Proceedings of the 2022 Winter Simulation Conference
B. Feng, G. Pedrielli, Y. Peng, S. Shashaani, E. Song, C.G. Corlu, L.H. Lee, E.P. Chew, T. Roeder, and P. Lendermann, eds.

PREDICTIVE MAINTENANCE POWERED BY MACHINE LEARNING AND SIMULATION

Mernout Burger
Csaba A. Boer
Edwin Straub
Yvo A. Saanen

TBA Group
Lange Kleiweg 12
Rijswijk, 2288 GK, THE NETHERLANDS

ABSTRACT

To optimize the balance between costs and reliability of cranes, it is important to perform maintenance when the risk of failures becomes high while possibly delaying planned maintenance when the crane shows no signs of possible problems. To accomplish this, we investigate the possibility of applying predictive maintenance for container-handling cranes. The application of predictive maintenance requires historical data collection and preprocessing of equipment sensor and maintenance data. To get a feeling of the possibilities and limitations of predictive maintenance for container-handling cranes, before investing time and money to collect operational data, we have used simulations to generate synthetic data for a few components of the cranes. Using the simulated crane data, a prediction model was trained to predict upcoming component failures. The results show that using simulation we can identify the possibilities and limitations of machine learning for predicting failures of components of the crane.

1 INTRODUCTION

Today, large amounts of data are produced by container terminals, either by equipment or *terminal operating software* (TOS) and *equipment control systems* (ECS) running on the terminal (Port Technology 2015). By analyzing long term operation data, we can recognize patterns either in planning, equipment dispatching, or even in equipment failure or breakdown. The large amount of data and the observed patterns allow us to predict events to happen in the near future and make corrective suggestions. In order to achieve this an artificial intelligence application, called *machine learning,* is widely applied (Tian et al. 2012; Kanawaday and Sane 2017; Carvalho et al. 2019)

In the last decade machine learning techniques won more and more ground in practice in different domains, such as healthcare, financial services, and automotive sectors, with great success (Géron 2017). However, in container terminal business, the use of artificial intelligence is still an undiscovered gold mine that offers an array of possibilities. Literature on predictive maintenance for container cranes is very limited. In (Szpytko and Salgado Duarte 2021) an optimization algorithm is proposed to schedule maintenance taking into account the crane characteristics (using digital twins) and the planning for (off)loading.

Research indicates that the characteristics of successful safety management are similar to those of successful quality management (Carvalho et al. 2019). Therefore, it is essential to maintain the quality of container terminal equipment using *predictive maintenance*. Next to having an impact on the customers' operations safety, predictive maintenance can have a significant impact on cost savings. Maintenance costs form a large percentage of the total cost in the product lifetime of a harbour crane (Gurning et al. 2021),

978-1-6654-7661-4/22/$31.00 ©2022 IEEE

hence reducing these costs will have a large impact. From a business point of view, predictive maintenance savings come in four forms:

- *Avoid or minimize the downtimes of equipment*. The users will achieve better performance by maximizing the reliability and availability rates of the equipment. Every 30 minutes breakdown results in a loss of 15 containers less performed and a crane gang idle during the breakdown time.
- *Optimize periodic maintenance operation planning*. By tracking failure patterns, a machine learning model can get realistic optimal data on maintenance planning. It could well fit in a plan to deliver optimized *overall equipment effectiveness* (OEE) programs.
- *Reduce warranty expenses*. Conducting periodic maintenance on the equipment, the product quality increases, and by this the warranty expenses are reduced, which again will lead to cost savings for equipment manufacturers.
- *Stronger market share*. The collected data is valuable for the equipment lifecycle decisions. It could provide crucial data on the lifetime of components to identify the weaknesses and strengths in the equipment lifecycles. Ultimately resulting in a stronger market share and position by delivering the best value for the investment made.

Typically, terminals have 10-15% additional machines to cater for unavailability due to breakdown or maintenance. When maintenance can be better predicted and planned, the percentage of additional equipment can be reduced. Overall, this will lower the cost of operation. Take a 1M *twenty foot equivalent unit* (TEU) operation with *rubber-tired gantry cranes* (RTGs). Typically operated by 20-25 RTGs, of which 10-15% (2-3) for the unavailability. Assume this can be reduced by one RTG. This will have an effect on a yearly basis of approx. 200,000 – 250,000 USD / year, or a reduction in *operating expenditures* (OPEX) of 42 US$c per container.

