

Application of Artificial Intelligence on the Centrifugal Pump Operation

Mohammed Twheed Khater

Seif Ibrahim Hassan

2024-07-03

This research explores the application of artificial intelligence (AI) in predicting the effects of impeller trimming on centrifugal pump performance, focusing on the DAB Model KDN 100-250 - 2 POLES pump. Centrifugal pumps are vital in various industrial applications, where optimizing their performance often involves impeller trimming—mechanically reducing impeller diameter to adjust flow rate and head. Traditional methods, like constant-area scaling, provide approximate solutions but suffer from inaccuracies due to oversimplification of the trimming process. In contrast, this study employs neural networks optimized with genetic algorithms to predict performance changes more accurately. The research showcases neural networks' effectiveness in modeling complex relationships between impeller dimensions and pump performance metrics, achieving high accuracy in predicting trimmed diameter and power output. Results indicate significant improvement over traditional methods, highlighting AI's potential to enhance efficiency and reduce energy consumption in industrial pump systems.

Other Formats

PDF

MS Word

Chapter 1 | Prediction of the Effect of Impeller Trimming on Centrifugal Pump Performance Using AI

Introduction

Centrifugal pumps are widely used across various industries due to their reliability and efficiency in transporting fluids. However, achieving optimal performance often requires fine-tuning, particularly through impeller trimming. This mechanical process involves reducing

the impeller diameter to adjust the pump's head and flow rate, aligning performance with specific system requirements. This research investigates the application of neural networks, optimized by genetic algorithms, to predict the effects of impeller trimming more accurately than conventional methods.

What is Trimming in Centrifugal Pump Impeller?

Impeller trimming is the process of mechanically reducing the diameter of the pump impeller as shown in Figure 1. This adjustment directly influences the pump's head and flow rate, thereby modifying its performance characteristics.



Figure 1: impeller trimming process using lathe

Why Trimming?

System design engineers often round up system capacity and losses to account for construction changes, potential expansions, fluid viscosity changes, and pipe roughening over time. This practice leads to pumps running at higher pressures and flow rates than necessary, leading not only to wasting energy but also causing issues like cavitation and premature wear (Hydraulic Institute (2022)). Instead of throttling back powerful pumps or opening discharge bypass lines that recirculate a portion of the flow back to the suction, leading to energy loss, or using variable speed drives to reduce pump speed and head, a more efficient and cost-effective solution is impeller trimming. This process involves machining the impeller to reduce its diameter, thereby lowering the flow rate, pressure, and power consumption of the pump. While trimming can slightly reduce pump efficiency due to increased internal recirculation, the overall energy savings typically outweigh this drawback.

Manufacturers provide performance curves to guide impeller trimming, ensuring that the pumps operate efficiently within specified limits.

Energy Consumption

Centrifugal pumps are often responsible for a significant portion of the energy consumption in industrial settings. Trimming the impeller to match the exact system requirements can greatly reduce the energy consumption of the pump. By operating more efficiently, the pump uses less power, leading to substantial energy savings, where each 1 kW in pump delivery corresponds to 6 kW in power station energy.

Market Availability

The pumps available in the market may not always fit specific system requirements precisely. Typically, pumps are designed for a range of operations and may be larger or smaller than needed for a particular application. Impeller trimming allows for customizing the pump's performance to meet these specific needs, ensuring that the pump operates at optimal efficiency.

Energy Saving and Environmental Impact

As illustrated in Figure 2 (Hickok (1985)), energy flows through a power system experiencing significant losses at each stage, from the initial fuel input (e.g., a boiler) to the usable work performed by equipment. While minimizing thermal losses at the power plant (through combined cycle or cogeneration) offers substantial gains, focusing solely on the front-end might seem counterintuitive. After all, if only 15-20 kW of usable work remains from a 100 kW fuel input, why prioritize efficiency in motors and driven equipment? The answer lies in the leverage – every 1 kW saved at the utilization stage translates to a 6 kW reduction in fuel consumption at the power plant (Hickok (1985)). This dramatic impact underscores the importance of identifying and minimizing losses within process drives.

Traditional Trimming Methods

The essential issue in impeller trimming is understanding how the head and flow rate relate to a trimmed impeller diameter. This problem has been explored extensively when a centrifugal pump transports water. Previous studies aimed to find the relation between the head and the flow rate at a given operating point with an approximate trimmed impeller diameter. Affinity laws were employed to determine the size of the trimmed impeller that generates the head required by the piping system at the operating flow rate. The trimmed pump is strictly not similar to the initial pump because only the impeller outer diameter is modified, while all other dimensions remain unchanged. Despite this, the assumption of valid similarity was applied previously by several researchers and pump experts, which may lead to significant errors in their predictions (Elgohary, Khalil, and Shaito (2013)).

The affinity laws for centrifugal pumps are given by the following equations:

