International Journal of Mechatronics and Applied Mechanics
2026, Issue 25, Vol. II

EFFECT OF NORMALIZATION METHODS ON PREDICTION THE LOAD TORQUE OF DIRECT CURRENT SHUNT MOTOR BY USING ARTIFICIAL NEURAL NETWORK

Sertac Savas[1], Mehmet Safa Bingo[2], Ahmet Kirnap[1], Sahin Yildirim[1*]
[1]Mechatronics Engineering Department, Engineering Faculty, Erciyes University, Kayseri, 38039, Turkey
[2]Mechatronics Engineering Department, Engineering Faculty, Nigde Ömer Halisdemir University, Nigde, 51240, Turkey
E-mail: sahiny@erciyes.edu.tr

Abstract - Direct current (DC) motors are electrical machines that are widely used in industrial applications and offer various advantages. These motors are especially preferred in different sectors such as robotic systems, factories, textile industry and machine tools due to their ability to operate in wide speed ranges, low cost and easy controllability. For DC shunt motors to operate efficiently, it is critical to accurately predict the load torque of the motor. In this study, the effect of normalization methods on the prediction of the load torque of a DC shunt motor is investigated. For this purpose, firstly, the DC shunt motor is modeled in MATLAB Simulink environment and a data set is created by operating the motor under different voltage (V_a) and load torque (τ_L) values. Then, the obtained data are normalized with 5 different normalization methods. These normalization methods play an important role in making the data suitable for the artificial neural network (ANN) model. The designed ANN is trained using the normalized data and the effects of different normalization techniques on ANN performance are analyzed in detail. Performance is evaluated using metrics such as error rates and learning rates, with the results presented in both tables and graphical representations.

Keywords: DC Shunt Motor, Normalization, ANN, Load Torque.

1. Introduction

Direct current (DC) motors are widely used in automation and robotic systems due to their wide speed control range and ease of control. Especially shunt DC motors attract attention with their stable speed and torque characteristics. Effective control of these motors requires accurate modelling and appropriate control strategies. Artificial neural networks (ANNs) are among the data-driven methods frequently preferred for this purpose. Data preprocessing, especially normalization techniques, play an important role in improving ANN performance. There are numerous studies on the effects of these techniques in the literature.

Beltran et al. developed a velocity control method that is independent of unmeasurable external disturbances to ensure stable operation of nonlinear shunt DC motors under variable load torque and parameter uncertainties. With the integration of artificial neural networks and dynamic error compensators, the adaptability of the system to dynamic uncertainties is increased and the dependence on predetermined system parameters is eliminated. The obtained analytical and numerical results demonstrate the robustness of the proposed control method [1]. Tapia-Olvera et al. proposed an adaptive speed control method based on B-spline neural networks. The proposed method aims to provide speed control by only measuring the speed of the motor without the need for any other data. Experimental results showed the effectiveness compared to other methods [2]. Akshata et al. developed ANN-based models for speed and torque estimation of DC shunt motors. The prediction results of the feed-forward ANN trained with experimental data provided high prediction accuracy [3]. Beltran-Carbajal et al. proposed an algebraic method for online estimation of parameters and variable mechanical torque for shunt DC motors. Dynamic control gains were calculated with a B-spline ANN. Simulations confirmed the rapid and accurate prediction capability of the method [4]. Choudhary et al. developed an ANN-based model for speed estimation of DC shunt motors.

dx.doi.org/10.17683/ijomam/issue25/v2.11

The network trained with the Levenberg-Marquardt algorithm successfully predicted the motor speed [5]. Valluru and Singh in their study obtained better performance in speed control of shunt DC motor using NARMA L-2 neural controller when compared to conventional PID controller. This study highlights the superiority of neural network based controllers in nonlinear systems [6]. Antar and Allu proposed an ANN-based model reference adaptive system for sensorless speed and torque control. Simulations showed that the method gives better results under different speed and load conditions [7].

Aksu et al. examined the effect of different normalization methods on the performance of artificial neural networks. Min-max normalization method showed better performance, especially for large sample sizes. The study reveals the critical role of normalization methods in model performance [8]. Bhanja and Das examined the effect of normalization methods used in deep neural networks for time series forecasting. The Tanh Estimator normalization method gave more effective results with lower error rates compared to other methods [9]. Sola and Sevilla examined the importance of input data normalization for ANN applications in nuclear power plants. They stated that the appropriate normalization method reduces prediction errors and decreases the training time [10]. Simtharakao and Sutivong examined the effect of normalization methods used for Bitcoin price prediction. Based on the RMSE and MAPE measures, the normalization method that gave the best result with the GRU model was sliding window normalization [11]. Kim studied the normalization of input and output vectors in backpropagation ANNs and stated that different normalization methods affect the prediction accuracy [12]. Djordjević et al. analyzed the effect of normalization on ANN performance in photoacoustic applications. They emphasized that Min-Max normalization shows the best prediction performance [13]. Sedghisigarchi et al. compared three different nonlinear control methods for speed control of a DC shunt motor. The input-output linearization method gave better results than the other methods. Meanwhile, the fuzzy logic based controller provided effective control without the need for a mathematical model [14]. Alawan and Al-Furaiji designed an online PID controller supported with genetic algorithms for speed control of a DC shunt motor. Simulations showed that the controller provides speed stability under sudden load changes [15]. Chiley et al. stated that their proposed online normalization method offers less memory usage and wider application area compared to batch normalization [16]. Puheim and Madarasz investigated the normalization methods of the input and output data of the artificial neural network based controller used in the inverse kinematics of the robotic arm [17].

Beyond these application specific studies, several foundational works have established data normalization as a decisive factor in NN training. LeCun et al. showed that centring and scaling the network inputs conditions the error surface and accelerates gradient based learning [26]. Jayalakshmi and Santhakumaran reported that statistical normalization applied prior to backpropagation markedly improves both the classification accuracy and the convergence speed of the network [27]. Ioffe and Szegedy introduced batch normalization, which stabilizes the distribution of layer activations during training and has since become a standard component of deep architectures [28]. More recently, Singh and Singh carried out a broad empirical comparison and demonstrated that no single normalization technique is universally optimal; the best choice is problem and data dependent [29]. Taken together, these studies confirm that normalization is not a routine preprocessing step but a design decision that directly governs model accuracy, training time and numerical stability.

