Smart Temperature Control Using Neuro-Fuzzy Model

Kelvin N. Nnamani, Ken Aghaegbunam Akpado, Augustine C.O. Azubogu

Abstract: ***The temperature control model employs a neuro-fuzzy approach with a defined universe of discourse encompassing temperature (20°C-50°C), humidity (30%-90%), and fan speed (20%-70%). Membership functions were established, utilizing generalized bell functions for temperature and humidity, along with trapezoidal functions for fan speed. A rule base comprising nine rules was developed, incorporating temperature and humidity as linguistic input variables and fan speed as the linguistic output variable. In the data preprocessing phase using Python, 60% of the dataset was designated for training, while 40% was set aside for testing with the scikit-learn model. A convolutional neural network (CNN) was created using TensorFlow's Keras API, featuring 64 neurons, ReLU activation, and two input shape features. The model underwent training for 100 epochs with the Adam optimizer and a batch size of 16, achieving a training loss of 0.9951 and a test loss of 1.0239. The closely matched and relatively low values of both training and test loss indicate that the model is not overfitting and has successfully captured the underlying patterns. For instance, when the current temperature and humidity were set to 35°C and 65%, the recommended fan speed was 48%. Moreover, predicted fan speeds were 20.14%, 35.21%, and 43.64% for temperature and humidity settings of (35°C, 45%), (45°C, 75%), and (55°C, 85%), respectively.***

Keywords: Temperature, Humidity, Fan Speed, Gbell Membership Function, Trapezoidal Membership Function, TensorFlow's Keras API, Scikit-learn, Convolutional Neural Network (CNN).

Nomenclature:
ANNs: Artificial Neural Networks
MATLAB: Matrix Laboratory Software
ANFIS: Artificial Neuro-Fuzzy Inference System
HVAC: Heating, Ventilation, and Air Conditioning Systems
SST: Sea Surface Temperature
Gbell: Generalized Bell Membership Function
ReLU: Rectified Linear Unit; a Non-Linear Activation Function for Deep Neural Networks.
CNN: Convolutional Neural Network
L_{train}: Training Loss
L_{test}: Testing (Validation) Loss
$L_{MSE_{train}}$: Mean Squared Error for Training Loss
$L_{MSE_{test}}$: Mean Squared Error for Test Loss
$L_{MAE_{train}}$: Mean Absolute Error for Training Loss
$L_{MAE_{test}}$: Mean Absolute Error for Test Loss
RMSE: Root Mean Square Error
N_{train} and N_{train}: Number of Training and Testing Samples
API: Application Programming Interface
Adam: Adaptive Moment Estimation for Neural Network Optimization
IoT: Internet of Things
MAE: Mean Absolute Error
MSE: Mean Squared Error

Manuscript received on 30 September 2025 | Revised Manuscript received on 11 October 2025 | Manuscript Accepted on 15 October 2025 | Manuscript published on 30 October 2025.
*Correspondence Author(s)

Kelvin N. Nnamani*, Department of Electronic and Computer Engineering, Nnamdi Azikiwe University, Awka (Anambra), Nigeria. Email ID: kn.nnamani@pg.unizik.edu.ng, ORCID ID: 0000-0002-5484-7237

Prof. Ken Aghaegbunam Akpado, Department of Electronic and Computer Engineering, Nnamdi Azikiwe University, Awka (Anambra), Nigeria. Email ID: k.akpado@unizik.edu.ng

Prof. Augustine C.O. Azubogu, Department of Electronic and Computer Engineering, Nnamdi Azikiwe University, Awka (Anambra), Nigeria. Email ID: ac.azubogu@unizik.edu.ng

I. INTRODUCTION

In recent years, the demand for efficient temperature control systems has grown significantly due to the increasing need for energy conservation and enhanced comfort in both residential and industrial environments. Traditional temperature control methods often rely on fixed algorithms that do not adapt to changing conditions, leading to inefficiencies and discomfort. To improve this, researchers are exploring neuro-fuzzy models, which blend ANNs with fuzzy logic systems [1]. These models highlight the adaptive learning capabilities of ANNs, enabling them to enhance performance over time and making them particularly effective for temperature control in fluctuating environments [2]. This can be applied to grain storage, where maintaining optimal conditions prevents spoilage. By utilizing fuzzy logic controllers with triangular membership functions and employing tools such as MATLAB and Arduino, we can significantly enhance the efficiency of temperature regulation. In addition, a robust fuzzy modelling approach utilizes 'if-then' rules to effectively manage temperature and humidity, employing the centroid method for improved accuracy and performance [3].

