This is an Author Accepted Manuscript version of the following chapter: S. Roysson, T. A. Sitompul, and R. Lindell, Using Artificial Neural Network to Provide Realistic Lifting Capacity in the Mobile Crane Simulation, published in Proceedings of the 22nd Engineering Applications of Neural Networks Conference, edited by L. Iliadis, J. Macintyre, C. Jayne, and E. Pimenidis, 2021, Springer reproduced with permission of Springer. The final authenticated version is available online at: https://doi.org/10.1007/978-3-030-80568-5_37.

Using Artificial Neural Network to Provide Realistic Lifting Capacity in the Mobile Crane Simulation

Simon Roysson[1], Taufik Akbar Sitompul[1,2][0000-0002-1930-4181], and Rikard Lindell[1][0000-0003-3163-6039]

[1] School of Innovation, Design, and Engineering, Mälardalen University, Västerås, Sweden
[2] CrossControl AB, Västerås, Sweden
s.roysson@bahnhof.se, {taufik.akbar.sitompul, rikard.lindell}@mdh.se

Abstract. Simulations are often used for training novice operators to avoid accidents, while they are still polishing their skills. To ensure the experience gained in the simulation be applicable in real-world scenarios, the simulation has to be made as realistic as possible. This paper investigated how to make the lifting capacity of a virtual mobile crane behave similarly like its real counterpart. We initially planned to use information from the load charts, which document how the lifting capacity of a mobile crane works, but the data in the load charts were very limited. To mitigate this issue, we trained an artificial neural network (ANN) using 90% of random data from two official load charts of a real mobile crane. The trained model could predict the lifting capacity based on the real-time states of the boom length, the load radius, and the counterweight of the virtual mobile crane. To evaluate the accuracy of the ANN predictions, we conducted a real-time experiment inside the simulation, where we compared the lifting capacity predicted by the ANN and the remaining 10% of the data from the load charts. The results showed that the ANN could predict the lifting capacity with small deviation rates. The deviation rates also had no significant impact on the lifting capacity, except when both boom length and load radius were approaching their maximum states. Therefore, the predicted lifting capacity generated by the ANN could be assumed to be close enough to the values in the load charts.

Keywords: Neural network · Virtual reality · Mobile crane · Lifting capacity · Realism

1 Introduction

Cranes are typically used for lifting and moving objects from one place to another. Cranes come in different types and sizes, which range from bigger fixed-position cranes that should be assembled first, such as tower cranes, to smaller cranes that can be mobilized immediately, also known as 'mobile cranes'. Although modern cranes are increasingly equipped with information systems that are designed to assist operators performing their work, operators' knowledge and experience still play a vital role for ensuring safe lifting operations [18]. Due to the dangerous nature of crane operations, it is common for novice operators to train using simulations [10]. Therefore, any accident could be avoided, while novice operators are still polishing their skills. The advent of virtual reality (VR) offers opportunities to provide immersive training to novice operators, which could enhance their learning experience [26]. It is also essential that the simulation is made as realistic as possible, thus the learning experience from the simulation is transferable to real-world scenarios [12].

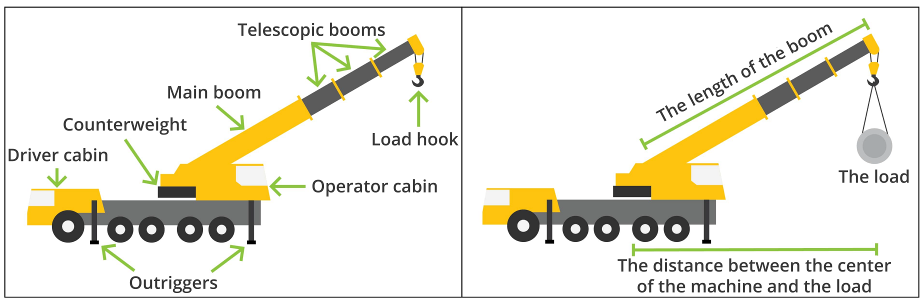

Fig. 1. The left image shows some parts of mobile cranes that are relevant to know in this paper. The right image shows the concepts of boom length and load radius in mobile crane operations.

In this paper, we investigated the aspect of realism with focus on the real-time lifting capacity of a virtual mobile crane in a simulator built in the Unity game engine[3], which is a common platform for making VR applications[4]. Mobile cranes were chosen as the use case, as they contributed the majority of crane-related accidents [15]. The real-time lifting capacity refers to the maximum weight that the crane can currently lift based on two dynamic parameters: (1) how far the boom is currently extended (the boom length) and (2) the current distance between the lifted load and the crane's center (the load radius). The left image in Figure 1 illustrates the parts of mobile cranes that are relevant to know in this paper, while the right image in Figure 1 depicts the concepts of boom length and load radius in mobile crane operations. The lifting capacity is decreasing when the boom length and the load radius are increasing, and vice versa. The lifting capacity in different states of boom length and load radius are usually documented in a 'load chart' provided by crane manufacturers (see [20, pp. 23–47] for some examples of the load chart). Mobile crane operators are strongly advised to refer to the load chart before any lifting operation, since exceeding the limit will make the mobile crane lose its balance. Operating cranes beyond

[3] https://unity.com/
[4] https://unity3d.com/unity/features/multiplatform/vr-ar

permissible operational limits is also one of the causal factors of crane-related accidents [21].

