Enhancing Genetic Algorithm-Based Process Parameter Optimisation through Grid Search-Optimised Artificial Neural Networks

1st Yusuf Furkan Ugurluoglu
School of Engineering
Lancaster University
Lancaster, UK
y.ugurluoglu@lancaster.ac.uk

2nd Darren Williams
School of Engineering
Lancaster University
Lancaster, UK
d.williams8@lancaster.ac.uk

3rd Min Xia
School of Engineering
Lancaster University
Lancaster, UK
m.xia3@lancaster.ac.uk

***Abstract*—In recent years, genetic algorithms (GAs) have been combined with artificial neural networks (ANNs) to increase the search space for desired outputs. However, a research gap remains in optimising the ANN before incorporating it into the GA. In this study, the ANN was optimised through grid search, using parameters such as learning rates, number of epochs, batch sizes, number of neurons, activation functions, and optimisers. Furthermore, the optimisation approach was designed to be applied to the additive manufacturing process, where data is collected on LPBF process parameters such as laser power, scan speed and hatch distance. The results demonstrated that this approach improved the performance of GA-ANN parameter optimisation, including faster convergence and higher objective function value.**

***Index Terms*—Machine learning, Process optimisation, Neural network, Genetic algorithm, Parameter optimisation**

I. Introduction

Additive manufacturing, a technology that enables the creation of complex 3D structures layer-by-layer, has seen significant advancements in recent years. Laser Powder Bed Fusion (LPBF) is a popular additive manufacturing process that uses laser energy to fuse metal powder particles to create functional metal parts selectively. To optimise the LPBF process parameters such as laser power, scan speed, and hatch distance, various optimisation techniques have been explored. Genetic Algorithms (GAs) and Artificial Neural Networks (ANNs) have gained significant attention due to their potential to solve complex problems in engineering, finance, biology, and other fields [1]. ANNs are computational models inspired by the structure and function of the human brain [2], while GAs are optimisation algorithms inspired by natural selection and genetics [3].

ANNs are machine learning models inspired by the structure and function of biological neurons. ANNs comprise layers of interconnected nodes, or "neurons." The input layer receives input data, which is then processed by one or more hidden layers before the output layer generates a prediction or classification [4].

There is a large body of research on applying ANNs and GAs to optimisation problems. This includes the use of ANNs to approximate the fitness function of a GA and the use of GAs to optimise the structure and weights of ANNs. In this research, ANN was used to approximate the fitness function of a developed GA.

Grid search is an effective method for optimising ANN design by systematically trying different hyperparameter combinations to identify the best configuration. The performance of the ANN architecture can be considerably enhanced before being fed into a GA, which will also profit from the enhanced architecture [5], [6]. This method can enhance a neural network's performance while simultaneously reducing its complexity and computational requirements [7].

Manufacturing process optimisation is critical for high-quality products and increased efficiency. Genetic algorithms (GAs) can be used to optimise a variety of production processes, including additive manufacturing (AM) and friction stir welding (FSW) [5]. GAs can handle complex objective functions and search many process parameters for the best solution. GAs can be used in additive manufacturing to improve printing process parameters such as layer thickness and print speed to generate high-quality prints. Similarly, GAs can be utilised in FSW to optimise process parameters such as plunge depth, tool rotational speed, welding speed and tool geometry to generate high-quality welded joints. Overall, GAs are a versatile and strong tool for optimising production processes, and their use can lead to higher product quality, efficiency, and cost-effectiveness [8].

Several studies have combined GAs and ANNs to optimise various tasks, including feature selection, parameter optimisation, and network topology optimisation [9]. Nonetheless, most of these studies focused on optimising the GA while assuming that the ANN is already optimised. This assumption may not hold in practice because the performance of the ANN depends on its architecture and hyper-parameters, such as learning rate, number of epochs, batch size, number of neurons, activation function, and optimiser.

