BIU Journal of Basic and Applied Sciences 10(1): 13 – 28, 2025.
©Faculty of Science, Benson Idahosa University, Benin City, Nigeria
ISSN: 2563-6424

NEW APPROACH FOR SPAM DETECTION USING ADAPTIVE NEURO-FUZZY INFERENCE SYSTEM (ANFIS)

***AHUBELE, B. O. AND INYANG, I. A.**
Department of Computer Science, Faculty of Science, Benson Idahosa University, Benin City, Edo State, Nigeria
*Corresponding author: bahubele@biu.edu.ng

ABSTRACT

Spam refers to unwanted bulk messages sent to numerous recipients without their consent. These messages may consist of commercials for products and services, business opportunities, or misleading material meant to fool people into divulging financial or personal information. Spam can take many different forms, including spam on social media networks, emails, texts, and more. Spam primarily aims to reach as many people as possible to promote goods, services, or dubious schemes. In this study, we developed an adaptive neuro fuzzy system for classifying email spam. The Adaptive Neuro-Fuzzy Inference System (ANFIS) is a hybrid model for spam detection that combines the advantages of fuzzy logic and neural networks. The ANFIS model is a powerful tool for classifying spam due to its ability to effectively handle the complexities of spam detection. The ANFIS model developed was trained on a preprocessed dataset that classifies emails into spam and not spam. The class data was decategorized. transformed to a float and retrained to address this. Starting with 3 membership functions and a hybrid optimization algorithm in the range of 100, 500, and 1000 epochs. The minimum Root Mean-Squared Error (RMSE) obtained occurred after the 1000th epoch as 0.191166 with four membership functions. A training accuracy of 80.88% was achieved. After testing the model with 30% of previously unseen test data, a training error of 0.25145 was obtained, representing a prediction accuracy of 74.86%. The model's performance was evaluated using statistical metrics such as Mean Squared Error (MSE) and RMSE. RMSE is a metric that indicates how close the predicted values are to the actual values by calculating the square root of the average squared differences between the predicted and observed values. A lower RMSE indicates better model performance. Accuracy, on the other hand, refers to the percentage of emails correctly classified by the model, where a higher accuracy indicates better classification performance. The model achieved a minimum RMSE of 0.392044 with three membership functions and 500 iterations.

KEYWORDS: *Spam detection, Ham, Adaptive Neuro-Fuzzy Inference System (ANFIS), Machine Learning, Filtering*

INTRODUCTION

In recent times, unwanted commercial bulk emails called spam have become a huge problem on the internet. The person sending the spam messages is referred to as the spammer (Dada et al, 2019). They usually gather emails addresses and phone numbers, from different websites, chat rooms and viruses (Awad and Foqaha, 2016). These unwanted messages usually prevent the user from making complete and effective use of time, storage capacity and network bandwidth. Sometimes these messages have resulted in users suffering untold financial loss by falling for internet scam sponsored by spammers pretending to be from reputable companies like banks with the intention of collecting important information like passwords, bank details and credit/debit card numbers. (Dada *et al.*, 2019).

Spam prevents the user from making full and good use of time, storage capacity, and network bandwidth. The huge volume of spam mail flowing through the computer networks has destructive effects on the memory space of email servers, communication bandwidth, CPU power, and user time (Fonseca *et al.*, 2016). The menace of spam email is on the increase on a yearly basis and responsible for over 77% of the whole global email traffic (Kaspersky Lab Spam Report, 2023). According to a report from Kaspersky Lab, in 2015, the volume of spam emails being sent was reduced to a 12-year low. Spam email volume fell below 50% for the first time since 2003. In June 2015, the volume of spam emails went down to 49.7%, and in July 2015 the figures were further reduced to 46.4%, according to anti-virus software developer Symantec. This decline was attributed to a reduction in the number of major botnets responsible for sending spam emails in the billions. Malicious spam email volume was reported to be constant in 2015.