The research question in this paper is "How can we make the virtual mobile crane in the Unity game engine behave similarly to a real mobile crane in terms of the real-time lifting capacity?". For example, if a specific mobile crane model can only lift an object of up to 10 tonnes in certain states of boom length and load radius, then the virtual mobile crane should not be able to lift an object heavier than 10 tonnes in the corresponding states. In other words, if the real-time lifting capacity is 10 tonnes and the virtual mobile crane is lifting an 11-tonne object, then the virtual mobile crane should collapse. It is also important that the virtual mobile crane should not collapse in arbitrary conditions, as novice operators would be clueless whether they perform the correct thing or not. We expect that there would be two possible benefits of having this mechanism: (1) improving the realistic aspect of the simulation and (2) providing relevant learning experience to novice operators.

We used two official load charts [20, pp. 23 and 29] from a crane manufacturer as a source of information on how the lifting capacity of a mobile crane should work, and thus we planned to replicate the mechanism in the Unity game engine. However, achieving such objective was later found to be difficult, since the load charts document the lifting capacity in some states of boom length and load radius only. This would make the simulation less realistic, since the lifting capacity would jump when the virtual mobile crane were moving between the documented states in the load charts. In a real mobile crane, the lifting capacity gradually increases or decreases as the crane progressively moves. In addition, we also could not find reliable sources of information regarding the formula behind the load charts. To solve this problem, we used 90% of random data from the load charts to train an Artificial Neural Network (ANN) to predict the lifting capacity based on the real-time states of boom length and load radius of the virtual mobile crane. With this approach, we could predict the lifting capacity in any states of boom length and load radius using the ANN, although the information documented in the load charts was limited. We then compared the lifting capacity between the ANN predictions and the remaining 10% of the data in the load charts to evaluate the accuracy of the ANN.

2 Related Work

This section is divided into two parts. The first part describes how realism in crane simulations has been investigated, while the second part describes how ANN has been used to support realism in VR applications.

2.1 Realism in Crane Simulations

The realistic aspects of crane simulations have been investigated since two decades ago with various focuses, such as simulation setup [10], depth perception [12], resemblance of scenarios and virtual objects with their real counterparts [7,13], physics [3], and immersion [6,26].

Huang and Gau [10] focused on providing a realistic simulation setup that consisted of panels and controls that could be found in the mobile crane's cabin, as well as a motion platform that could mimic the cabin's movement. Despite having a realistic setup, their users commented that the developed simulation was deemed to be less realistic due to three factors: (1) the scenario was not the same as the real-world scenario, (2) the lack of depth perception, and (3) the visual quality of the simulation. The lack of depth perception in crane simulations was then addressed by Juang et al. [12] by providing stereoscopic view, where individual images were generated to each eye, and kinesthetic vision, which simulated the operator's head movement. These additional features improved operators' confidence due to the realistic and intuitive behavior of the depth perception.

There were two studies that focused on the realistic aspect of scenarios and virtual objects inside crane simulations. Fang et al. [7] focused on making realistic scenarios and virtual objects that closely resembled their real counterparts by integrating data from Building Information Modelling (BIM) and real-time location tracking inside the simulation. Although they aimed to make the virtual cranes as realistic as possible, their study was limited to the visual appearance of the virtual cranes. Kan et al. [13] also aimed to make realistic scenarios by taking into account causal factors of crane-related accidents and safety guidelines when designing their simulation. Their simulation could be used to generate realistic and dynamic crane-operating scenarios, which could expose novice operators to possible hazardous situations that may occur in real operations.

Due to the immersion that the technology could provide, the use of VR for crane simulations has also been investigated in two studies. Dong et al. [6] developed a VR overhead crane simulation and Patrão and Menezes [26] developed a VR tower crane simulation. While VR could provide immersion to the user, the immersion could be broken if there is a mismatch between what the user is expecting and what the simulation is generating, due to the additional attention to detail that the immersion could provide compared to a non-VR simulation [29].

Contrast to the studies mentioned above, Chi and Kang [3] developed a simulation that focused on the physical aspect of two collaborating cranes, which could be used for planning purposes. Using their simulation, problems could be discovered before carrying out the real operation. The crane's physics was split into kinematic and dynamic rigid bodies. The cabin and the boom were made kinematic, which means they are objects without mass and force that could be manipulated by transformation matrices. The cable and the hook were made dynamic rigid bodies, which mean they were affected by mass and force. This approach made it possible to simulate some physical behaviours, such as object collisions and swinging cables. Although they considered the crane's physics in the simulation to be realistic enough, the technical evaluation that was carried out did not evaluate how realistic the crane's physics was.