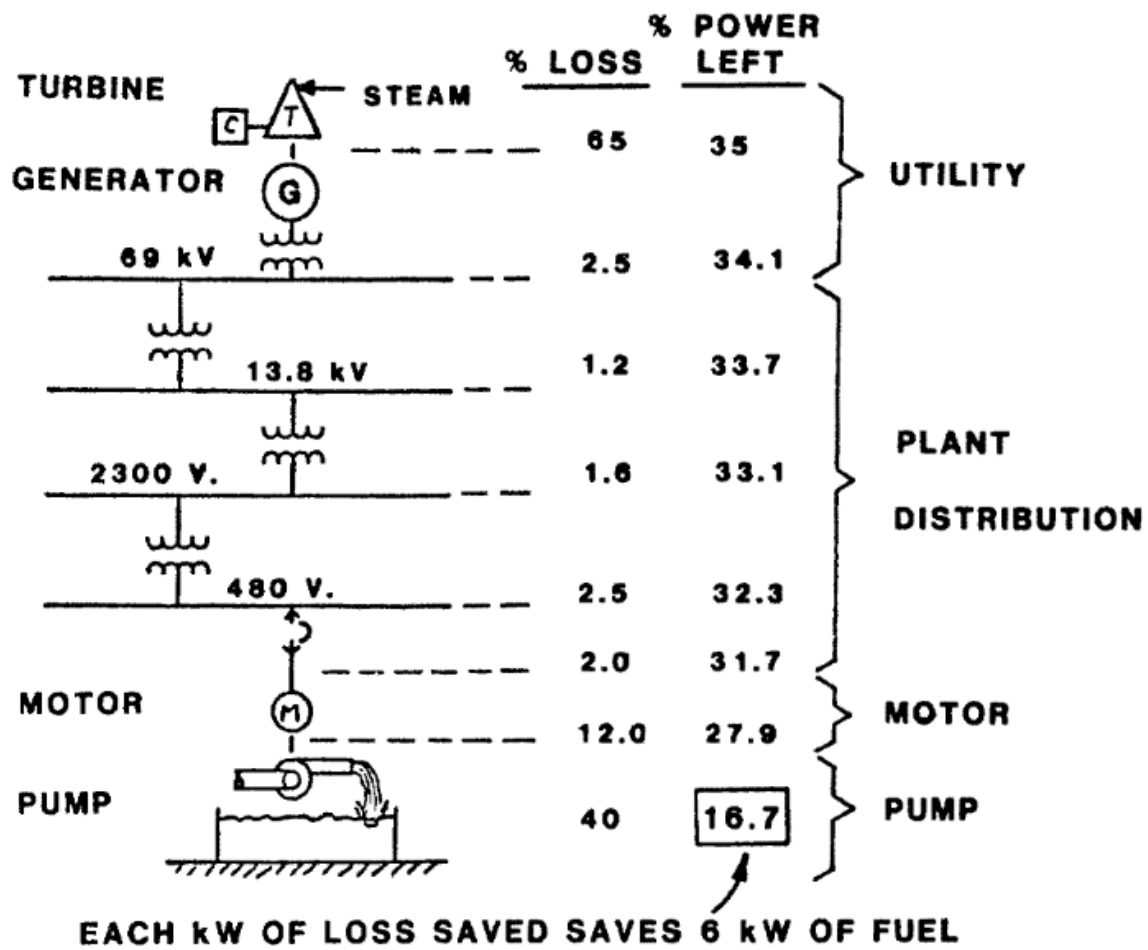


Figure 2: Energy flow and typical losses through power station

$$\text{Flow rate: } \frac{Q'}{Q} = \frac{D'_2}{D_2}$$

$$\text{Head: } \frac{H'}{H} = \left(\frac{D'_2}{D_2} \right)^2$$

$$\text{Power: } \frac{P'}{P} = \left(\frac{D'_2}{D_2} \right)^3$$

where: - Q and Q' are the original and new flow rates. - H and H' are the original and new heads. - P and P' are the original and new powers. - D_2 and D'_2 are the original and trimmed impeller diameters.

One of the common trimming methods is constant-area scaling , which assumes that the trimmed impeller maintains a constant area (Detert Oude Weme et al. (2018)), thereby ensuring proportional changes in both flow rate and head. This method involves reducing the impeller diameter in a manner that preserves the proportional relationship between the original and the trimmed impeller's performance metrics. The equations governing constant-area scaling are as follows:

$$\text{Constant-area scaling: } \frac{Q'}{Q} = \frac{D'_2}{D_2}, \quad \frac{H'}{H} = \left(\frac{D'_2}{D_2} \right)^2$$

where:

- Q' and H' represent the flow rate and head after trimming, respectively.
- D'_2 and D_2 represent the diameter of the trimmed and untrimmed impeller, respectively.

In this project, constant-area scaling serves as a benchmark to compare against the predictive capabilities of neural networks. By validating the neural network models against this traditional method, we can assess the accuracy and reliability of data-driven approaches in predicting trimmed pump diameter for a given operating point (Q , H).

Data Extraction and Digitization

Pump performance data supplied by manufacturer **DAB**¹ for pump **Model KDN 100-250 - 2 POLES**, including Q-H Figure 3 as and Q-power Figure 4 curves for different diameters, were digitized using on online tool **webplotdigitizer**² and analyzed using MATLAB. This data provided a foundation for training and validating the neural network models we developed in this project.

¹here is the manufacturer website : <https://www.dabpumps.com/en>

²it works online and can be downloaded to work in pc , here is the website link : <https://automeris.io/>

ambient temperature: +40°C

= 2900 1/min

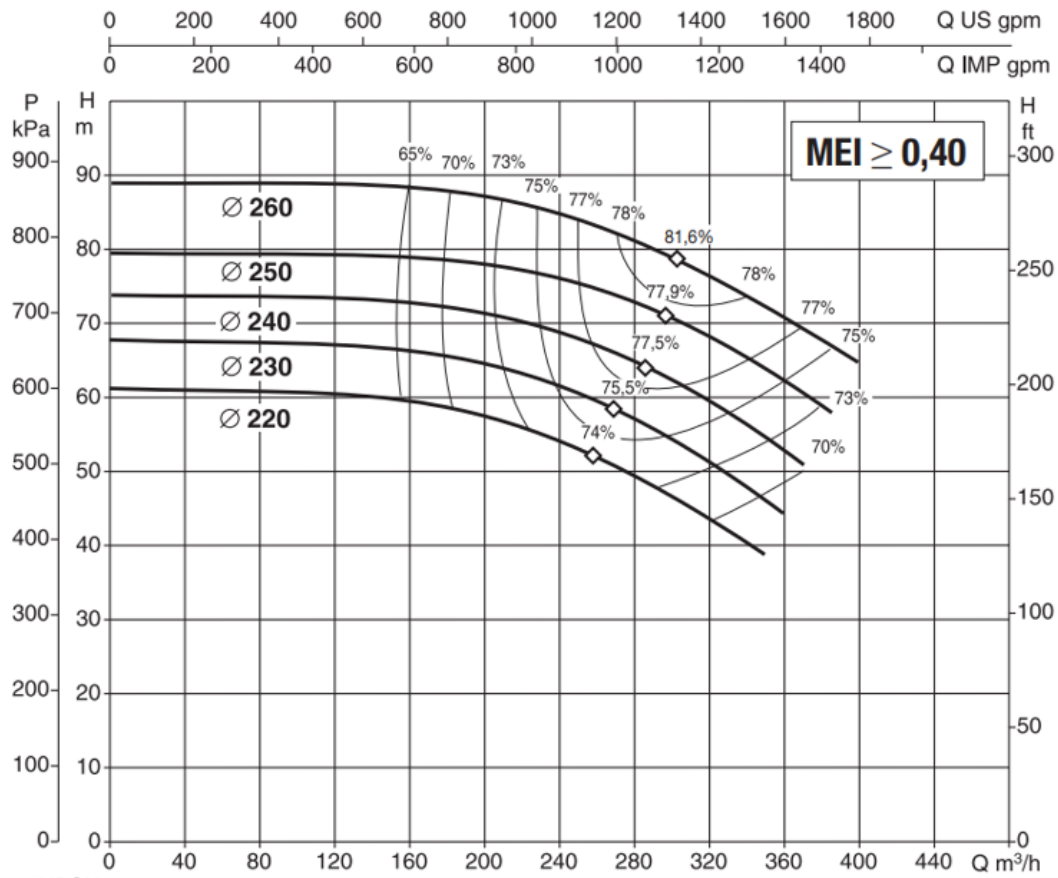


Figure 3: Q,H curves supplied by manufacturer DAB

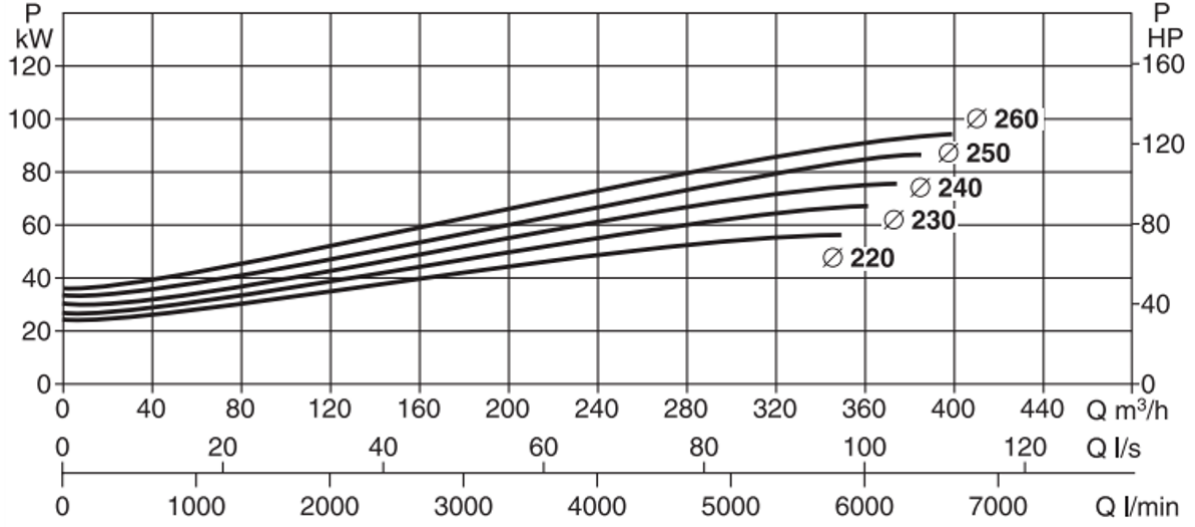


Figure 4: Q,P curves supplied by manufacturer DAB

after digitization we managed to re-produce $Q - H$, $Q - P$ and $Q - \eta$ curves and extract the best efficiency points as shown in Figure 5, Figure 6 and Figure 7:

now we can generate as much training points as we want for testing and validating neural nets we build and we also calculated the best efficiency points using :

$$P = \frac{Q * \rho * g * H}{\eta}$$

where:

- η is the pump efficiency (-)
- ρ is the fluid density (kg/m^3)
- g is the acceleration due to gravity (m/s^2)

Neural Networks

Neural networks are a class of machine learning models inspired by the structure and function of the human brain. They consist of interconnected layers of nodes, or neurons, where each neuron receives input, processes it through a weighted sum, applies a non-linear activation function, and passes the output to the next layer. The primary goal of a neural network is to learn the mapping from input data to output predictions by adjusting the weights during training, which is done using a backpropagation algorithm.

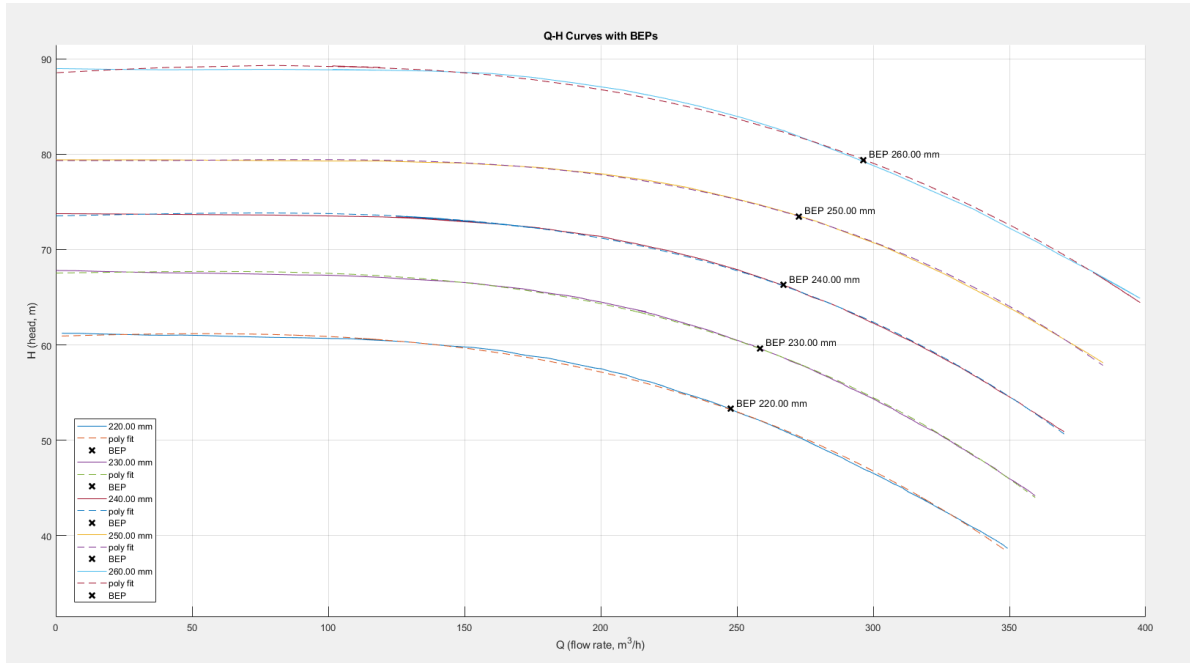


Figure 5: Q, H curves after digitization using webplotdigitizer and fitting using using matlab

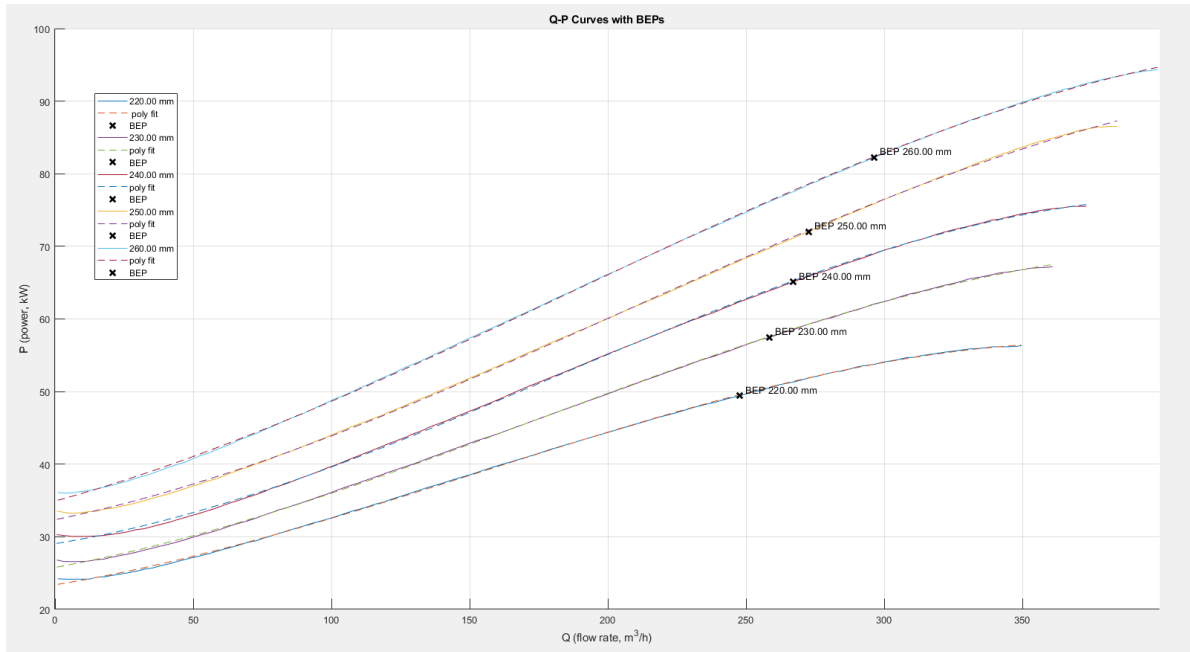


Figure 6: Q, P curves after digitization using webplotdigitizer and fitting using using matlab