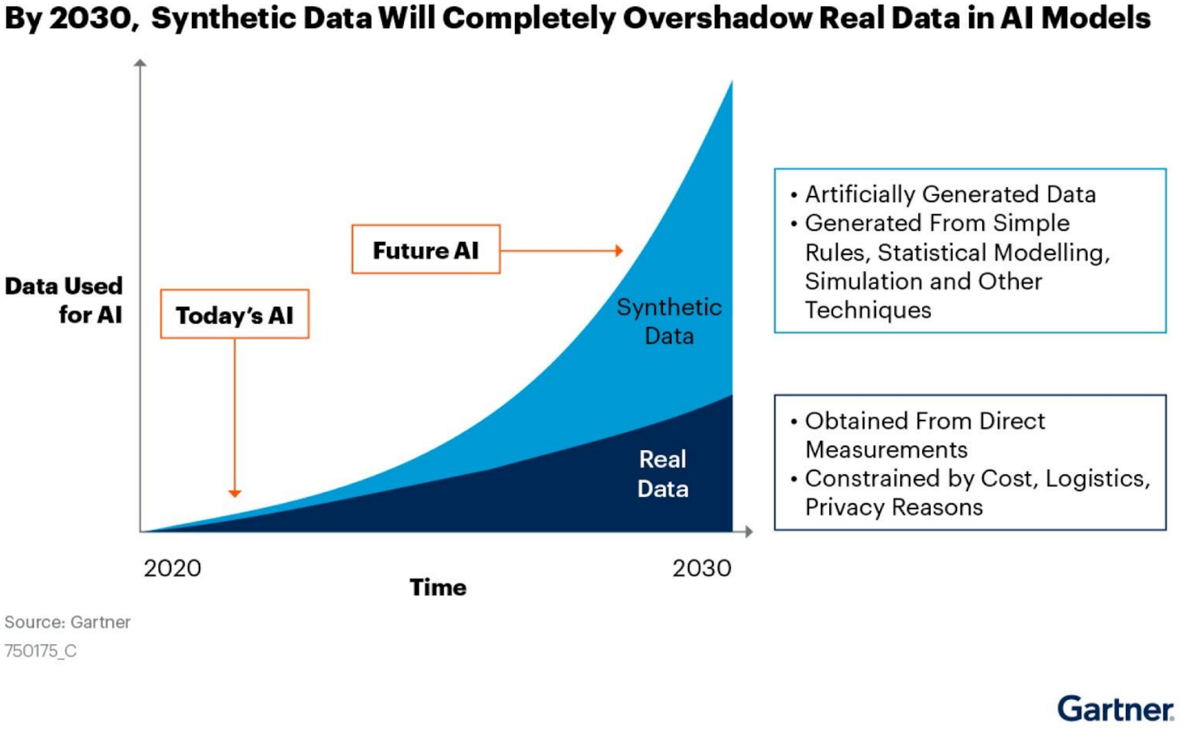

Figure 1. Prediction of using synthetic data (Gartner 2021).

Our research is a feasibility study to assess the potential of predictive maintenance for container cranes. We have conducted experiments with currently available data from mobile harbour cranes. This data contained the times when cranes broke down (but without details which component was broken), and the available sensor data mainly consisted of alarms to check certain components (which often is too late to prevent breakdowns in operation). Both the quality and quantity of data was not sufficient to predict maintenance days before a component would fail, for that we expect to need raw sensor data and detailed maintenance reports.

Since real data is not readily available, it will require a significant investment to make it available, and the research described in this paper is performed to assess if this investment would be worth it. We use simulation (Law and Kelton 2000) to generate synthetic data based on our long-term experience in container terminal business. After collecting the synthetic data, we used different machine learning strategies, such as logistic regression, neural networks, K-nearest neighbors models, and genetic algorithms to predict the equipment component failures. Besides introducing predictive maintenance based on sensor data for container cranes, the use of genetic algorithms to optimize a prediction model is uncommon but effective.

The paper is structured as follows. First, we describe the simulation model of a mobile harbor crane which will be used to generate synthetic equipment data. After that, we describe different scenarios for data generation considering different sensors, different down-time for cranes and different time periods. After collecting the data, we will mainly focus on a predictive machine learning model based on genetic algorithms and use the synthetic data for predictive maintenance. Finally, we present the results considering the different scenarios, and conclude if it is wise to start collecting real operational data and apply the prepared predictive maintenance for it.

2 SIMULATION MODEL

Container terminals come in different shapes and sizes (Van Ham and Rijsenbrij 2012). We have focused on the modelling of waterside operations using *mobile harbor cranes* (MHCs) for handling vessels.

2.1 Container Terminal Operation

Container terminals are a complex, multimodal system which has a yard with multiple storage blocks (Van Ham and Rijsenbrij 2012). Containers can arrive and depart with different container carriers, such as vessels, barges, trucks and trains. Cranes are used to (off)load containers from/onto these carriers, and often placed on (manual or automated) trucks that will drive the container from and to the yard. At these storage blocks another (type of) crane will move the container between the storage location and the truck. For the purpose of simulating the effect of wear and tear on cranes, a simplified model of container terminal operations is used.

We have simulated simplified operations at the *waterside* of a container terminal. Here large vessels are berthed at the *quay*, on which one or more quay cranes are handling the discharging and loading of containers from/to the vessel. Per quay crane several work lists are planned that describe which containers should be discharged (moved from the vessel to the yard) and loaded (moved from the yard to the vessel). One specific type of quay crane is the *mobile harbor crane* (MHC), which is flexible in its movement since it has tires (as opposed to quay cranes that are fixed on rails along the quay).

When a vessel arrives, it is assigned a number of cranes based on the number of containers to move, and the availability of the cranes. At quiet times one can assign the desired number of cranes to a vessel so that the work is expected to be finished within the scheduled time. But when there are many vessels at the same time, the number of available MHCs might not be enough to give each vessel the desired number of cranes, leading to delays in the schedule (Agerschou et al. 2004).

For the experiments we create a realistic number of container moves for 4 years of operation. The work is created randomly (following some settings) during the simulations, as described next. The simulator creates random vessel visits, such that sometimes there are no vessels and sometimes a few at the same time (as also happens in reality). Work is generated for each vessel visit, called a *vessel work list*, which consists of containers that should be loaded onto the vessel (randomly selected from the current yard inventory), and containers that will arrive with the vessel. Based on the number of planned moves, a number of quay cranes are assigned to the vessel (such that the work can be completed within a few hours), and the work is divided over these cranes. Each container move contains information about the *weight* of the container, and the *duration* for handling the move. These values are also generated randomly within reasonable bounds, and they will affect the rate of deterioration of the MHC components.