Looking at the studies that specifically mentioned which tools that were used for developing their simulations, none of them explicitly stated that the tools being used as the main limitation. Chi and Kang [3] used Open dynamics engine for simulating physics and OpenGL for rendering. Dong et al. [6] also used two different tools: Bullet Physics Library was used for simulating physics and OpenAR for rendering. Juang et al. [12] used a tool called SimCrane 3D+ for both rendering and simulating physics. Kan et al.[13] used the Unity game engine as the single tool for both rendering and simulating physics.

Our approach has some similarities with the prior work. Firstly, related to the study that investigated the physics of the virtual crane [3], this study also focused on how the states of boom length and load radius affect the lifting capacity. Secondly, identical to Kan et al. [13], we also used the Unity game engine for our simulation. Furthermore, since almost of the mentioned prior studies did not use a game engine to implement their simulations, this study could further demonstrate how a game engine, such as Unity game engine, could be utilized for crane-related research.

2.2 Using Artificial Neural Network to Support Realism in Virtual Reality

The use of ANNs to support realism in VR applications could be tracked back to Caudell et al. [2], where they used the ANN to generate head silhouettes in teleconference VR applications. The ANN was trained using 3D data and videos of a person talking and making various facial expressions, although only in the form of silhouettes. Olszewski et al. [25] used the ANN to predict how the avatar's mouth should be animated based on the user's mouth movement. The ANN was trained using videos of mouth movements from ten people. Slightly different from the previous examples, Seele et al. [30] used the ANN to guide the eye movement of the avatar. The ANN was trained using eye tracking data of a person who looked at different visual stimuli.

There were four studies that used ANNs for generating realistic feedback based on what the user does. Specifically for VR port simulations, García-Fernández et al. [8] used the ANN to generate realistic impact of two colliding virtual containers based on their positions and the point of collision. They trained the ANN using collision models on different points of collision. In the context of VR welding simulations, Yang et al. [34] used the ANN to predict the form of virtual welding beads based on the user's performance. Their ANN was trained with welding data from both simulation and practical experiments. In the context of VR tennis games, Hambli et al. [9] used the ANN to predict how much force feedback should be given to the user's hand based on the deformation of the virtual racket, which collided with the virtual tennis ball. The ANN was trained using object deformation data generated from a finite element simulation. Wu et al. [33] used the ANN to generate haptic feedback when typing on a virtual keyboard. Data from vibration sensors on a physical keyboard were used train the ANN.

Using ANNs to facilitate interaction techniques in VR has also been investigated in three studies. In the context of VR drum games, Rosa-Pujazón et al. [27] used the ANN to determine whether the user's fast hand movements matched with specified drum-hitting gestures. The ANN was trained with data of drum-hitting motion from three people. Specifically for VR sword-fighting games, Dehesa et al. [5] used the ANN to check whether the user's movement could match with the predefined sword-fighting movements. The ANN was trained using motion-capture data from one person. Kang et al. [14] also used the ANN for a similar purpose, for example, when the user positioned his/her hands like holding a sword, the avatar would then be equipped with a virtual sword. They trained the ANN with hand gesture images.

The last examples used ANNs to render realistic virtual objects and environments. Iwahori et al. [11] used two ANNs to generate virtual objects with color reflection. The first ANN collected the 3D object of target objects and their reflection factors, while the the second ANN estimated how virtual objects should be colored. The rendering results showed that the generated virtual objects looked very similar to the image of corresponding real objects. Lastly, Tang and Xiao [32] used the ANN to mimic human vision, where some areas of the virtual environment were blurred based on where the user's gaze was. The ANN was trained using a schematic eye model.

3 Mobile Crane Simulation

In this study, we used an available mobile crane simulation that was purchased from the Unity Assets Store. The simulation was then modified, so that we could implement the ANN and conduct the experiment. This section provides a brief description about the initial simulation, the modifications that we made on the simulation, and how we designed the ANN.

3.1 Initial Simulation

To speed up the implementation, we purchased an available mobile crane simulation from the Unity Asset Store[5], thus we did not have to develop everything from scratch. The simulation included two fully-functioning mobile cranes, but the one that we used in this study was the one named 'HTR1045' in the downloaded project. Although not the exact replica, the virtual mobile crane had a very close resemblance with the Liebherr's LTC 1050-3.1 compact mobile crane [20]. The virtual environment was made of a flat ground without additional weather conditions. We kept this initial setup, since a mobile crane should be operated on the solid flat ground due to safety recommendations [28,24]. As described in Section 1, the lifting capacity is based on two dynamic factors: the boom length and the load radius. However, the Liebherr's LTC 1050-3.1 crane could be used with two counterweight configurations: 4.8 tonnes and 6.5 tonnes [20]. The counterweight also plays an important role here, since heavier counterweights mean that the crane could lift heavier objects, and vice versa. Another factor that

[5] https://assetstore.unity.com/packages/3d/vehicles/land/crane-simulator-v-2-designer-150285

could affect the lifting capacity is wind [19], but we excluded this factor in this study, since the information in the load charts is applicable only if the wind speed does not exceed 9 m/s. Therefore, the factors in the simulation that affected the lifting capacity were the counterweight being used, the boom length, and the load radius.

