Science & Technology Development Journal – Engineering and Technology 2025, 28(4):3840-3848

Open Access Full Text Article

Research Article

Prediction of the wear rate of polyester composite materials using artificial neural networks and fuzzy inference systems

Luong Cuong Hoang[1], Duc Hao Nguyen[2,3], Duc Lan Vu[2,3], Hoa Son Vo[2,3], Van Hieu Le[2,3,*]

ABSTRACT

Glass fiber reinforced polyester (GFRP) composites are widely used in applications that require high durability and excellent corrosion resistance. In Vietnam, lightweight GFRP materials are primarily used in the production of boat hulls and bows, automobile and motorcycle chassis, as well as various industrial and household devices due to their high corrosion resistance. Accurately predicting the wear rate of GFRP composites is of significant practical importance as it provides important guidance for the experimental design and production of these materials. In this study, artificial intelligence (AI) techniques, specifically artificial neural networks (ANN) and fuzzy inference systems (FIS), were used to predict the wear rate of GFRP composites. Both methods are well-suited to modeling nonlinear systems. Experimental results from trained ANN and adaptive neuro-fuzzy inference system (ANFIS) models were used to determine the manufacturing parameters that yielded the lowest wear rate for GFRP composites. Specifically, a wear rate of 18.241 × 10^{-6} mm^3/Nm was obtained using a composition of 30 wt.% $CaCO_3$, 8 wt.% P, 11 wt.% GF, 5 wt.% GB, and 1 wt.% Al, with a compressive load of 10 N and a rotation speed of 200 rpm.

Key words: ANN network, ANFIS network, GFRP, material wear rate

[1]*Faculty of Physics – Engineering Physics, University of Science, Ho Chi Minh, Viet Nam*

[2]*Faculty of Materials Science and Technology, University of Science, Ho Chi Minh, Viet Nam*

[3]*Vietnam National University Ho Chi Minh City, Ho Chi Minh, Viet Nam*

Correspondence

Van Hieu Le, Faculty of Materials Science and Technology, University of Science, Ho Chi Minh, Viet Nam

Vietnam National University Ho Chi Minh City, Ho Chi Minh, Viet Nam

Email: lvhieu@hcmus.edu.vn

History

- Received: 21-06-2024
- Revised: 21-05-2025
- Accepted: 08-07-2025
- Published Online: 12-10-2025

DOI :

https://doi.org/10.32508/stdj.v28i4.4324

INTRODUCTION

Glass fiber reinforced polyester (GFRP) composites are hybrid materials composed of polyester polymers reinforced with glass fibers [1,2]. This combination produces a lightweight material with excellent mechanical properties. The demand for materials with improved wear resistance has grown rapidly due to robust industrialization, modernization, and international integration. Improving the durability and wear resistance of materials not only boosts their operational efficiency and reduces maintenance costs but also supports environmental protection and workplace safety [3]. In ad dition, materials with high wear resistance can help optimize production performance and lower operational costs; this is crucial for businesses competing in international markets, where product quality and efficiency are critical factors.

GFRP composites are highly valued for their wear resistance across several applications; however, the specific rate of wear often depends on several factors, including the type and proportion of glass fibers, the quality of the polyester matrix, and the ambient environmental and operating conditions [1]. Due to the wide range of input conditions that can affect the performance of the final product, optimizing these input conditions to achieve the lowest wear rate is a key focus of current research. This study proposes a machine learning approach involving artificial neural networks (ANNs) and adaptive neuro-fuzzy inference systems (ANFIS) to optimize input parameters for the experimental production of GFRP materials with reduced wear rate. The experimental data used to construct the model were obtained from previous work [4].

MATERIALS AND METHODS

Materials and Experimental Parameters

The components used to produce the GFRP composites investigated in this study are presented in Table 1 [4].

In this study, orthophthalic polyester resin was used as the matrix for the GFRP samples. Additives used in the GFRP composites include glass beads (GB), calcium carbonate ($CaCO_3$), and aluminum oxide, while glass fibers (GF) were used as the reinforcement material. Additional input conditions considered for wear rate testing included load (L) and rotational speed (S). The output variable of interest was the wear rate (WR) of the resultant material.