2.2 Mobile Harbor Cranes

We want to investigate the possibilities and limitations of predictive maintenance for mobile harbor cranes. For this we will simulate the wear and tear of its components, and create sensor data that has a correlation with the health level of the component.

An MHC can drive to the desired location along the vessel where it is planned to (off)load containers, but during the actual movement of the containers it will be standing still. The crane will continuously pick up a container (from the vessel or the quay), rotate to the other side, and drop off the container at the desired location (on the vessel or the quay). To perform these actions, several components are active that allow the container to be lifted and moved. The MHC status is logged each sampling time, with possible values

- OFF: the crane is not active
- ON: the crane is active but currently not working
- WORKING: the crane is performing a move
- BROKEN DOWN: the crane is damaged and requires maintenance

Examples of components of an MHC are the hoist motor, its gearbox, hoist cables, the trolley, and its engine. The investigation focusses on the hoist motor and gearbox, components for which relatively often failures are reported. For both components a *health level* is specified, which is a value representing the condition or quality of the component. When the health level becomes low the *chance of failures* increases. During simulation the health level of a component will go down over time (simulating the effect of corrosion) and due to usage (wear and tear dependent on the container weight). After maintenance the health level will be restored fully.

For both the gearbox and hoist motor two sensors are modelled that are related to the component, representing measurements such as temperature, power consumption, weight, etcetera. The sensor values are correlated to the component health, but fluctuate in value due to the correlation with the workload and ‘random noise’ representing external disturbances and signal noise. An example of sampled data for the four sensors is shown in Figure 2, where the day index is shown on the horizontal axis and the measurement value is given on the vertical axis.

The level of degradation of a component is not measurable directly, but (as shown by the trend in the signals in Figure 2) are expected to be indirectly measurable by one or more sensors. Yet we don’t expect this to be directly usable, but rather affected by the workload and external factors that will cause fluctuations in the measurements. The challenge at the learning stage for creating the predictive maintenance model is to (automatically) select the sensors that have a strong correlation to the component’s health level with little disturbance from the workload or external factors.

The failure of a component is simulated by specifying a health level value around which the component will break down. Besides the health level, also the workload is taken into account (a component is more likely to break down while lifting heavy containers), and a random component is added to model for example weather conditions (strong winds and extreme temperatures can also add to the likeliness of a failure). As a result, the component will not fail at exactly the given health level, but sometimes below and sometimes above this threshold value.

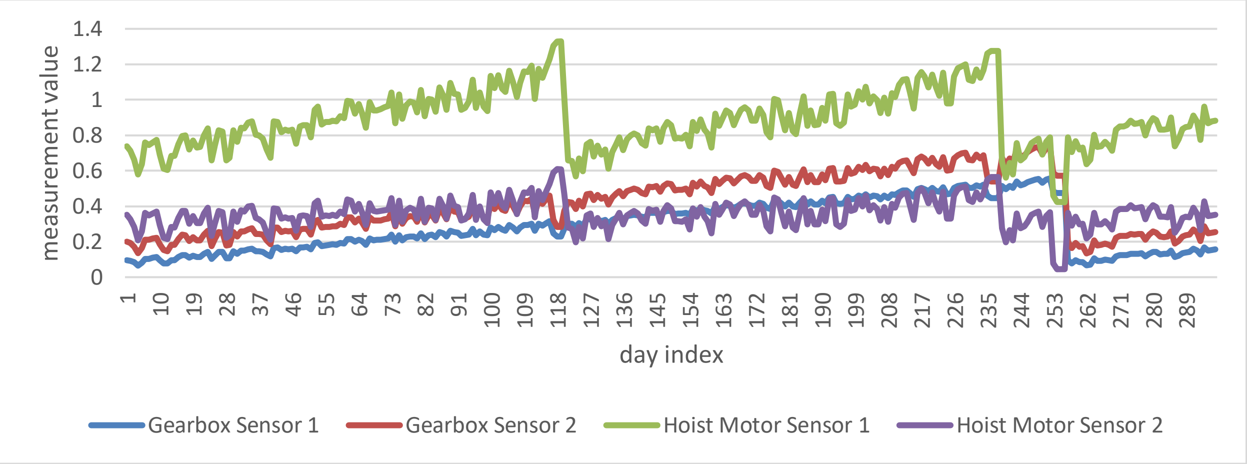

Figure 2: An example of the daily averages for the four sensors. All values go up as the component's health goes down; workload and noise cause fluctuations in the measurements.

Once a component fails, we say that the entire MHC has broken down, and it will be out of operation for about 4 days; again we use a randomly generated period required for the maintenance to complete. Once the maintenance is done, the health level of the broken component is restored fully, and the MHC is available again to be selected for operation.

3 SYNTHETIC DATA GENERATION USING SIMULATION

3.1 Data Collection

The sensor and failure data is generated by simulating MHC operations for a period of 4 years. For every minute we store the sample values s(k), which contains the values of the 4 sensors and the state of the MHC. We created data sets with different percentages (1%, 2%, 4%, 8% and 16%) of down-time for a crane, meaning the crane will be broken-down for about 3.5, 7, 14, 29 or 58 days per year on average.

3.2 Quality of Data

The expectation is that predictions become better when

- The percentage of down-time is higher (more examples of failures in the training data)
- Predictions are made per component (link a specific component failure to sensor data from that component)

A high down-time is obviously undesirable for the real cranes; what will be demonstrated is that a few examples of component failures are required to learn from, otherwise one cannot make good predictions. That is, if a component only wears out once every few years it might not be feasible to use predictive maintenance for it.