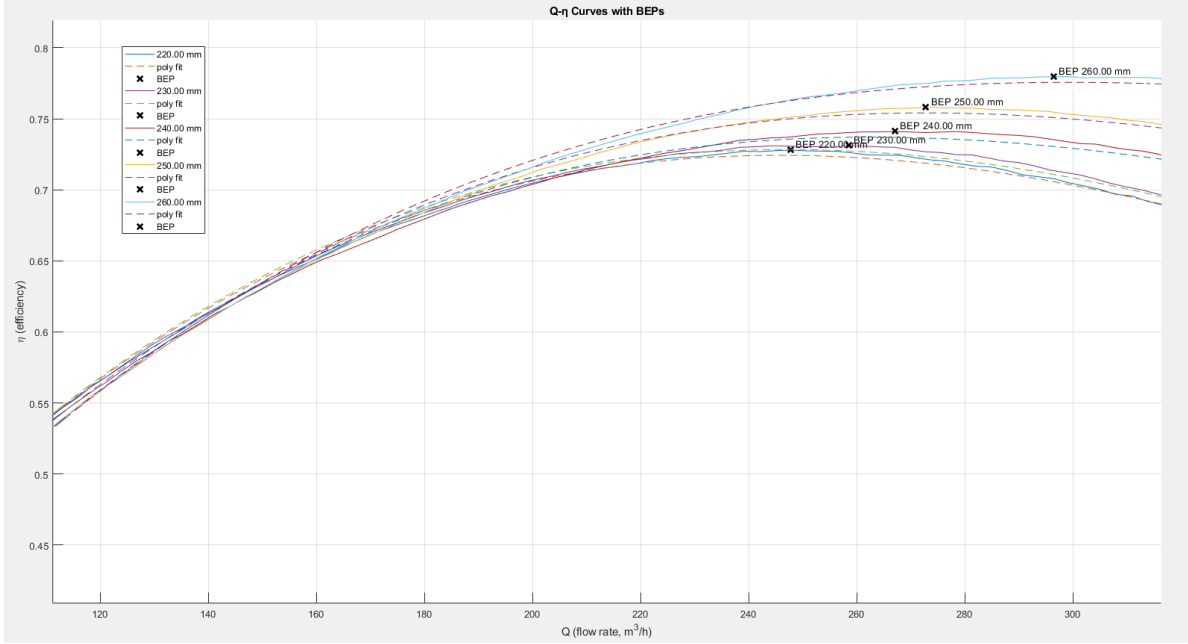


Figure 7: Q, η curves after digitization using webplotdigitizer and fitting using using matlab

In general, a neural network is composed of an input layer, one or more hidden layers, and an output layer. The layers are connected by weights that are adjusted during the training process to minimize the error between the predicted output and the actual target values. The training involves feeding the network with input data and targets, calculating the error, and using optimization algorithms like gradient descent to update the weights.

Usage in the Project

In our project, we employ neural networks to predict the trimmed pump diameter and power output based on given operating points, such as flow rate (Q) and head (H). The neural network is designed with multiple hidden layers to capture the complex relationships between the input and output variables.

We evaluated hyperparameters by building and training the neural network, then calculating the average mean square error (MSE) across training, validation, and testing datasets. The data is divided into 70% for training, 15% for validation, and 15% for testing. This division ensures that the model is trained effectively and its performance is validated on unseen data.

Despite the Adam optimization algorithm being the most cited across various knowledge domains due to its efficiency and effectiveness, we found that the ‘trainlm’ (Levenberg-Marquardt backpropagation) training method was the best option available in MATLAB for our relatively

small dataset. This method, ‘trainlm’, is particularly effective for training medium-sized networks quickly. We compared several training functions, including ‘trainbr’, ‘trainrp’, ‘traincgb’, ‘traincgf’, ‘traincgp’, ‘traingdx’, and ‘trainoss’, and found ‘trainlm’ to yield the most reliable results.

Similarly, for activation functions, we evaluated options such as ‘tansig’ (tangent sigmoid) and ‘logsig’ (logistic sigmoid). We found ‘tansig’ to be most effective in introducing non-linearity into the network.

$$f(x) = \text{tansig}(x) = \frac{2}{1 + \exp(-2x)} - 1$$

Data Processing and Division

Data preprocessing is a crucial step to ensure the model’s efficiency and accuracy. We used ‘removeconstantrows’ to eliminate constant rows that do not contribute to learning and ‘mapminmax’ to normalize the data, ensuring that all input features are scaled within a specific range.

The performance of the network was evaluated using the mean squared error (MSE), which measures the average squared difference between predicted and actual values. The formula for MSE is:

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

where y_i is the actual value and \hat{y}_i is the predicted value.

Additionally, the coefficient of determination, R^2 , was used to assess the goodness of fit of the model. The formula for R^2 is:

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2}$$

where \bar{y} is the mean of the actual values.

To ensure the robustness of the model, the data was divided randomly into training, validation, and testing sets using the ‘dividerand’ function. The division mode was set to ‘sample’, with 70% of the data allocated for training, 15% for validation, and 15% for testing. This approach allowed the model to be trained and validated on different subsets of data, enhancing its performance on unseen data.

Determining Optimal Parameters with Genetic Algorithm

Based on our pump curves dataset, we decided to work exclusively with deep feedforward neural networks. However, to ensure optimal performance, we needed to determine the following parameters: - Number of hidden layers - Number of hidden neurons in each layer - Activation function - Training algorithm

To achieve this, we employed a genetic algorithm (GA) to optimize these parameters. The GA evaluated different configurations of the neural network to find the combination that minimized the MSE. The 'trainlm' method, with the 'tansig' activation function, proved to be the most effective in our evaluations.

Results and Analysis

Our optimal neural network architectures were [2, 7, 29, 17, 1] for predicting power from flow rate and diameter (Q,D) \rightarrow P here the input layer is 2-neurons for the two inputs Q and D and 1 for output P and 3 hidden layers first is 7 neurons then 29 then 17 as shown in Figure 8 , and [2, 5, 15, 1] for predicting diameter from flow rate and head (Q,H) \rightarrow D as shown in Figure 9 , with the 'trainlm' training method , 'tansig' activation function and 355 epochs for all of them yielding the best results.

The results show that R for all NNs in all stages is close to 1 as shown in Figure 11 for the (Q,D) \rightarrow P NN model and Figure 13 for the (Q,H) \rightarrow D NN model , and the average of mse among all models we trained was 0.004 as shown in Figure 10 and Figure 12 . The results indicate a good agreement between the model values and the experimental values. Similar results were obtained when using the designed Neural Network for another centrifugal pump (TACO pump) by (Elgohary, Khalil, and Shaito (2013)).

Genetic Algorithm

volutionary algorithms are a class of optimization techniques inspired by the process of natural selection, a cornerstone of Charles Darwin's theory of evolution. These algorithms evolve a population of candidate solutions over multiple generations to solve complex optimization problems. Among these, the genetic algorithm (GA) is one of the most popular and effective methods. GAs mimic the process of natural evolution, utilizing operations such as selection, crossover, and mutation to iteratively improve a population of solutions.