Recent studies have proposed optimising the ANN before incorporating it into the GA to solve this problem; before applying a GA for feature selection in a classification task,

[10] optimised the hyper-parameters of an ANN using a grid search technique. According to their findings, Random Search and Grid Search are promising and effective optimisation strategies for this task. Sometimes, the small population of solutions used at the outset and the expensive goal functions employed by these searches can result in slow convergence or execution time. In this study, they propose using the Support Vector Machine as a machine learning model and optimising it with four distinct algorithms—the Ant Bee Colony Algorithm, the GA, the Whale optimisation, and the Particle Swarm Optimisation—to determine the computational cost of SVM following hyper-tuning. The GA was found to have lower temporal complexity than other algorithms in this study.

Other studies have proposed more sophisticated techniques for optimising ANNs before their incorporation into GAs. For instance, they [11] proposed a hybrid method combining differential evolution and backpropagation for optimising ANNs before using a GA for feature selection. In terms of the precision of feature selection, their findings demonstrated that the hybrid strategy outperformed other methods. Simulation results demonstrate that the GA-SVM hybrid achieves comparable classification accuracy and consistency to other well-established algorithms. The results validate the classification accuracy improvements and demonstrate the classifier's potential for future data mining applications.

In addition to optimising ANNs before incorporating them into GAs, other studies have investigated the use of GAs to optimise ANNs directly. [12] For instance, when comparing the performance of the models, optimised ANNs reduced the mean squared error (MSE) over testing sets by up to 90%, whereas optimised SVM's led to a reduction of up to 70%, with the latter's greatest advantage being computational efficiency and consistency across the various GA runs. The GA converged in minutes, which is more effective than trial-and-error methods. This study demonstrates that more attention should be paid to the effect of machine learning model architectures on the accuracy of the models, as the computational cost of the GA is more than justified by its high accuracy. In addition, research has been conducted on enhancing the performance of GAs by incorporating ANNs as fitness functions [13].

While previous studies have examined the combination of ANNs and GAs, the specific aspect of optimising ANNs before integrating them with gas is still a new research topic. This study will bridge this gap by developing a grid search strategy for pre-optimising ANNs before incorporating them into GAs to solve optimisation challenges. We hypothesise that by enhancing the initial configuration of the ANN, the performance of the GA-optimised ANN can be improved, resulting in more efficient and effective optimisation outcomes.

In this study, we propose a novel approach for parameter optimisation of an Artificial Neural Network (ANN) by combining grid search and genetic algorithm techniques. Our approach aims to enhance the performance of the genetic algorithm-optimised ANN by pre-optimising the ANN through grid search. We optimise the ANN's hyperparameters, such as learning rates, number of epochs, batch sizes, number of neurons, activation functions, and optimisers. This systematic grid search allows us to identify the optimal combination of hyperparameters to improve the ANN's convergence speed and objective function values.

To validate the effectiveness of our approach, we utilise data from the Friction Stir Welding (FSW) process, as the availability of data from the target additive manufacturing process was limited. Despite the difference in process context, our findings demonstrate the efficacy of the pre-optimisation method in enhancing the performance of the genetic algorithm-optimised ANN. We achieve faster convergence and higher objective function values by incorporating grid search to fine-tune the ANN before the genetic algorithm optimisation.

Our approach provides a valuable contribution to the field, emphasising the importance of pre-optimisation in the ANN-GA integration. By systematically exploring the hyperparameter space and leveraging the power of genetic algorithms, we establish a robust framework for parameter optimisation that can be applied to various optimisation problems, including additive manufacturing processes.

II. Methodology

A. Experimental approach

There were two optimisation approaches in this study: process parameter optimisation with GA and ANN optimisation with grid search.

In the process parameter optimisation, an ANN was trained using the process data, and then a pre-trained ANN model was used in the fitness function evaluation step. The flow chart of the integration of an ANN and a GA can be seen in Figure 1.