Beforehand, we do not know which sensors will be able to predict an upcoming failure for which component. One of the strengths of machine learning is that -in theory- it can find such patterns for you, but it might need a lot of data (and examples of failures) before it can make good predictions. In practice, where we want to find a predictive maintenance model within reasonable time (so that it can be used to save costs for a large part of the crane's lifetime), one needs to combine domain knowledge (which sensors might be related to which component failure) with the strength of machine learning. This research will demonstrate the difference in learning rate between using little unrelated data (component-wise predictions) and much unrelated data (crane-wise predictions).

3.3 Quantity of Data

For creating a good prediction for component failures, maintenance and sensor data is needed that covers a period with multiple break-downs. Since failures are not expected to occur often, in practice it could take years to collect data from one MHC that contains enough breakdowns. This means that the crane might almost be out of its warranty period before one would be able to predict failures.

To speed up the process it is expected that combining the measurements from different (but similar) cranes for the training is possible, thereby reducing the data gathering period by the number of MHCs in use. That is, if for example a certain component failure is likely to occur once per year, then by using the data of 12 MHCs we would expect that component failure to occur once per month (after some time when the cranes are not all new anymore). If 4 breakdowns are needed to learn to predict an upcoming breakdown, this can now be achieved after about 4 months (12 cranes) rather than 4 years (1 crane).

4 PREDICTIVE MAINTENANCE MODEL

For predictive maintenance a model is needed that will tell how likely it is that a component failure will occur soon, based on the current sensor values. We trained our model to predict a failure up to 7 days ahead.

4.1 Prediction Model

To predict a failure, a logistic function (Peng et al. 2002; Buis 2007) is used with the vectors of sensor data *s(k)* as input given as

$$\mathrm{P}_{\text{failure}}(s(k)) = \frac{1}{1+\exp(-r*s(k))}. \tag{1}$$

Here *r* is a multiplier vector (the prediction model) that needs to be found such that the value of $\mathrm{P}_{\text{failure}}(s(k))$ becomes higher than a chosen threshold when a failure is reported within 7 days from now.

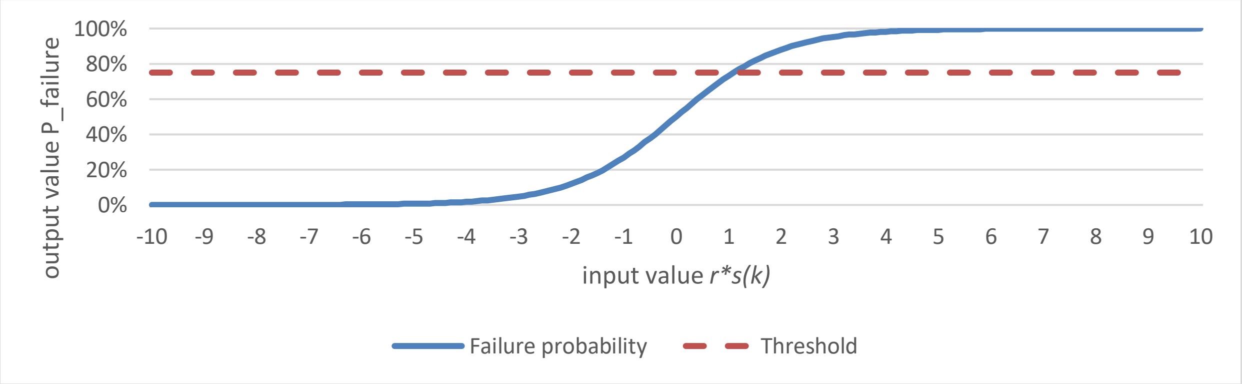

Figure 3: A logistic function (s-shaped line) is used to model the failure probability of components, where probability values above a certain threshold should warn that a failure occurs within 7 days from now.

We have used a threshold of 75%, such that when $\mathrm{P}_{\text{failure}}(s(k)) > 75\%$ it means a component failure is expected within 7 days, and the MHC should be taken out of operation and maintenance should be performed (see Figure 3). This leads to four possible cases for the predictions:

- **True positive**: predicting a failure while a failure was also reported within 7 days
- **False positive**: predicting a failure while no failure was reported within 7 days
- **True negative**: not predicting a failure while also no failure was reported within 7 days
- **False negative**: not predicting a failure while a failure was reported within 7 days

A *true positive* means a breakdown would be prevented in operation by taking out the MHC for maintenance in time; a *true negative* means the prediction correctly tells it is safe to operate the crane. A *false positive* means the crane is taken out of operation for maintenance too early, and a *false negative* results in a breakdown in operation because the component failure was not predicted; both negatives will result in larger costs. The quality of maintenance predictions is specified using two notions:

- **Precision**: percentage of failure predictions that were correct
  - Calculation: #True positives / (#True positives + #False positives)
  - Impact: the higher the precision, the less unnecessary inspections
- **Recall**: percentage of failures that were predicted
  - Calculation: #True positives / (#True positives + # False negatives)
  - Impact: the higher the recall, the less breakdowns in operation

4.2 Training Approach

To find a vector *r* that would result in both a high precision and high recall, we have tested several learning strategies: logistic regression, neural networks, K-nearest neighbors models, and genetic algorithms (Holland 1975; Géron 2017). The latter gave the best results, and this approach is described next.

The challenge is to find a vector *r* such that the value of the logistic function in equation (1) becomes high when a failure is about to happen, while staying low when the system is healthy. For a single MHC we will have K measurements of sensor values *s(k),* where K is the product of the measurement period and the number of samples taken per unit time.