The key steps involved in the genetic algorithm include as shown Figure 14 : 1. **Initialization:** Creating an initial population of potential solutions with random hyperparameters. 2. **Selection:** Evaluating the fitness of each individual in the population based on the mean squared error (MSE) of the neural network's predictions. 3. **Crossover:** Combining pairs of individuals to produce offspring with mixed characteristics, promoting the inheritance of good traits. 4. **Mutation:** Introducing random changes to some individuals to maintain

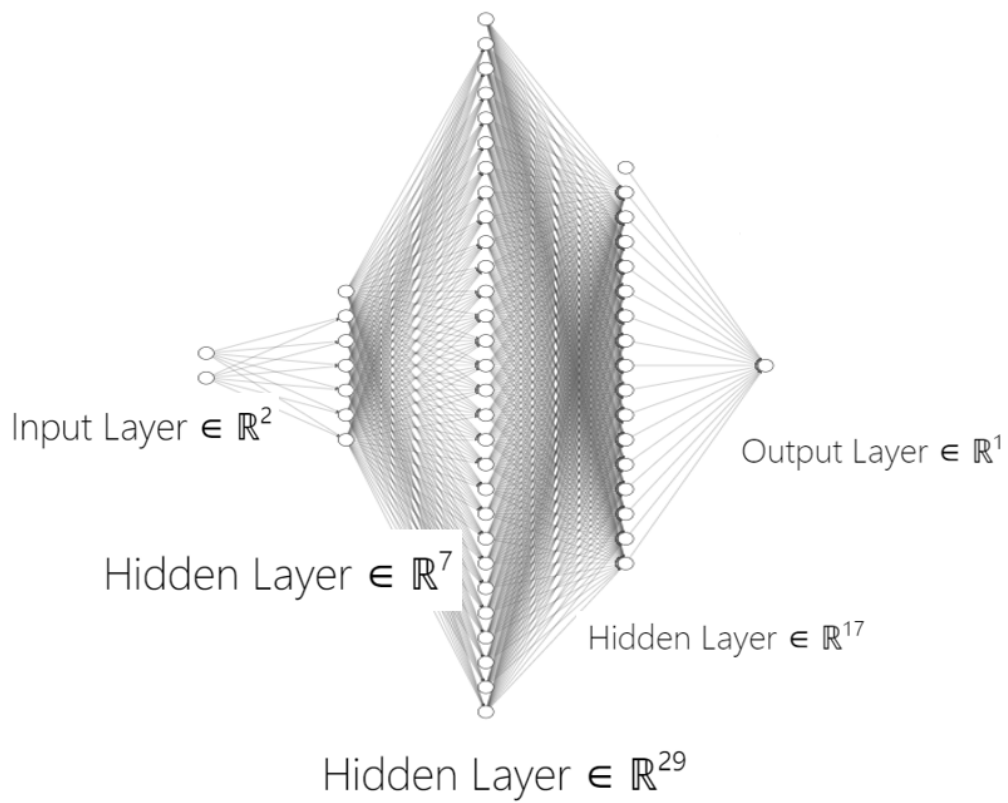


Figure 8: best architecture for $(Q,D) \rightarrow P$, for DAB pump Model KDN 100-250 - 2 POLES

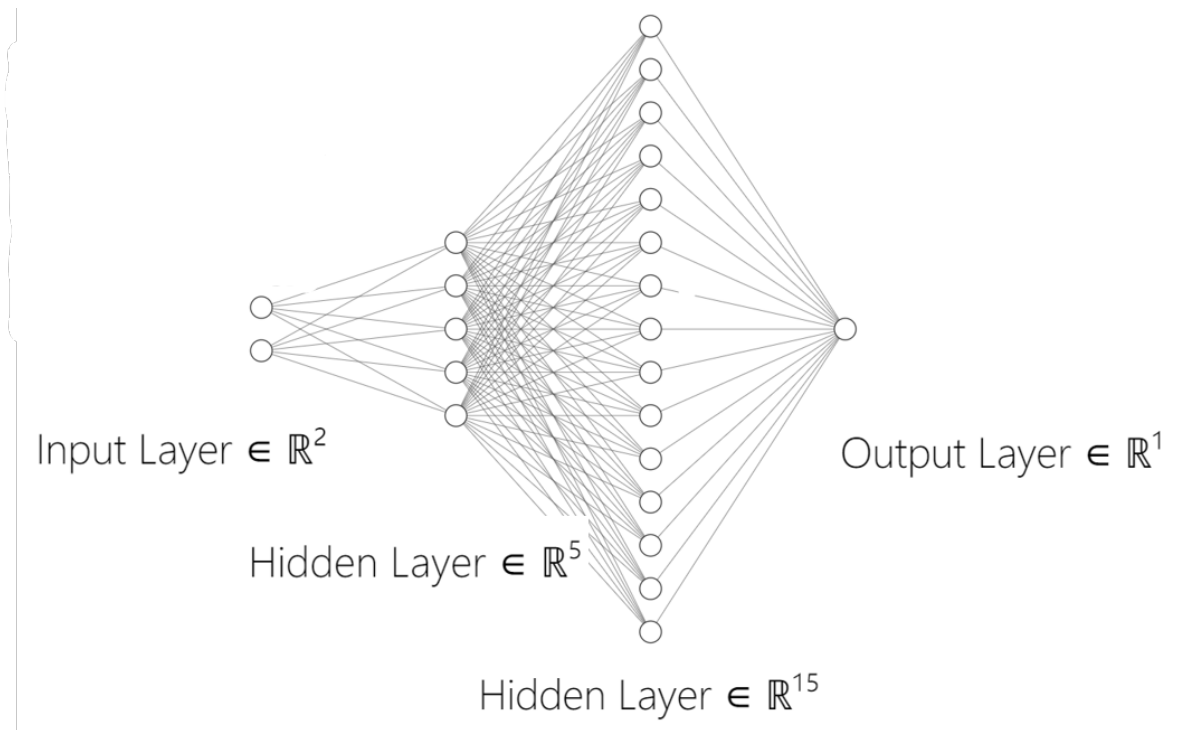


Figure 9: best architecture for $(Q,H) \rightarrow D$, for DAB pump Model KDN 100-250 - 2 POLES

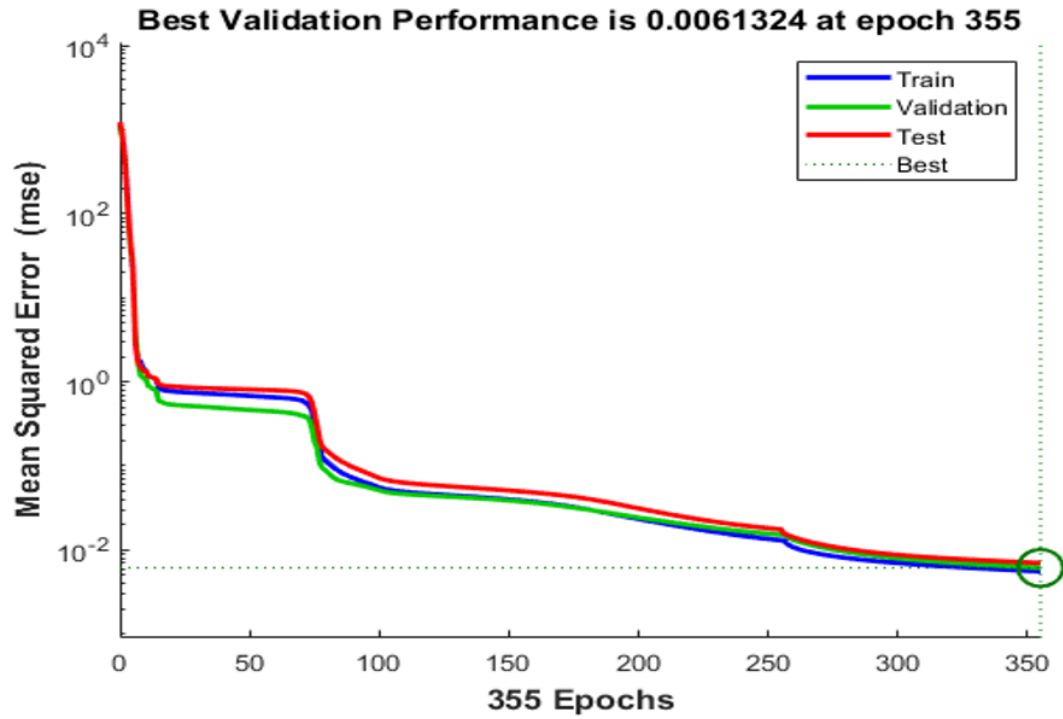


Figure 10: Training, validation and testing MSE for our best Neural network model for (Q,D)
 -> P for DAB pump Model KDN 100-250 - 2 POLES