Despite this extensive body of work, two gaps remain. First, most comparative normalization studies target classification or generic time series benchmarks, whereas the regression based prediction of electromechanical quantities such as the load torque of a DC machine has received little attention. Second, the existing DC motor studies that employ ANNs adopt a single, fixed scaling scheme and do not examine how the choice of normalization influences the resulting model. To the best of the authors' knowledge, no prior study has systematically compared multiple normalization methods for ANN based load torque prediction of a DC shunt motor across a range of training epochs. The present work addresses this gap by evaluating the raw data together with five normalization methods (Z-Score, Min-Max, Median, D-Min-Max and Linear) under identical network and training conditions, thereby isolating the effect of the scaling method itself. The scientific contribution is therefore twofold: (i) it provides the first controlled, quantitative comparison of normalization methods for this specific regression problem, and (ii) it identifies, with statistical support, the scaling strategies that yield the fastest convergence and the lowest prediction error for DC motor applications.

In this study, the effect of different normalization methods on the prediction of the load torque of a DC shunt motor with ANN is investigated in detail. In the literature, normalization techniques have been shown to significantly affect ANN performance in various fields; however, studies on the effects of these techniques on DC motor systems are limited. The data obtained by modeling the DC shunt motor under different voltage and load torque values in MATLAB Simulink environment are prepared as input to the ANN model using five different normalization methods. Thus, by comparatively analyzing the effects of each normalization technique on model accuracy, learning speed and error rates, it is aimed to determine the most appropriate preprocessing method for motor control applications.

To achieve this aim, the study is carried out in the following consecutive steps:

1) The DC shunt motor is modelled in the MATLAB/Simulink environment using its governing differential equations (Eqs. 1–3).

2) A data set is generated by operating the model under systematically varied armature voltage (V_a) and load torque (τ_L) values, and the corresponding V_a, I_a, ω and τ_L values are recorded.

3) The data set is divided into training and testing subsets and is scaled using the raw form together with the five normalization methods (Z-Score, Min-Max, Median, D-Min-Max and Linear).

4) A feed-forward backpropagation ANN with a 3–6–4–1 architecture is designed.

5) The network is trained separately with each data set using the Levenberg–Marquardt algorithm for 25, 100, 1000 and 10000 epochs.

6) The performance obtained with each scaling case is evaluated using the MSE and R^2 metrics and is compared statistically in order to determine the most suitable preprocessing method for DC motor load torque prediction.

2. Material and Method

2.1. DC Shunt Motor and Model

There are many different types of DC motors. These varieties can be listed as shunt, separated excitation, series, and compound [18]. In this study, a shunt motor with armature and field windings connected in parallel is used. DC shunt motor provides important features where the rotor speed does not change considerably as the load torque varies from zero to its nominal value [4].

The armature and the field windings are connected in parallel configuration in a DC shunt motor. So, the armature current and field current are supplied by the same source [19]. Figure 1 shows a schematic diagram of the DC shunt motor at this study.

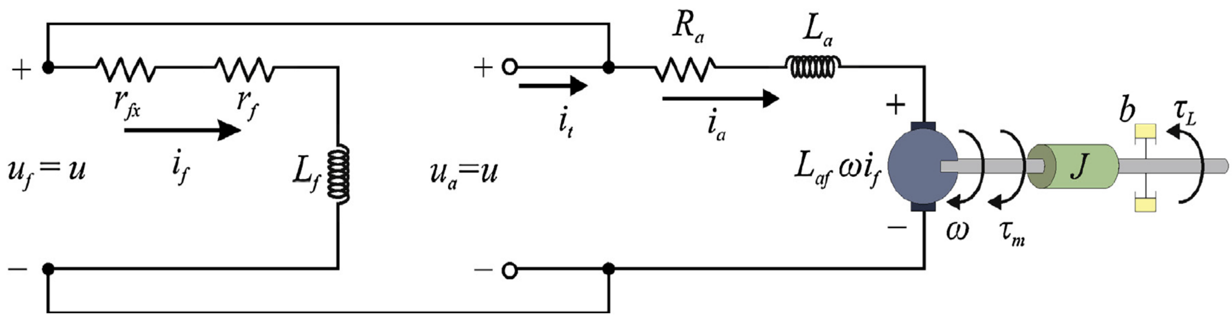

Figure 1: Schematic diagram of DC shunt motor [1]

While defining the parameters given in Figure 1, two main headings can be used as field windings and armature windings. E_a is the back EMF voltage, k is the machine constant, J is the moment of inertia, and B is the friction coefficient. L_f is the field winding inductance, R_f is the field resistance, R_{fx} is an external variable resistance, if is the field current, and V_f is the field voltage. R_a is the armature resistance, L_a is the armature winding inductance, V_a is the armature voltage, and i_a is the armature current. w is the rotor speed and τ_L is the load torque [2].

The differential equations obtained using the schematic diagram of the DC shunt motor are given separately as Equation 1, Equation 2 and Equation 3.

$$\frac{di_a}{dt} = \frac{V_a - R_a.i_a - E_a}{L_a} \tag{1}$$

$$\frac{di_f}{dt} = \frac{V_f - R_f.i_f}{L_f} \tag{2}$$

$$\frac{dw}{dt} = \frac{k.i_f.i_a - T_L - B.w}{j} \tag{3}$$

The DC shunt motor model used in this study is simulated using MATLAB Simulink software. The 3 differential equations given above are used to obtain this Simulink model. Simulink model of the DC shunt motor is given in Figure 2.