Integrating neuro-fuzzy models into temperature control systems can significantly enhance energy efficiency and comfort levels. These models reduce energy consumption [4] by dynamically adjusting heating and cooling based on real-time data. By using ANFIS methods in MATLAB to model the cutting temperature of steel, we can identify key processing parameters and improve machine efficiency [5]. This is vital for smart buildings and industries focused on minimizing energy costs.

In a greenhouse compartment, a geothermal system powered by fossil fuels generates electricity to regulate temperatures for optimal crop growth [6]. This approach reduces costs while enhancing yield. The system utilizes ANFIS combined with backpropagation and least squares algorithms, using if-then rules and membership functions to determine control parameters for improved efficiency.

II. LITERATURE REVIEW

A novel neuro-fuzzy algorithm, designed by [4], aims to enhance temperature control systems and meet the growing demand for adaptive intelligent control mechanisms, making it applicable in HVAC systems and smart home technologies. The authors effectively combine neural networks and fuzzy logic to model complex, nonlinear relationships in temperature control, utilizing well-defined membership functions for temperature and humidity along with a comprehensive rule base.

Also, the accurate prediction of SST using ANFIS was highlighted in [7] to model air temperature and evaporation data in Çanakkale. Implemented in MATLAB, the model employed Gaussian membership functions. With 75% of the data for training and 25% for testing, the predictions show improved regression and correlation coefficients close to unity, indicating strong performance and potential for further applications.

In the paper authored by [8], an adaptive neural-fuzzy controller was developed for temperature control in an egg hatchery, specifically maintaining a range of 35°C. This system was also implemented using MATLAB, considering inputs such as the number of eggs and the current temperature, to produce the desired output: temperature. ANFIS was used to train, test, and validate the neural network model. Their analysis revealed that the controller significantly increases the incubator's temperature to accommodate between 85 and 95 eggs effectively.

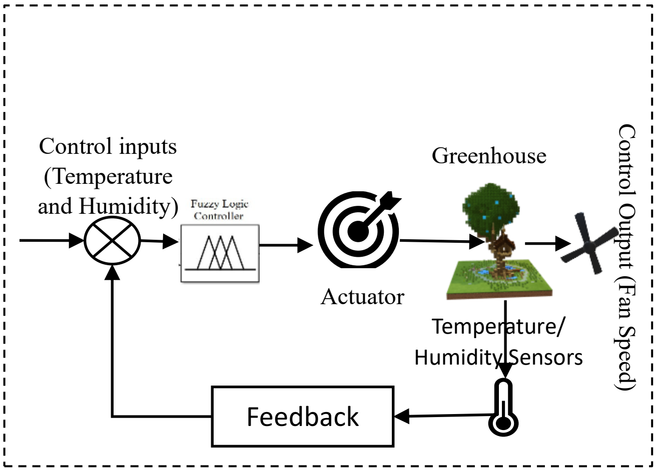
III. METHODOLGY

The Python libraries were first installed and imported.

A. Defining the Fuzzy Variables and the Universe of Discourse

The universe of discourse for three variables—temperature, humidity, and fan speed —is explored using NumPy, a popular library in Python for numerical computations. The Proposed Feedback Fuzzy controller of a typical Temperature controller is shown in Fig.1. This is done by setting the universe of discourse for this implementation:

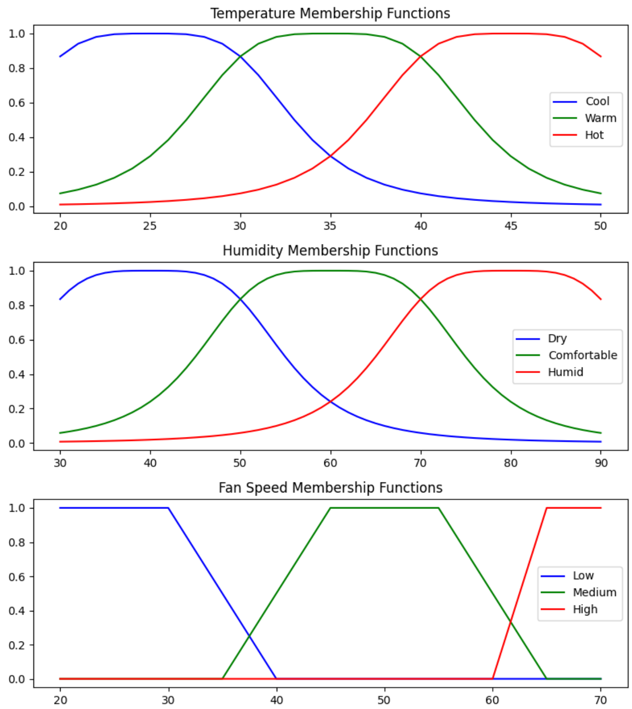
i. *Temperature:* 20°C to 50°C
ii. *Humidity:* 30% to 90%
iii. *Fan Speed:* 20% to 70%



[Fig.1: Block Diagram for a Typical Fuzzy Smart Temperature Controller]

B. Setting Up the Membership Functions

The Gbell membership functions were configured for the input variables (temperature and humidity) as shown in Fig.2, while trapezoidal membership functions were used for the output variable (fan speed). This setup was chosen to provide clearer operational ranges for effective control and decision-making in the system.



[Fig.2: Membership Functions for the Temperature, Humidity and Fan Speed]

C. Defining the Rule Base for the Fuzzy Variables

The rule base consists of nine intuitive rules that dictate the fan speed based on the combinations of temperature and humidity conditions. The rules are as follows:

i. If the temperature is cool and the humidity is dry, then the fan speed should be set to low.
ii. If the temperature is cool and the humidity is comfortable, then the fan speed should also be set to low.
iii. If the temperature is cool and the humidity is humid, the fan speed remains low.
iv. If the temperature is warm and the humidity is dry, the fan speed should be set to medium.
v. If the temperature is warm and the humidity is comfortable, the fan speed should be set to medium.
vi. If the temperature is warm and the humidity is humid, then the fan speed should be set to high.
vii. If the temperature is hot and the humidity is dry, the fan speed should be set to high.
viii. If the temperature is hot and the humidity is comfortable, the fan speed should also be set to high.
ix. If the temperature is hot and the humidity is humid, the fan speed remains high.

Retrieval Number: 100.1/ijainn.F110705061025
DOI: 10.54105/ijainn.F1107.05061025
Journal Website: www.ijainn.latticescipub.com

Published By:
Lattice Science Publication (LSP)

D. Preprocessing the Temperature, Humidity and Fan Speed Dataset for Training

The linguistic input variables (X) - temperature and humidity - and the linguistic output variable (y) - fan speed - were extracted from the dataset using the iloc method.

E. Splitting of the Dataset for Training and Testing

i. *Purpose:* Splits the dataset into initial training and testing sets

ii. *Parameters:*
- X: Features data (temperature and humidity)
- y: Target variable (fan speed)
- test_size=0.4: 40% of data allocated to testing, 60% to training
- random_state=0: Ensures reproducible splits across runs

iii. *Result:*
- X_train, y_train: 60% of data for training
- X_test, y_test: 40% of data for testing

F. Splitting of the Dataset for Validation

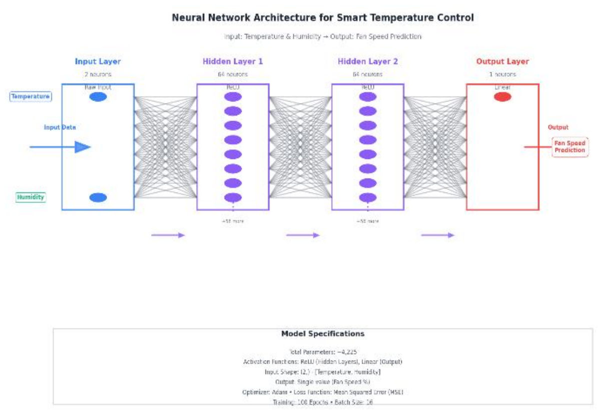
i. *Purpose*: Further splits the testing set into validation and final testing sets

ii. *Operation*: Takes the 40% test set and splits it again

iii. *Result*:
- **X_valid, y_valid:** Validation set (40% of 40% = 16% of original data)
- **X_test, y_test:** Final test set (60% of 40% = 24% of original data)

