Intelligent Scanning Collision Avoidance Device with Risk Assessment

Malik Haddad[1], Jamal Shamieh[2], David Sanders[3], Amir Gharavi[4], Giles Tewkesbury[3], Mohamed Hassan-Sayed[5] and Martin Langner[6]

[1] Northeastern University London, London United Kingdom
[2] America University of Madaba, Madaba, Jordan
[3] University of Portsmouth, Portsmouth, United Kingdom
[4] University College London (UCL), London, United Kingdom
[5] University of Southampton, Southampton, United Kingdom
[6] Chailey Heritage Foundation, North Chailey, United Kingdom
malik.haddad@nulondon.ac.uk

Abstract. This paper presents an Intelligent Scanning Collision Avoidance Device (Intelligent-SCAD) which is used to detect obstacles in a powered wheelchair surroundings to avoid related risk consequences. The Intelligent-SCAD provides a safe direction for the wheelchair. Inputs to the Intelligent-SCAD originate from a single rotating ultrasonic transducer fixed to the wheelchair. Readings from the ultrasonic transducer are used to train and test different Artificial Intelligence (AI) algorithms. The AI algorithms used were: Artificial Neural Network, Decision Tree, optimised Tree and optimised K-Nearest Neighbour. An algorithm is selected based on a compromise between accuracy and complexity. The optimised K-Nearest Neighbour algorithm provided the highest testing accuracy and relatively straightforward operation when compared with the other algorithms used. The new device applies optimised K-Nearest Neighbour to predict a safe direction for a wheelchair. The user can override the new system if necessary.

Keywords: Artificial Intelligence, Collision Avoidance, Risk, Disabled, Steer, Wheelchair.

1 Introduction

The work presented in this paper is part of broader research conducted by the authors aiming to improve mobility, reduce risk and enhance the quality of life of disabled powered wheelchair users, increasing their self-confidence and reliance [1], suppressing risk and avoiding related risk consequences [2].

The number of people diagnosed with disability worldwide is on the rise. The type of disability is shifting from mostly physical to a more complex mix of physical/cognitive disability. New systems to address that shift in disability are required. Powered mobility is becoming more acceptable and useful to support individuals with disability [3]. Powered mobility is expected to revolutionize the quality of life of people with disabilities in the next two decades. Many researchers have presented novel approaches for navigating powered mobility [3-6] by creating Human Machine Interfaces [7], intelligent collision avoidance systems and intelligent controllers [8], sensors and sensor fusion [9], Deep Learning [3,4], expert systems [10], and image processing and computer vision [11,12] and they have analysed the behaviour of powered wheelchair drivers to improve mobility [13].

Langner [3] created a Scanning Collision Avoidance Device (SCAD) that used a single rotating ultrasonic transducer to detect obstacles in the wheelchair surroundings by sending ultrasonic pulses through stepped periods. The distance from a detected obstacle was determined by measuring the time of flight required by a pulse to be sent and reflected to the receiver [3]. The area in front of the wheelchair was divided into six sectors: Extreme Left, Mid Left, Front 1, Front 2, Mid Right and Extreme Right. The area scanned by the SCAD and sector division is shown in Figure 1.

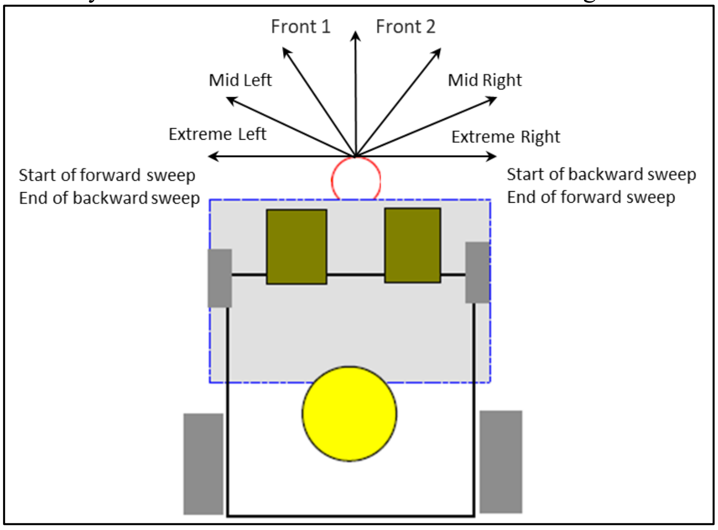

Fig. 1. The area scanned by the SCAD and sector division [3].