3.2 Modified Simulation

This subsection describes the modifications that we made on the initial simulation, which enabled us to train the ANN and conduct the experiment.

Establishing a Metric Measurement. We had to establish a constant metric system in the simulation in order to conduct the experiment, since the load charts in [20, pp. 23 and 29] were documented in meters. From the load charts, we found that the maximum boom length for the LTC 1050-3 crane is 36 meters. We defined a new conversion rate by dividing the virtual mobile crane's maximum boom length in the Unity coordinates with the LTC 1050-3 crane's maximum boom length, which gave us a conversion rate of 0.88264784. By multiplying the Unity coordinates with the new conversion rate, we could establish a measurement that constantly defined the length of one meter in the simulation. This calculation was always used for measuring boom length and load radius of the virtual mobile crane.

Implementing the Artificial Neural Network. We used a python ANN Library called Keras[6] to train the ANN model. We also used a Unity plugin called Noedify[7], which allowed us to import the trained model from Keras into the Unity game engine. In order to do so, we wrote two scripts: (1) a python script for training the ANN using Keras and (2) a Noedify script for exporting the trained ANN to a text file. The scripts read a specified text file that contained the training data, trained the model using the provided data, and then exported the model to another specified text file. See the illustration of the process in Figure 2.

The ANN structure in this study was designed after a trial-and-error process, where we tried different options, such as different number of nodes in the hidden layer, different number of hidden layers, and different activation functions, to see which configuration that provided the highest accuracy. We started the process with a three-layer structure: one input layer, one hidden layer, and one output layer. During the trial-and-error process, the accuracy of the ANN was improving when the number of nodes within the hidden layer was increased up to 100. When the hidden layer went beyond 100 nodes, the accuracy stopped improving. In an attempt to improve the accuracy further, another hidden layer was added, but

[6] https://keras.io/

[7] https://assetstore.unity.com/packages/tools/ai/noedify-easy-neural-networks-161940

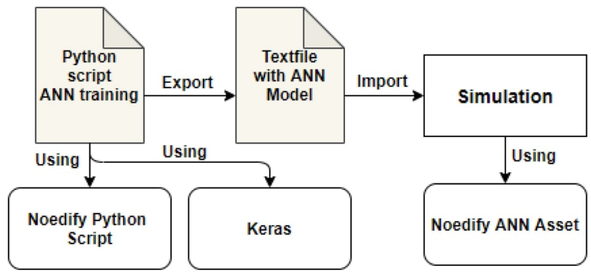

Fig. 2. The process of exporting and importing the trained ANN model from Keras to the simulation.

the accuracy was decreased. We also tested other activation functions, such as the Sigmoid function [22], but the accuracy was lower than when the Exponential Linear Unit (ELU) function [4] was used.

The final ANN structure was a three-layer structure consisted of three input nodes (counterweight being used, current boom length, and current load radius), 100 nodes in the hidden layer that used the ELU function, and one output node for the predicted lifting capacity (see Figure 3). We defined the input nodes this way, since the load charts documented the lifting capacity based on these three factors. We finally trained the model with 6000 iterations and an optimizer called Adam [16] with a learning rate of 0.001. The process was almost completely automated, as we only had to provide the training data, which contained 90% of the randomly selected data from the load charts of both 4.8-tonne and 6.5-tonne counterweights [20, pp. 23 and 29].

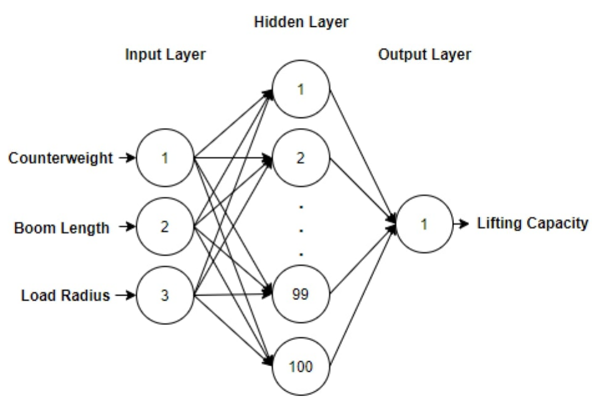

Fig. 3. The structure of the ANN that we used to predict the real-time lifting capacity. The prediction was made based on which counterweight configuration being used and the current states of boom length and load radius.