G. Creating a Convolutional Neural Network Model for Training

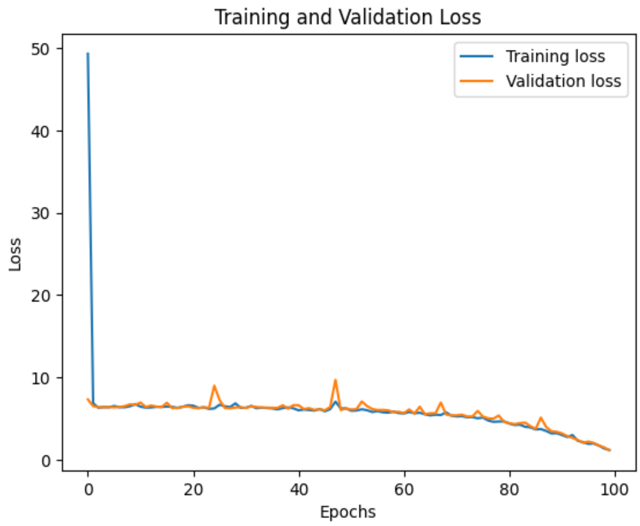
A neural network model is created using TensorFlow's Keras API. The model consists of three layers: an input layer with 64 neurons and ReLU activation, a hidden layer with 64 neurons and ReLU activation, and an output layer with a single neuron and linear activation. The input shape is defined as having two features. This can be demonstrated in the proposed Neural Network Architecture for Smart Temperature Control, as shown in Fig. 3.



[Fig.3: Proposed Neural Network Architecture for Smart Temperature Control]

H. Training of the Model using the fit Method

The model is trained using the **fit** method for 100 epochs with Adam optimizer and a batch size of 16, utilizing training data (X_train, y_train) and validation data (X_valid, y_valid). After training, the training and validation loss are plotted against the number of epochs, with appropriate labels and a legend, and then displayed. The data visualization of the training set based on the training loss of 1.1808 and validation loss of 1.1458 at 100 epochs is shown in Fig. 4.

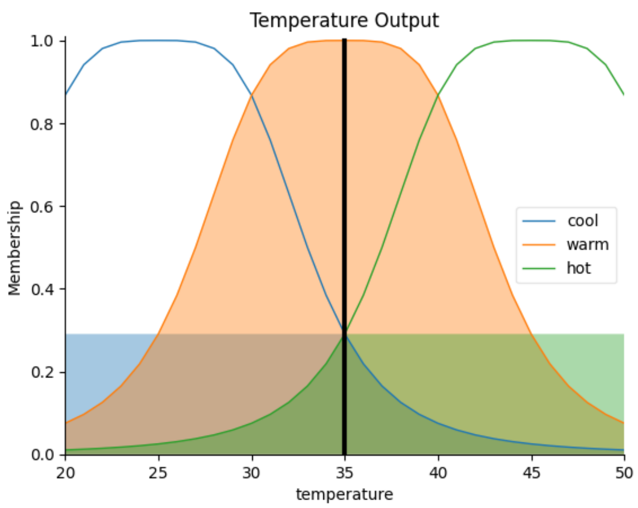


[Fig.4: Training and Validation Loss Versus 100 Epochs]