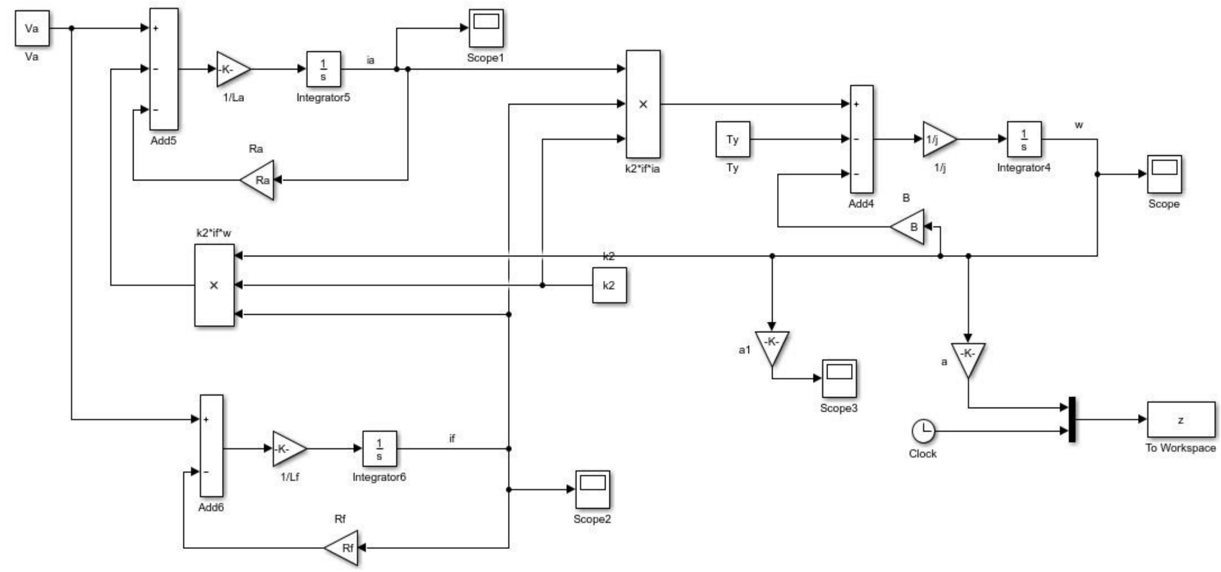

Figure 2: Simulink diagram of DC shunt motor model

The parameters of the modeled DC shunt motor are given in Table 1.

Table 1. DC shunt motor parameters

Initial	Symbol	Unit	Value
Rated Power	P_n	Hp	5
Rated Source Voltage	V_a	Volt	240
Armature Winding Resistance	R_a	Ohm (Ω)	0.6
Armature Winding Inductance	L_a	Henry	0.012
Moment of Inertia	J	kg.m^2	1
Coefficient of Friction	B	N.m.s	0
Excitation Winding Resistance	R_f	Ohm (Ω)	240
Excitation Winding Inductance	L_a	Henry	120

A dataset is created to train and test the ANN using the given Simulink model. V_a and τ_L values are changed to obtain the dataset. V_a, I_a, w and τ_L values are saved while creating the dataset.

In total, 60 input and output samples were generated from the Simulink model. For every operating point, the steady state values of the armature voltage (V_a), the armature current (I_a) and the angular velocity (ω) were taken as the three ANN inputs, while the corresponding load torque (τ_L) was taken as the single target output. Identical training and testing subsets were used for every scaling case, so that any difference in performance can be attributed solely to the normalization method and not to a change in the data.

2.2. Normalization Methods

Nowadays, as the popularity of artificial intelligence and artificial learning has increased, interest in data science has also increased and the operations to be made on raw data have gained great importance. In this direction, before the training of artificial neural networks (ANN), pre-treatment steps such as smoothing, generalization of the data, attribute construction, and normalization are performed [20]. The normalization process is applied to raw data to prepare the appropriate data set in ANN training. Scaling of ANN inputs and outputs significantly affects network performance, otherwise, network training may be inefficient [21, 22].

Normalization makes the distribution in the data set regular. Excessively large or small values can be seen instantaneously in the data set. Such data, and especially irregular data, can mislead the network. In this respect, avoiding redundancy and inconsistency in the data will increase accuracy and speed up the process. A different scaling method can be used for each problem.

There are many different normalization methods in the literature [23, 24]. The normalization methods commonly used in the literature and used in the study are reviewed below. The effects of the methods on the artificial neural network training and test performance were observed by applying the methods to the shunt motor data set.

2.2.1. Z-Score Normalization

This method is used in fixed situations where the values of a data set do not change. It is applied to eliminate the distances between the data and reduce the endpoints in cases where some values are too small than 0 and some are much larger in the data set.

The distribution is usually obtained between -1.5 and +1.5 with this normalization method. The mean value (μ_i) and standard deviation (σ_i) of the data set are calculated and the normalization process is performed using Equation 4. In the equation, x_i refers to the input data value and x^' refers to the normalized data set.

$$x' = \frac{x_i - \mu_i}{\sigma_i} \tag{4}$$

2.2.2. Linear Normalization

Another normalization method is the linear normalization. This approach is achieved by finding the maximum value of the data set (x_max) and dividing the values in the data set by this maximum value. The formula of the method is given in Equation 5.

$$x' = \frac{x_i}{x_{max}} \tag{5}$$

2.2.3. Min-Max Normalization

Another common normalization method is the min-max normalization. This method finds the minimum (x_{min}) and maximum (x_{max}) values of the raw data set and normalizes each input value linearly between 0 and 1 with the formula given in Equation (6). The difference between any two values or the proportionality of the magnitudes in the values does not change with this method.

$$x' = \frac{x_i - x_{min}}{x_{max} - x_{min}} \tag{6}$$

2.2.4. Median Normalization

In this normalization method, the median value of the raw data ($Median_x$) is found and the values are normalized by dividing the values in the data set by this number.

This situation is given in Equation (7). The most advantageous aspect of the median normalization method over other normalization methods is that it is not significantly affected by excessive deviations.

$$x' = \frac{x_i}{Median_x} \tag{7}$$

2.2.5. D-Min-Max Normalization

The last normalization method is also called the adjustable min-max normalization method. The method is applied by Equation (8), which normalizes all data from 0.1 to 0.9. The data set takes a dimensionless form with this normalization.

$$x' = 0.8 \cdot \frac{x_i - x_{min}}{x_{max} - x_{min}} + 0.1 \tag{8}$$

2.3. Feed-Forward Neural Network

Feed-forward backpropagation artificial neural network is used to test the ANN training performance of our data set normalized with different methods. The artificial neural network consists of 1 input layer with 3 linear neurons, 2 hidden layers with 6 and 4 non-linear neurons, and 1 output layer with 1 non-linear neuron. The structure of the artificial neural network is shown in detail in Figure 3. V_a is the armature voltage, I_a is the armature current and w is the angular velocity value of the shunt motor, which are the input values of the artificial neural network. T_y, the output of the artificial neural network, is the load torque.

