DETECTION CLASSIFIERS

Three supervised learning algorithms have been investigated for the purpose of contact detection. LR and SVM are well known in the literature on supervised learning, and LSTM is known to perform well on sequential data such as time-series Karim et al. (2018). The sensor data considered in this work is sequential and this is hence the main motivation for studying LSTM in this context. The contact detection task is a binary classification problem as mentioned previously. Either the ferry has, or has not, been in contact with the quay.

2.1 Logistic Regression (LR)

LR is described thoroughly in Dreiseitl and Ohno-Machado (2002). A logistic function is the basis for this classifier and often modeled as a function

$$f(\mathbf{x}) = \frac{1}{1+e^{-h(\mathbf{x})}} \tag{1}$$

where the function $h(\mathbf{x})$ is defined as

$$h(\mathbf{x}) = \beta_0 + \beta_1 x_1 + ... + \beta_n x_n = \boldsymbol{\beta}^\top \mathbf{x} \tag{2}$$

and n is the number of input variables. $\mathbf{x} = [1, x_1, ..., x_n]^\top$ is the input vector and $\boldsymbol{\beta}$ is a vector of coefficients determined during supervised training. The output of $f(\mathbf{x})$ can be interpreted as the probability of a specific outcome in the binary classification problem, and the function is optimized to use coefficients that maximizes the probability of correct outcome based on the training data.

2.2 Support Vector Machine (SVM)

The SVM is also a well-known classifier, for example described in Noble (2006). In a binary classification task, SVM uses a hyperplane to distinguish between the two classes. The hyperplane is modeled as

$$\boldsymbol{\beta}^\top \mathbf{x} = 0 \tag{3}$$

The weights are determined in training where the hyperplane is designed to give the best possible separation between the outcomes. A perfect separation is rarely possible for experimental data, and thus an optimization problem is solved in training to find the best separation. SVM is a geometric classifier without providing a probability score directly. However, it is possible to quantify the reliability of the result based on how far an input vector is from the hyperplane which separates the outcomes. The most common method for this is Platt Scaling as described in Platt (1999).

2.3 Long Short-Term Memory (LSTM)

The LSTM algorithm is a different approach and is based on deep learning and artificial neural networks.

The method was originally described in Hochreiter and Schmidhuber (1997). LSTM is a recurrent neural network which means that is has a memory that is useful for time-series and sequential data. The outcome of one classification event can therefore affect future outcomes through a feedback connection. LSTM was originally designed to counteract the vanishing gradient problem that often arise in recurrent networks with long-term dependencies. The key is that only a recent time window of relevant data is kept in the memory so that previous irrelevant data do not affect the current outcome. In a contact detection task, it is only a short window of a few tenths of a second that is of interest.

The LSTM classifier has three main parts that decide the outcome. The first component is the cell state (long-term memory) describing the data stored in the network currently. The second component is the output of the previous point in time (previous hidden state). The final component is the input data at the current time step. These components will together determine the classification outcome of the current data. The flow of the algorithm can be summarized as:

(1) The forget gate decides which part of the long-term memory that is relevant based on the current data, and which part of the memory that can be removed.
(2) The long-term memory is extended with the relevant part of the new data.
(3) Find the new hidden state (outcome of classification problem) based on the new data.

Each layer in the network is tuned during training as usual for neural networks.

3. SENSORS AND EXPERIMENTAL DESIGN

This section presents the sensor suite used to collect experimental data. Data preparation and the data labeling process for supervised machine learning are also described.

3.1 Sensor suite

A sensor payload was developed to collect data. The sensor suite included the following sensors:

- Adis 16490 MEMS IMU measuring specific force and angular rate at a frequency of 250 Hz with specifications given in Analog Devices (2022).
- Two uBlox Neo M8T global navigation satellite system (GNSS) receivers equipped with two Harxon HX-GS288A antennas, ublox (2022).
- IDS UI-5260FA-C-HQ electro-optical camera capturing images at 10 Hz. The camera was used to identify when contact with the quay occurred for the labeling process.

The sensor payload was mounted in a pelicase strapped to the top floor on the starboard side of the ferry. The location was chosen due to ease of access.