The work presented in this paper aims to develop the original SCAD by introducing advanced machine learning algorithms for obstacle detection to improve the accuracy of the SCAD, accommodate the changing nature of disabilities, save costs by developing the equipment already in use instead of replacing them, saving time and improving user convenience by using the same original approaches in [3] instead of introducing new devices that would require training.

2 The Intelligent SCAD (Intelligent-CAD)

A new Intelligent Scanning Collision Avoidance Device (Intelligent-SCAD) was created. It used the K-Nearest Neighbour algorithm to provide a safe direction for a powered wheelchair. Inputs to the Intelligent-SCAD were readings from a SCAD head used in the original SCAD. An electronic circuit was inserted between the SCAD head and the Control box. The electronic circuit used a voltage divider circuit, an op-amp isolation circuit and a Raspberry Pi. Python programming language was used to create a program that analysed readings from the SCAD head. Readings were used to identify the location of obstacles in the surroundings. The program was installed onto the Raspberry Pi.

The program considered three readings from the SCAD head. The readings were: start of the forward scan, start of the backward scan and echo if an obstacle was detected.

The SCAD operated using a 5 Volts power supply, however, such voltage would have damaged the inputs of the Raspberry Pi. A high-impedance voltage divider was used to reduce the 5 Volts used in the SCAD to 3.3 Volts compatible with the Raspberry Pi input voltage similar to the circuit used in [6,7]. An Op-Amp isolation circuit was installed to provide isolation between the SCAD electronic circuit and the Raspberry Pi. Three inputs were used from the SCAD:

- Start Forward Scan: This input identified the start of the forward sweep cycle of the stepper motor. A falling edge on this input identified the start of a forward sweep.
- Start Backward Scan: This input identified the start of the backward sweep of the stepper motor. A rising edge on this input identified the start of a backward sweep.
- Echo: This pin was used to receive echo reflected from detected obstacles. A rising edge on this pin identified that an echo was received.

Figure 2 shows the three inputs from the SCAD, voltage divider and the Op-Amp isolation circuits, and outputs. The outputs were used as inputs to the Raspberry Pi.

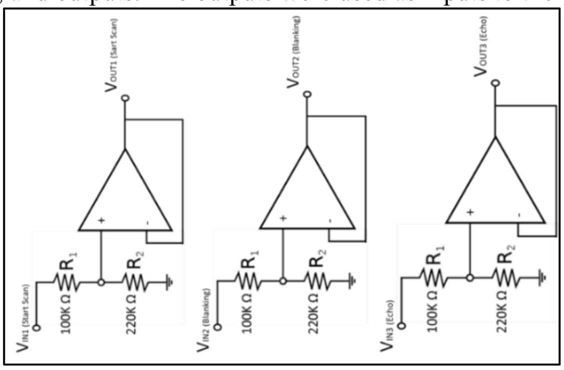

Fig. 2. The schematic diagram of the voltage divider and Op-Amp isolation circuits [14].

Figure 3 shows the Start of the Forward Scan, the Start of the Backward Scan and the Echo signal received when objects were detected in different sectors.

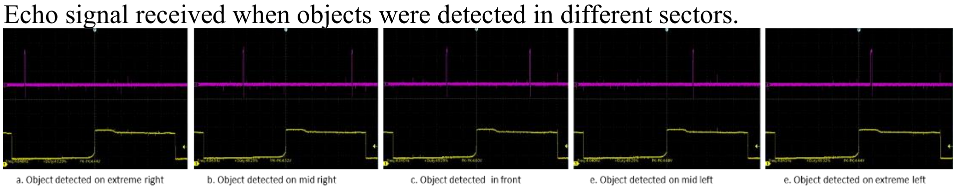

Fig. 3. A screenshot of the signals received from the SCAD head.