Implementing a Logging Feature. We also added a logging feature that could be used for recording four kinds of information from the virtual mobile crane: (1) the boom length, (2) the load radius, (3) the predicted lifting capacity from the ANN, and (4) the corresponding lifting capacity based on the load charts. This approach enabled us to directly compare the lifting capacity based on the ANN predictions and the load charts. The type of counterweight being used was also written as the file name when exporting the data into a file. We designed the logging feature to be manually controlled by the user in the run-time, where the user operated the virtual mobile crane into specific states of boom length and load radius, and then pressed a button on the keyboard in order to save the data. When the user finished logging everything, another button on the keyboard was pressed to store all the saved data into a file with CSV format.

4 Experimental Procedure

Before conducting the experiment, we randomly divided the data from the load charts [20, pp. 23 and 29] into the training dataset and the testing dataset with a ratio of 90%:10% by using another python script. This ratio was chosen, as the data from the load charts were very limited. Since the experiment was done inside the simulation, we had to know which data were assigned into the testing dataset. The script for dividing the data also exported a text file that contained the data inside the testing dataset. We then trained the ANN described in Section 3.2 with the training dataset. After that, we imported the trained model into the simulation.

We started the experiment by running the simulation, selecting which counterweight configuration that would be used, and then operating the virtual mobile crane into the lifting mode. We then operated the virtual mobile crane, so that the boom length and the load radius were moved to the states of boom length and load radius in the testing dataset. Once we were sure that the virtual mobile crane has been moved to the designated state of boom length and load radius, we pressed a button on the keyboard to save the data, and then the graphical user interface (GUI) in the simulation would show the next state to be logged. We repeated this process until all states in the testing dataset have been logged. After that, we pressed another button on the keyboard to export the logged data into a CSV file. The whole process was repeated twice, since there were two different counterweights (4.8 tonnes and 6.5 tonnes) that could be used by the virtual mobile crane.

5 Results

To calculate the overall prediction accuracy of the ANN, we used Mean Absolute Error (MAE) and Root Mean Square Error (RMSE), which are common metrics for measuring the accuracy of ANNs [1]. We took all the values in the testing dataset and the prediction values from the ANN, and then calculated the metrics. Note that the load charts for both 4.8-tonne and 6.5-tonne counterweights document the same states of boom length and load radius, and thus the predictions were made based on the same states. The MAE of the ANN predictions was 0.16 tonnes, while the RMSE was 0.24 tonnes.

Although the MAE and the RMSE could indicate the overall accuracy of the ANN predictions, it is also interesting to see how the predictions varied among different states of the virtual mobile crane. We took the values from the testing dataset and the ANN predictions, and then calculated the mean deviation for each load radius. The same calculation could also be done based on each boom length, but we decided to calculate based on the load radius, since it was more documented in the load charts. Figure 4 illustrates that the deviations seem to be leaning towards the negative side, except when the load radii were small. This means that the predicted lifting capacity generally tends to be slightly lower than the values in the load charts. This suggests that, in the current simulation, the virtual mobile crane could not lift heavier objects than what the real counterpart could lift. In practice, lifting something heavier than the lifting capacity would make the mobile crane collapse. Therefore, from a safety point of view, it is more favorable if the ANN predictions are lower, if the predicted values could not be made precisely the same as the ones in the load charts.

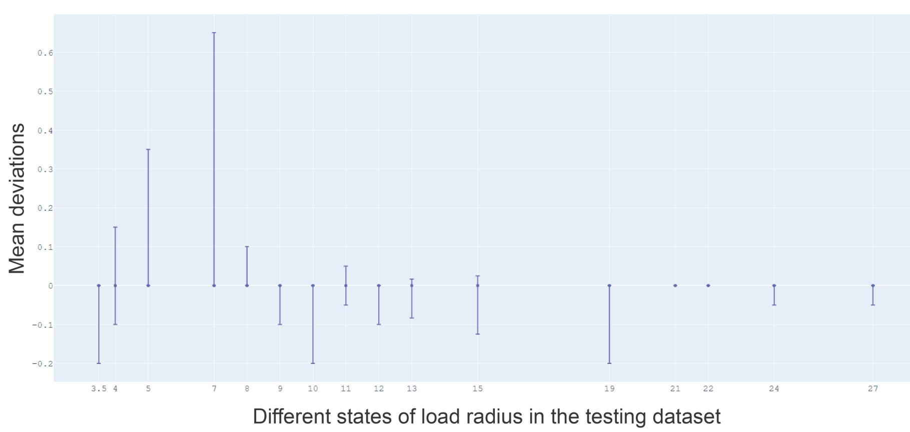

Fig. 4. The mean deviations of the predicted lifting capacity based on various states of load radius in the testing dataset. The x-axis shows whether the deviations were higher or lower than the values from the load charts. The y-axis shows the mean deviations of the predicted lifting capacity for each state of load radius.