The ANN optimisation approach was utilised using the grid search method. In other words, it was optimised using grid search before incorporating the ANN into the GA. Grid search is a common technique in machine learning used to search for the optimal combination of hyperparameters. In this study, the hyperparameters selected for optimisation were learning rate, number of epochs, batch size and activation function. The range of values for each hyperparameter can be seen in Table I.

TABLE I
Range of hyperparameters for grid search optimisation of the ANN

Parameters	Search Space		
Learning Rate	0.0001	0.001	0.01
Number of Epochs	50	100	150
Batch Size	32	64	128
Activation Function	ReLU	Sigmoid	Tanh

The grid search algorithm exhaustively searched for the optimal combination of hyperparameters within the specified range. The performance of the ANN was evaluated on the validation set for each combination of hyperparameters, and the combination that achieved The mean squared error (MSE) was selected as the optimal combination.

After the ANN was optimised, it was integrated into the GA for further optimisation. The GA used a binary string

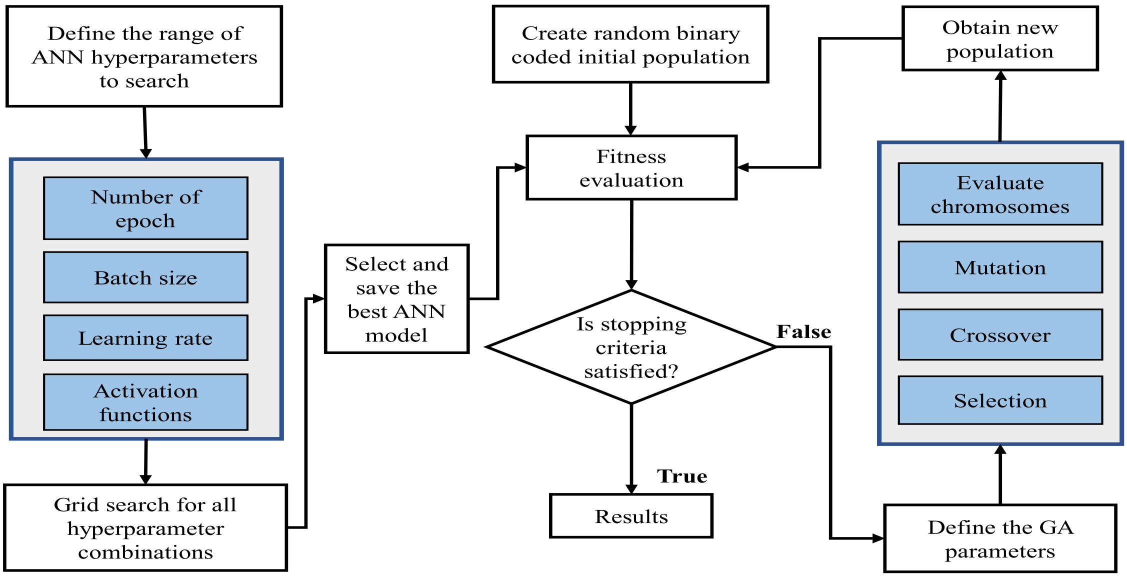

Fig. 1. This figure illustrates the methodology employed in this study for parameter optimisation. The flowchart showcases two main components: the grid search of the Artificial Neural Network (ANN) and the genetic algorithm (GA) model. The grid search is performed to optimise the ANN's hyperparameters, while the GA incorporates a pre-trained ANN model in the fitness function calculation.

representation of the weights of the ANN as the chromosome. The population size was set to 50, and the mutation rate was set to 0.01. The fitness function used for the GA was the accuracy of the ANN on the validation set. The GA was run for 100 generations, and the best individual from the final generation was selected as the solution.

All experiments were conducted on a machine with an Intel Core i7-7700HQ processor and 16GB RAM. The software used for implementation was Python 3.7 with the Keras library for building and training the ANN and the GA. The experiments were performed using TensorFlow.