The Levenberg-Marquardt (trainlm) algorithm is used which is particularly fast in artificial neural network training. Also, the activation function of nonlinear neurons in the hidden layer is logsig, and the activation function of the neuron in the output layer is purelin. These functions are given in Equation 9 and 10, in the order they are mentioned.

$$f(n) = \frac{2}{1 + e^{-2n}} - 1 \tag{9}$$

$$g(n) = n \tag{10}$$

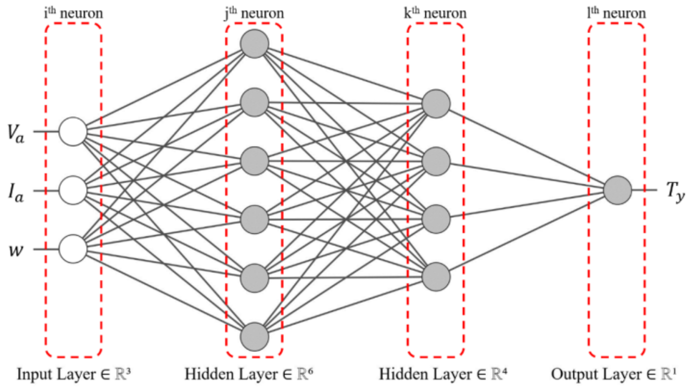

Figure 3: Schematic representation of the feed-forward neural network

The output signal of the first hidden layer, the second hidden layer, and the output layer of the proposed neural network can be described in the Equation 11, 12 and 13, respectively [25].

$$m_j = f\left(\sum_{j=1}^{6}\sum_{i=1}^{3} w_{ij} \cdot x_i\right) \tag{11}$$

$$n_k = f\left(\sum_{k=1}^{4}\sum_{j=1}^{6} w_{jk} \cdot m_j\right) \tag{12}$$

$$y_l = g\left(\sum_{l=1}^{1}\sum_{k=1}^{4} w_{kl} \cdot n_k\right) \tag{13}$$

In Equation 11 m_j is the output of the j^{th} neuron in the first hidden layer, w_{ij} is the weight value of the connection between the input layer and the first hidden layer, and x_i is the input values of the ANN. In Equation 12, n_k is the output of the kth neuron of the second hidden layer and w^{jk} is the weight value of the connection between the first hidden layer and the second hidden layer. Likewise, in Equation 13, y_l is the output of the l^{th} neuron in the output layer. Since our artificial neural network has an output, our output expression is only y_1. Finally, w_{kl} is the weight value of the connection between the second hidden layer and the output layer.

3. Simulation Results

The designed ANN model has 3 inputs and 1 output. This model consists of input layer, 2 hidden layers and output layer. There are 6 neurons in the first hidden layer and 4 neurons in the second hidden layer. Log-sigmoid activation function (logsig) is used in the first and second hidden layers. Purelin activation function, which is a linear activation function, is used in the output layer.

The ANN model structure is given in Table 2. 60 data samples are used for the training and testing process. Levenberg-Marquardt backpropagation algorithm (trainlm) is preferred for the training process.

Table 2. ANN model structure

Layer	Number of Neurons	Activation Function
Input	3	-
1. Hidden	6	Logsig
2. Hidden	4	Logsig
Output	1	Purelin

The parameters used in the training processes of the ANN are given in Table 3. The performance analysis is repeated with different epoch values as 25, 100, 1000, and 10000 for the training process performed with the raw data set and normalized data sets, and the results are obtained as Mean Squared Error (MSE) and R^2. Flow diagram of DC shunt motor data set generation and ANN training process are given schematically in Figure 4. Also, ANN training and testing procedures in the study are carried out in the MATLAB program.

Table 3. ANN training parameters

Performance Function	MSE
Maximum Number of Iterations (Epochs)	25/100/1000/10000
Target Error Value	1×10^{-10}
Learning Rate (η)	0.01
Training Algorithm	Trainlm

The data obtained from the Simulink model of the DC shunt motor model are normalized according to Z score normalization, min-max normalization, median normalization, D-min-max normalization and linear normalization methods. ANN is trained with raw data and normalized data. ANN training MSE values and R^2 for 25, 100, 1000 and 10000 epochs are given in Table 4.

MSE of 25 epoch of ANN's are given for Raw Data, Z-Score Normalization, Min-Max Normalization, Median Normalization, D-Min-Max Normalization, Linear Normalization respectively in Figure 5, Figure 6, Figure 7, Figure 8, Figure 9, and Figure 10.

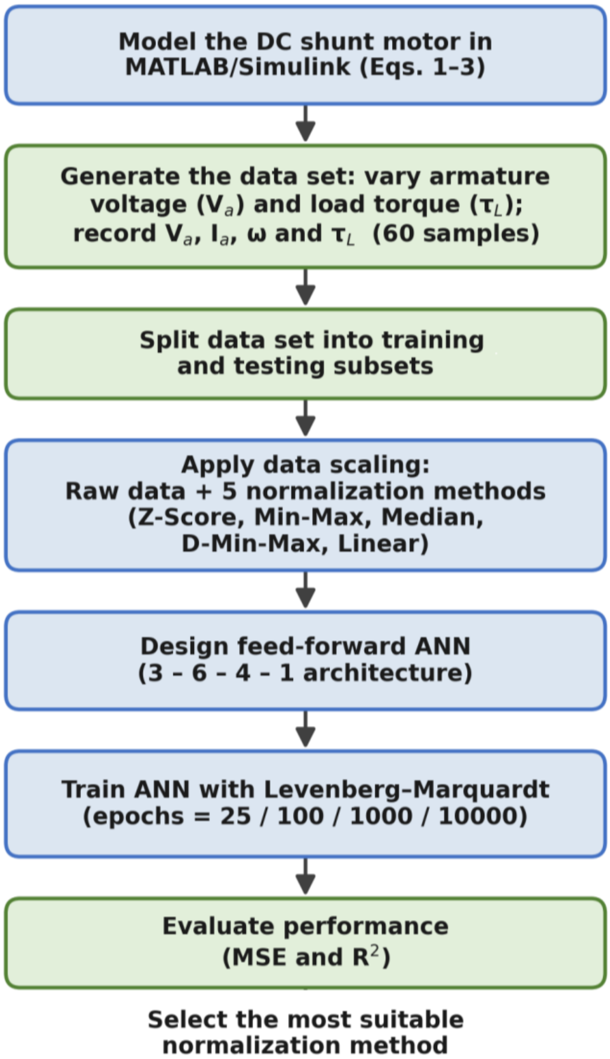

Figure 4: Flow diagram of DC shunt motor data set generation and ANN training process