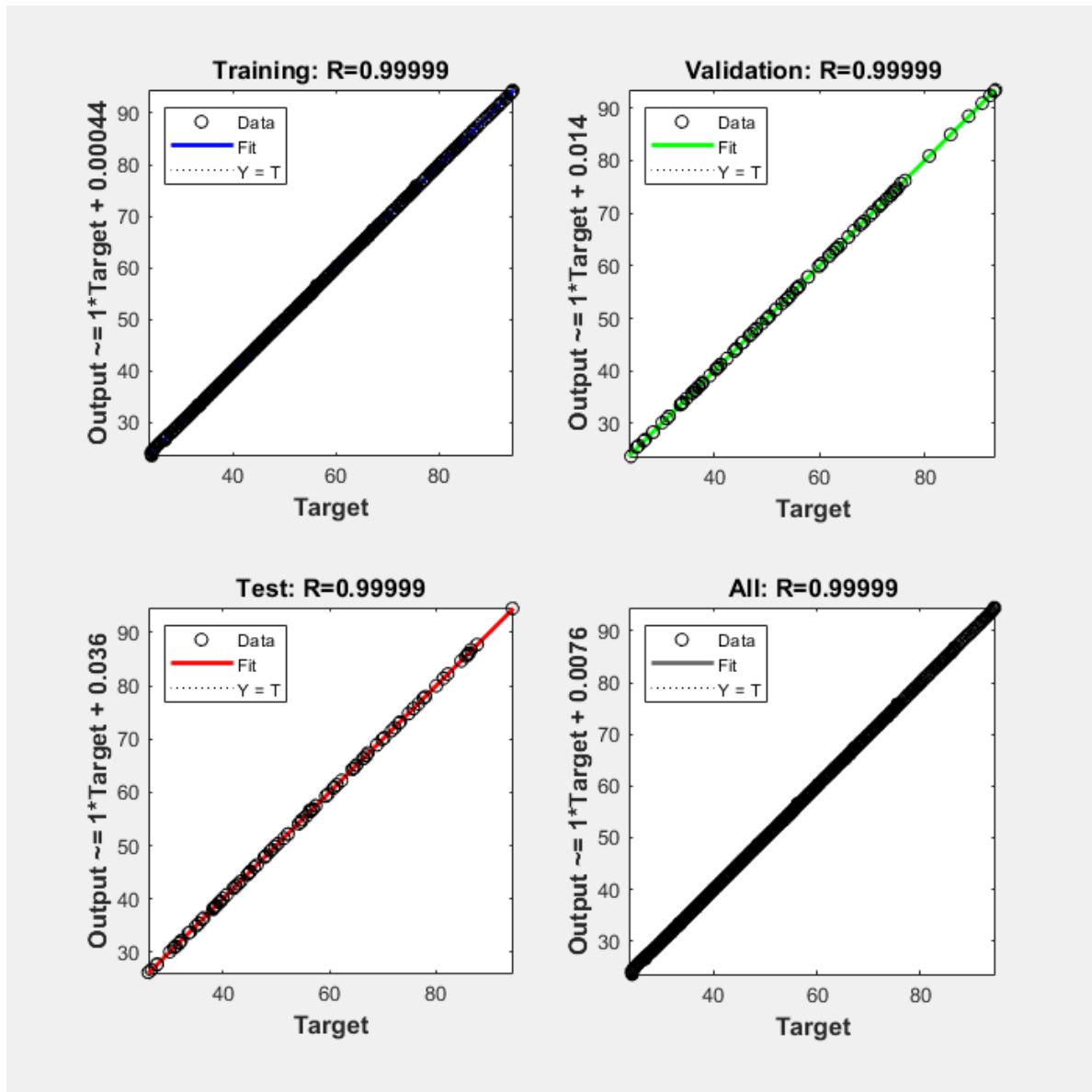


Figure 11: Regression Curve (Q,D) \rightarrow P, its a Comparison between predicted and actual values for DAB pump Model KDN 100-250 - 2 POLES

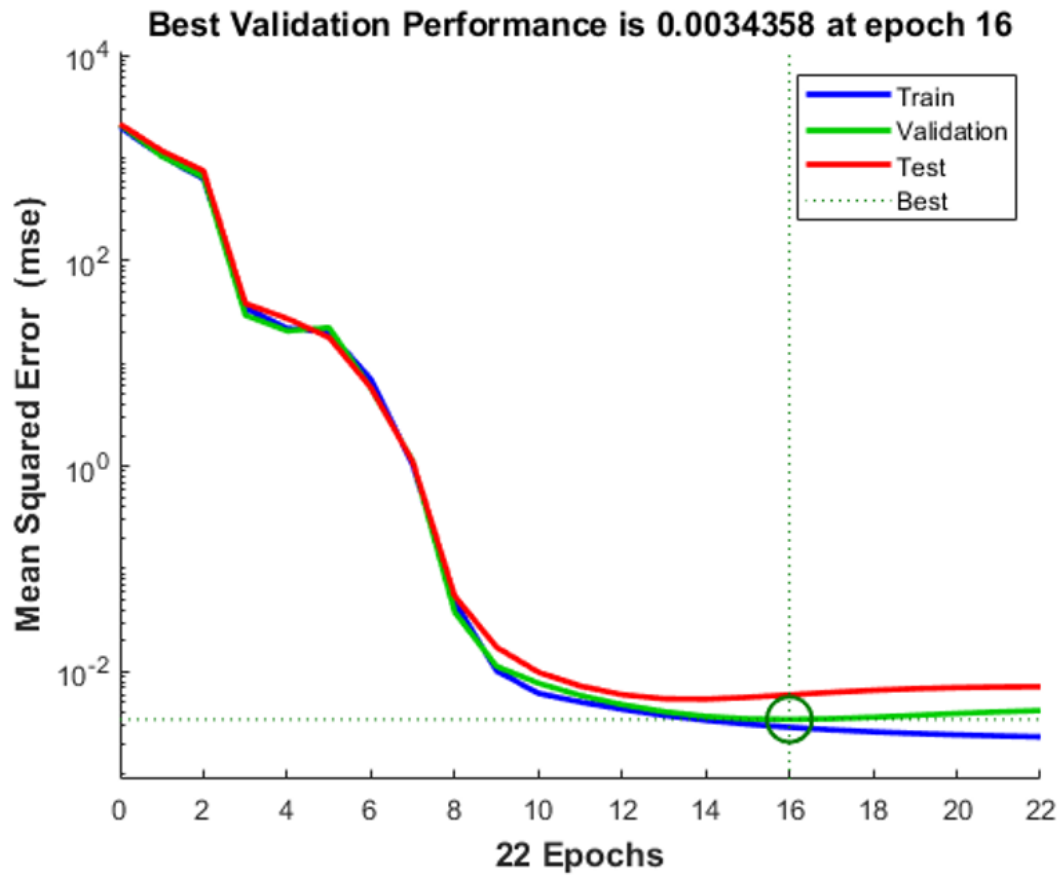


Figure 12: Training, validation and testing MSE for our best Neural network model for (Q,H)
 -> D for DAB pump Model KDN 100-250 - 2 POLES