Latest statistics from Ellis (2024) show that spam messages accounted for 45.6% of e-mail traffic worldwide, and the most familiar types of spam emails were healthcare and dating spam. Spam results in unproductive use of resources on Simple Mail Transfer Protocol (SMTP) servers since they must process a substantial volume of unsolicited emails. The task of spam filtering is to automatically rule out unsolicited mails from a user's mail stream (Abdelrahim *et al.*, 2000). The task of spam filtration is to determine whether the mail is spam or not (Kaur and Sarangal, 2009). To effectively handle the threat posed by email spam, leading email providers such as Gmail, Yahoo Mail, and Outlook have employed the combination of different machine learning (ML) techniques, such as neural networks, in their spam filters. These ML techniques have the capacity to learn and identify spam mails and phishing messages by analyzing loads of such messages throughout a vast collection of computers. Since machine learning has the capacity to adapt to varying conditions, Gmail and Yahoo Mail spam filters do more than just check junk emails using pre-existing rules.

LITERATURE REVIEW

Spam filtering involves the identification of dangerous incoming mails from attackers or marketers. There exist several spam filtering methods. Some of the efficient and state-of-the-art approaches include content-based spam filtering techniques, case-based spam filtering, heuristic or rule-based spam filtering techniques, and previous likeness-based spam filtering techniques.

Machine learning (ML) algorithms have been extensively applied in the field of spam filtering. Substantial work has been done to improve the effectiveness of spam filters for classifying emails as either ham (valid messages) or spam (unwanted messages) through ML classifiers. They can recognize distinctive characteristics of the contents of emails. Many significant works have been done in the field of spam filtering using techniques that do not possess the ability to adapt to different conditions and problems that are exclusive to some fields, e.g., identifying messages that are hidden inside a stego image. Most of the machine learning algorithms used for classification of tasks were designed to learn about inactive objective groups. Barreno *et al.* (2006) posited that when these algorithms are trained on data that has been poisoned by an enemy, it makes the algorithms susceptible to several different attacks on the reliability and accessibility of the data. As a matter of fact, manipulating as little as 1% of the training data is enough in certain instances (Nelson *et al*., 2008).

Neuro Fuzzy System (NFS) was proposed by Jango in 1991. Neuro-fuzzy results in a hybrid intelligent system that synergizes artificial neural networks and fuzzy logic techniques by combining the human-like reasoning style of fuzzy systems with the learning and connectionist structure of neural networks (Zhang *et al*., 2023). Combining these two technologies into an integrated system appears to be a promising path toward developing intelligent systems capable of capturing qualities characterizing the human brain. However, fuzzy logic and neural networks generally approach the design of intelligent systems from quite different angles. Each having their own sets of strengths and weaknesses, most attempts to combine these two technologies have the goal of using each technique's strengths to cover the weaknesses of the other.

Neural networks are essentially low-level computational algorithms that sometimes perform well in pattern recognition tasks. On the other hand, fuzzy logic provides a structural framework that uses and exploits the low-level capabilities of neural networks (Makhsoos *et al*., 2009). This neural network learning algorithm implies that a fuzzy system with linguistic information in its rule base can be updated or adapted using numerical information to gain an even greater advantage over a neural network that cannot make use of linguistic information. Also, the architecture uses a fuzzy system to represent knowledge in an interpretable manner and the learning ability of a neural network to optimize its parameters (Zhang *et al*., 2023). The main strength of neuro-fuzzy systems is that they are universal approximations with the ability to seek interpretable IF-THEN rules (Lughofer *et al*., 2022).

Bhowmick and Hazarika (2016) presented a broad review of some of the popular content-based e-mail spam filtering methods. The study focused mostly on machine learning algorithms for spam filtering. They surveyed the important concepts, efforts, effectiveness, and the trend in spam filtering. They discussed the fundamentals of e-mail spam filtering, the changing nature of spam, the tricks of spammers to evade spam filters of e-mail service providers (ESPs) and examined the popular machine learning techniques used in combating the menace of spam.

This research of Beaman *et al.* (2022) investigates the use of machine learning techniques applied to email header information for detecting anomalies such as spam and phishing emails. The study demonstrates that analyzing email headers alone can effectively identify malicious emails, highlighting the potential of anomaly detection methods in enhancing email security.