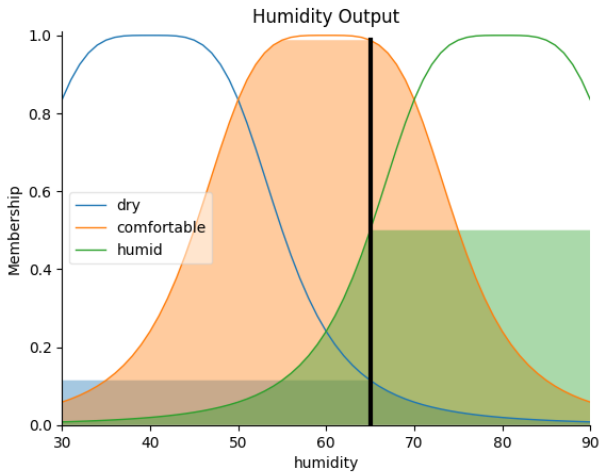
IV. ANALYSIS AND RESULTS

A. Fan Speed Control

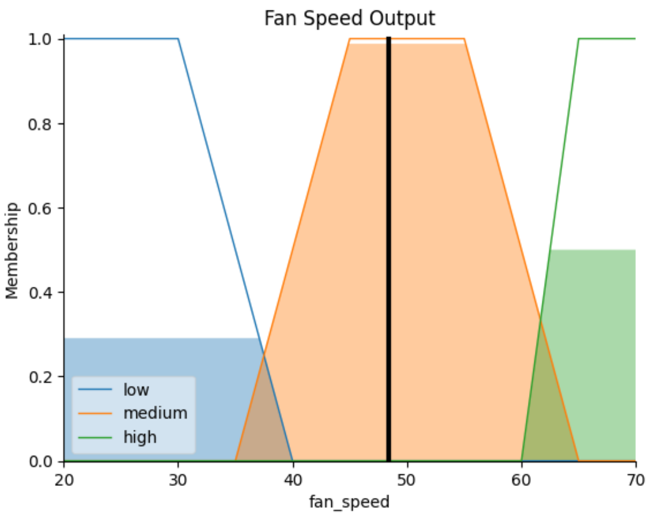
The Fan speed control based on Temperature and Humidity is shown in Fig.5 and Fig. 6, respectively. By setting the current Temperature at Comfortable and Humidity at Warm states as 35°C and 65% respectively, the recommended Fan Speed operating at medium state becomes 48.38462454923027 % with a degree of membership function close to unity as shown in Fig. 7. The surface 3D viewer is also shown in Fig. 8. Also, the Temperature, Humidity and Fan speed distribution is shown in Fig. 9. In contrast, the 3D plot is shown in Fig. 10.



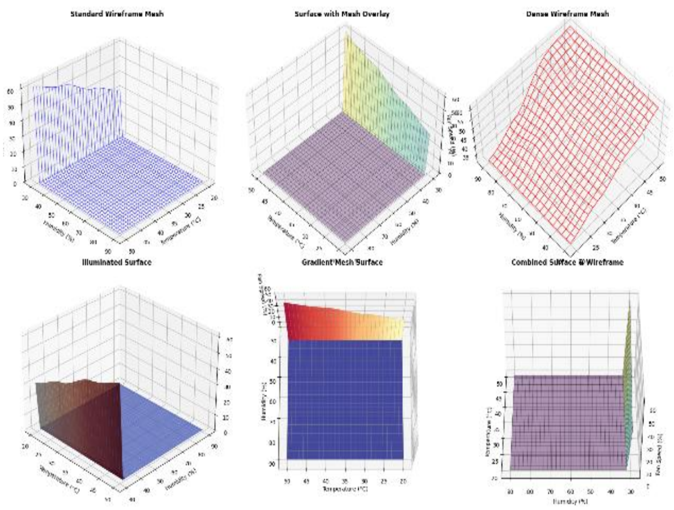
[Fig.5: Fan Speed Control Based on Temperature at 35°C]



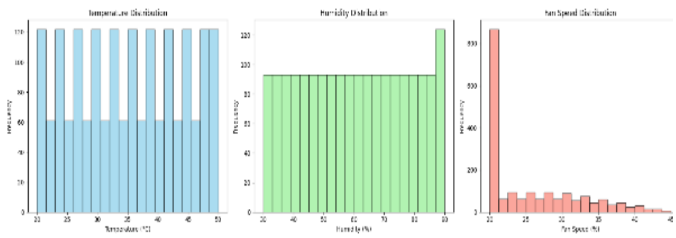
[Fig.6: Fan Speed Control Based on Humidity at 65%]