Table 4. MSE and R^2 of ANN for load torque

Normalization Methods	25 Epoch		100 Epoch		1000 Epoch		10000 Epoch	
	MSE	R^2	MSE	R^2	MSE	R^2	MSE	R^2
Raw Data	0.016031	0.99983	0.0020457	0.99998	0.00012834	1	6.3205e-5	1
Z-Score Normalization	1.15e-6	1	9.8175e-7	1	2.1795e-7	1	3.5487e-8	1
Min-Max Normalization	1.1075e-6	0.99999	2.6122e-7	1	6.0327e-9	1	3.5849e-9	1
Median Normalization	6.9212e-5	0.99962	1.843e-5	0.9999	5.5545e-8	1	6.5015e-9	1
D-Min-Max Normalization	1.0662e-7	1	5.9328e-8	1	1.5082e-8	1	5.7046e-9	1
Linear Normalization	3.0053e-5	0.99962	1.9786e-6	1	6.605e-8	1	3.8347e-9	1

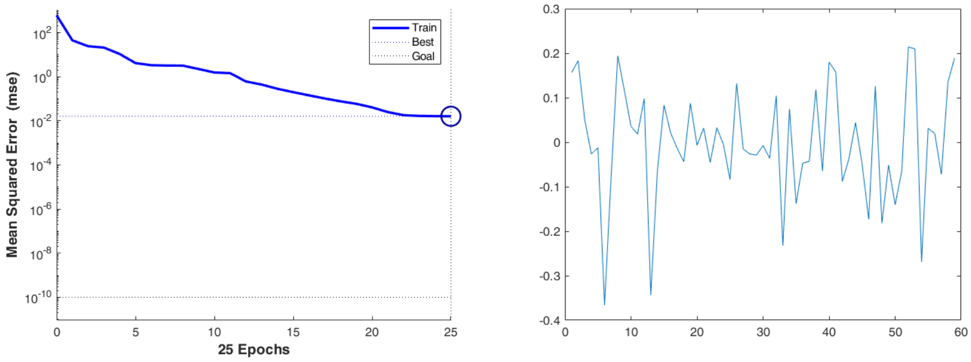

Figure 5: For raw data a) MSE - epoch b) amount of error – data

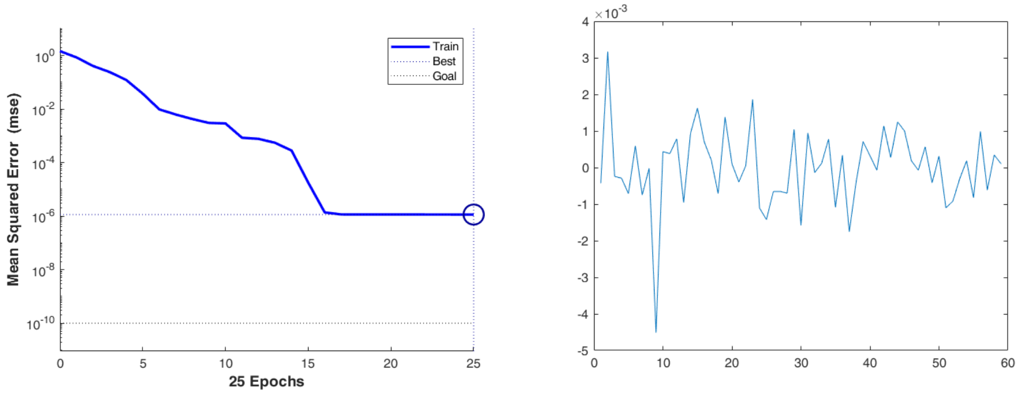

Figure 6: For z-score normalization a) MSE - epoch b) amount of error – data

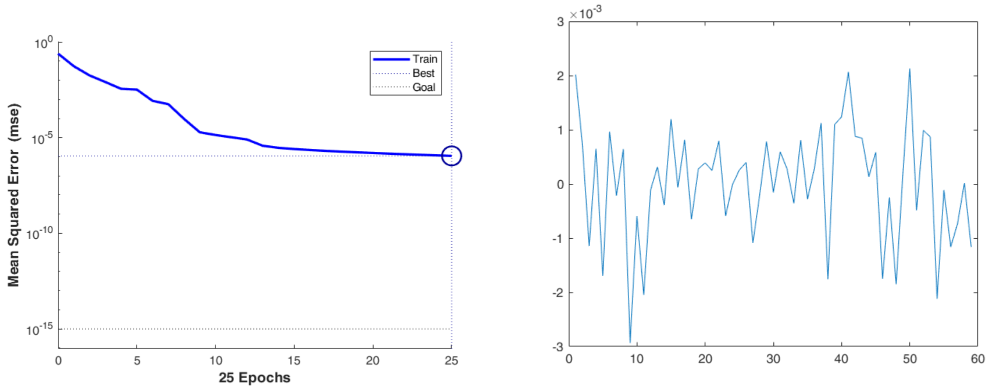

Figure 7: For min-max normalization a) MSE - epoch b) amount of error – data

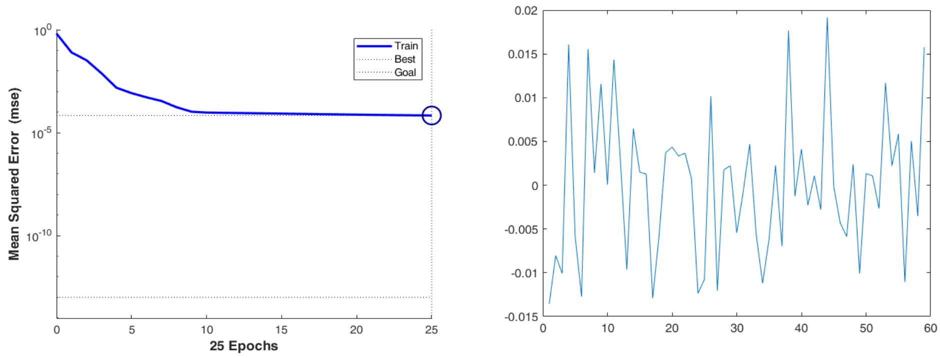

Figure 8: For median normalization a) MSE - epoch b) amount of error – data

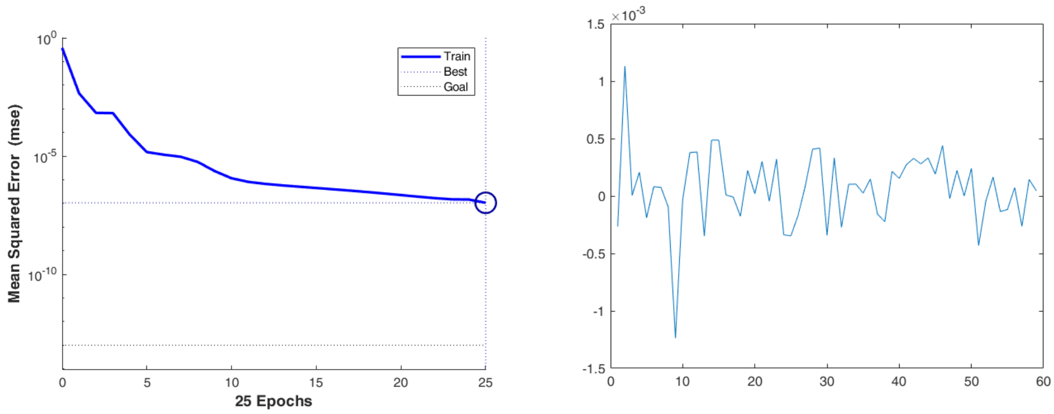

Figure 9: For D-min-max normalization a) MSE - epoch b) amount of error – data

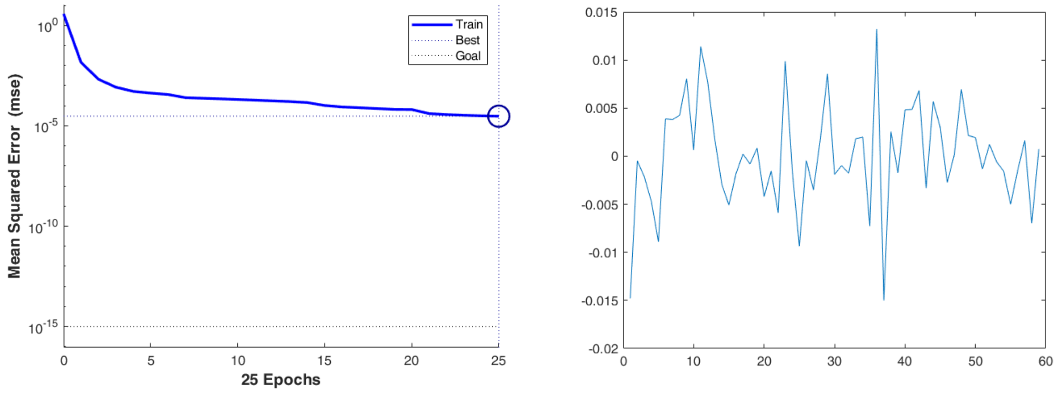

Figure 10: For linear normalization a) MSE - epoch b) amount of error – data

In model training with raw data, the MSE value has a relatively high value of 0.016031 for 25 epochs. However, as the number of epochs increased by 1000 and 10000, the MSE value decreased by 1.2834e-4 and 6.3205e-5 respectively. The R^2 value was 0.99983 for 25 epochs and 1 for 10000 epochs. In the training with raw data, it was observed that the error value of the model decreased to very low levels with a high number of epochs. In D-Min-Max normalization, the MSE values were considerably low in all epochs. In D-Min-Max normalization, the R^2 value remains constant at 1 in all epochs. This shows that D-Min-Max normalization optimizes the data for the model in a very effective way.