Python program identified the start time of the forward sweep cycle, the start time of the backward sweep and the time an echo was received. A setting was prepared to collect training and testing datasets used to train and test the intelligent algorithms. Small plastic cones were considered as obstacles as shown in Figure 4.

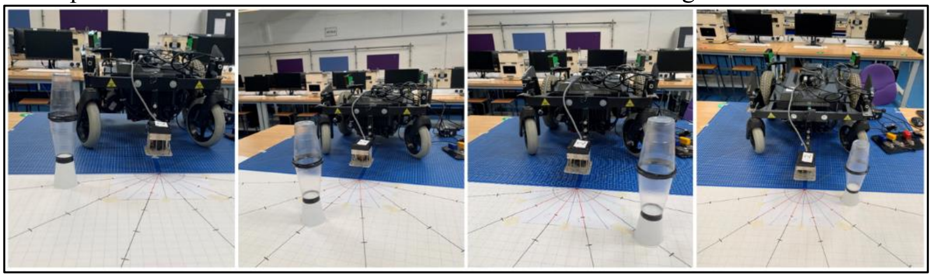

Fig. 4. The setting used to collect training and testing datasets used to train and test the intelligent algorithms.

The small plastic cones were placed at different locations and distances from the SCAD head. The locations considered were: Extreme Left, Mid Left, Front 1, Front 2, Mid Right and Extreme Right. Distances considered were 15, 25, 35 and 45 cm away from the SCAD head. The time needed for echoes to be reflected from the cones during the forward sweep cycle and backward sweep cycle were recorded. 16231 echo times were recorded. A (4 by 16231) matrix was created. Table 1 shows the structure of the matrix.

Table 1. The structure of the Matrix used to train and test the intelligent algorithms.

Sweep Direction	Distance (cm)	Time (msec.)	Location
Forward/Backward	15/25/35/45		Extreme Left Mid Left Front 1 Fron 2 Mid Right Extreme Right

3 Training and Testing the Intelligent Algorithms

The (4 X 16231) matrix was imported to a MATLAB platform and used as training and testing sets. The matrix was split into two sets of 12000 and 4231 for training and testing sets respectively. Training and testing of intelligent algorithms were conducted using the (4 by 16231) matrix. A (4 by 12000) matrix was used for training and a (4 by 4231) matrix was used for testing.

MATLAB was used to create four intelligent algorithms:

3.1. BiLSTM (Bidirectional Long-Short Memory) Neural Network
3.2. Fine Tree classification model
3.3. Optimised Tree classification model

3.4. Optimised K-Nearest Neighbour (KNN) classification model

3.1 BiLSTM Neural Network

The BiLSTM Neural Network used is shown in Figure 5. It considered five layers:

1. Sequence Input Layer with three inputs.
2. BiLSTM Layer (100 hidden units)
3. Fully Connected Layer with six nodes.
4. Softmax Layer.
5. Classification Layer.

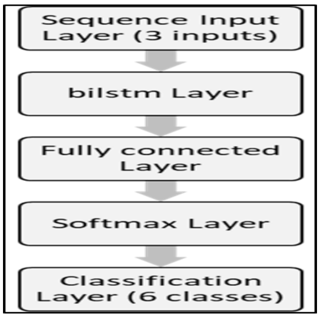

Fig. 5. The BiLSTM Neural Network structure.

Training and testing of the BiLSTM Network was conducted using the (4 by 16231) matrix. A (4 by 12000) matrix was used for training and a (4 by 4231) matrix was used for testing. Figure 6 shows Network training progress with an initial learning rate of 0.01 and 100 epochs, as Network training progressed. The Network accuracy increased and Network training loss decreased.