The initial experimental parameters for model construction were obtained from previous work [4] and are presented in Table 2. Seven input parameters—$CaCO_3$, P, GF, GB, Al, L, and S—were assigned the variables X1, X2, X3, X4, X5, X6, and X7, respectively, while the WR was used as the output parameter (Y).

Cite this article : Hoang L C, Nguyen D H, Vu D L, Vo H S, Le V H. **Prediction of the wear rate of polyester composite materials using artificial neural networks and fuzzy inference systems**. *Sci. Tech. Dev. J. – Engineering and Technology* 2025; 28(4):3840-3848.

Table 1: [4] **Material components of individual GFRP samples**

Sample	Matrix	Reinforcement	Filler 1	Filler 2	Filler 3
D1	20 wt.% orthophthalic polyester	40 wt.% GF	10 wt.% GB	30 wt.% $CaCO_3$	None
T1	20 wt.% orthophthalic polyester	11 wt.% GF	1 wt.% Alumina	68 wt.% $CaCO_3$	None
T2	20 wt.% orthophthalic polyester, 8 wt.% polystyrene tensile additive	11 wt.% GF	1 wt.% Alumina	60 wt.% $CaCO_3$	None
T3	20 wt.% orthophthalic polyester, 8 wt.% polystyrene tensile additive	11 wt.% GF	5 wt.% GB	56 wt.% $CaCO_3$	None
T4	20 wt.% orthophthalic polyester	11 wt.% GF	1 wt.% Alumina	5 wt.% GB	63 wt.% $CaCO_3$
T5	20 wt.% orthophthalic polyester, 8 wt.% polystyrene tensile additive	11 wt.% GF	1 wt.% Alumina	5 wt.% GB	55 wt.% $CaCO_3$

A total of 24 experimental data points were used, of which 75% were used for training and 25% were used for validation.

Artificial Neural Network (ANN) Model

ANNs are computational models inspired by the structure and function of biological brains[5–8]. The structure of the ANN used in this study consists of the following layers[9]:

Input Layer: Each neuron in the input layer receives an input value X_i that is passed to the neurons in the next layer without modification. This input is typically represented as a vector $X = [x_1, x_2, ..., x_n]$, where n is the number of features.

Hidden Layers: In the hidden layers, each neuron j computes a weighted sum of the inputs ($\boldsymbol{z}_j$) from all neurons i in the previous layer, in addition to a bias term ($\boldsymbol{b}_j$). Subsequently, an activation function ϕ is applied to produce the output $\boldsymbol{h}_j$.

The formula for the weighted sum of inputs and the bias for a neuron j in the hidden layer is as follows[5,9,10]:

$$z_j = \sum_i (w_{ij}, x_i) + b_j$$

where w_{ij} is the weight connecting neuron i in the previous layer with neuron j in the current layer, x_i is the output from neuron i in the previous layer (or the network input if i is in the input layer), and b_j is the bias for neuron j.

The output h_j of neuron j after applying the activation function ϕ is:

$$h_j = \phi(z_j)$$

The activation function ϕ can be selected from one of several types of functions, including the sigmoid, Tanh, and ReLU functions.

Output Layer: Like the hidden layers, each neuron in the output layer computes a weighted sum of the inputs from all neurons in the previous layer and applies an activation function (which may differ from the function used in the hidden layer) to produce the final output of the network. Typically, a linear activation function (e.g., purelin) is used for the output layer.

Training Process: The ANN is trained by using a backpropagation algorithm and an optimization algorithm such as gradient descent[5,7,11] to iteratively adjust the weights (w_{ij}) and biases (b_j) in the layers to minimize the error between the predicted and actual values.

Learning Rate: The learning rate is a crucial hyperparameter in ANN training that controls the step size during weight updates, influences the convergence, and helps minimize the loss function. This parameter is defined by the equation[2,4]:

$$w_{t+1} = w_t - \eta \nabla L(w_t)$$

where w_t and w_{t+1} are the weight values at time t and t +1, η is the learning rate, and $\nabla L(\nabla L(w_t))$ is the gradient of the loss function with respect to the weights w_t.