3.2 Experimental data

The experimental data was collected on a passenger ferry between Vanvikan and Trondheim in Norway. Data from eight different crossings from Vanvikan to Trondheim have been collected. The data was collected across several hours on two different days. Significant tidal variations were present, and the weather varied. However, it was mostly calm and cloudy without much influence from waves or wind. It is mainly the data close to the quay that is of interest. A contact detection algorithm will only be initiated close to the dock to limit the risk of false positives and hence data from open waters is not relevant. Figure 1 displays the passenger ferry used to collect experimental data at the dock in Trondheim.

Fig. 1. Passenger ferry used to collect data.

The operator steered the ferry so that the front could glide with the starboard side along the quay. Consequently, the ferry was pushed out from the dock after the initial contact and numerous discrete contacts could occur before the ferry stopped completely. The initial contact is considered as the most important instance to detect, but it is also beneficial to identify subsequent occasions with contact. The contact point on the ferry influences how the ferry reacts to the collision and therefore the measurements. The accelerometers are for example affected by where the impact occurs with respect to the IMU position. Moreover, contact far from the center would lead to a yaw moment. The selection of sensor placement is outside the scope of this paper but will most likely have an impact on the results.

3.3 Data used for classification

In addition to the measurements from the sensors listed in the previous section, estimates from an inertial navigation system are also utilized. A multiplicative extended Kalman filter (MEKF), as described in Fossen (2021), was implemented to estimate the navigation states of the ferry using the IMU and the dual-antenna GNSS receiver setup. This sensor combination enables the possibility to estimate the position, velocity, acceleration and attitude. Therefore, estimates of these states are also available for all ferry crossings. The classifiers are based on the following measurements/states which are assumed to contain the most valuable information:

(1) Raw specific force measured in the body frame at 250 Hz using the IMU.
(2) Raw angular rate measured in the body frame at 250 Hz using the IMU.
(3) Heading estimates from the MEKF with an update rate of 250 Hz.
(4) GNSS velocity measurements directly from one receiver. The measurements were upsampled to 250 Hz

to correspond to the rate of the other measurements/states.

The ferry dynamics are in general slow with time constants of a few seconds. However, the IMU measurements were contaminated with high-frequency noise most likely caused by vibrations from the engines. Consequently, the raw IMU data was low pass filtered. A cut-off frequency of 10 Hz was chosen to reduce the noise without filtering out low-frequency motion that is expected to be relevant during contact.

3.4 Labeling and Training data

Robust labeling is key with a limited set of training data. In this work, labeling of when contact occurred was conducted with the help of images captured during the docking process and inspection of the IMU data. Contact was detected through manual inspection of the images with an expected accuracy of a single image frame. Images were captured with a frame rate of 10 Hz, and thus the time accuracy of each contact is expected to have a worst-case accuracy of 0.1 s. In addition, the IMU data was inspected within the corresponding period to verify the labeling on occasions where a visible response was seen in the IMU data. The time of a single IMU sample was chosen as the time of contact with the quay.

It is important to emphasize that labeling of sequential data is a different process than labeling of images for object detection. Contact with the quay can occur at several consecutive samples due to the high sample rate of the sensors and therefore a window of positive samples exists around each contact. A window of 12 IMU samples before and 24 samples after the time of contact was labeled as a positive sample. With an IMU rate of 250 Hz, this corresponds to a contact length of approximately 0.14 s. Every sample within this window was labeled to originate from contact with the quay.

The main hypothesis was that a repeatable pattern should be present before and after contact with the quay. This means that the period just before and after impact can contain crucial information. Using measurements from a single time instant can therefore prevent the classifier from learning the typical behavior during contact. Consequently, a sliding window of data is needed to capture the behavior. A sliding window of 38 consecutive measurements was used as feature vector to train the classifiers. If one of the 38 measurements included contact, the entire time window was classified as contact during labeling. The size of the sliding window was motivated by the work presented in Becker and Ebner (2019), where a sliding window of 19 samples was used to detect collisions for a mobile robot using LR. The size of the sliding window has not been further investigated in this work but was chosen after testing different values initially.

Table 1 describes the number of instances with contact in each dataset. The number varies in each dataset because different operators used different strategies when approaching the quay. This is therefore the natural behavior when manually docking the passenger ferry. Moreover, the motion and forces experienced on the ferry during docking were different in each dataset due to varying environmental conditions and the strategy of the human operator.