The dataset used in the study was obtained from a friction stir welding process, which contains 50 experiments with 8 input parameters and 5 output features [14].

B. Grid Search Procedure

The grid search systematically optimises this study's Artificial Neural Network (ANN) hyperparameters. The procedure consists of the following steps:

- Hyperparameters Considered: The grid search focuses on essential hyperparameters, including alearning rate, number of epochs, batch size, and activation function (see Table I).
- Defining the Search Space: Predefined ranges are established for each hyperparameter based on prior knowledge and literature review. The hyperparameters considered in this study included the learning rate, number of epochs, batch size, and activation function. We considered values of 0.0001, 0.001, and 0.01 for the learning rate and 50, 100, and 150 for the number of epochs. The batch size was explored with 32, 64, and 128 values. Additionally, we considered three different activation functions: ReLu, Sigmoid, and Tanh.
- Sampling the Search Space: A regular grid pattern is used to systematically evaluate each hyperparameter combination within the defined search space.
- Performance Evaluation: The ANN's performance is assessed for each hyperparameter combination using the mean absolute error (MAE) as the predefined objective function. MAE captures the discrepancy between predicted and actual values, measuring prediction accuracy.
- Selection of Optimal Hyperparameters: The optimal set of hyperparameters is determined based on the MAE performance metric. The combination yielding the lowest MAE is selected as the optimised set.

The grid search procedure allows for identifying the optimal hyperparameters for the ANN, resulting in improved performance. The ANN can improve accuracy and predictive capability by fine-tuning the hyperparameters.

C. ANN Modelling and Evaluation Criteria

Data from the friction stir welding process was used to train a three-hidden-layer neural network with multiple inputs and outputs (see Figure 2) [14]. The neural network was built with eight input and five output neurons corresponding to input parameters and output features. The dataset was pre-processed using normalisation and feature scaling techniques to ensure all input variables were on the same scale. This pre-processing phase allows the neural network to converge faster during training by preventing input features from disproportionately affecting the network's weights.

The neural network was trained using the backpropagation algorithm, using mean squared error (MSE) as the loss function. MSE is a common loss function in regression problems since it penalises larger errors more than smaller ones. The dataset was divided into training and validation sets at random,

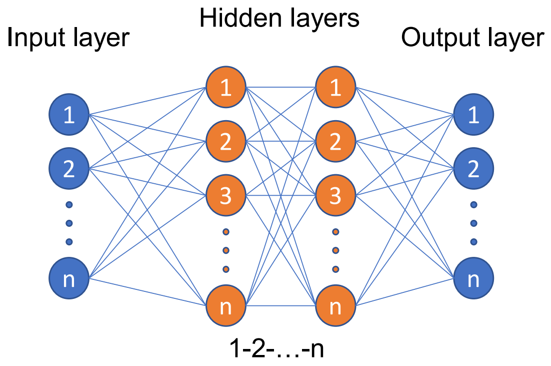

Fig. 2. The architecture of the multiple inputs, multiple outputs ANN. This study used eight input neurons, five output neurons and three hidden layers in the ANN architecture.

with a 67/33 split. This division ensures that the model gets trained on enough data while allowing for an independent evaluation of the model's performance.

The neural network architecture consisted of three fully connected hidden layers, 160, 480 and 256 neurons in the first, second and third hidden layers, respectively. The input layer included 8 neurons, while the output layer had 5 neurons. The neural network can learn complicated representations of the input information due to the inclusion of three hidden layers. In contrast, the variable number of neurons in each layer allows the model to be more flexible in its ability to fit the data. The output layer of the neural network was made up of five neurons that were activated using the Sigmoid function. The Sigmoid activation function is appropriate for regression problems, as it scales the output between 0 and 1 and can provide probability-like interpretations of the predictions.