Karthika and Visalakshi (2015) proposed an approach that combined and implemented Support Vector Machine (SVM) and Ant Colony Optimization (ACO) algorithms for spam classification. The proposed technique is a hybrid model that relies on selecting the features of emails for their classification. The SVM basically works as the classifier, i.e., the classification algorithm, while the feature selection of emails is implemented by the ACO algorithm for efficiency and accuracy. The proposed method permits more than one mail to be classified at a time with an improved speed of execution. The proposed method performs better than some state-of-the-art (i.e., K-Nearest Neighbor-KNN, Naïve Bayes-NB, and SVM) classification methods in terms of accuracy, precision, and recall. Adopting the ACO algorithm for feature selection offers improved efficiency in spam mail classification. The drawback of their approach is low performance.

There are many approaches for spam detection and filtering. The spammer's creativity results in new spam emails that break filter rules. Therefore, learning-based adaptive detection becomes a key issue to cope with spam. The combination of learning-based adaptive detection systems filters out spam emails better. The main aim of this study is to generate a low error rate using a combination of the Adaptive Neuro-Fuzzy Inference System with the Genetic Algorithm, where the Genetic Algorithm tunes the fuzzy rule base.

METHODOLOGY

To employ Adaptive Neuro-Fuzzy Inference Systems (ANFIS) for our email spam classification problem using MATLAB, the methodology adopted is of the waterfall method, which includes data preparation, data pre-processing, feature extraction, partitioning data, and ANFIS model building and evaluation. The research methodology is depicted in Figure 1.

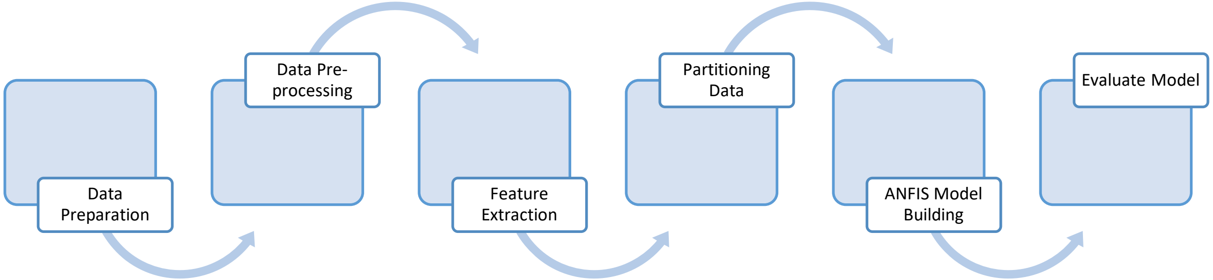
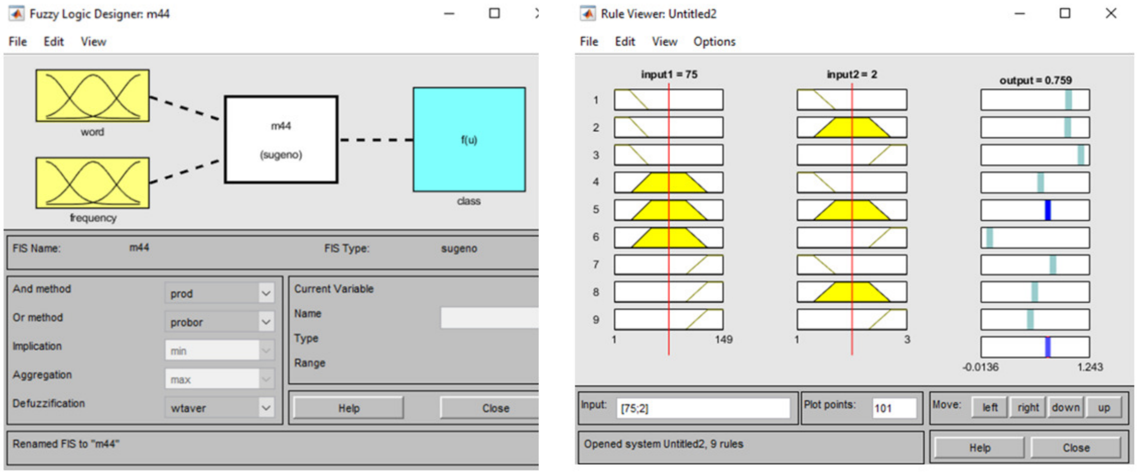