Table 1. Datasets and number of contacts

Dataset	Number of contacts
1-8	4, 5, 4, 5, 2, 1, 3, 4
Total	28

Another consideration worth highlighting is that the number of positive samples (contact samples) is much smaller than the number of negative samples. Contact only occurs a few times for each crossing (assuming that the ferry is pushed out of the quay a couple of times before contact is maintained) and in a very short period. Therefore, the training data will be dominated by negative samples, and this will also affect how the results should be assessed. Data captured earlier than the final five seconds before the initial contact in each data set were neglected. Even with this strategy, only 5 % of the data originates from positive samples. This means that classification accuracy must be interpreted with care since classifying all samples as negative would give an accuracy of 95 %.

4. EXPERIMENTAL RESULTS

This section presents experimental results using the data presented in Section 3. The Python framework *scikit-learn* was used to implement LR and SVM and *tensorflow* was used to implement LSTM. Detailed tuning of hyperparameters in the training phase is not within the scope and will only be discussed if assessed to be relevant for the interpretation of the results. The default parameters for LR and SVM can be found in LR configuration (2022) and SVM configuration (2022), respectively. The LSTM was implemented with two dropout layers to prevent overfitting, one hidden LSTM layer and one dense output layer. This is a simple configuration but considered to be sufficient for showcasing LSTM in this context.

4.1 Results

Datasets 1-8 mentioned in Table 1 form the basis for the results. To simplify the training process, all data from the sensors have been combined into a single time series (one for each sensor). Figures 2 and 3 show the IMU data in one dataset after low-pass filtering. Contact with the quay occurs at five instances in this dataset, specifically at 957 s, 977 s, 983 s, 997 s, and 1002 s. The time instances are marked in the figures as vertical lines. A notable change can be seen in the accelerometer data along the x and y axes at these times, but there are other similar changes that do not originate from contact. The measured angular velocity does not have the same notable response. However, a change in the sign of the yaw rate (bottom plot in Figure 3) can be seen in some of the instances with contact. This makes sense since the vessel is steering towards the dock and is pushed away before it tries to steer towards the dock again. The magnitude of the yaw rate during contact can potentially say something about where contact appears since it depends on the length of the moment arm in yaw. The magnitude of the measurements vary between the different contacts as illustrated in Figures 2 and 3. This underlines that difficulty of finding general threshold limits and significant differences in the magnitude were also experienced in between the datasets.

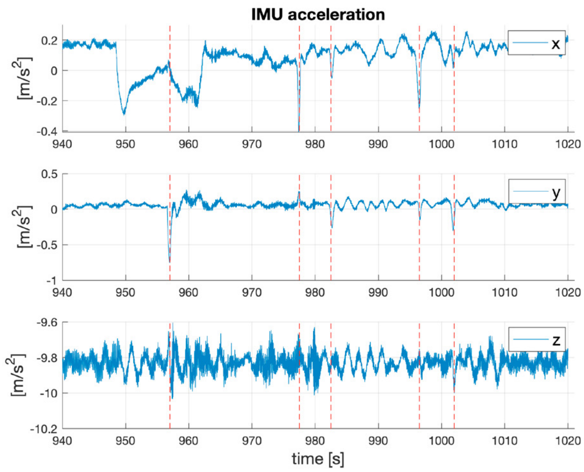

Fig. 2. Low pass filtered acceleration measurements. Contact times are marked with vertical red lines.

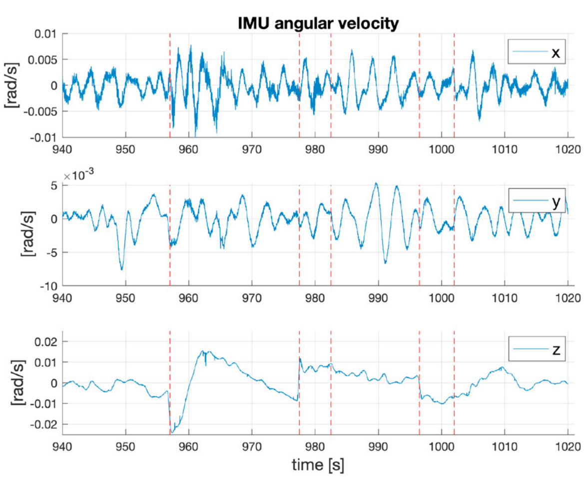

Fig. 3. Low pass filtered angular rate measurements. Contact times are marked with vertical red lines.