The network was optimised using several hyperparameters such as learning rate, epochs, batch size, and activation function in hidden layers. These hyperparameters were changed to provide the best neural network performance as evaluated by the MAE metric. The MAE metric measures the network's prediction accuracy by calculating the absolute difference between predicted and actual values. The neural network may generalise to new data and improve its prediction ability in the process parameter optimisation approaches by optimising these hyperparameters.

The Mean Absolute Error (MAE) was used as an evaluation metric in this study to assess the performance of the Artificial Neural Network (ANN) model in predicting the best process parameters for friction stir welding. MAE was chosen above other measures such as R-squared (R2), RMSE, accuracy, recall, and F1-score because it provides a more intuitive and easily interpretable measure of the difference between predicted and actual values. In addition, the MAE is less sensitive to outliers and extreme values in the data, which is important in welding process parameters with a large range of values.

The mathematical definition of MAE is as follows:

$$MAE = \left(\frac{1}{n}\right)\sum_{k=1}^{n}|(y_i - \widehat{y_i})| \tag{1}$$

where y_i represents the actual value of the experiment, $\widehat{y_i}$ represents the predicted value, and n is the total number of samples.

D. Genetic algorithm optimisation procedure

The genetic algorithm used a tournament selection approach with a tournament size of three. The uniform crossover approach was employed for the crossover procedure, in which each element of the offspring was randomly selected with equal probability from the corresponding elements of the two parents. The mutation operation modified the value of a hyperparameter at random with a 0.1 chance.

The genetic algorithm's ideal hyperparameters were used to train a final ANN model. The resulting model's performance was evaluated on a separate test data set, and the findings were compared to the original ANN model and other benchmark models.

TABLE II
LIST OF SETTINGS AND PARAMETERS USED FOR GENETIC ALGORITHMS.

Parameter	Value
Population size	50
Mutation probability	0.05
Crossover probability	0.7
Number of generations	100
Response to be maximised	Fitness function

Overall, the optimisation approach implemented in this work combines the advantages of grid search and evolutionary algorithms to find the best process parameters.

III. RESULTS AND DISCUSSION

The optimised ANN was trained on the training set using the hyperparameters selected by the grid search method. The training process was run for the entire grid search parameters. The optimised ANN was then integrated into the GA for further optimisation.

Figure 3 demonstrates the effect of activation function, learning rate, number of epochs, and batch size on MAE. The results showed that the only considerable parameters were the activation function and learning rate, which resulted in changes in MAE. Furthermore, the ReLU function showed the best performance with 0.45 MAE compared to the Sigmoid function with 0.71 MAE. According to the findings, optimising the learning rate parameter may result in lower MAE values in this model by decreasing the MAE by 22%. Overall, the results indicate that optimising ANN parameters of activation function and learning rate can considerably impact model performance, emphasising the necessity of selecting proper parameters during model construction.

The grid search results revealed that the artificial neural network (ANN) performance was highly dependent on the choice of hyperparameters. The ANN model achieved the highest performance among the tested combinations with a

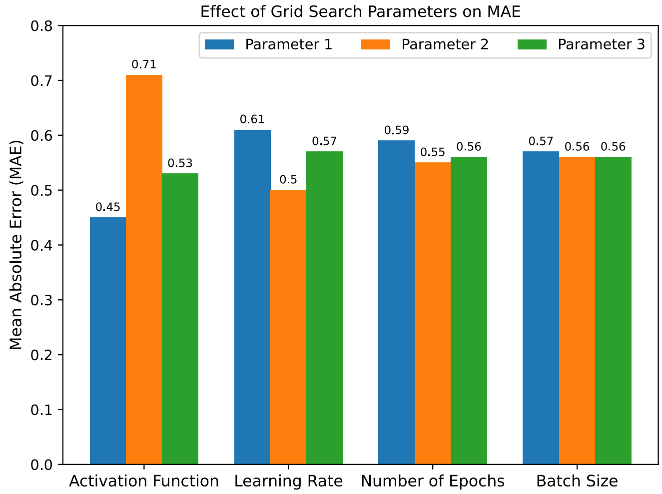

Fig. 3. The figure showcases the influence of grid search parameters on the Mean Absolute Error (MAE). The x-axis displays the hyperparameters: Activation Function, Learning Rate, Number of Epochs, and Batch Size. Each bar is divided into three sections representing different parameter values. The MAE values highlight the impact of parameter configurations on the accuracy of the ANN model.