6 Discussion

The presented results in Section 5 suggest that the lifting capacity of the virtual mobile crane was quite close to the real crane, as the MAE was 0.16 tonnes and the RMSE was 0.24 tonnes. Although the overall deviations in both metrics were rather small, they could affect the lifting capacity differently depending on the states of the virtual mobile crane. To put things into perspective, we present some lifting capacity values from the load charts [20, pp. 23 and 29] in Table 1 and show how the MAE value of 0.16 tonnes would influence the lifting capacity in different states of boom length and load radius. For simplification purposes, we only used the MAE value in Table 1, since RMSE squared the deviations before they were averaged.

Table 1. The values inside the brackets show how the deviation based on the MAE value would influence the lifting capacity (in percentages) in some states of boom length and load radius.The white rows show some of the lifting capacities (in tonnes) in the 4.8-tonne load chart [20, pp. 29], while the grey rows show some of the lifting capacities (in tonnes) in the 6.5-tonne load chart [20, pp. 23].

Load radius	Boom length (in meters)					
(in meters)	8.2	13.8	19.3	24.9	30.4	36
3.0	44.4 [0.36%]	40.0 [0.4%]	30.0 [0.53%]	20.4 [0.78%]		
3.0	44.4 [0.36%]	40.0 [0.4%]	30.0 [0.53%]	20.4 [0.78%]		
9.0		12.5 [1.28%]	12.5 [1.28%]	11.9 [1.34%]	10.7 [1.49%]	7.8 [2.05%]
9.0		13.6 [1.17%]	13.6 [1.17%]	12.5 [1.28%]	11.2 [1.42%]	7.8 [2.05%]
15.0			5.6 [2.8%]	5.6 [2.8%]	5.5 [2.90%]	5.2 [3.07%]
15.0			6.3 [2.53%]	6.2 [2.58%]	6.1 [2.62%]	5.8 [2.75%]
21.0				3.2 [5.0%]	3.1 [5.16%]	2.8 [5.71%]
21.0				3.6 [4.44%]	3.5 [4.57%]	3.2 [5.0%]
27.0					1.9 [8.42%]	1.6 [10.0%]
27.0					2.2 [7.27%]	1.9 [8.42%]
33.0						0.8 [20.0%]
33.0						1.1 [14.54%]

As shown in Table 1, the deviation rate based on the MAE value would affect the lifting capacity differently. For example, when the boom length is 8.2 meters and the load radius is 3 meters, the deviation rate would only influence 0.36% of the lifting capacity. The influence is increasing as the boom length and the load radius are increasing, even though the influence is mostly negligible. Only when the boom length and the load radius are approaching their maximum states, the overall deviation rate would influence more than 10% of the lifting capacity (see the bottom-right numbers in Table 1). Taking this into account, it could be assumed that the ANN predictions were realistic enough for most of the states, except when both boom length and load radius were approaching their maximum states.

Using this approach, we could make the virtual mobile crane to collapse if the user lifts something heavier than the predicted lifting capacity in any state of boom length and load radius. Another use of the predicted lifting capacity is that we could also simulate the Load Moment Indicator (LMI), which is a supportive system inside the crane's cabin that shows similar information as what is documented in the load chart [23]. Instead of presenting arbitrary visual information to the operator, the presented virtual information would resemble what the real LMI would show. For example, Kvalberg [17] investigated the use of transparent displays for presenting the lifting capacity of off-shore cranes, but the information presented had to be manually inserted by the user. Therefore, researchers could also use the simulation modified in this study as a tool to investigate new visualization approaches in mobile cranes. Due to the hazardous nature of heavy machinery operations, it is also common to evaluate new technologies in simulations before they are installed in real machines [31].

7 Limitations and Future Work

Although there are many machine learning algorithms that could be used for predicting missing data, this study was limited to what was compatible with the Unity game engine. To the best of our knowledge, Unity game engine so far supports reinforcement learning[8] and neural network[9] only. Therefore, we could not benchmark the ANN against other prediction algorithms in this study. There are three studies presented in Section 2.2 that explicitly used the Unity game engine [33,30,14] and the proposed ANNs were also not benchmarked against other prediction algorithms. However, the limitations that prevented them from doing so were not explicitly written in the papers.

As described in Section 3.2, we mostly used the overall accuracy when we were exploring which ANN configuration that provided the highest accuracy. Therefore, during the trial-and-error process, we were not aware that the deviation rates were slightly higher when the boom length and the load radius were short. This was probably because the differences between one documented lifting capacity and the next ones were high in these conditions, and thus leading to slightly higher deviation rates. Nevertheless. it would be interesting to investigate how the ANN accuracy in general and in specific states could be improved by further exploring different ANN configurations.

In this study, we used a virtual mobile crane that had a close resemblance to the LTC 1050-3.1 crane, and thus we used the official load charts for this mobile crane model. To further investigate the generalizability of the proposed ANN, it would also be interesting to use different mobile crane models and their corresponding official load charts. Therefore, we could further investigate to what extent the proposed ANN would also be applicable for different mobile crane models.