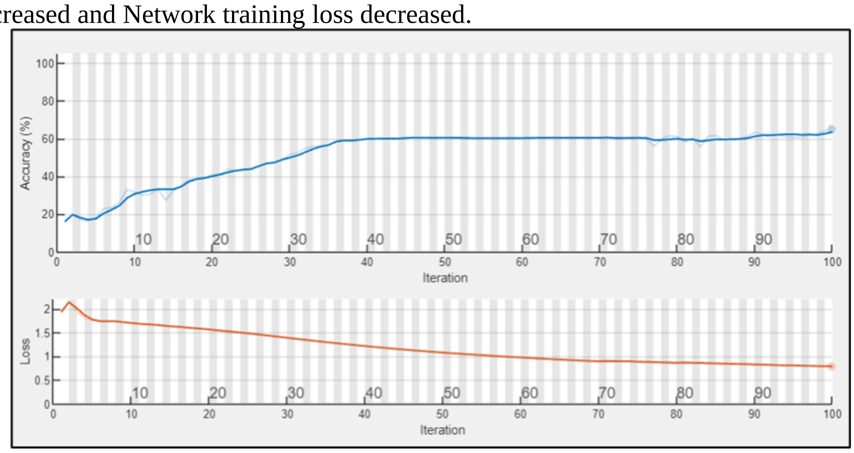

Fig. 6. BiLSTM Neural Network training progress.

Network training accuracy reached 65.22% and Network testing accuracy reached 64.81% when tested with the testing set Figure 7 shows the resulting confusion chart.

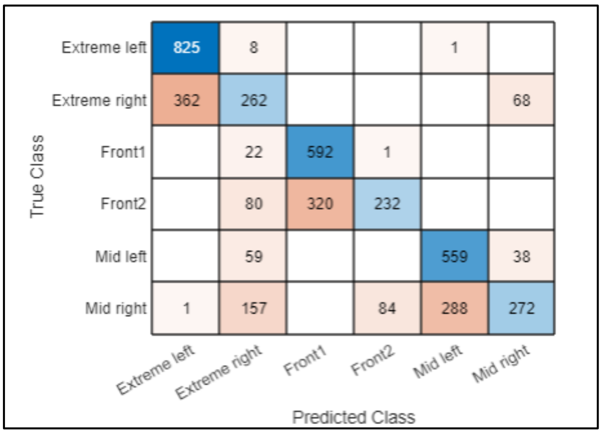

Fig. 7. The confusion chart produced from testing the BiLSTM Neural Network.

3.2 Decision Tree Classification Model

Default settings in MATLAB for the Decision Tree classification model were used to create the Decision Tree classification model used in this paper.

Training and testing of the Decision Tree model was conducted using the (4 by 16231) matrix. A (4 by 12000) matrix was used for training and a (4 by 4231) matrix was used for testing. By the end of training, Model training accuracy reached 99.8% and testing accuracy reached 99.8% when tested with the testing set. Figure 8 shows the resulting confusion chart.

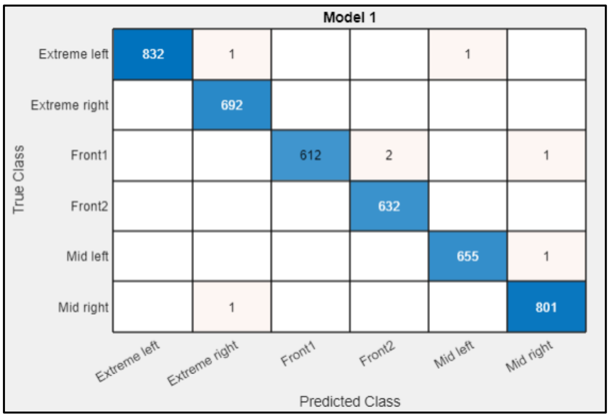

Fig. 8. The confusion chart produced from testing the Decision Tree model.