A closer inspection of Figures 5–10 reveals clear differences in both the convergence behaviour and the final accuracy of the six cases. In each figure, subplot (a) shows the evolution of the MSE with the training epoch, while subplot (b) shows the distribution of the residual error over the individual data samples. Two features are used to compare the methods: the epoch at which the MSE curve flattens (convergence speed) and the error level at which it settles (steady state accuracy).

For the raw data (Figure 5), the MSE curve starts at a comparatively high level (1.6×10^{-2}) and decreases only gradually, requiring several thousand epochs before it approaches the 10^{-5} range, and its residual error plot is the widest of the six cases. This slow and comparatively unstable descent is a direct consequence of the disparate physical scales of the inputs: the armature voltage (of order 10^2), the armature current (of order 10^0–10^1) and the angular velocity (of order 10^2) span several orders of magnitude, which distorts the error surface and forces the Levenberg–Marquardt algorithm to take very small effective steps along the poorly scaled directions.

When the inputs are normalized, the three variables are mapped onto a common bounded interval, the conditioning of the problem improves, and the optimizer converges far more rapidly. The Min-Max and D-Min-Max methods (Figures 7 and 9) show the fastest and smoothest convergence: their MSE curves fall below 10^{-6} within the first tens of epochs and their residual error plots are the narrowest and most symmetric. Because both methods map the data linearly onto a fixed range ([0, 1] for Min-Max and [0.1, 0.9] for D-Min-Max), they preserve the relative spacing of the samples while removing scale differences; the slightly compressed range of D-Min-Max additionally keeps all targets away from the saturated tails of the logsig activation functions, which explains its consistently low error and its R^2 of 1 at every epoch.