A total of 28 contact instances exists in the eight datasets, which is a small amount. The implications of this for machine learning are discussed later. Remember that a sliding window is used and that a single sample with contact is enough to label the entire sliding window as contact with the quay. Consequently, the data includes more than 28 positive samples, but the data originates from 28 unique contacts. The data is split into a training and test set where the training set consists of 11 contacts and the test set of 17 contacts. Cross validation was used during training.

LR vs SVM

The first part of the results is a comparison between LR and SVM with data from all sensors listed in Section 3.3. A chosen set of classification results are presented in Table 2. The classification score threshold is a manually chosen value between 0 and 1. Classification values above the threshold is classified as a positive sample (contact). In binary classification, a classification threshold of 0.5 is typically chosen. However, for imbalanced data sets or in scenarios where misclassification is critical, it might be necessary to decrease or increase the threshold. A threshold below 0.5 typically increases the chance of classifying the positive samples correctly at the cost of more false positives. A threshold above 0.5 typically means that fewer positive samples will be detected, but that the number of false positives decreases.

The receiver operation characteristic curves (ROC) for LR and SVM using default hyperparameters are shown in Figures 4 and 5, respectively. The figures indicate that SVM is more accurate in the training phase and that is also the case in the results using the test set. The bottom two rows in Table 2 show the results on the test set when the parameters in the training phase were tuned to improve the accuracy in training. Even though the accuracy was improved in training, only one more contact was detected in the test set using LR and SVM (at the cost of a false positive for SVM). Other parameter choices were also tested and led to increased accuracy in training. However, the classification results on the test set were not improved compared to the default configuration. This underlines the difficulty of using training accuracy for tuning of the classifier with a small training set.

Table 2. Contact detection results using LR and SVM. TP = true positives, FP = false positives, and FN = false negatives

Method	Classification score threshold	TP	FP	FN
LR - default hyperparameters	≥ 0.8	8	0	9
SVM - default hyperparameters	≥ 0.8	11	0	6
LR - tuned hyperparameters	≥ 0.8	9	0	8
SVM - tuned hyperparameters	≥ 0.8	12	1	5

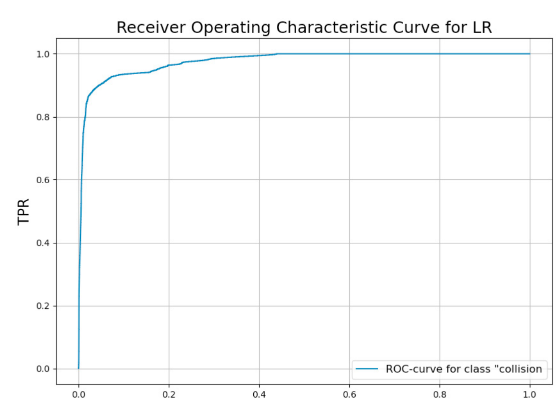

Fig. 4. Receiver operating characteristic (ROC) for logistic regression with default hyperparameters. TPR = True positive rate and FPR = false positive rate.

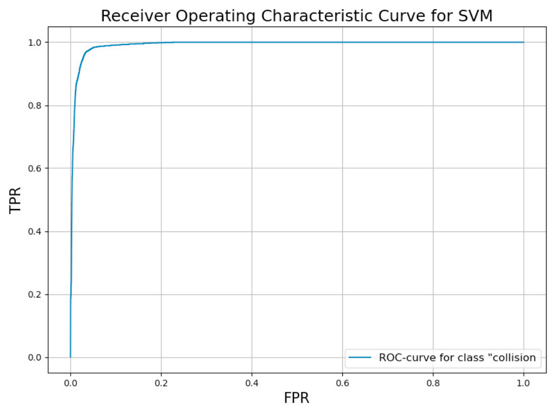

Fig. 5. Receiver operating characteristic (ROC) for the support vector machine classifier with default hyperparameters. TPR = True positive rate and FPR = false positive rate.