Fig. 1: Study Methodology Adopted

The Proposed Model Architecture

The ANFIS model developed (shown in figure 2) was trained on a preprocessed dataset that classifies emails into spam and not spam. Training based on binary classification resulted in a model with an RMSE that was not very acceptable because it was not near zero. Which resulted in this approach focusing on fuzzification in the data preparation and pre-processing, class data were transformed from binary (spam/non-spam) to fuzzy sets, allowing emails to have degrees of membership in both categories. Features like word frequency were extracted, with text converted to fuzzy representations for compatibility with the ANFIS model. Data normalization, missing value imputation using fuzzy logic, and feature selection based on fuzzy metrics were key pre-processing steps. Starting with 3 membership functions and a hybrid optimization algorithm in the range of 100, 500, and 1000 epochs. The minimum RMSE obtained occurred after the 1000th epoch as 0.191166 with four membership functions. This represents a training accuracy of 80.88%. After testing the model with the previously unseen test data (30%), a training error of 0.25145 was obtained. This represents a prediction accuracy of 74.86%. This shows an improvement of previous works carried out by other authors, including Solanki and Bagde (2024), who pointed out that Naïve Bayes and Random Forest models for the detection of spam social engineering attacks are 64% and 65%, respectively.



FIS Name: m44

FIS Type: sugeno

And method: prod

Or method: probor

Implication: min

Aggregation: max

Defuzzification: wtaver

Renamed FIS to "m44"

input1 = 75

input2 = 2

output = 0.759

Input: [75;2]

Plot points: 101

Opened system Untitled2, 9 rules

Fig. 2: The Proposed Model Architecture

Advantages of the Proposed Model

The proposed spam filtering classification approach using ANFIS offers the following advantages:

i. By integrating fuzzy logic, which deals with uncertainty and imprecision, with neural networks that can learn from data, ANFIS can effectively classify emails as spam or ham based on features like email content, sender information, and metadata.

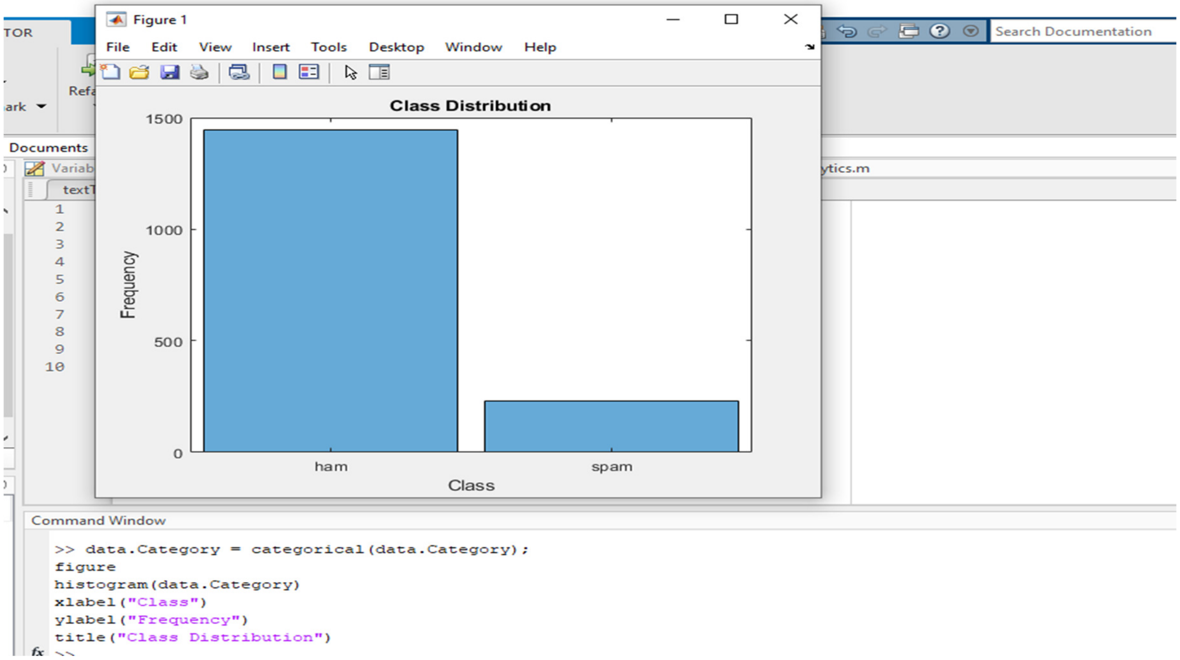
ii. The proposed approach outperforms the existing techniques in terms of classification accuracy.