Z-Score normalization (Figure 6) also converges quickly and reaches a very low error, but its curve is marginally less smooth than that of the two linear methods. This is expected, because Z-Score centres the data to zero mean and unit variance rather than to a fixed interval, so a small number of samples can still fall outside the [−1.5, 1.5] band and briefly drive the activation functions towards saturation before the network adapts.

The Median and Linear methods (Figures 8 and 10) exhibit intermediate behaviour: both converge substantially faster than the raw data but settle at a slightly higher error and a marginally lower R^2 than the Min-Max family. Median normalization divides the data by the median value and is highly robust to outliers, but it does not bound the data to a fixed interval, so the effective input range and hence the conditioning of the problem depends on the particular distribution of the samples. Linear normalization divides only by the maximum value and therefore leaves the lower part of the range uncompressed, which limits the improvement in conditioning compared with the twopoint scaling of the Min-Max methods.

Across all methods the MSE decreases monotonically as the number of epochs is increased from 25 to 10000, confirming that longer training continues to refine the mapping. The key observation, however, is that the normalized cases reach at 25 epochs an accuracy that the raw data attains only after 10000 epochs; that is, appropriate scaling reduces the required training effort by roughly two to three orders of magnitude. The practical implication is that the benefit of normalization is greatest when the training time or computational budget is limited, which is precisely the situation encountered in real time motor control applications.

To determine whether the performance differences observed in Table 4 are statistically significant rather than incidental, a nonparametric Friedman test was applied to the MSE values. Each of the four epoch settings (25, 100, 1000 and 10000) was treated as a block, and the six data scaling cases were ranked within every block (rank 1 = lowest MSE). The test gives $\chi^2(5) = 16.14$ with $p = 0.006$, which is well below the 0.05 significance level; the null hypothesis that all scaling cases perform equally is therefore rejected, confirming that the choice of normalization method has a statistically significant effect on the prediction error.

Table 5. Mean ranks of the data scaling cases from the Friedman test (rank 1 = best)

Data-scaling method	Mean rank (MSE)	Overall order
Min-Max Normalization	1.50	1
D-Min-Max Normalization	1.75	2
Linear Normalization	3.50	3
Z-Score Normalization	4.00	4
Median Normalization	4.25	5
Raw Data	6.00	6

Table 5 lists the resulting mean ranks. The two Min-Max-based methods obtain the best (lowest) mean ranks, followed by the Linear, Z-Score and Median methods, while the raw data is ranked last in every block. A Nemenyi post-hoc test (critical difference CD = 3.77 for six methods and four blocks at $\alpha = 0.05$) indicates that the raw data differs significantly from the best performing methods, whereas the five normalization methods are statistically comparable to one another. In other words, the decisive and statistically significant improvement comes from applying a normalization method at all; among the normalization methods, Min-Max and D-Min-Max are consistently the strongest, which is in agreement with the MSE and R^2 values reported in Table 4.

4. Discussion and Conclusions

In this study, the impact of different normalization methods on ANN performance for predicting the load torque of a DC shunt motor is systematically investigated. The results show that data normalization plays an important role in reducing the error rates of the ANN model, especially in terms of MSE and R^2 error metrics. Although training the ANN model with raw data showed an acceptable performance, it is observed that the performance of the model improved slowly with higher MSE values in the beginning. However, as the number of epochs increased, the MSE values decreased and the R^2 value reached 1 at the end of 10000 epochs. This shows that there is a strong relationship between the predicted and actual load torque. However, raw data is not always the optimal approach and the results revealed that appropriate normalization techniques significantly improve the performance.

Among the different normalization methods, the D-Min-Max normalization gave the most promising results and consistently low MSE values are obtained in all epochs. This method maintained an R^2 value of 1 throughout the training process, indicating that D-Min-Max normalization optimizes the data for ANN training by effectively scaling and regularizing it. This suggests that D-Min-Max normalization is particularly suitable for situations where the input data has non-uniform ranges and needs to be adjusted to a narrow scale. On the other hand, the median and linear normalization methods are slightly less effective than the Z-Score and Min-Max normalizations. However, the results obtained with these normalization methods also produced estimates with acceptable error values. The increase in the number of epochs led to a decrease in the MSE value for these two normalization methods. However, the R^2 values of these methods are slightly lower compared to the other normalization methods. This study shows that normalization methods are effective in improving MSE and R2 values and increasing the convergence speed in ANN based predictions.

Future work will extend this study in several directions. First, the approach will be validated on experimental data measured from a physical DC shunt motor test bench, rather than on simulation data alone, in order to confirm its practical applicability. Second, the feed forward network will be compared with deep learning models such as long short-term memory (LSTM) and convolutional neural networks (CNN) that may better capture the dynamic behaviour of the machine. Finally, the comparison will be broadened to additional motor types and operating conditions so that more general guidelines for data preprocessing in motor control applications can be established.

References

[1] Beltran-Carbajal, F.; Tapia-Olvera, R.; Lopez-Garcia, I.; Guillen, D.; Adaptive dynamical tracking control under uncertainty of shunt DC motors. Electric Power Systems Research. 164, 70-78 (2018). https://doi.org/10.1016/j.epsr.2018.07.033

[2] Tapia-Olvera, R.; Beltran-Carbajal, F.; Aguilar-Mejia, O.; Valderrabano-Gonzalez, A.; An Adaptive Speed Control Approach for DC Shunt Motors. Energies. 9, 11, 961 (2016). https://doi.org/10.3390/en9110961

[3] Akshata, R.; Netravati, H.; Shweta, P.; Veena, N.; SR, K.; Development of Artificial Neural Network based Speed and Torque Prediction Models of a DC Shunt Motor. International Journal of Technology and Science. 3, 2, 22-26

[4] Beltran-Carbajal, F.; Tapia-Olvera, R.; Aguilar-Mejia, O.; Favela-Contreras, A.; Lopez-Garcia, I.; An online algebraic estimation approach of parameters and variable mechanical torque in shunt DC motors. International Transactions on Electrical Energy Systems. 28, 1, e2474 (2018). https://doi.org/10.1002/etep.2474

[5] Choudhary, P.; Bepari, B.; Kumar, A.; Ghosh, A.; Speed Prediction of DC Shunt Motor by using Artificial Neural Network. International Journal of Emerging Technology and Advanced Engineering. 4, 5, 214-218 (2014).