Sensor combinations for classification using SVM

The second part of the results will investigate the usefulness of each sensor. Several sensor combinations will be evaluated on SVM, which proved to be more accurate than LR in the first part of the results. Table 3 summarizes the main results using different sensor combinations. Note that the threshold for the classification score has been lowered in this case.

Table 3. Contact detection using SVM and different sensor combinations. TP = true positives, FP = false positives, and FN = false negatives

Sensors	Classification score threshold	TP	FP	FN
Benchmark - all sensors	≥ 0.8	12	1	5
Benchmark - all sensors	≥ 0.2	15	8	2
Accelerometer	≥ 0.2	14	1	3
Angular rate	≥ 0.2	2	1	15
Acceleration + Angular rate (IMU)	≥ 0.2	15	3	2
Heading + velocities	≥ 0.8	16	14	1
IMU + heading	≥ 0.2	16	2	1
IMU + velocities	≥ 0.2	16	7	1
Acceleration, heading, and velocity	≥ 0.2	17	7	0

There are several noteworthy findings in Table 3. The angular rate sensors are not useful alone. Only one combination of sensors detects all contacts at a cost of many false positives. The best combination is perhaps using the IMU measurements together with heading measurements, which detects 16 out of 17 contacts with two false positives. Moreover, the false positives in this case occur immediately after a true contact and is not a severe issue. It is more problematic with false positives before contact with the quay occurs since the control strategy might change after the initial contact. Several other classification score thresholds were also evaluated but only the most interesting results have been included in Table 3.

LSTM

The final result is a short study on LSTM. LSTM needs more training data than LR and SVM. Consequently, the training set was changed to include 17 out of 28 contacts and 11 contacts remained in the test set. In other words, six contacts from the previous results were moved from the test set to the training set. The LSTM was tested on input data from all sensors. The method managed to detect 10 out of 11 contacts in the test set with three false positives. The classification score threshold was 0.8. This is comparable to the best results using SVM. Note that the single missing contact could not be detected at a classification score of 0.2 either. The same contact was also hard to detect with SVM. However, it was detected when using acceleration, heading and velocity measurements as shown in the bottom row of Table 3.

4.2 Discussion

The results in this study indicate that supervised learning can be a promising approach for detecting contact with the quay. However, there are some shortcomings that should be addressed. One weakness is the small amount of data used in this work. It is time consuming to collect and label relevant data, and the data was collected on one ferry in one harbor on two different days. Therefore, it is hard to generalize the results to other ferries and locations without collecting more data. Moreover, it is likely that sensor mounting affects the measurements and therefore also the classifiers. Consequently, sensor placement and the effect it has on the performance is a topic that must be studied for generalization of the results. The classifiers developed in this work were also tested on data from another location (without any relevant training for that location). The docking maneuver was different in this location and the ferry docked head on instead of on the starboard side. The results were naturally not of the same accuracy and indicate that data from the location of interest (or at least from the same type of docking maneuver) should be included in the training phase. In addition, data from varying weather conditions should also be collected to make the classifiers more robust.

The focus in this work has been to investigate if supervised machine learning methods can be used for contact detection. Therefore, other supervised learning algorithms should also be evaluated in the future. Moreover, the accuracy presented here could most likely be improved by tuning the hyperparameters in the training phase further. The number of hidden layers in the LSTM could also be explored further. Nevertheless, the results have proven that it is possible to detect contact with the quay in a repeatable manner without a significant number of false positives. The acceptable number of false positives is a topic for discussion and might differ with the type of ship and docking procedure. Moreover, as mentioned previously, false positives straight after a true positive is a much smaller issue than false positives far from a true positive. When false positives occur is therefore a topic that should be studied more thoroughly. This is also closely related to the classification score threshold that is chosen for a detection to be accepted.

5. CONCLUSION

The results presented in this research indicate that supervised learning is a viable strategy for detecting contact with the quay. Several sensor combinations and methods have been investigated showing varying performance and accuracy. Acceleration measurements combined with angular rate measurements and heading measurements proved to be a particularly interesting sensor combination without many false positives. The tradeoff between false positives and false negatives is very important and might vary with the type of mission. Failing to detect contact is critical in some operations while other missions might struggle if contact is detected too early. Consequently, the timing of false positives and negatives is key and a topic for further research. Moreover, gathering data from different ships, weather conditions, and harbors are needed to investigate how much a detection algorithm can be generalized. The methods chosen in this research are not necessarily the best ones for the application so other methods should also be studied in the future.