The ANFIS Model

ANFIS networks are hybrid models that combine neural networks and fuzzy inference systems, designed to simulate Takagi-Sugeno fuzzy inference systems[12,13]. These models leverage the advantages of both methods to determine the optimal parameters of the fuzzy

Table 2: Experimental parameters used in the model [4]

Dataset		Input data							Output data
Sample Name	Sample Number	$CaCO_3$ (wt%)	(wt.%)	GF (wt.%)	GB (wt.%)	Al (wt.%)	L (N)	S (rpm)	WR (mm^3/Nm) $*10^{-6}$
DI	1	30	0	40	10	0	10	100	700
DI	2	30	0	40	10	0	10	200	583
DI	3	30	0	40	10	0	20	100	929
T1	5	68	0	11	0	1	10	100	235
T1	6	68	0	11	0	1	10	200	237
T1	7	68	0	11	0	1	20	100	605
T2	9	60	8	11	0	1	10	100	34
T2	10	60	8	11	0	1	10	200	42
T2	11	60	8	11	0	1	20	100	684
T3	13	56	8	11	5	0	10	100	52
T3	14	56	8	11	5	0	10	200	35
T3	15	56	8	11	5	0	20	100	115
T4	17	63	0	11	5	1	10	100	139
T4	18	63	0	11	5	1	10	200	163
T4	19	63	0	11	5	1	20	100	154
T5	21	55	8	11	5	1	10	100	25
T5	22	55	8	11	5	1	10	200	29
T5	23	55	8	11	5	1	20	100	554
M	4	30	0	40	10	0	20	200	777
DI	8	68	0	11	0	1	20	200	446
T1	12	60	8	11	0	1	20	200	83
T2	16	56	8	11	5	0	20	200	41
T3	20	63	0	11	5	1	20	200	261
T4	24	55	8	11	5	1	20	200	490

system, thereby improving its inference and prediction capabilities. A standard ANFIS system consists of five layers as follows [13,14]:

Layer 1: Fuzzification Layer

The nodes in this layer transform the input values into fuzzy values using membership functions, such as Gaussian, bell-shaped, and triangular functions. Each node corresponds to a fuzzy rule and produces outputs defined as follows [13,15]:

$$O_{1,i} = \mu_{A_i}(x) \; for \; input \; x$$
$$O_{1,j} = \mu_{B_j}(y) \; for \; input \; y$$

where $O_{1,i}$ and $O_{1,j}$ are the outputs of the i and j nodes in Layer 1, and $\mu_{A_i}(x)$ and $\mu_{A_j}(y)$ are the corresponding fuzzy functions.

Layer 2: Evaluation of Rules Layer

Each node in this layer represents a fuzzy rule. The output of each node is the product of the attribute values from the previous layer, which define the weight of the rule. The outputs of this layer are normalized by dividing each output by the total weight of all rules. The formula for calculating the weight of each rule is as follows:

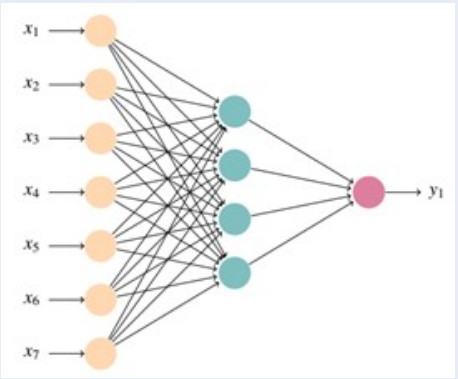

Figure 1: An ANN architecture with seven inputs, five hidden neurons, and one output

$O_{2,k} = w_k = \mu_{A_k}(x) \times \mu_{B_k}(y)$

where w_k is the weight of rule k.

Layer 3: Normalization Layer

The nodes in this layer normalize the rule weights obtained from the previous layer to ensure that the total weight of all rules is equal to 1. The output of each node is thus the normalized weight of a specific rule. The normalization formula is defined as follows:

$Q_{3,k} = \overline{w_k} = \frac{w_k}{\sum_k w_k}$

where $\overline{W_k}$ is the normalized weight of rule k.

Layer 4: Defuzzification Layer

Each node in this layer generates a fuzzy output based on the normalized weights from Layer 3. These outputs are typically expressed as linear or polynomial functions of the inputs. The formula determining the output function of each rule is as follows:

$O_{4,k} = \overline{W_k} \times f_k = \overline{W_k}(p_k x + q_k y + r_k)$

where f_k is the output function of rule k, typically represented as a linear equation of the inputs with coefficients p_k, q_k, and r_k.