3.3 Optimised Decision Tree Classification Model

Default settings in MATLAB for the optimised Decision Tree classification model were used to create the optimised Tree classification model used in this paper.

Training and testing of the optimized Decision Tree model was conducted using the same (4 by 16231) matrix. The same (4 by 12000) and (4 by 4231) matrices were used for training and testing respectively. By the end of training, Model training accuracy reached 99.8%. and testing accuracy reached 99.8% when tested with the testing set. Figure 9 shows the resulting confusion chart.

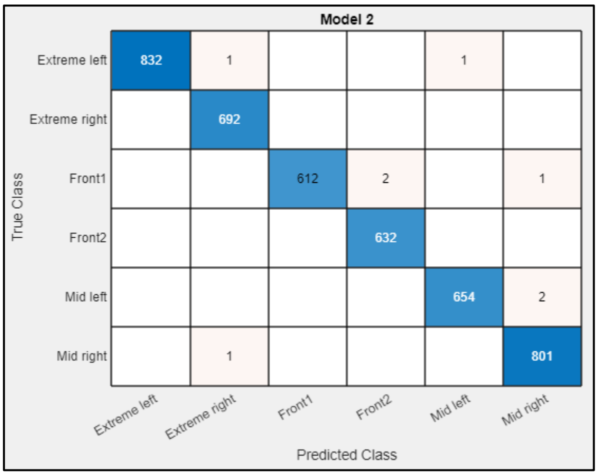

Fig. 9. Confusion chart produced from testing the optimised Decision Tree model.

3.4 Optimised K-Nearest Neighbour (KNN) Classification Model

Default settings in MATLAB for the optimised KNN classification model were used to create the optimised KNN classification model used in this paper.

Training and testing of the optimised KNN model was conducted using the same (4 by 16231) matrix. The same (4 by 12000) and (4 by 4231) matrices were used for training and testing respectively. By the end of training, model training accuracy reached 99.9% and testing accuracy reached 99.9% when tested with the testing set. Figure 10 shows the resulting confusion chart.

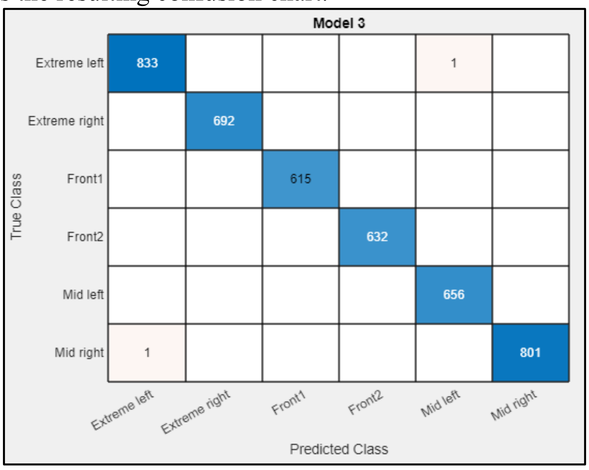

Fig. 10. Confusion chart produced from testing the optimised KNN model.

4 Discussion

The BiLSTM Neural Network achieved 64.81% accuracy when tested using the testing set. The BiLSTM had the lowest accuracy when compared to other intelligent algorithms considered in this paper. Moreover, the BiLSTM Neural Network required the longest training time when compared to the other algorithms.

Real-world testing revealed BiLSTM poor performance when obstacles were introduced into the wheelchair's surroundings, as depicted in Figure 11. Different locations for the obstacles were tested: Figure 11.A for the Mid Left sector, Figure 11.B for the Front 1 sector, Figure 11.C for the Extreme Left sector, Figure 11.D for the Mid Right sector, Figure 11.E for the Extreme Right sector, and Figure 11.F for the Front 2 sector. Despite these tests, it became evident that the BiLSTM model suffered from underfitting, requiring further hyperparameter tuning and extended training for accuracy improvement.