0.001 learning rate, 100 epochs, 32 batch size, and ReLU activation function. In contrast, the worst performing ANN was found to have a learning rate of 0.01, 50 epochs, 32 batch size, and Sigmoid activation function. To further investigate the impact of the activation function on model performance, we selected two ANN models with the same parameters except for the activation function. The ANN model with ReLU activation achieved an MAE of 0.35, while the Sigmoid activation model had a much higher MAE of 0.63. These results suggest that the choice of activation function can significantly impact ANN performance and should be carefully considered when designing and training ANN models.

To evaluate the performance of the selected ANN models, they were used to evaluate the fitness function of a genetic algorithm (GA). The GA had a population size of 50, a mutation probability of 0.05, and a crossover probability of 0.7, and it was run for 100 generations. A dataset contained eight input features, including plunge depth (PD), tool rotational speed (RPM), welding speed (WS), tool geometry (TG), shoulder diameter (SD), pin diameter (PnD), tool pin length (TPL), and dwell time (DT), as well as five output features, including ultimate tensile strength (UTS in MPa), yield strength (YS in MPa), ductility (% EL.), bending angle (BA in degree), and nugget zone hardness (HRD in HV). To evaluate the effectiveness of the ANN model as a fitness function, we compared two ANN models with identical parameters except for the activation function. One model used the ReLU activation function, while the other used the sigmoid activation function. Both models had a 0.001 learning rate, 100 epochs, and 32 batch sizes.

The convergence rate of the two ANN models was also assessed. A plot of the fitness function values over 100 generations revealed that the ANN model with the ReLU activation function reached a fitness value of 94, but the model with the sigmoid activation function got only an 85. This implies that the ReLU activation function produced better optimisation results. Furthermore, an evaluation of the convergence rate revealed that the ANN model with the ReLU activation function converged faster than the model with the sigmoid activation function. This is seen in Figure 4, which shows that the ReLU model has a steeper convergence rate than the sigmoid model. These results emphasise the importance of selecting an appropriate activation function when designing and implementing ANN-based optimisation algorithms.

In summary, the proposed method was effective in improving the time to convergence and reaching to maximum fitness value compared to using an optimised GA with a randomly initialised ANN. The optimised ANN was able to reduce the search space required by the GA and improve the convergence speed. The results of this study provide insights for future work in the field of metaheuristic optimisation and artificial neural networks.

The proposed method in this study aimed to optimise the ANN before integrating it into the GA for classification tasks. The results of the experiments showed that the proposed method was effective in improving the classification accuracy compared to using a GA with a randomly initialised ANN. One of the advantages of the proposed method is that it can reduce the search space required by the GA. By optimising the ANN before integrating it into the GA, the search space for the GA is reduced, which can lead to faster convergence and better performance. This can be particularly useful when dealing with larger datasets or more complex classification tasks, including facial recognition systems and big data in health science [15], [16].

However, there are still some limitations to the proposed method. The grid search method used in this study is a brute-force approach that can be computationally expensive, especially for larger datasets or more complex ANN architectures. Additionally, the proposed method only considers limited hyperparameters for the ANN. There may be other hyperparameters or combinations of hyperparameters that could lead to better performance. In future work, Model Predictive Control (MPC) can be utilised to overcome these limitations since MPC is the ability to get a fast response and feedback control [17], [18].