Layer 5: Composite Output Layer

The overall output of the ANFIS network is generated by the final layer, which aggregates all outputs from Layer 4. This output is obtained by multiplying the sum of the outputs by their normalized weights and is defined by the following formula:

$O_5 = \sum_k \overline{w_k} f_w$

In ANFIS networks, training can be conducted exclusively using backpropagation algorithms or in combination with optimization algorithms, such as the least squares estimator (LSE). These algorithms are used to update the parameters of the membership functions and the fuzzy inference rules. Through this process, the ANFIS network optimizes the fuzzy inference system to approximate complex functions, thus allowing the network to model intricate relationships within the data, thereby enhancing its predictive accuracy and efficiency.

The sub-clustering method was used to identify fuzzy rules from input data through Matlab software [16,17]. The ANFIS sub-clustering process consists of three main stages:

• Creation of sub-clusters: This was achieved by employing clustering techniques such as C-Means clustering or fuzzy C-Means clustering to partition the input space into clusters that correspond to a fuzzy rule that is typically characterized by Gaussian, bell-shaped, or sigmoidal functions.

• Model training: The ANFIS model is trained based on the fuzzy rules established in the first stage. This process involves the fine-tuning of parameters, attribute functions, and weights associated with each fuzzy rule to improve predictive capabilities and optimize model performance.

• Model evaluation: The trained model is validated using a validation dataset that was set aside prior to the training process. Evaluation criteria include accuracy, coverage, and other indices such as the coefficient of determination (R^2), mean squared error (MSE), and root mean squared error (RMSE).

This structured approach allows the ANFIS network to effectively learn and model complex relationships within the data, thus providing a robust framework for applications that require nuanced data interpretation and prediction.

Evaluation Methods: Several metrics were used to evaluate the performance of the ANN and the ANFIS networks. Several commonly used evaluation metrics are presented below [18–20]:

The MSE [21] measures the average squared difference between actual and predicted values and is defined as:

$MSE = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \widehat{Y}_i)^2$

where n is the number of data samples used in the evaluation set, Y_i is the actual value of sample i, and $\widehat{Y}_i$ is value predicted by the model for sample i.

Similarly, the RMSE [20–22] measures the deviation between the predicted values and the actual values. A smaller RMSE is indicative of a more reliable model. It is defined as follows:

$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(Y_i - \widehat{Y}_i)^2}$

The R^2 [20,23] value is used to evaluate how well a regression model fits the actual data, with values ranging from 0 to 1. A higher R^2 score indicates a better fit. It is defined as:

$R^2 = 1 - \frac{\sum_{i=1}^{n}(Y_i - \widehat{Y}_i)^2}{\sum_{i=1}^{n}(Y_i - \overline{Y})^2}$

RESULTS AND DISCUSSION

ANN Model

The number of neurons in the hidden layer of an ANN is a critical parameter that influences the structure and performance of the model. Each hidden neuron receives inputs from the previous layer, applies an activation function, and transmits its output to the next layer.

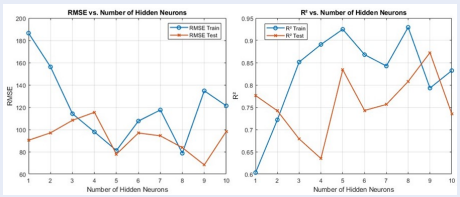

Figure 2: Identification of the optimal number of hidden neurons.

A hidden layer with five neurons was selected as it yielded the lowest RMSE on the test sets (~80), indicating high prediction accuracy (Figure 2). The corresponding R^2 values were both high and well-balanced (~0.92 and ~0.83 for the training and test set, respectively), suggesting that the model generalizes well without evidence of overfitting.

Of the activation functions evaluated in this investigation, *tansig* achieved the best performance on the test set; specifically, it had the lowest RMSE and the highest R^2 value, indicating that it was more generalizable compared to the *logsig*, *radbas,* and *purelin functions*. Indeed, the test RMSE of *tansig* was approximately 20–40 units lower than that of the other functions, while its R^2 exceeded 0.85. More detailed results are presented in Figure 3. Based on *this analysis, tansig was selected* as the optimal activation function for the proposed model.