Furthermore, we have information about the period that a crane was broken down; the start of that period is when the failure occurred, and we want to predict the failure 7 days prior to it. The symbol f_i is used to indicate the time step *k* at which the period of the *i*-th failure started, and p_i to indicate the time step 7 days before. The time step s_i indicates when the MHC started working again after the *i*-th breakdown. Using the definitions above and using *F* to indicate the number of failures in a dataset, the goal is to find a vector *r* that results in values

$$P_{failure}\big(s(k)\big)\begin{cases}< 0.75\ when\ s_i \le k < p_i\ for\ all\ i \in \{1,\dots,F\}\\ \ge 0.75\ when\ p_i \le k < f_i\ for\ all\ i \in \{1,\dots,F\}\end{cases}$$

Note that we ignore the periods from f_i to s_i where the crane was broken down; during this period the MHC is mostly off (meaning there are no measurements) or on for testing and tuning during maintenance (unreliable measurements).

4.2.1 Genetic Algorithm

We have used a *genetic algorithm* (Holland 1975; Alajmi and Wright 2014; Katoch et al. 2020) to find a good vector *r* for prediction of failures. Each element in *r* is a multiplier for a specific element in *s(k).* Some sensors might have a strong correlation to a certain component failure, while others are unrelated. Using a genetic algorithm, a good vector *r* for all failures can be found step-wise by combining vectors that can predict one or a few failure types correctly.

The algorithm uses a set of vectors r_p representing the *population* of P candidate solutions. Initially, we create P random vectors r_p (satisfying some bounds). For each candidate solution r_p the fitness is calculated as a weighted sum of steps k where $\mathrm{P_{failure}}(s(k))$ would result in false positives or false negatives; the lower the value of this sum the better the prediction of Equation (1) with $r = r_p$.

The best few candidate solutions (the *elite*) are kept in the population for the next generation, and new candidate solutions are created. A part of the new generation is created randomly (as during the initialization), while the other part is based on the elite solutions using crossover and mutation. For *crossover* two (distinct) member solutions are randomly selected from the elite, and the first few values from the first vector and the last values from the second vector are used; how many values are used from the first is randomly chosen for each individual crossover action. During *mutation* a member from the elite is selected randomly, and a few of its values are modified (where the number of changed values, the indices of the changed values, and the new values are all random as well).

Selecting the elite from a population, and creating a new population using these elite candidate solutions is done a pre-specified number of times. The final solution for vector r will then be the candidate solution r_p with the lowest value for the fitness function, indicating that it is the best solution found to predict upcoming component failures.

4.2.2 Levels of Complexity

We specified four test scenarios using simulated data from two components (gearbox and hoist motor), each with two sensors

- Gearbox-related sensors GBS1 and GBS2
- Hoist motor-related sensors HMS1 and HMS2

Using the sensor data and failure reports four training scenarios are specified with an expected increasing level of complexity for obtaining a reliable failure prediction:

- **Level 1**: The first level trains a prediction model for gearbox failures using only the gearbox sensor data. Since the sensors are related to the component it is expected that training is easy, and predictions should be quite good.
- **Level 2**: The second level trains a prediction model for hoist motor failures using only data from the hoist motor sensors. Since the sensors are related to the component it is expected that training is easy, and predictions should be quite good. But since the hoist motor data has more 'noise' than the gearbox data (see Figure 2), it is expected to be more difficult to make good predictions.
- **Level 3**: For the third level we train a model to predict gearbox failures, but this time use the data of all 4 sensors. This means that 2 out of 4 sensors are expected to have no predictive value for the gearbox breakdown, and we expect a lower-quality predictor because training time is wasted on those two sensors as well.
- **Level 4**: The fourth level generalizes the gearbox and hoist motor failures into a crane failure, and train the prediction model using the data from all four sensors. The model cannot learn to ignore a part of the sensor data as in Level 3; sometimes the crane failure can be predicted by the gearbox sensors, and sometimes it can be predicted by the hoist motor sensors. We expect this to be the most challenging scenario, with the lowest-quality predictions.

5 RESULTS

Using the simulation data described in Section 3 logistic models are trained to predict failures as described in Section 4. The resulting quality of the predictions is discussed in this section.

5.1 Levels of Complexity

Using the MHC simulator sensor and failure data was generated for a period of 4 years with different failure rates: 1%, 2%, 4%, 8% and 16% down-time. For each failure rate a logistic model was trained using a different ratio of training and testing data: 1/3, 2/2 and 3/1 years of data is used for training/testing.

At low failure rates there are only a few examples of failures that can be used to learn how to predict an upcoming failure; at high failure rates there are many examples of failures that can be used during training to find a good model for prediction. When using 1 year of data for training (and the remaining 3 years for testing) there will be less failures to learn from than when using 2 or 3 years for training (and the remaining 2 or 1 year(s) for testing).

5.1.1 Results for Scenario 1 – Gearbox Only

When predicting gearbox failures using only data from the two gearbox sensors (which have a relatively low level of noise), the precision and recall for different failure rates and training-testing ratios are shown in Figure 4. The results show that (even when using 3 years of training data) a failure rate of only 1% does not provide enough failure examples to predict failures; both the precision and recall are 0% meaning that there were no true positives. At 2% failure rate the precision is good (above 80%) but the recall is poor, even when using 2 or 3 years of data for training the model. At 4% failure rate the recall is 100% for all three training-testing ratios; this means that there were no false negatives, all failures would be prevented by predictive maintenance. At 8% failure rate the precision is 100% for all three training-testing rations; this means that there were no false positives; all predicted failures were resulting in actual breakdowns. At 16% failure rate both the precision and recall are good (above 80%).