IV. Conclusion

In conclusion, the findings of this study show that adjusting an artificial neural network's (ANN) activation function and learning rate can considerably impact its performance. The grid search strategy employed in this work successfully selected the best hyperparameters for the ANN model, which was then integrated into a genetic algorithm (GA) for further optimisation. The results suggest that an optimised ANN can improve convergence speed and achieve maximum fitness values compared to using a GA with a randomly initialised ANN.

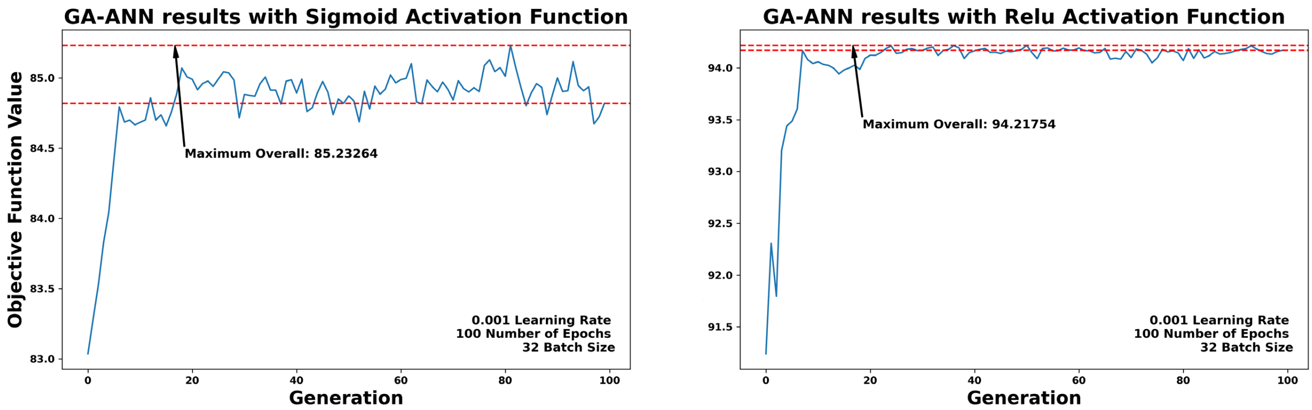

Fig. 4. This figure presents a comparison of the Genetic Algorithm-optimised Artificial Neural Network (GA-ANN) results using two different activation functions: ReLu (on the right) and Sigmoid (on the left). The x-axis represents the number of generations, while the y-axis indicates the objective function value. The graph on the right demonstrates significantly improved performance regarding both the objective function value and time to convergence, highlighting the superiority of the GA-ANN with the ReLu activation function over the Sigmoid activation function.

The study results also demonstrated that the activation function used significantly impacts ANN performance. The ReLU activation function performed better than the sigmoid activation function, leading to reduced MAE and higher optimisation results in the GA. However, it is important to acknowledge the limitations of using grid search and GA parameters. The predefined ranges may not encompass the optimal values, and potential interactions between parameters could not be explicitly investigated. Further exploration and experimentation with alternative parameter configurations are necessary to optimise the performance of the ANN and GA comprehensively. Additionally, it is worth noting that the grid search strategy employed in this work is computationally expensive, highlighting the need for future research to address this limitation.

Overall, the proposed method provides insight into future research in metaheuristic optimisation and artificial neural networks. Further research could look into using more efficient hyperparameter optimisation methods or investigating the effect of other hyperparameters on ANN performance. Furthermore, the proposed method could be expanded to other classification challenges or applied to regression tasks.

Acknowledgment

This research was funded by UK Research and Innovation (reference number 76940) and Smart Eureka (reference number S0309).

References

[1] O. I. Abiodun, A. Jantan, A. E. Omolara, K. V. Dada, N. A. E. Mohamed, and H. Arshad, "State-of-the-art in artificial neural network applications: A survey," Heliyon, vol. 4, no. 11, p. e00938, 2018, doi: 10.1016/j.heliyon.2018.e00938.

[2] M. G. M. Abdolrasol et al., "Artificial neural networks based optimisation techniques: A review," Electron., vol. 10, no. 21, 2021, doi: 10.3390/electronics10212689.