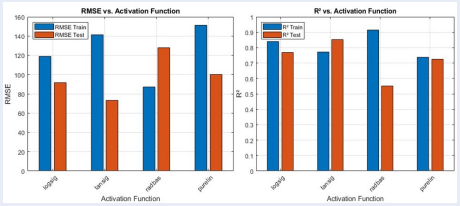

Figure 3: Identification of the optimal activation function

Across the range of evaluated learning rates, a learning rate of 0.3 yielded the best performance on the test set. Specifically, it exhibited the lowest RMSE (~65) and the highest R^2 (~0.88), indicative of efficient learning and strong generalizability. Other learning rates ranging from 0.01 to 0.5 resulted in less stable or less accurate results. The detailed results of this evaluation are illustrated in Figure 4. Consequently, a learning rate of 0.3 was selected for the proposed model.

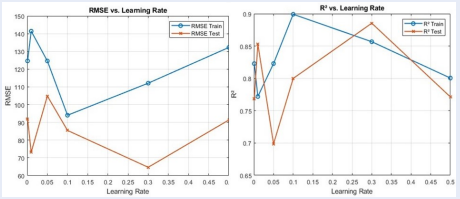

Figure 4: Identification of the optimal learning rate.

Of the variety of optimization algorithms tested in this study, trainscg was found to achieve a favorable balance between accuracy and generalizability. Figure 5 shows that this algorithm achieved low RMSE values on both the training and test sets, indicative of efficient convergence and consistent performance across both datasets. In addition, the corresponding R^2 values were both high and well-balanced (~0.87 and ~0.86 for the training and test sets, respectively), suggesting that the model effectively captures the underlying data patterns without overfitting. Compared to other algorithms such as traingd—which exhibited significantly poorer RMSE and R^2 scores—and trainbr, the trainscg algorithm offered superior stability and predictive power, making it the most suitable optimization algorithm for this regression task.

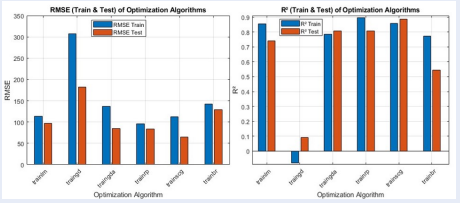

Figure 5: Identification of the ideal optimization algorithm.

The evaluation of different epoch settings shows that the model performed best at 250 epochs. Under these conditions, the test RMSE reached its lowest value (~65), while the R^2 approached 0.89, reflecting high prediction accuracy and strong generalizability (Figure 6). Consequently, a total of 250 epochs was selected as the optimal training setting for this task.

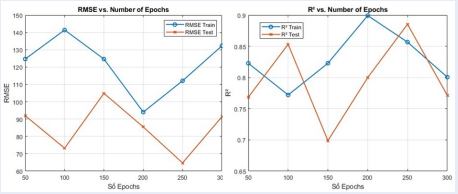

Figure 6: Identification of the optimal number of epochs.

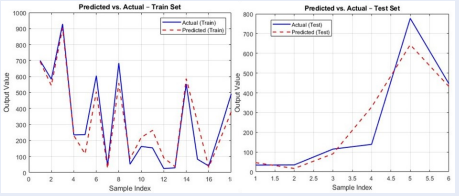

Figure 8: Predicted vs. actual values on the training and testing sets.

Based on systematic hyperparameter tuning, the neural network was configured with five hidden neurons, a tansig activation function, a learning rate of 0.3, the trainscg training algorithm, and 250 training epochs. This configuration achieved the best overall performance, with low RMSE and consistently high, well-balanced R^2 values on both training and testing datasets, indicative of strong predictive accuracy and strong generalizability without any evidence of overfitting.

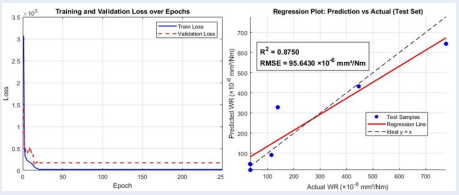

Figure 7: a) Training loss (blue line) and validation loss (red dashed line) curves; b) Regression plot highlighting the differences between the actual and predicted WR values.