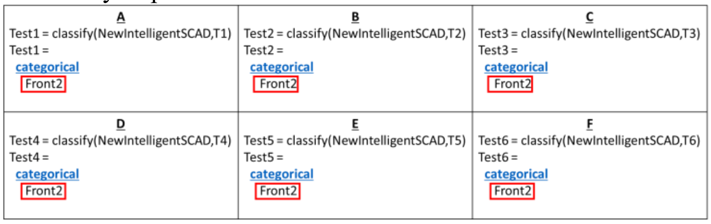

Fig. 11. The outcome of the BiLSTM Neural Network when tested in a real-world environment.

In contrast, the Decision Tree model proved highly accurate, achieving a 99.8% testing accuracy with minimal training time. Figure 12 demonstrates its successful performance in a real-world setting, indicating the locations of obstacles in various sectors surrounding the wheelchair. The model's proficiency is evident in Figure 12.A for the Mid Left sector, Figure 12.B for the Front 1 sector, Figure 12.C for the Extreme Left sector, Figure 12.D for the Mid Right sector, Figure 12.E for the Extreme Right sector, and Figure 12.F for the Front 2 sector. The Decision Tree classification model accurately identified the positions of all six obstacles in the wheelchair's surroundings.

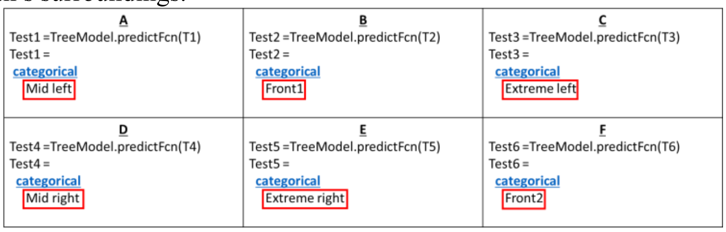

Fig. 12. The outcome of the Decision Tree model when tested in a real-world environment.

Furthermore, the optimised Decision Tree model, despite a longer training time, achieved the same remarkable 99.8% accuracy when tested with the dataset. Figure 13 illustrates its performance under real-world conditions, successfully identifying obstacle locations in different sectors: Figure 13.A for the Mid Left sector, Figure 13.B for the Front 1 sector, Figure 13.C for the Extreme Left sector, Figure 13.D for the Mid Right sector, Figure 13.E for the Extreme Right sector, and Figure 13.F for the Front 2 sector. The optimised Decision Tree classification model consistently recognised the positions of all obstacles in the wheelchair's surroundings.

A Test1 = TreeOptModel.predictFcn(T1) Test1 = categorical Mid left	B Test2 =TreeOptModel.predictFcn (T2) Test2 = categorical Front1	C Test3 =TreeOptModel.predictFcn(T3) Test3 = categorical Extreme left
D Test4 =TreeOptModel.predictFcn(T4) Test4 = categorical Mid right	E Test5 =TreeOptModel.predictFcn(T5) Test5 = categorical Extreme right	F Test6 =TreeOptModel.predictFcn(T6) Test6 = categorical Front2

Fig. 13. The outcome of the optimised Decision Tree when tested in a real-world environment.

Lastly, the optimised K-Nearest Neighbour (KNN) model demonstrated the highest accuracy among all intelligent algorithms considered, reaching 99.9% in testing, though it required more training time than the Decision Tree and optimized Decision Tree models. Figure 14 showcases the model's performance in a real-world environment, successfully pinpointing obstacle locations in different sectors: Figure 14.A for the Mid Left sector, Figure 14.B for the Front 1 sector, Figure 14.C for the Extreme Left sector, Figure 14.D for the Mid Right sector, Figure 14.E for the Extreme Right sector, and Figure 14.F for the Front 2 sector. The optimised KNN classification model consistently and accurately identified the positions of all six obstacles surrounding the wheelchair.