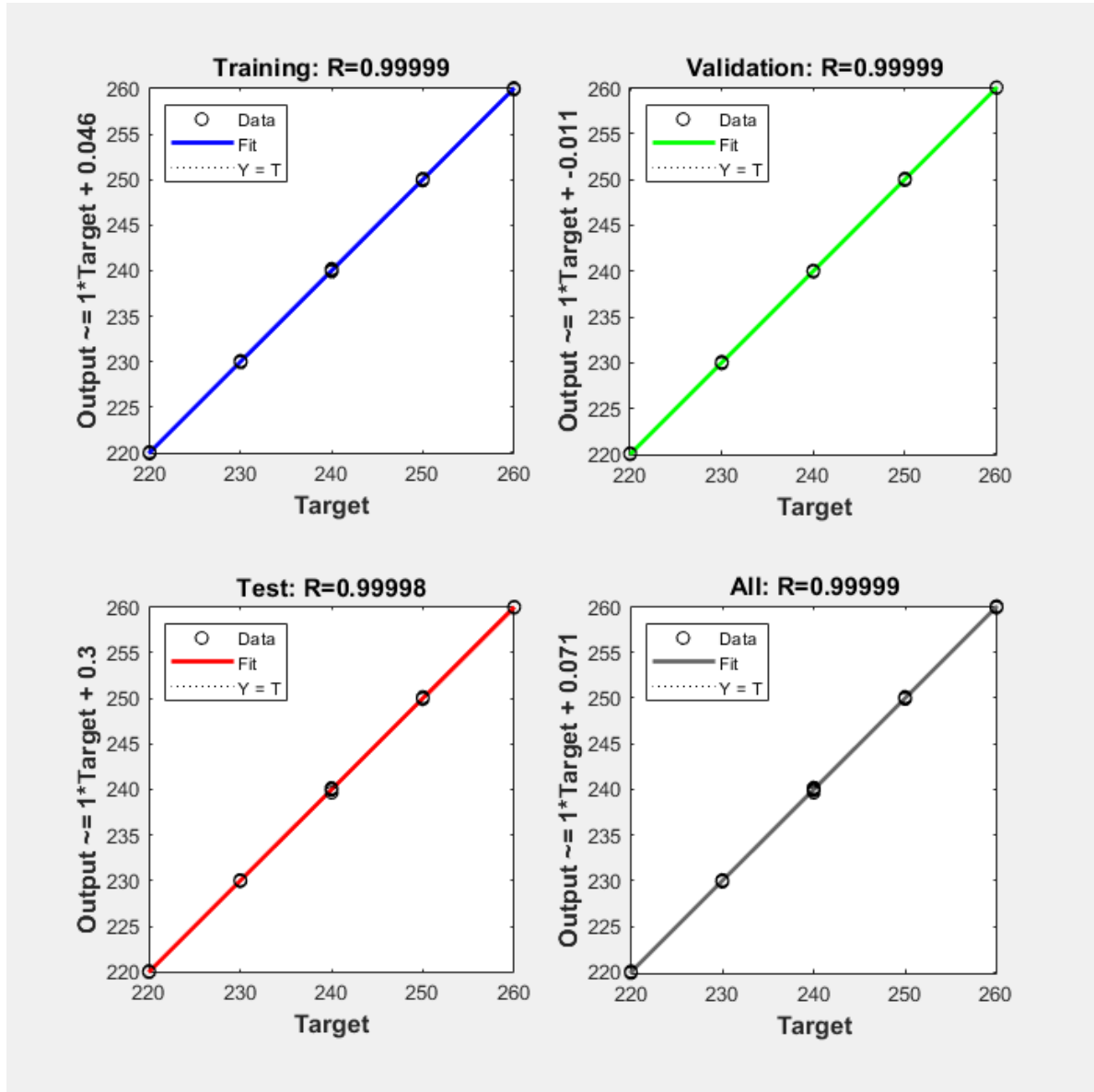


Figure 13: Regression plots for actual and predicted results for trimmed impeller diameter using our final best Neural network model (Q,H)-> D for DAB pump Model KDN 100-250 - 2 POLES

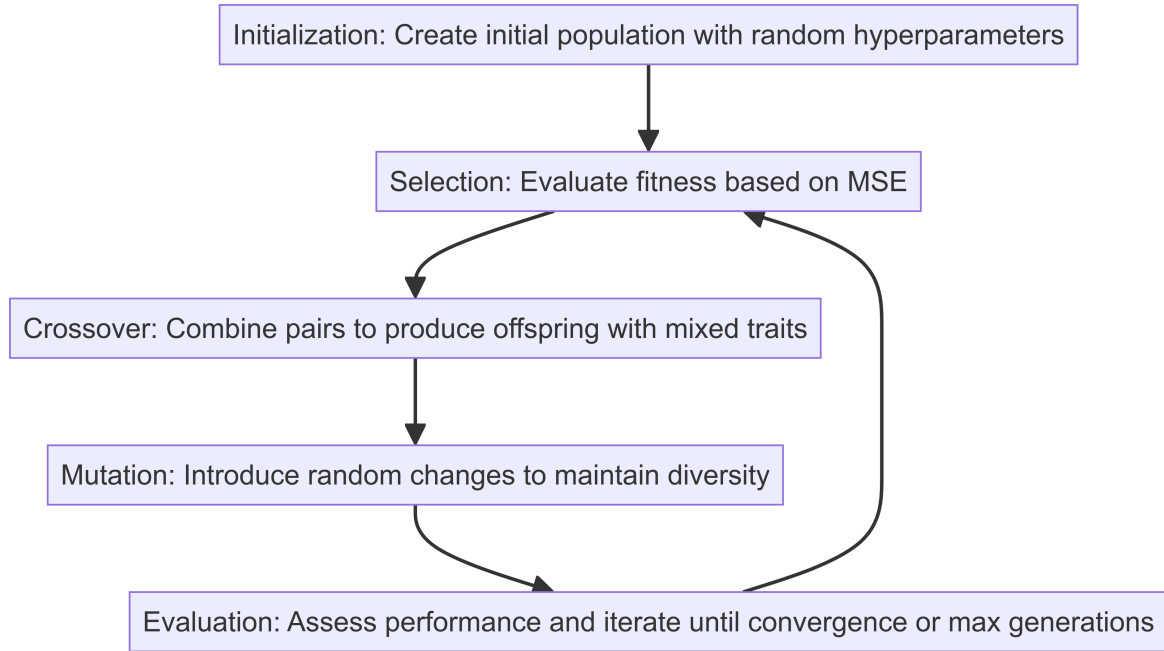


Figure 14: genetic algorithm general working sequence

genetic diversity and explore new solutions. 5. **Evaluation:** Assessing the performance of the new population and iterating through the selection, crossover, and mutation steps until convergence or a predefined number of generations is reached.

Application of Genetic Algorithm in Our Project

In our project, we leveraged a genetic algorithm to optimize the hyperparameters of our deep feedforward neural networks. This approach was chosen due to its robustness in exploring large, complex search spaces and finding near-optimal solutions for problems with many variables. Our objective was to determine the optimal number of hidden layers, the number of neurons in each layer, the activation function, and the training algorithm for our neural networks.

Methodology

We began by defining the bounds for our hyperparameters. These bounds constrained the possible configurations of our neural network to a feasible range. Specifically, we set the following bounds:

Number of hidden layers: between 2 and 17
Number of neurons in each hidden layer: between 9 and 95
Training algorithm options
Activation function options
These bounds ensured that the

GA explored a diverse set of neural network architectures without venturing into impractically large or small configurations.