Figure 7a shows the training process over 250 epochs. Both the training loss and validation loss decrease rapidly and then stabilize, indicating that the model learns effectively, converges quickly, and does not suffer from overfitting. Figure 7b shows that the regression line closely follows the ideal y = x line, indicating that the predicted WR values are consistent with the actual values. The final model achieved a high R^2 value of 0.875 and a low RMSE of approximately 95.64 $\times 10^{-6}$ mm^3/Nm on the test set, demonstrating accurate predictions and strong generalization capability.

Figure 8 illustrates the relationship between the predicted and actual values for both the training and test sets. In the training set, the predicted curve closely matches the actual data, suggesting that the model has effectively captured the underlying patterns in the data. Despite some slight deviations at points with higher variability, the predictions remain stable and consistent with actual trends in the test set, suggesting that the model generalizes well to unseen data without overfitting.

ANFIS Model

The ANFIS network was optimized by focusing on key parameters such as the number of sub-clusters, fuzzy membership functions, and fuzzy rules. Figure 9 presents the overall structure of the ANFIS model used in this study. The network consists of seven input variables: $CaCO_3$, P, GF, GB, Al, L, and S, all of which influence the WR.

Each input is connected to a set of fuzzy membership functions (inputmf), which translate the input values into linguistic levels (e.g., low, medium, and high). These membership functions feed into the rule layer, where fuzzy "if–then" rules are formed by combining input conditions. The blue nodes represent logical "AND" operations used to determine rule activation. The rule outputs are then passed to the output membership functions (outputmf), which are mapped to fuzzy output values.

Finally, these outputs are aggregated and defuzzified to produce a single output value: the predicted WR. The ANFIS network used in this study consists of 18 sub-clusters, 18 fuzzy rules, and 18 membership functions for each variable. The "gaussmf" function was chosen as the membership function for the input layer, while the "linear" function was applied to the output layer. Detailed results of this investigation are presented in Table 3.

Figure 10 shows the fuzzy rule base that was automatically generated through the sub-clustering process. A total of 18 rules were created, with each rule corresponding to a data cluster. Each rule integrates all seven input variables (in1 to in7), which are mapped to their respective membership functions and used to infer a specific output. This well-defined, comprehensive ruleset demonstrates the model's ability to learn complex nonlinear relationships between the inputs and the output, while also ensuring complete coverage of the input space.

Table 3: Results generated from the ANFIS model.

	Variable Name	Membership Function	Number of Sub-Clusters	Sub-Cluster Classification
Input	$CaCO_3$	gaussmf	6	8, 13, 14, 16, 17, 18
		gaussmf	2	17, 18
	GF	gaussmf	2	8, 18
	GB	gaussmf	3	8, 17, 18
	Al	gaussmf	2	13, 18
	L	gaussmf	2	16, 18
	S	gaussmf	2	16, 18
Output	WR	linear	18	1, 2, …17, 18

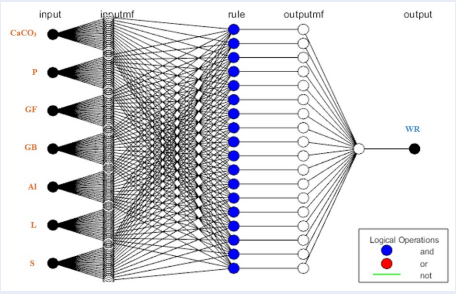

Figure 9: Structure of the ANFIS Network used to predict WR.

Figure 10: The fuzzy rule base of the ANFIS model generated using sub-clustering.

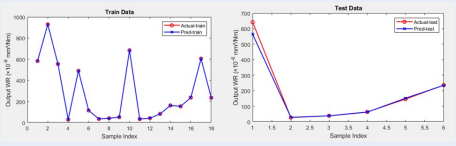

Figure 11: Comparison between the actual values and the values predicted by the ANFIS model.

Figure 11 shows that the predictions generated by the ANFIS model applied to the training samples were consistent with the actual values. In addition, the ANFIS model yielded more accurate WR predictions compared to the ANN model when applied to the test set. Specifically, the R^2 value of the ANFIS model was 0.9778, which is higher than the R^2 value of 0.875 obtained by the ANN model.

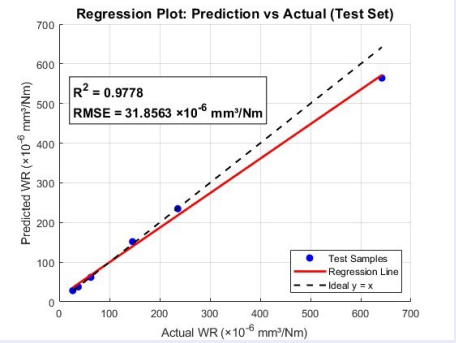

Figure 12: Regression plot of predicted vs. actual WR with ANFIS model applied to the test set.