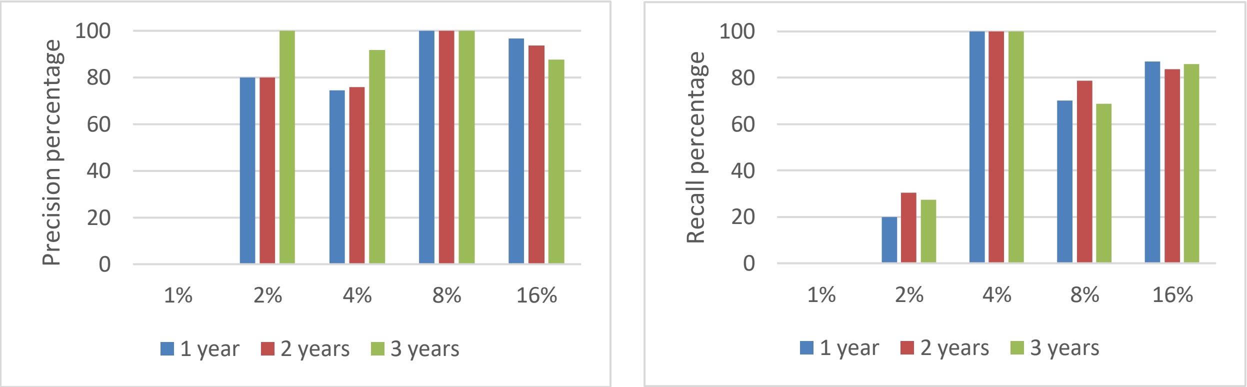

Figure 4: Precision and recall values for Scenario 1 at different failure rates and training-testing ratios.

5.1.2 Results for Scenario 2 – Hoist Motor Only

When predicting hoist motor failures using only data from the two hoist motor sensors (which have a relatively high level of noise), the precision and recall for different failure rates and training-testing ratios are shown in Figure 5.

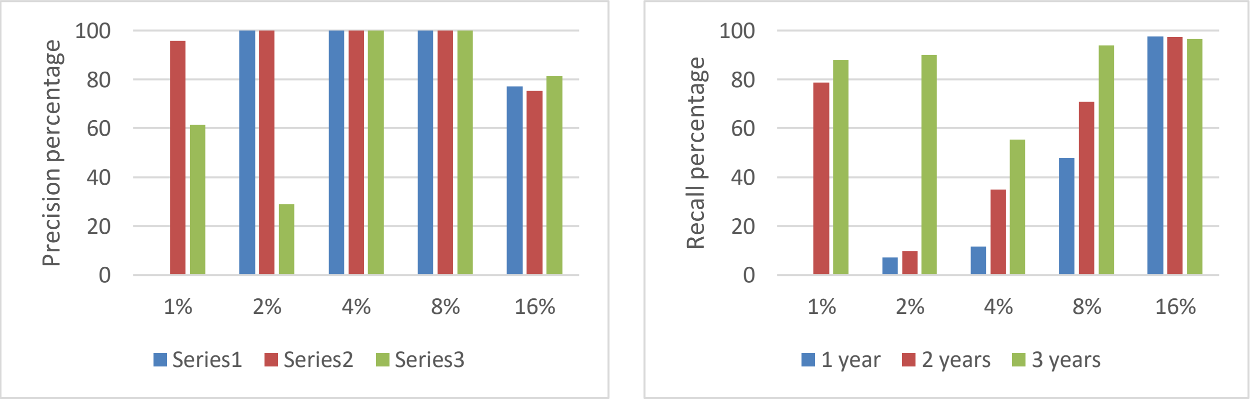

Figure 5: Precision and recall values for Scenario 2 at different failure rates and training-testing ratios.

At 1% failure rate the precision and recall is 0% when using just 1 year of data for training. With 2 years of training data both precision and recall are good, but the precision drops again when using 3 years of training data and one year of testing data. At 2 % failure rate the precision is perfect for 1 and 2 years of training, but again there is a large drop when using 3 years for training and 1 year for testing. It means that in the last year the model finds several false positives, possibly causes by high level of noise in the signals.

The precision is very good at higher percentages of failure rate, and for the recall the percentage is increasing with both the increases in failure rate and the amount of training data. The latter is what would be expected; when there are more examples of failures to learn from the quality of the prediction increases.

5.1.3 Results for Scenario 3 – Predict Gearbox Failures using All Sensor Data

When predicting gearbox failures using the data from all four sensors (where the two hoist motor sensors have no relation to the health level of the gearbox), the precision and recall for different failure rates and training-testing ratios are shown in Figure 6. Again at 1% failure rate there are too few failure examples to train the model to obtain true positives, resulting in precisions and recalls of 0%.