The MATLAB `ga` parameters were carefully chosen to balance exploration and exploitation based on best practices :

- Population Size: 17 - This parameter defines the number of candidate solutions (individuals) in each generation. A larger population size increases diversity but requires more computational resources.
- Max Generations: 13 - This parameter sets the maximum number of generations the GA will run. Each generation represents a complete cycle of selection, crossover, and mutation.
- Crossover Fraction: 0.8 - This parameter determines the fraction of the population selected for crossover (mating) to produce the next generation. A higher crossover fraction encourages mixing of genetic material, promoting diversity.
- Constraint Tolerance: 0.000991 - This is the tolerance level for constraint violations. Solutions that exceed this tolerance are considered infeasible.
- Fitness Limit: 0.000991 - This parameter sets a threshold for the fitness function. Once a solution reaches this fitness level, the GA terminates, assuming it has found a sufficiently optimal solution.
- Elite Count: 2 - This parameter specifies the number of top-performing individuals that are directly passed to the next generation without undergoing crossover or mutation, ensuring that the best solutions are preserved.
- Display: 'iter' - This option controls the display of the GA's progress. Setting it to 'iter' enables iterative display, showing updates after each generation.
- Parallel Execution: Enabled - This option allows the GA to use parallel processing, significantly improving computational efficiency by distributing the workload across multiple processors.

To ensure reproducibility, a random seed (4826) was set for the random number generator. The optimization process involved evaluating the fitness of each candidate solution using a predefined fitness function, which assessed the performance of the neural network configuration by training it on our dataset and calculating the mean squared error (MSE).

The GA iteratively improved the population of neural networks by selecting the best-performing individuals, performing crossover to combine their features, and introducing mutations to explore new configurations. This evolutionary process continued until the GA converged to a solution that met our fitness criteria.

Comparison of NN-Based vs Constant Area Scaling Method

The neural network models were compared with traditional constant-area scaling methods. By removing a complete diameter curve from the training data, the networks were tested for

their predictive accuracy. The AI-based approach proved more effective, particularly when the removed diameter was not the smallest or largest, highlighting the robustness of neural network predictions.

Analysis of Final Results

The Table 1 below presents the final results of the neural network predictions compared to traditional methods. The AI-based approach consistently outperformed the constant-area scaling method, offering more precise and reliable predictions of pump performance post-trimming. also it shows that training the aforementioned NNs on the full dataset without removing any complete single (Q,H) curve or (Q,P) curve will yield a better results but it also performed well even when a complete diameter curve is removed for the training data set which means our chosen architectures are capturing the topology of the data very well.

Table 1: Final Results

Diameter Removed	Avg MSE	Train Per- formance	Validation Performance	Test Per- formance	MSE Deleted Diameter	MSE BEPs	Score
QHD Results							
NaN	0.00392850	0.0028838	0.0043358	0.0059646	NaN	0.0054126	0.0033419
220	0.030093	0.0091516	0.027962	0.063279	62.305	14.482	27.842
230	2.1664	0.0049077	0.74021	6.7737	73.774	11.816	34.281
240	0.00439620	0.0016072	0.0019346	0.010973	0.19113	0.0241550	0.085218
250	0.00088785	0.00096387	0.00075212	0.00090432	56.083	19.901	26.414
260	0.00170870	0.0008838	0.0024402	0.002194	45.295	5.7397	19.267
QDP Results							
NaN	0.00614750	0.0055729	0.0061324	0.007011	NaN	0.0149410	0.0057486
220	0.005917	0.0048547	0.0034999	0.0090041	290.19	88.169	133.71
230	0.00136120	0.0011326	0.001204	0.0018558	0.55901	0.0480940	0.23393
240	0.084471	0.01403	0.018891	0.22013	103.78	16.451	42.746
250	0.071431	0.01074	0.014046	0.19024	118.19	32.507	50.596
260	0.04818	0.013519	0.034681	0.096339	62.805	12.708	36.864
QDH Results							
NaN	0.00146830	0.0013138	0.0014464	0.0016446	NaN	0.0013515	0.0012684
220	0.00170050	0.00098342	0.0020148	0.0021028	0.014518	0.0041745	0.0027077
230	0.029849	0.0080034	0.02848	0.052156	65.481	18.064	27.468

Diameter Removed	Avg MSE	Train Per- formance	Validation Performance	Test Per- formance	MSE		
					Deleted Diameter	MSE BEPs	Score
240	0.018048	0.0041386	0.016647	0.041358	45.752	12.684	23.218
250	0.00072173	0.0005285	0.00055663	0.0010802	0.064235	0.0049490	0.0036413
260	0.00514710	0.0021419	0.0072202	0.0060793	24.932	4.9345	11.144

Where :

-Removed Diameter: is the Diameter corresponding to the removed complete curve from the training dataset and used for validation where the mse error based on it is in column MSE Deleted Diameter.

- **AvgMSE (Average Mean Squared Error):** Indicates the average error between the predicted and actual values. Lower values signify better model accuracy.
- **Train Performance:** Reflects the model's performance on the training dataset. Lower values indicate better fitting to the training data.
- **Validation Performance:** Indicates the model's performance on the validation dataset, used to tune hyperparameters and prevent overfitting.
- **Test Performance:** Represents the model's performance on the test dataset, providing an unbiased evaluation of the model's accuracy.
- **MSE Deleted Diameter:** Shows the mean squared error when the specific diameter is excluded from the training data, assessing the model's robustness.
- **MSE BEPs (Best Efficiency Points):** Highlights the model's accuracy at the best efficiency points, crucial for optimizing pump performance.
- **Score:** A composite metric combining various performance indicators to provide an overall assessment of the model's effectiveness. it is calculated as :

$$S = 0.05 \cdot Validation + 0.35 \cdot Test + 0.40 \cdot MSE_{Deleted} + 0.20 \cdot MSE_{BEPs}$$

Conclusion

This study demonstrates the superiority of AI-based models in predicting the effects of impeller trimming on centrifugal pump performance. The neural network models, optimized through genetic algorithm matlab toolbox **ga**, consistently outperformed traditional constant-area scaling methods. These findings underscore the potential of AI to enhance pump efficiency, offering significant energy savings and operational improvements.

- Detert Oude Weme, D. G. J., M. S. Van Der Schoot, N. P. Kruyt, and E. J. J. Van Der Zijden. 2018. “Prediction of the Effect of Impeller Trimming on the Hydraulic Performance of Low Specific-Speed Centrifugal Pumps.” *Journal of Fluids Engineering* 140 (8): 081202. <https://doi.org/10.1115/1.4039251>.
- Elgohary, Mohamed A, M Farid Khalil, and Ali A Shaito. 2013. “Application of Artificial Neural Network for the Prediction of Trimmed Impeller Size of a Centrifugal Pump.”
- Hickok, Herbert N. 1985. “Adjustable Speed—A Tool for Saving Energy Losses in Pumps, Fans, Blowers, and Compressors.” *IEEE Trans. On Ind. Applicat.* IA-21 (1): 124–36. <https://doi.org/10.1109/TIA.1985.349672>.
- Hydraulic Institute. 2022. “Trimming Impellers to Reduce Energy Consumption.” September 27, 2022. <https://www.pumps.org/2022/09/27/trimming-impellers-to-reduce-energy-consumption/>.