A Test1 =OptKNNModel.predictFcn(T1) Test1 = categorical Mid left	B Test2 =OptKNNModel.predictFcn(T2) Test2 = categorical Front1	C Test3 =OptKNNModel.predictFcn(T3) Test3 = categorical Extreme left
D Test4 =OptKNNModel.predictFcn(T4) Test4 = categorical Mid right	E Test5 =OptKNNModel.predictFcn(T5) Test5 = categorical Extreme right	F Test6 =OptKNNModel.predictFcn(T6) Test6 = categorical Front2

Fig. 14. Outcome of the optimised KNN when tested in a real-world environment.

5 Risk Assessment and Mitigation

This section presents the risk overview and mitigation.

5.1 Risk Overview

There is no single activity or system that has no risk embedded in it [15]. Risks can be inherent in almost whatever activity or system is developed or used. As such, wheelchair users are not an exception and are subject to technical failures, improper operation, software and/or hardware malfunctioning, or other manufacturing defects. The wheelchair may encounter such failures individually or collectively.

Amongst many different types of risks, wheelchair users face mainly what is called pure risk. That is to say, “Loss or No Loss.” In other words, “Injury or No Injury.”

This situation could hold the manufacturer and or the sales company liable for compensation or other penalties according to the laws of specific countries. What complicates the situation is that such failures can put the life or health of the wheelchair user in jeopardy.

The direct implication is that wheelchair risks need to be mitigated in order to minimise the probability and impact of risks that wheelchair users are subjected to.
The way forward is to implement the five-phase risk management cycle in order to identify, assess (evaluate and estimate), mitigate such risks and monitor and control any residual risk.

5.2 Risk Mitigation

In principle, there are two established approaches to applying the risk management cycle [15, 16]:

1. The reactive approach: This is to take necessary actions once the risk manifests itself. However, this approach is not successful in the case of a wheelchair since the damage has occurred and users need to mitigate the risk by going to a hospital, as an example.
2. The proactive approach: This is to identify the risk before it manifests itself, and take corrective actions to mitigate it.

The direct implication is to mitigate wheelchair impact risks using an intelligent system similar to the Intelligent-SCAD setup presented in this paper.

6 Conclusions and Suggested Future Work

A new Intelligent Scanning Collision Avoidance Device (Intelligent-SCAD) was created and presented in this paper. The new device successfully identified the location of obstacles in a wheelchair surroundings.

Four intelligent algorithms were used to identify the location of obstacles in the wheelchair's surroundings. Results from testing the intelligent algorithms were compared and an analysis was conducted to determine the best compromise between accuracy and complexity. The optimised K-Nearest Neighbour (KNN) model achieved the highest accuracy. It provided 99.9% accuracy when tested against the testing set and achieved the lowest loss compared to the other intelligent algorithms used in this paper. Also, the KNN model provided straightforward operation.

The Intelligent-SCAD successfully identified the location of all obstacles in the wheelchair's surroundings when tested in a real-world environment.

Data used to train and test the intelligent algorithms used in this paper was the raw data collected from placing an obstacle in a wheelchair surrounding and calculating the time needed for an echo signal to be sent and reflected from the obstacle. During the backward sweep, ghost echoes were received especially when the detected obstacle was in close proximity to the wheelchair at the start of both the forward sweep cycle and the backward sweep cycle. These ghost echoes were from pulses transmitted during the previous sweep. Data pre-processing could be conducted to eliminate the problem caused by ghost echoes received from previous transmissions.

Conducting data pre-processing could improve the accuracy of the BiLSTM Neural Network used in this paper.

The new approach will be used with other intelligent systems created by the authors [11-13] to improve powered wheelchair users' mobility, avoid related risks, enhance their quality of life and reduce the need and cost of carers.

Future work will consider using mathematically inexpensive Machine Learning and Artificial Intelligence algorithms to detect driving patterns and intelligently avoid obstacles in the wheelchair's surroundings.

Data collected in this paper will be used to improve the accuracy of the Deep Learning Collision Avoidance systems presented in [3,4].