[6] Valluru, S. K.; Singh, M. Trajectory control of DC shunt motor by NARMA Level-2 neuro controller, In Proceedings of 2016 IEEE 1st International Conference on Power Electronics, Intelligent Control and Energy Systems (ICPEICES), 4-6 July 2016, (2016). 10.1109/ICPEICES.2016.7853600

[7] Antar, R. K.; Allu, A. A.; Sensorless speed/torque control of DC machine using artificial neural network technique. Tikrit Journal of Engineering Sciences. 23, 3, 55-62 (2016).

[8] Aksu, G.; Güzeller, C. O.; Eser, M. T.; The Effect of the Normalization Method Used in Different Sample Sizes on the Success of Artificial Neural Network Model. International Journal of Assessment Tools in Education. 6, 2, 170-192 (2019). 10.21449/ijate.479404

[9] Bhanja, S.; Das, A.; Impact of data normalization on deep neural network for time series forecasting. arXiv preprint arXiv:1812.05519. (2018).

[10] Sola, J.; Sevilla, J.; Importance of input data normalization for the application of neural networks to complex industrial problems. IEEE Transactions on Nuclear Science. 44, 3, 1464-1468 (1997). 10.1109/23.589532

[11] Simtharakao, S.; Sutivong, D. Exploring Normalization Techniques in Neural Networks for Bitcoin Candlestick Price Prediction, In Proceedings of 2023 International Conference on Artificial Intelligence in Information and Communication (ICAIIC), 20-23 Feb. 2023, (2023). 10.1109/ICAIIC57133.2023.10067086

[12] Kim, D.; Normalization methods for input and output vectors in backpropagation neural networks. International Journal of Computer Mathematics. 71, 2, 161-171 (1999). 10.1080/00207169908804800

[13] Djordjević, K. L.; Jordović-Pavlović, M. I.; Ćojbašić, Ž. M.; Galović, S. P.; Popović, M. N.; Nešić, M. V.; Markushev, D. D.; Influence of data scaling and normalization on overall neural network performances in photoacoustics. Optical and Quantum Electronics. 54, 8, 501 (2022). 10.1007/s11082-022-03799-1

[14] Sedghisigarchi, K.; Hasanovic, A.; Feliachi, A.; Davari, A. Evaluation of three algorithms for nonlinear control of a DC shunt motor, In Proceedings of Proceedings of the 33rd Southeastern Symposium on System Theory (Cat. No.01EX460), 20-20 March 2001, (2001). 10.1109/SSST.2001.918555

[15] Alawan, M. A.; Al-Furaiji, O. J. M.; Numerous speeds loads controller for DC shunt motor based on PID controller with online parameters tuning supported by genetic algorithm. Indonesian Jour-

nal of Electrical Engineering and Computer Science. 21, 1, 64-73 (2021).
[16] Chiley, V.; Sharapov, I.; Kosson, A.; Koster, U.; Reece, R.; Samaniego de la Fuente, S.; Subbiah, V.; James, M.; Online normalization for training neural networks. Advances in Neural Information Processing Systems. 32, (2019).
[17] Puheim, M.; Madarász, L. Normalization of inputs and outputs of neural network based robotic arm controller in role of inverse kinematic model, In Proceedings of 2014 IEEE 12th International Symposium on Applied Machine Intelligence and Informatics (SAMI), 23-25 Jan. 2014, (2014). 10.1109/SAMI.2014.6822439
[18] Umans, S.; Fitzgerald, A.; Kingsley, C., Electric Machinery: Seventh Edition. McGraw-Hill: 2013.
[19] Verma, P.; Gupta, P.; Kumar, B. Graphical user interface (GUI) to study DC motor dynamic characteristics, In Proceedings of 2017 International conference of Electronics, Communication and Aerospace Technology (ICECA), 20-22 April 2017, (2017). https://doi.org/10.1109/ICECA.2017.8212746
[20] Shalabi, L. A.; Shaaban, Z.; Kasasbeh, B.; Data Mining: A Preprocessing Engine. Journal of Computer Science. 2, 9, (2006). https://doi.org/10.3844/jcssp.2006.735.739
[21] Suarez-Alvarez, M. M.; Pham, D.-T.; Prostov, M. Y.; Prostov, Y. I.; Statistical approach to normalization of feature vectors and clustering of mixed datasets. Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences. 468, 2145, 2630-2651 (2012). https://doi.org/10.1098/rspa.2011.0704
[22] Mustaffa, Z.; Yusof, Y. A comparison of normalization techniques in predicting dengue outbreak, In Proceedings of International Conference on Business and Economics Research, Kuala Lumpur, Malaysia, 2011, (2010).
[23] Patro, S.; Normalization: A preprocessing stage. arXiv preprint arXiv:1503.06462. (2015).
[24] Saranya, C.; Manikandan, G.; A study on normalization techniques for privacy preserving data mining. International Journal of Engineering and Technology (IJET). 5, 3, 2701-2704 (2013).
[25] Eski, I.; Erkaya, S.; Savas, S.; Yildirim, S.; Fault detection on robot manipulators using artificial neural networks. Robotics and Computer-Integrated Manufacturing. 27, 1, 115-123 (2011). https://doi.org/10.1016/j.rcim.2010.06.017
[26] LeCun, Y.; Bottou, L.; Orr, G. B.; Müller, K.-R.; Efficient BackProp. In Neural Networks: Tricks of the Trade; Springer: Berlin, Heidelberg, 9-50 (1998). https://doi.org/10.1007/3-540-49430-8_2
[27] Jayalakshmi, T.; Santhakumaran, A.; Statistical normalization and back propagation for classification. International Journal of Computer Theory and Engineering. 3, 1, 89-93 (2011). https://doi.org/10.7763/IJCTE.2011.V3.288
[28] Ioffe, S.; Szegedy, C. Batch normalization: accelerating deep network training by reducing internal covariate shift, In Proceedings of the 32nd International Conference on Machine Learning (ICML), Lille, France, 448-456 (2015).
[29] Singh, D.; Singh, B.; Investigating the impact of data normalization on classification performance. Applied Soft Computing. 97, 105524 (2020). https://doi.org/10.1016/j.asoc.2019.105524