[8] https://unity.com/products/machine-learning-agents

[9] https://docs.unity3d.com/Packages/com.unity.barracuda@1.0/manual/index.html

8 Conclusion

This paper presents another use of ANN to support realism in the virtual environment, where we investigated how a virtual mobile crane could have a realistic lifting capacity behavior that mimics its real counterpart. Although the official load charts contained relevant information on how the lifting capacity of a mobile crane should behave, the available information was limited. We trained the ANN using the training dataset from the load charts, to predict the lifting capacity of the virtual mobile crane. This approach enabled us to predict the lifting capacity in any states of boom length and load radius. We then compared the ANN predictions and the testing dataset from the load charts to evaluate the ANN accuracy and we found that the deviation rates measured in Mean Absolute Error and Root Mean Squared Error were relatively small. We also found that the prediction deviations did not provide significant impacts on the lifting capacity for most of the states, except when the boom length and the load radius were approaching their maximum states. Therefore, we could assume that the predicted lifting capacity made by the ANN to be close enough to the load charts.

Acknowledgements

This research has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 764951.

References

1. Botchkarev, A.: A new typology design of performance metrics to measure errors in machine learning regression algorithms. Interdisciplinary Journal of Information, Knowledge, and Management **14**, 45–76 (2019). https://doi.org/10.28945/4184
2. Caudell, T.P., Janin, A.L., Johnson, S.K.: Neural modeling of face animation for telecommuting in virtual reality. In: Proceedings of IEEE Virtual Reality Annual International Symposium. pp. 478–485. IEEE (1993). https://doi.org/10.1109/VRAIS.1993.380741
3. Chi, H.L., Kang, S.C.: A physics-based simulation approach for cooperative erection activities. Automation in Construction **19**(6), 750 – 761 (2010). https://doi.org/10.1016/j.autcon.2010.03.004
4. Clevert, D.A., Unterthiner, T., Hochreiter, S.: Fast and accurate deep network learning by exponential linear units (ELUs). arXiv e-prints (2015)
5. Dehesa, J., Vidler, A., Lutteroth, C., Padget, J.: Touché: Data-driven interactive sword fighting in virtual reality. In: Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems. p. 1–14. CHI '20, ACM, New York, NY, USA (2020). https://doi.org/10.1145/3313831.3376714
6. Dong, H., Xu, G., Chen, D.: Research on overhead crane training system and its construction based on virtual reality. In: 2010 International Conference on Artificial Intelligence and Education (ICAIE). pp. 206–209. IEEE (2010). https://doi.org/10.1109/ICAIE.2010.5641436

7. Fang, Y., Teizer, J., Marks, E.: A framework for developing an as-built virtual environment to advance training of crane operators. In: Construction Research Congress 2014. pp. 31–40. ASCE (2014). https://doi.org/10.1061/9780784413517.004
8. García-Fernández, I., Martín-Guerrero, J.D., Pla-Castells, M., Soria-Olivas, E., Martínez-Durá, R.J., Muñoz-Marí, J.: Crane collision modelling using a neural network approach. Expert Systems with Applications **27**(3), 341–348 (2004). https://doi.org/10.1016/j.eswa.2004.05.002
9. Hambli, R., Chamekh, A., Bel Hadj Salah, H.: Real-time deformation of structure using finite element and neural networks in virtual reality applications. Finite Elements in Analysis and Design **42**(11), 985 – 991 (2006). https://doi.org/10.1016/j.finel.2006.03.008
10. Huang, J.Y., Gau, C.Y.: Modelling and designing a low-cost high-fidelity mobile crane simulator. International Journal of Human-Computer Studies **58**(2), 151 – 176 (2003). https://doi.org/10.1016/S1071-5819(02)00129-5
11. Iwahori, Y., Kawanaka, H., Woodham, R.J., Funahashi, K.: Neural network based modeling and color rendering for mixed reality. In: Proceedings of the 17th International Conference on Pattern Recognition (ICPR 2004). pp. 198–201 Vol.3. IEEE (2004). https://doi.org/10.1109/ICPR.2004.1334502
12. Juang, J., Hung, W., Kang, S.: SimCrane 3D+: A crane simulator with kinesthetic and stereoscopic vision. Advanced Engineering Informatics **27**(4), 506 – 518 (2013). https://doi.org/10.1016/j.aei.2013.05.002
13. Kan, C., Zhang, P., Fang, Y., Anumba, C., Messner, J.: A taxonomic analysis of mobile crane fatalities for CPS-based simulation. In: 17th International Conference on Computing in Civil and Building Engineering. pp. 1–8. Ex Ordo (2018)
14. Kang, T., Chae, M., Seo, E., Kim, M., Kim, J.: DeepHandsVR: Hand interface using deep learning in immersive virtual reality. Electronics **9**(11) (2020). https://doi.org/10.3390/electronics9111863
15. King, R.A.: Analysis of crane and lifting accidents in North America from 2004 to 2010. Master's thesis, Massachusetts Institute of Technology (June 2012), http://hdl.handle.net/1721.1/73792
16. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv e-prints (2014)
17. Kvalberg, J.L.: Head-up display in driller and crane cabin. Master's thesis, Norwegian University of Science and Technology (June 2010), http://hdl.handle.net/11250/260185
18. Lei, Z., Taghaddos, H., Hermann, U., Al-Hussein, M.: A methodology for mobile crane lift path checking in heavy industrial projects. Automation in Construction **31**, 41 – 53 (2013). https://doi.org/10.1016/j.autcon.2012.11.042
19. Liebherr: Influence of wind on crane operation (2017), https://www.liebherr.com/shared/media/mobile-and-crawler-cranes/brochures/\wind-influences/liebherr-influence-of-wind-p403-e04-2017.pdf
20. Liebherr: Technical Data - Compact Crane LTC 1050-3.1 (2020), https://www.liebherr.com/external/products/products-assets/916432/liebherr-260-ltc-1050-3-1-td-260-01-defisr07-2020.pdf
21. Milazzo, M.F., Ancione, G., Brkic, V.S., Valis, D.: Investigation of crane operation safety by analysing main accident causes. In: Walls, L., Revie, M., Bedford, T. (eds.) Risk, Reliability and Safety: Innovating Theory and Practice, pp. 74–80. Taylor and Francis, London, UK (2017)
22. Mitchell, T.: Machine Learning, chap. 4 - Artificial Neural Networks, pp. 81–128. McGraw-Hill Education, New York, NY, USA (1997)