Table 4 shows that the ANFIS model outperformed the ANN model in both accuracy and generalizability. Specifically, the ANFIS model achieved a higher R^2 value of 0.9778 (compared to 0.875 for the ANN) and a significantly lower RMSE of 31.8563 (compared to 95.643 for the ANN). These results indicate that the ANFIS model provides a more suitable and effective approach for predicting WR.

Prediction of WR using the ANFIS model

The superior ANFIS model was used to optimize the input parameters to obtain the lowest possible WR.

Table 4: Comparison of the predictive performance of the ANN and ANFIS models.

Model	R^2	RMSE ($x10^{-6}mm^3/Nm$)
ANN	0.875	95.643
ANFIS	0.9778	31.8563

Table 5: Predictions of the ANFIS model.

	Input_pred							Output_pred
	$CaCO_3$		GF	GB	Al	L	S	WR
	(wt%)	(wt.%)	(wt.%)	(wt.%)	(wt.%)	(N)	(rpm)	$(mm^3/Nm)*10^{-6}$
Random input samples	30	5	20	3	1	10	150	18.138
GA	30	8	11	5	1	10	200	18.241

Two approaches were employed: 1) generating random input samples and evaluating their corresponding outputs using the optimized ANFIS model, and 2) applying a genetic algorithm (GA) to search for the optimal input conditions that minimize the WR. Detailed results of both approaches are presented in Table 5.

Table 5 presents the predicted minimum WR obtained using the random input sampling and GA-based optimization approaches. The trained ANFIS model was used to determine the predicted WR corresponding to each input configuration. The predicted WR values from both approaches were very close, with a difference of only ~1.0×10^{-7} mm^3/Nm, highlighting the stability and reliability of the ANFIS model for predicting wear behavior. Although random sampling yielded a slightly lower WR (18.138 vs. 18.241), the GA approach offers a more systematic and targeted optimization process, making it particularly valuable in practical applications.

CONCLUSIONS

This study investigated the WR of GFRP materials using the ANFIS model. An evaluation of the ANFIS model demonstrates that it can be used to optimize input parameters even with a limited dataset. The results indicated that the lowest WR (18.241×10^{-6} mm^3/Nm) obtained corresponds to the following input conditions: 30% $CaCO_3$, 8% P, 11% GF, 5% GB, 1% Al, a compressive load of 10 N, and a rotational speed of 200 rpm.

This study also conducted a comparative analysis of the ANN and ANFIS models, revealing that the ANFIS model is a promising tool for optimizing production processes and material synthesis compared to the ANN model due to its higher R^2 value of 0.9778, compared to the R^2 value of 0.875 obtained by the ANN model.

ABBREVIATIONS

Glass Fiber Reinforced Polyester (GFRP)
Artificial Intelligence (AI)
Artificial Neural Network (ANN)
Fuzzy Inference System (FIS)
Adaptive Neuro-Fuzzy Inference System (ANFIS)
Glass beads (GB), calcium carbonate ($CaCO_3$), Glass fibers (GF)
Load (L), rotational speed (S), wear rate (WR)
Coefficient of determination (R^2), mean squared error (MSE), and root mean squared error (RMSE)
Genetic Algorithm (GA)

COMPETING INTERESTS

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

AUTHORS' CONTRIBUTIONS

Cuong Luong Hoang: Data curation; Formal analysis; Investigation; Methodology; Software; Writing - original draft; review & editing. Hao Duc Nguyen: Data curation; Funding acquisition; Investigation; Writing - original draft. Lan Duc Vu: Data curation; Formal analysis; Investigation; Resources; Software; Writing - original draft. Hoa Son Vo: Investigation. Van Hieu Le: Conceptualization; Investigation; Methodology; Project administration; Supervision; Validation; Writing - review & editing.

ACKNOWLEDGEMENTS

The authors would like to acknowledge the research collaborations of Multifunctional Materials Laboratory (Faculty of Materials Science and Technology), the Faculty of Physics and Engineering Physics, University of Science, Vietnam National University Ho Chi Minh City for providing instrumental support and facilitating experimental research this project.