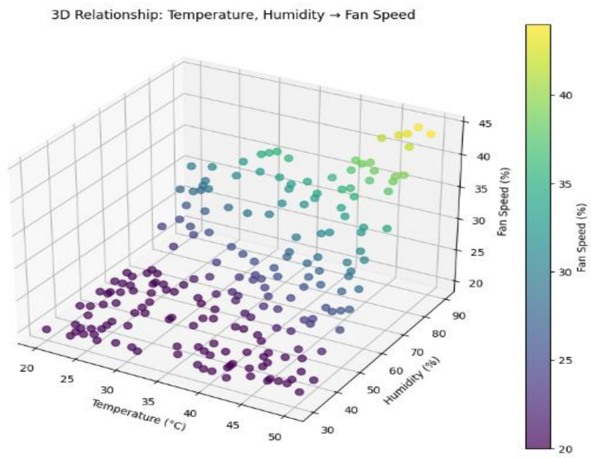
[Fig.7: Fan Speed with Recommended Operation at 48%]



[Fig.8: Surface 3D Viewer of Fan Speed Control based on Temperature and Humidity]



[Fig.9: Temperature, Humidity and Fan Speed Distribution]



[Fig.10: 3D Temperature, Humidity and Fan Speed Distribution]

B. Loss in the CNN Model

i. For a convolutional neural network (CNN) Model, the loss is the average error between the predicted output $\hat{y}$ and the time target y, across all samples. Loss serves as a valuable numerical indicator of how well a neural network model's predictions align with the actual target values. It guides the optimizer during training by highlighting the discrepancies between predicted values and exact outcomes. According to [9], two widely recognized metrics for evaluating model performance are the Mean Absolute Error (MAE) and the Root Mean Squared Error (RMSE). RMSE is defined as the square root of the Mean Squared Error (MSE), which provides a valuable measure for understanding prediction values.

Where $\hat{y}_i = test\_predictions$ and $y_i = y\_test$

- Training Loss:

$$L_{train} = \frac{1}{N_{train}} \sum_{i=1}^{N_{train}} \ell(y_i, \hat{y}_i) \quad \ldots \quad (1)$$

- Testing(validation)Loss:

$$L_{test} = \frac{1}{N_{test}} \sum_{i=1}^{N_{test}} \ell(y_i, \hat{y}_i) \quad \ldots \quad (2)$$

- N_{train} and $N_{test} = number\ of\ training\ and\ testing\ samples$
- $\ell(y_i, \hat{y}_i) = given\ error\ function\ (e.g\ MSE, MAE)$

ii. *Mean Squared Error (MSE):* MSE highlights the squared differences between predicted and actual values.

$$\ell_{MSE}(y_i, \hat{y}_i) = (y_i - \hat{y}_i)^2$$

- *MSE Training Loss*:

$$L_{MSE_{train}} = \frac{1}{N_{train}} \sum_{i=1}^{N_{train}} (y_i - \hat{y}_i)^2 \quad \ldots \quad (3)$$

- *MSE Testing Loss*:

Retrieval Number: 100.1/ijainn.F110705061025
DOI: 10.54105/ijainn.F1107.05061025
Journal Website: www.ijainn.latticescipub.com

Published By:
Lattice Science Publication (LSP)

$$L_{MSE_{test}} = \frac{1}{N_{test}} \sum_{i=1}^{N_{test}} (y_i - \hat{y}_i)^2 \quad \dots \qquad (4)$$

iii. *Mean Absolute Error (MAE):* MAE measures the average absolute difference between the predicted output and the time target.

$$\ell_{MAE}(y_i, \hat{y}_i) = |y_i - \hat{y}_i|$$

- *MAE Training Loss*:

$$L_{MAE_{train}} = \frac{1}{N_{train}} \sum_{i=1}^{N_{train}} |y_i - \hat{y}_i| \quad \dots \qquad (5)$$

- *MAE testing Loss*:

$$L_{MAE_{test}} = \frac{1}{N_{test}} \sum_{i=1}^{N_{test}} |y_i - \hat{y}_i| \quad \dots \qquad (6)$$