23. Neitzel, R.L., Seixas, N.S., Ren, K.K.: A review of crane safety in the construction industry. Applied Occupational and Environmental Hygiene **16**(12), 1106–1117 (2001). https://doi.org/10.1080/10473220127411
24. Occupational Safety and Health Branch: Code of practice for safe use of mobile cranes (September 2017), https://www.labour.gov.hk/eng/public/os/B/MobileCrane.pdf
25. Olszewski, K., Lim, J.J., Saito, S., Li, H.: High-fidelity facial and speech animation for VR HMDs. ACM Transactions on Graphics **35**(6), 221:1–221:14 (2016). https://doi.org/10.1145/2980179.2980252
26. Patrão, B., Menezes, P.: An immersive system for the training of tower crane operators. In: 2nd Experiment@ International Conference. pp. 158–159. exp.at'13, IEEE (2013). https://doi.org/10.1109/ExpAt.2013.6703052
27. Rosa-Pujazón, A., Barbancho, I., Tardón, L.J., Barbancho, A.M.: Fast-gesture recognition and classification using Kinect: an application for a virtual reality drumkit. Multimedia Tools and Applications **75**, 8137–8164 (2016). https://doi.org/10.1007/s11042-015-2729-8
28. Safe Work Australia: General guide for cranes (December 2015), www.safeworkaustralia.gov.au/system/files/documents/1703/general-guide-for-cranes.pdf
29. Sanchez-Vives, M.V., Slater, M.: From presence to consciousness through virtual reality. Nature Reviews Neuroscience **6**, 332–339 (2005). https://doi.org/10.1038/nrn1651
30. Seele, S., Misztal, S., Buhler, H., Herpers, R., Schild, J.: Here's looking at you anyway! How important is realistic gaze behavior in co-located social virtual reality games? In: Proceedings of the Annual Symposium on Computer-Human Interaction in Play. p. 531–540. CHI PLAY '17, ACM, New York, NY, USA (2017). https://doi.org/10.1145/3116595.3116619
31. Sitompul, T.A., Wallmyr, M.: Using augmented reality to improve productivity and safety for heavy machinery operators: State of the art. In: the 17th International Conference on Virtual-Reality Continuum and Its Applications in Industry. pp. 8:1–8:9. VRCAI '19, ACM, New York, NY, USA (2019). https://doi.org/10.1145/3359997.3365689
32. Tang, N., Xiao, S.: Real-time human vision rendering using blur distribution function. In: Proceedings of the 14th ACM SIGGRAPH International Conference on Virtual Reality Continuum and Its Applications in Industry. p. 39–42. VRCAI '15, ACM, New York, NY, USA (2015). https://doi.org/10.1145/2817675.2817686
33. Wu, C.M., Hsu, C.W., Lee, T.K., Smith, S.: A virtual reality keyboard with realistic haptic feedback in a fully immersive virtual environment. Virtual Reality **21**, 19–29 (2007). https://doi.org/10.1007/s10055-016-0296-6
34. Yang, U., Lee, G.A., Kim, Y., Jo, D., Choi, J., Kim, K.H.: Virtual reality based welding training simulator with 3D multimodal interaction. In: 2010 International Conference on Cyberworlds. pp. 150–154. IEEE (2010). https://doi.org/10.1109/CW.2010.68