iii. ANFIS is more robust and flexible in spam classification tasks compared to static fuzzy logic systems due to its ability to learn from feedback, adjust its parameters and rules, and adapt to changing spam patterns and evolving spam techniques.

iv. Ability to handle nonlinear relationships between input features and output, robustness to noisy data, and interpretability of the fuzzy rules.

RESULTS

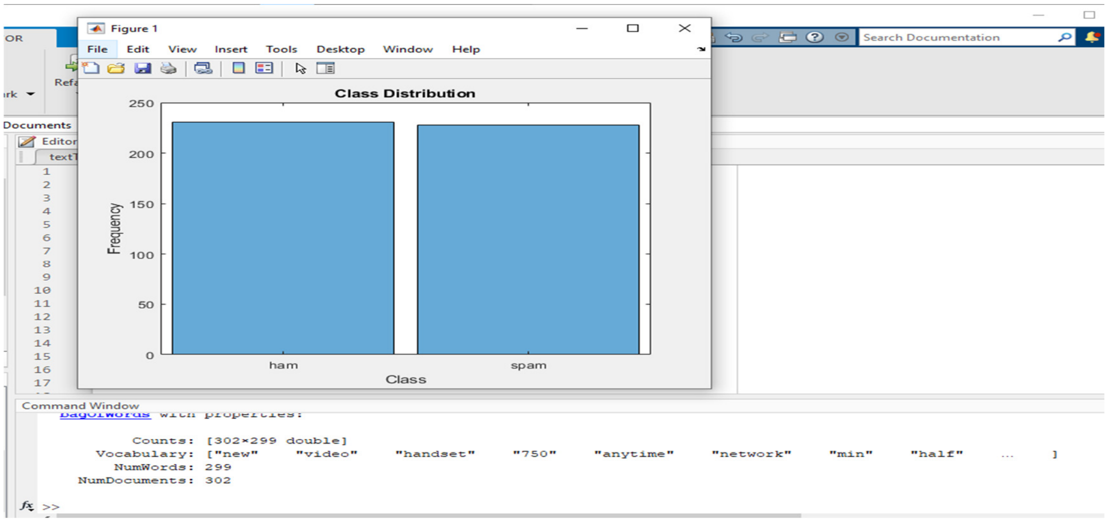
The dataset, after categorization and preprocessing, was examined based on the class distributions of spam and ham after classification into 0 and 1, respectively; this is illustrated in Figure 3 Analysis, which shows an uneven distribution between the various class distributions.



```
>> data.Category = categorical(data.Category);
figure
histogram(data.Category)
xlabel("Class")
ylabel("Frequency")
title("Class Distribution")
fx >>
```