C. Evaluation of the Mean Absolute Error (MAE) and Mean Squared Error (MSE) of the Training Model

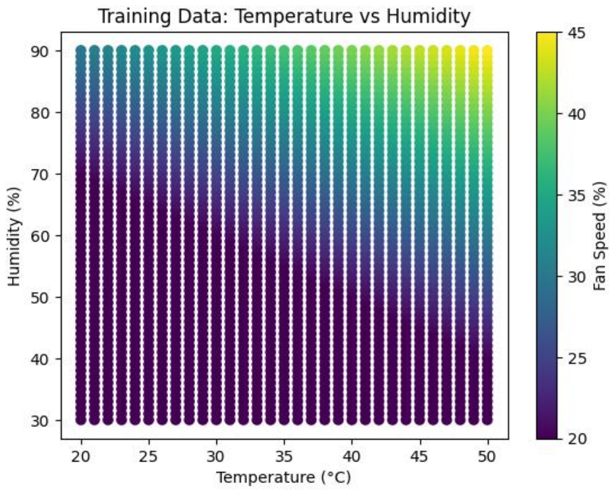
The training set shows a Mean Absolute Error (MAE) of 0.7589 and a Mean Squared Error (MSE) of 0.9951, resulting in a test loss of 1.0239. These low values suggest effective learning, and the comparable test error indicates that the model generalizes well to unseen data. This positive performance lays the groundwork for further improvement and more reliable predictions.

D. Evaluation of the Mean Absolute Error (MAE) and Mean Squared Error (MSE) of the Test Model

The test model displays a training set MAE of 0.7722 and an MSE of 1.0239, resulting in a training loss of 0.9951. The similarity between the training loss and test MSE shows good generalization without overfitting. In addition, the test MAE aligns with the training metrics, confirming prediction accuracy. The model's fast processing times highlight its strong performance and readiness for optimization.

E. Predicted Fan Speed for Temperature and Humidity Control

By adjusting the temperature and humidity settings to 35°C and 45%, 45°C and 75%, and 55°C As shown in Fig. 11, the predicted outputs for fan speeds are 20.14%, 35.21%, and 43.64%, respectively. This data highlights the relationship between target input and fan speed performance, thereby optimizing ventilation systems.



[Fig. 11: Predicted Fan Speed for Temperature and Humidity Control]

V. CONCLUSION AND RECOMMENDATIONS

A. Conclusion

The optimal control of temperature and humidity using neuro-fuzzy systems facilitates effective decision-making and the design of inference systems. The assigned membership function provides a model controller that manages precise control of fan speed as the output. In addition, the application of CNN enhances the adaptability and ventilation in a given environmental setup. Error evaluations highlight the controller's efficiency in predicting the output fan speed when the desired temperature and humidity values are set.

B. Recommendations

i. Future work should focus on refining the membership functions and rule base to improve accuracy and responsiveness through experimentation with different shapes and parameters.

ii. Utilizing a larger, more diverse dataset for training can enhance the model's generalization capabilities, enabling it to handle various environmental conditions better.

iii. Implementing the model in real-time will provide valuable insights into its performance, allowing for continuous monitoring and adjustments based on real-world data.

iv. Exploring integration with IoT technologies could improve functionality, enabling remote monitoring, control, and comprehensive data collection.

v. Creating a user-friendly interface will enhance interaction and control for end-users, making the model more accessible for diverse applications, including HVAC and smart homes.

DECLARATION STATEMENT

After aggregating input from all authors, I must verify the accuracy of the following information as the article's author.

- **Conflicts of Interest/ Competing Interests:** Based on my understanding, this article has no conflicts of interest.
- **Funding Support:** This article has not been funded by any organizations or agencies. This independence ensures that the research is conducted with objectivity and without any external influence.
- **Ethical Approval and Consent to Participate:** The content of this article does not necessitate ethical approval or consent to participate with supporting documentation.
- **Data Access Statement and Material Availability:** The adequate resources of this article are publicly accessible.
- **Author's Contributions:** The authorship of this article is contributed equally to all participating individuals.

REFERENCES

1. Maya-Rodriguez, M. C. (2023). Temperature Control of a Chemical Reactor Based on Neuro-Fuzzy Techniques. In Energies 16(17)