This paper has investigated motion sensors typically present on ships. An interesting topic for future work is fusion of data from additional sensor technologies, such as exteroceptive sensors or dedicated ranging sensors. For larger vessels, it might be difficult to place cameras and LIDARs so that the quay can be detected and tracked accurately within the sensor range and field of view. However, if this is solved, other sensors might provide complementary information that can be used for contact detection. Another topic of interest is to study the effect of IMU placement and if multiple IMUs can improve the accuracy. Finally, investigating if it is possible to use the same classifier without additional training on other vessels of the same type is an interesting topic.

ACKNOWLEDGEMENTS

The authors are grateful for the cooperation with Kongsberg Maritime and the feedback received from them during this work, in particular, Sverre Torben and Øivind Kjerstad. We are also grateful for the cooperation with FosenNamsos that has given us access to acquire data on their vessel.

REFERENCES

Amin, M.S., Nasir, S.S., Reaz, M.B.I., Ali, M.A.M., and Chang, T.G. (2014). Preference and placement of vehicle crash sensors. *Technical Gazette*, 10(4), 889–896.

Analog Devices (2022). URL `https://www.analog.com/en/products/adis16490.html#`. Last accessed 2022-02-22.

Bajo, A. and Simaan, N. (2010). Finding lost wrenches: Using continuum robots for contact detection and estimation of contact location. In *IEEE International Conference on Robotics and Automation*, 3666–3673.

Becker, F. and Ebner, M. (2019). Collision detection for a mobile robot using logistic regression. In *Proceedings of the 16th International Conference on Informatics in Control, Automation and Robotics - Volume 2: ICINCO*, 167–173.

Dreiseitl, S. and Ohno-Machado, L. (2002). Logistic regression and artificial neural network classification models: a methodology review. *Journal of Biomedical Informatics*, 35(5), 352–359.

Fossen, T. (2021). *Handbook of Marine Craft Hydrodynamics and Motion Control, Second Edition*. John Wiley & Sons.

Hannan, M.A., Hussain, A., and Samad, S.A. (2011). Sensing systems and algorithms for airbag deployment decision. *IEEE Sensors Journal*, 11(4), 888–890.

Heyn, H.M. and Skjetne, R. (2015). Estimation of forces caused by ship-ice interaction using on-board sensor measurements. *Proceedings of the 23rd International Conference on Port and Ocean Engineering under Arctic Conditions*.

Heyn, H.M. and Skjetne, R. (2016). A system for measuring ice-induced accelerations and identifying ice actions on the CCGS Amundsen and a swedish Atle-class icebreaker. *International Conference on Offshore Mechanics and Arctic Engineering*.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8), 1735–1780.

Karim, F., Majumdar, S., Darabi, H., and Chen, S. (2018). LSTM fully convolutional networks for time series classification. *IEEE Access*, 6, 1662–1669.

Kockara, S., Halic, T., Iqbal, K., Bayrak, C., and Rowe, R. (2007). Collision detection: A survey. In *IEEE International Conference on Systems, Man and Cybernetics*, 4046–4051.

Kumar, N., Acharya, D., and Lohani, D. (2021). An IoT-based vehicle accident detection and classification system using sensor fusion. *IEEE Internet of Things Journal*, 8(2), 869–880.

Lee, S.i., Kim, H., Kim, U., and Lee, H. (2020). Concave wall surface tracking for aerial manipulator using contact force estimation algorithm. In *2020 20th International Conference on Control, Automation and Systems (ICCAS)*, 850–855.

LR configuration (2022). URL `https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html`. Last accessed 2022-03-03.

Noble, W.S. (2006). What is a support vector machine? *Nature biotechnology*, 24(12), 1565–1567.

Platt, J. (1999). Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers*, 10(3), 61–74.

SVM configuration (2022). URL `https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html`. Last accessed 2022-03-03.

ublox (2022). URL `https://www.u-blox.com/en/product/neolea-m8t-series`. Last accessed 2022-02-22.