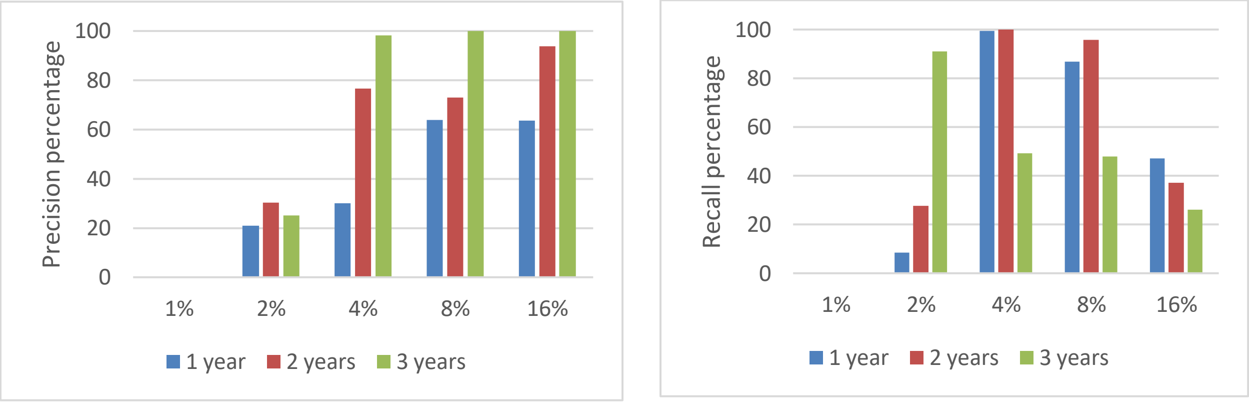

Figure 6: Precision and recall values for Scenario 3 at different failure rates and training-testing ratios.

For the precision one can see that by increasing the failure rate (number of failures per year) and by using more years for training the percentage of true positives over all positives increases. When training the model using more examples of failures there are less false positives in the testing phase.

For the recall we see that at 4% and 8% the values are good when using 1 or 2 years of training data, but the values drop when testing with only 1 year of data. And at 16% failure rates the number of false negatives has increased significantly, resulting in low recall values.

5.1.4 Results for Scenario 4 – Predicting Crane Failures using All Sensor Data

When predicting crane failures (which is either a gearbox failure or a hoist cable failure) using the data from all four sensors, the precision and recall for different failure rates and training-testing ratios are shown Figure 7. It was not possible for the genetic algorithm to find a vector *r* that results in a value higher than 0.75 for equation (1) only when either there was a gearbox failure or a hoist motor failure. This is as expected since -using the subscripts g and h to indicate the gearbox and hoist motor-related elements respectively- the multiplication in the logistics function equals

$$r * s(k) = r_g * s_{g(k)} + r_h * s_{h(k)}$$

That means that the prediction is always dependent on the health of both components. Hence for a certain health level of the gearbox quite different values will be obtained whether the health level of the hoist motor is high or low. The outcome of the failure prediction is dependent on the health level of both components, and hence it is impossible to find a vector *r* that will correctly predict the upcoming failure of one of the components independent of the health level of the other component.

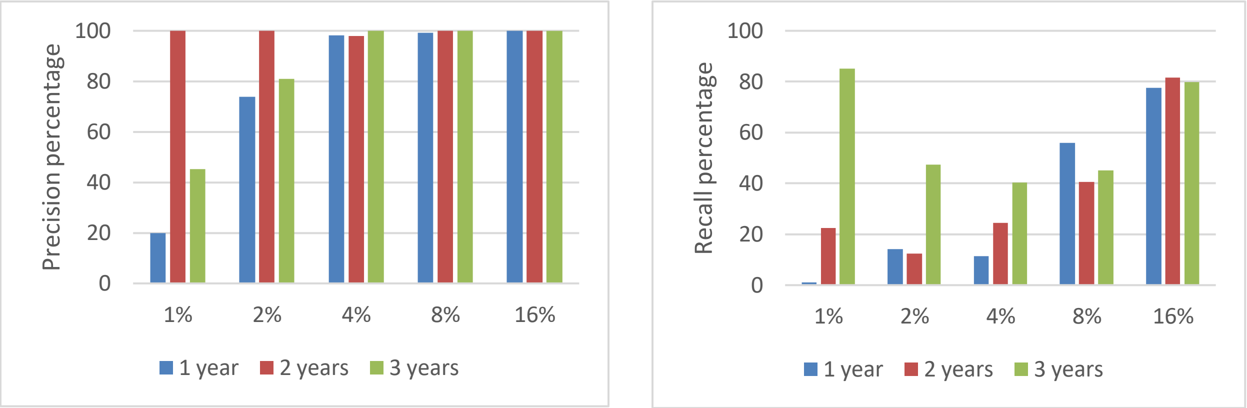

Figure 7: Precision and recall values for Scenario 4 at different failure rates and training-testing ratios.

6 CONCLUSIONS

To make an informed decision on investing in predictive maintenance for container-handling cranes, we have used simulation as a flexible, cost-effective and realistic tool to generate synthetic data. Using simulation we are able to identify the possibilities and limitations of machine learning for predicting failures of components of mobile harbor cranes.

6.1 Quality of Data

It was shown that it is important to consider failures on component-level. When trying to predict a general crane failure using all sensor information, the used model cannot distinguish well between an upcoming failure (caused by a single component) and the (not yet critical) reduction in health level of multiple components. This will cause too many false alarms for maintenance, and also does not provide an indication for which component maintenance is needed.

Focusing on predictive maintenance on component-level, we have shown that a better prediction model is created when only training the model with relevant sensor data. This means that it likely pays off to invest time with the crane experts to narrow down the sensors that would possibly have a correlation with a component's health level. Nevertheless, it can happen that some sensor information is uncorrelated to the failure and the training method should be able to handle this situation.

6.2 Quantity of Data

The tests done using the simulated data have confirmed that using more data to train the prediction model in general results in better predictions. Either having a higher fail rate (resulting in more examples of failures per year) or using data from a longer period (resulting in more examples of failures in total) for training will result in a more accurate model. But neither are desirable; we want high-quality cranes with as little failures as possible, and be able to provide predictions for maintenance well within the warranty period.