References

1. Sanders, D., Gegov, A., 2018. Using artificial intelligence to share control of a powered-wheelchair between a wheelchair user and an intelligent sensor system, EPSRC Project, 2019.
2. Gina, E., Bertocci and Jonathan E., 2020. Injury risk assessment of wheelchair occupant restraint systems in a frontal crash: A case for integrated restraints. Journal of Rehabilitation Research and Development, Rehabilitation Science and Technology, 5044 Forbes Tower, University of Pittsburgh, Pittsburgh, PA 15260
3. Haddad, M., Sanders, D., Tewkesbury, G., Langner, M. and Keeble, W. A novel collision avoidance approach for powered wheelchair steering using deep learning. Journal of Physics: Conference Series. (2021).
4. Haddad, M.J. and Sanders, D.A., 2020. Deep Learning Architecture to Assist With Steering a Powered Wheelchair. IEEE Transactions on Neural Systems and Rehabilitation Engineering: a Publication of the IEEE Engineering in Medicine and Biology Society, 28(12), pp.2987-2994.
5. Langner, M., 2012. Effort Reduction and Collision Avoidance for Powered Wheelchairs: SCAD Assistive Mobility System, Ph.D. dissertation, University of Portsmouth.
6. Haddad, M., Sanders, D., Langner, M., Bausch, N., Thabet, M., Gegov, A., Tewkesbury, G. and Ikwan, F. (2021) Intelligent Control of the Steering for a Powered Wheelchair Using a Microcomputer. In: Arai K., Kapoor S., Bhatia R. (eds) Intelligent Systems and Applications. IntelliSys 2020. Advances in Intelligent Systems and Computing, vol 1252. Springer, Cham.
7. Haddad, M., Sanders, D., Ikwan, F., Thabet, M., Langner, M. and Gegov, A. Intelligent HMI and Control for Steering a Powered Wheelchair Using a Raspberry Pi Microcomputer, 2020 IEEE 10th International Conference on Intelligent Systems (IS), 2020, pp. 223-228.
8. Haddad M, Sanders D, Langner M, Bausch N, Thabet M, Gegov A, Tewkesbury G and Ikwan F 2020 Intelligent control of the steering for a powered wheelchair using a microcomputer *Proc. of SAI Intelligent Systems Conference (Amsterdam)* pp 594-603
9. Haddad M, Sanders D, Tewkesbury G and Langner M 2021 Intelligent User Interface to Control a Powered Wheelchair using Infrared Sensors," *Proc. of SAI Intelligent Systems Conference (Amsterdam)* pp 640-649
10. Sanders D, Gegov A, Haddad M, Ikwan F, Wiltshire D and Tan YC 2018 A rule-based expert system to decide on direction and speed of a powered wheelchair *Proc. of SAI Intelligent Systems Conference (London)* pp 822-838

11. Haddad, M., Sanders, D., Langner, M. and Tewkesbury, G., 2022. One shot learning approach to identify drivers. In Intelligent Systems and Applications: Proceedings of the 2021 Intelligent Systems Conference (IntelliSys) Volume 1 (pp. 622-629). Springer International Publishing.
12. Haddad, M., Sanders, D., Tewkesbury, G., Langner, M., Fakhraddin Muhyaddin, S. and Ibrahim, M.: Computer-vision algorithms to steer powered wheelchairs. Journal of Physics: Conference Series. (2021).
13. Sanders D, Haddad M, Tewkesbury G, Bausch N, Rogers I and Huang Y (2020) Analysis of Reaction Times and Time-Delays Introduced into an Intelligent HCI for a Smart Wheelchair. In 2020 IEEE 10th International Conference on Intelligent Systems (IS), pp 217-222. IEEE.
14. Texas Instruments, Isolated Voltage-Measurement Circuit With ±250-mV Input. Availabe at: https://www.ti.com › lit › pdf › sbaa350, accessed on 22nd October 2023.
15. Rejda et al., Principles of Risk Management and Insurance. pp.22-27, 2014, 5th ed.
16. Hubbard, D., The Failure of Risk Management: Why It's Broken and How to Fix It. 2009, John Wiley & Sons. p. 46.