6187. MDPI. DOI: https://doi.org/10.3390/en16176187
2. Hemanth. D, Mahalingam. M Subramanian. K, and Vijayanandh.T (2023). Automatic Adjustable Exhaust System for Temperature Regulation in Granaries Using Fuzzy Logic, 17pages. DOI: https://dx.doi.org/10.2139/ssrn.4671533
3. Kelvin Ndubuisi Nnamani (2021). Fuzzy Modelling of Automatic Greenhouse Control. In International Research Journal of Modernization in Engineering Technology and Science (Volume: 03, Issue: 03. pp: 87-92). https://www.irjmets.com/paperdetail.php?paperId=147ce121ee9e623ab026a3e5bc1e2b88
4. P. N. Nwankwo, K. A. Akpado, and Christiana C. Okezie (2024). Development of an Advanced Neuro-Fuzzy Algorithm for an Intelligent Temperature Control System. In International Journal of Advances in Engineering and Management (IJAEM) (Volume 6, Issue 09, pp 769-791). https://www.ijaem.net/past-issue-volume.php?issueid=71&title=Volume%206%20,%20Issue%209,%20September%202024
5. Savkovic. B, Kovac. P, Dudic. B, Rodic. D, Taric. M and Gregus. M (2019). Application of an Adaptive "Neuro-Fuzzy" Inference System in Modelling Cutting Temperature during Hard Turning. MDPI. DOI: https://doi.org/10.3390/app9183739
6. Doaa M. A and Hanaa T. E (2017). Analysis and design of greenhouse temperature control using an adaptive neuro-fuzzy inference system. In Journal of Electrical Systems and Information Technology (volume 4, No.1, pp 34-48). DOI: https://doi.org/10.1016/j.jesit.2016.10.014
7. Semih Kale (2020). Development of an Adaptive Neuro-Fuzzy Inference System (ANFIS) model to predict sea surface temperature (SST). In International Journal of Oceanography and Hydrobiology (Volume 49, No.4, pp. 354-373). DOI: https://doi.org/10.1515/ohs-2020-0031
8. E. A Joseph, H.S. Okeke and M.A. Taiwo (2021). Development of Neuro-Fuzzy-Based Temperature Control for Egg Hatchery. In International Journal of Trend in Scientific Research and Development (IJTSRD)(Volume 5, Issue 4, pp: 598-601). https://www.ijtsrd.com/papers/ijtsrd41215.pdf
9. K. N. Nnamani, K. A. Akpado and S. U. Ufoaroh (2025). Integrated Neuro-Fuzzy Expert System for the Control of Hydro-Power Reservoir System. In International Research Journal of Modernization in Engineering Technology and Science (Volume 07, Issue 09. pp: 3476-3485). DOI: https://doi.org/10.56726/IRJMETS83194

AUTHOR'S PROFILE

Kelvin N. Nnamani is currently pursuing a Doctorate (PhD) in the Department of Electronic and Computer Engineering, with a focus on Computer and Control Systems Engineering, at Nnamdi Azikiwe University in Awka, Anambra State, Nigeria. He holds a Master of Engineering (M.Eng.) in Electronic and Computer Engineering from this esteemed institution. He is dedicated to advancing knowledge in Control Engineering, Artificial Neural Networks, and Embedded System Programming, and is committed to contributing innovative solutions through his research.

Professor Ken Aghaegbunam Akpado is a distinguished lecturer in the Department of Electronic and Computer Engineering at Nnamdi Azikiwe University, Awka. He possesses expertise in Computer and Control Engineering and has successfully supervised a considerable number of undergraduate and postgraduate students in this discipline. His research interests encompass process control systems, remote monitoring, intelligent solutions for medical diagnostics, and artificial intelligence. Furthermore, he has previously served as the Head of the Department of Electronic and Computer Engineering at UNIZIK. Currently, he holds the esteemed roles of visiting professor and external examiner at various universities throughout Nigeria.

Professor Augustine C.O. Azubogu is a highly respected member of the Department of Electronic and Computer Engineering at Nnamdi Azikiwe University in Awka, Nigeria. He holds a Bachelor of Engineering (B.Eng.) in Electronic Engineering from the University of Nigeria, Nsukka, a Master of Engineering (M.Eng.) in Electronics and Telecommunications from the University of Port Harcourt, and a Doctorate (PhD) in Wireless Communication Engineering and Digital Signal Processing from Nnamdi Azikiwe University. With over 25 years of experience in engineering and academia, Professor Azubogu has been with Nnamdi Azikiwe University since 2005. He currently serves as a professor and the Deputy Director of the Technology Incubation Centre/IP PTO, where he plays a crucial role in fostering technological innovation and advancement.