To increase the number of failures in the test data for components with a limited failure rate, we combined the data of several similar cranes. The results show that this is a reliable method to obtain good failure predictions relatively quickly. This means that we can provide a predictive maintenance model for a type of cranes after a few months of operations, and improve the quality of the predictions by updating the model periodically.

6.3 Validation Against Real Data

Since the available, real data is of low quality, validation of the results was not possible. Yet we believe that this research shows the potential of predictive maintenance based on sensor data for harbour cranes, and we conclude it is worth the investment to disclose real operational data for further research.

REFERENCES

Agerschou H., I. Dand, T. Sorensen, and T. Ernst. 2004. *Planning and Design of Ports and Maritime Terminals*. 2nd ed. London: Thomas Telford Ltd.

Alajmi A. and J. Wright. 2014. "Selecting the Most Efficient Genetic Algorithm Sets in Solving Unconstrained Building Optimization Problem". *International Journal of Sustainable Built Environment 3(1):* 18-26.

Buis M. L. 2007. "Predict and Adjust with Logistic Regression". *Sage Journals:* 18-26.

Carvalho T. P., Soares, F. A. A. M. N., Vita, R., and Francisco, P. 2019. "A Systematic Literature Review of Machine Learning Methods Applied to Predictive Maintenance". *Computers & Industrial Engineering 137*:106024

Gartner. 2021. https://www.gartner.com/document/4002912, accessed 22nd April.

Géron A. 2017. *Hands-On Machine Learning with Scikit-Learn and TensorFlow.* 1st ed. Sebastopol: O'Reilly.

Gurning, R.O.S, Simatupang, E.J., and Handani, D.W. 2021. "The Investigation of Life Cycle Costs of Mobile Harbour Crane: a Case Study in Berlian Terminal at Tanjung Perak Seaport Surabaya". In *IOP Conference Series on Earth and Environmental Science 649:012046*, September 2-3, Surabaya, Indonesia.

Holland, J.H. 1975. "Adaptation in Natural and Artificial Systems". *University of Michigan Press*, Ann Arbor, Michigan.

Katoch S., S.S. Chauhan, and V. Kumar. 2020. "A Review on Genetic Algorithm: Past, Present, and Future". *Multimedia Tools and Applications 80:* 1-36.

Kanawaday A. and A. Sane 2017. "Machine Learning for Predictive Maintenance of Industrial Machines Using IoT Sensor Data". In *Proceedings of the IEEE International Conference on Software Engineering and Service Sciences, ICSESS*, 87–90, November 24-26, Beijing, China.

Law, A. M., and W. D. Kelton. 2000. *Simulation Modeling & Analysis.* 3rd ed. New York: McGraw-Hill, Inc.

Nikolenko, S. 2021. *Synthetic Data for Deep Learning*. 1st ed. Spinger Optimization and Its Application: Springer, San Francisco, CA, USA.

Peng C-Y J., K. L. Lee, and G. M. Ingersoll 2002. "An Introduction to Logistic Regression Analysis and Reporting". *The Journal of Educational Research 96(1):* 3-14

Port Technology. 2015. The Internet of Things and the Container Port. https://www.porttechnology.org/news/internet_of_things_and_the_container_port/, accessed 22nd April.

Szpytko, J., and Salgado Duarte, Y. 2021. "A Digital Twins Concept Model for Integrated Maintenance: A Case Study for Crane Operation". *Journal of Intelligent Manufacturing 31*:1863-1881

Tian Z., D. Lin, and B. Wu 2012. "Condition Based Maintenance Optimization Considering Multiple Objectives". *Journal of Intelligent Manufacturing 23*: 333-340

Tremblay J., T. To, B. Sundaralingam, Y. Xiang, D. Fox, and S. T. Birchfield. 2018. "Deep Object Pose Estimation for Semantic Robotic Grasping of Household Objects". In *Proceedings of the 2nd Conference on Robot Learning*, edited by A. Billard, A. Dragan, J. Peters, and J. Morimoto, 306-316, October 29-31, Zurich, Switzerland.

Van Ham J.C. and J.C. Rijsenbrij. 2012. *Development of Containerization*. Amsterdam: IOS Press.

AUTHOR BIOGRAPHIES

MERNOUT BURGER is a Functional Expert at TBA Group. He earned his Ph.D. in the Department of Engineering Cybernetics at the Norwegian University of Science and Technology. His research interests include (non)linear control and estimation, artificial intelligence and machine learning, operations research, and the automation and optimization of container terminals. His e-mail address is mernout.burger@tba.group.

CSABA A. BOER is a Chief Product Owner at TBA Group. He is responsible for several products within the organization. He holds a Ph.D. in Computer Science and Logistics from Erasmus University Rotterdam on the topic of distributed simulation systems. His research interests include distributed virtual environments and simulation, machine learning, port simulation and emulation. His e-mail address is csaba.boer@tba.group.

EDWIN STRAUB is a Seniour Developer within TBA Group. He is responsible for several products within the organization. He holds a B.Sc. in Computer Science from Babes Bolyai University, Cluj Napoca. His research interests include equipment routing, decking strategies, and machine learning. His e-mail address is edwin.straub@tba.group.

YVO A. SAANEN is Managing Director and founder (1996) of TBA Group. He holds an M.Sc. in Systems Engineering and a Ph.D. on the design and simulation of robotized container terminals, both from Delft University of Technology. In addition, Yvo Saanen is a lecturer at Erasmus University in Maritime Economics and Logistics. His e-mail address is yvo.saanen@tba.group.