Fig. 3: Class Distributions in Dataset

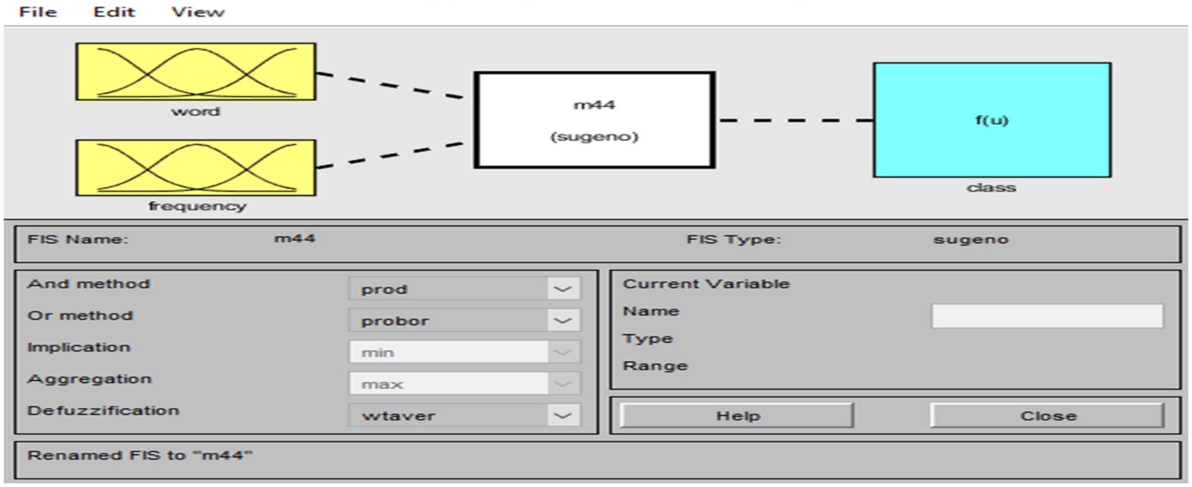
To resolve this discrepancy and normalize the class distributions, undersampling was undertaken. This resolved the dataset into the form represented below in Figure 4. It shows an equal distribution of the class sets.



```
bagOfWords with properties:

       Counts: [302×299 double]
   Vocabulary: ["new"    "video"    "handset"    "750"    "anytime"    "network"    "min"    "half"    …    ]
     NumWords: 299
 NumDocuments: 302

fx >>
```

Fig. 4: Class distribution in dataset after under sampling

After data preparation, preprocessing, and undersampling, the data was saved as a CSV file called feature_select.csv. The data consisted of three (3) columns, namely: Word, Frequency, and Class. Following this, the data was split into training and testing datasets, consisting of 70% and 30% of the dataset, respectively. The training dataset was used to train the ANFIS model, while the testing dataset was used to evaluate its performance. In order to create a fuzzy inference system, the ANFIS toolbox was used for modeling, and the inputs were subdivided into two (2) categories as shown in Figure 5.



FIS Name: m44 FIS Type: sugeno

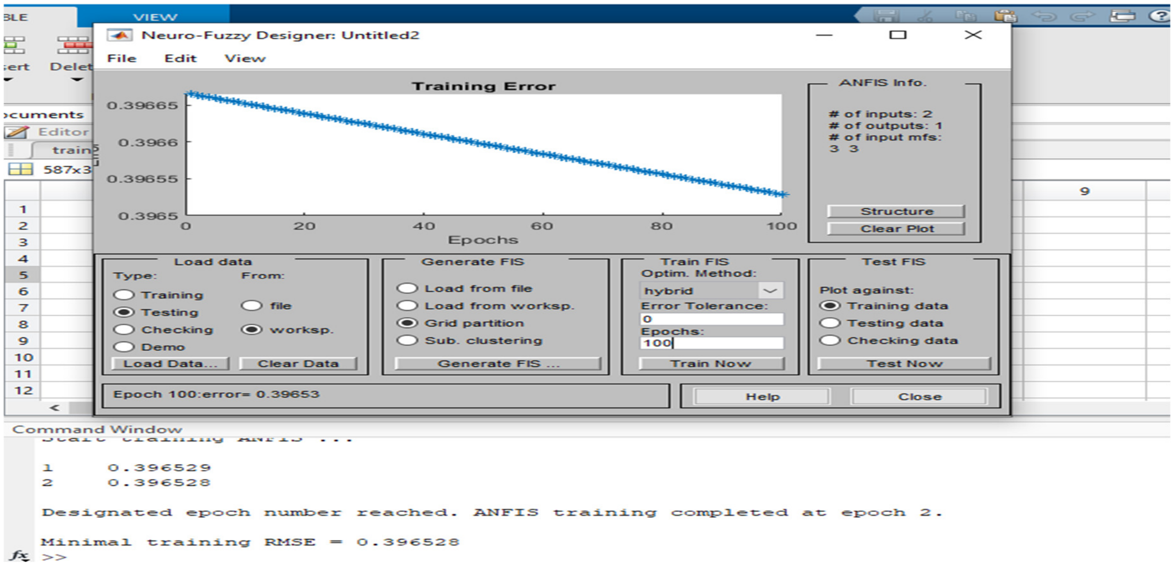
And method: prod
Or method: probor
Implication: min
Aggregation: max
Defuzzification: wtaver