[3] J. McCall, "GAs for modelling and optimisation," J. Comput. Appl. Math., vol. 184, no. 1, pp. 205–222, 2005, doi: 10.1016/j.cam.2004.07.034.

[4] S. Sharma, "Artificial neural network (ANN) in machine learning," Data Sci. Cent., 2017.

[5] G. Erdogan Erten, S. Bozkurt Keser, and M. Yavuz, "Grid Search Optimised Artificial Neural Network for Open Stope Stability Prediction," International Journal of Mining, Reclamation and Environment, vol. 35, no. 8, pp. 600–617, 2021.

[6] M. A. Mahmood, A. I. Visan, and C. Ristoscu, "Artificial Neural Network Algorithms for 3D Printing," pp. 1–27, 2021.

[7] C. K. J. Hou and K. Behdinan, "Dimensionality Reduction in Surrogate Modeling: A Review of Combined Methods," Data Sci. Eng., vol. 7, no. 4, pp. 402–427, 2022, doi: 10.1007/s41019-022-00193-5.

[8] 2nd International Conference on Information Technology and Management Innovation, ICITMI 2013, vol. 411–414. 2013.

[9] M. Kotyrba, E. Volna, H. Habiballa, and J. Czyz, "The Influence of GAs on Learning Possibilities of Artificial Neural Networks," Computers, vol. 11, no. 5, 2022, doi: 10.3390/computers11050070.

[10] Y. A. Ali, E. M. Awwad, M. Al-Razgan, and A. Maarouf, "Hyperparameter Search for Machine Learning Algorithms for Optimising the Computational Complexity," Processes, vol. 11, no. 2, 2023, doi: 10.3390/pr11020349.

[11] K. C. Tan, E. J. Teoh, Q. Yu, and K. C. Goh, "A hybrid evolutionary algorithm for attribute selection in data mining," Expert Syst. Appl., vol. 36, no. 4, pp. 8616–8630, 2009, doi: 10.1016/j.eswa.2008.10.013.

[12] M. Azzam, M. Awad, and J. Zeaiter, "Application of evolutionary neural networks and support vector machines to model NOx emissions from gas turbines," J. Environ. Chem. Eng., vol. 6, no. 1, pp. 1044–1052, 2018, doi: 10.1016/j.jece.2018.01.020.

[13] L. Ding, Y. Yue, K. Ahmet, M. Jackson, and R. Parkin, "Global optimisation of a feature-based process sequence using GA and ANN techniques," Int. J. Prod. Res., vol. 43, no. 15, pp. 3247–3272, 2005, doi: 10.1080/00207540500137282.

[14] N. F. Alkayem, B. Parida, and S. Pal, "Optimisation of Friction Stir Welding Process Parameters using Soft Computing Techniques," Soft Computing, vol. 21, pp. 7083–7098, 2017.

[15] S. Yucer, F. Tektas, N. Al Moubayed, and T. P. Breckon, "Measuring hidden bias within face recognition via racial phenotypes," in Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2022, pp. 995–1004.

[16] U. Demirbaga and G. S. Aujla, "Mapchain: A Blockchain-based Verifiable Healthcare Service Management in IoT-based Big Data Ecosystem," IEEE Transactions on Network and Service Management, 2022.

[17] T. Pamulapati, M. Cavus, I. Odigwe, A. Allahham, S. Walker, and D. Giaouris, "A Review of Microgrid Energy Management Strategies from the Energy Trilemma Perspective," Energies, vol. 16, no. 1, p. 289, Dec. 2022, doi: 10.3390/en16010289.

[18] M. Cavus, A. Allahham, K. Adhikari, M. Zangiabadi, and D. Giaouris, "Energy Management of Grid-Connected Microgrids using an Optimal Systems Approach," IEEE Access, pp. 1–1, Jan. 2023, doi: 10.1109/access.2023.3239135.