REFERENCES

1. A K, T Ö, A U. Comparative Study of the Mechanical Properties of Fiber-Reinforced Denture Base Resin. J Appl Polym Sci. 2009;113:716–720.
2. S Y, M C, F C, M F, Y L, et al. Effect of Different Environmental Conditions on Durabilities of Polyester- and Vinylester-Based Glass-Fiber-Reinforced Polymer Pultruded Profiles. Front Mater. 2022;9.
3. and Arslan F YI. Effects of fiber volume fraction and fiber orientation on the tribological behaviour of unidirectionally oriented glass fiber-polyester composites. Turk J Eng Environ Sci. 2000;24:181–191.
4. S Y, R I, E F. Estimation of adhesive wear behavior of the glass fiber reinforced polyester composite materials using ANFIS model. J Elastomers Plast. 2022;54(1):86–110.
5. Understanding Neural Networks: A Comprehensive Guide; 2023. Available from: https://dev.to/mariazayed/understanding-neural-networks-a-comprehensive-guide-28cg.
6. G S. Analytics Vidhya.; 2021. Available from: https://www.analyticsvidhya.com/blog/2021/09/introduction-to-artificial-neural-networks/.
7. How Do Neural Networks Work? Your 2024 Guide; 2024. Available from: https://www.coursera.org/articles/how-do-neural-networks-work.
8. and Montesinos A MLO, J C. Fundamentals of Artificial Neural Networks and Deep Learning. 2022;p. 379–452.
9. The Basic Structure of Neural Network; 2024. Available from: https://neuralnetwork101.com/structure.
10. ; 2024. Available from: https://deeplizard.com/lesson/dlb3izrlda.
11. ; 2024. Available from: https://brilliant.org/wiki/artificial-neural-network/.
12. S R, NK V. Developing deep fuzzy network with Takagi Sugeno fuzzy inference system. In: 2017 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE); 2024. p. 1–6. Available from: https://ieeexplore.ieee.org/document/8015718.
13. N W, H S, A S. ANFIS: Adaptive Neuro-Fuzzy Inference System-A Survey. Int J Comput Appl. 2015;123(13):32–38.
14. MNM S, N T, K H. Adaptive Neuro-Fuzzy Inference System: Overview, Strengths, Limitations, and Solutions. Springer International Publishing. 2017;p. 527–535.
15. CU Y, KC K. Adaptive Neuro-Fuzzy Inference System Predictor with an Incremental Tree Structure Based on a Context-Based Fuzzy Clustering Approach. Appl Sci. 2020;10(23):8495.
16. J G, J S, OS O, DS A, OE A, MO N, et al. Adaptive neuro-fuzzy inference system (ANFIS) approach for the irreversibility analysis of a domestic refrigerator system using LPG/TiO2 nanolubricant. Energy Rep. 2020;6:1405–1417.
17. Estimation of Mineral Resources with Machine Learning Techniques. In: Materials Proceedings; 2024. Available from: https://www.mdpi.com/2673-4605/5/1/122.
18. S A. MAE, MSE and RMSE as performance measures for regression models; 2023. Available from: https://www.makebitbyte.com/blog/rmse-mae-mse.
19. A K. MSE vs RMSE vs MAE vs MAPE vs R-Squared: When to Use? [Internet]. Analytics Yogi; 2024. Available from: ttps://vitalflux.com/mse-vs-rmse-vs-mae-vs-mape-vs-r-squared-when-to-use/.
20. Z B. RMSE vs. R-Squared: Which Metric Should You Use?; 2021. Available from: https://www.statology.org/rmse-vs-r-squared/.
21. Narayana KL, Yogesh, Kowshik P. Medical Insurance Premium Prediction Using Regression Models. International Journal for Research Trends and Innovation. 2023;8(4):1512–1517.
22. (PDF) Root-mean-square error (RMSE) or mean absolute error (MAE): when to use them or not; 2024. Available from: https://www.researchgate.net/publication/362110985_Root-mean-square_error_RMSE_or_mean_absolute_error_MAE_when_to_use_them_or_not.
23. K W, S C. An iterative approach to minimize the mean squared error in ridge regression. Comput Stat. 2015;30.