Current Variable
Name
Type
Range

Renamed FIS to "m44"

Fig. 5: ANFIS toolbox showing inputs (word) and (frequency) and output (class).

The Neuro Fuzzy system used was fitted with the training data and used to train the model with three (3) membership functions and a number of epochs set to 100. The results of the training are shown in Figure 6. The minimum root mean square error after the 100th epoch was 0.39628.



# of inputs: 2
# of outputs: 1
# of input mfs:
3 3

Load data — Type: ○ Training ● Testing ○ Checking ○ Demo; From: ○ file ● worksp.

Generate FIS — ○ Load from file ○ Load from worksp. ● Grid partition ○ Sub. clustering

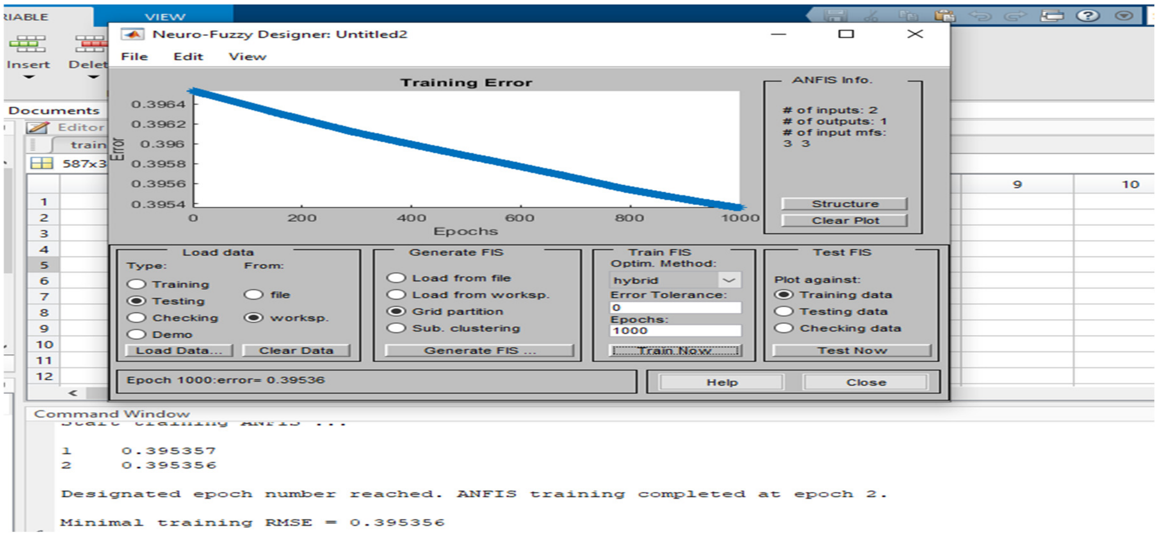
Train FIS — Optim. Method: hybrid; Error Tolerance: 0; Epochs: 100

Test FIS — Plot against: ● Training data ○ Testing data ○ Checking data

Epoch 100:error= 0.39653

```
Start training ANFIS ...

1     0.396529
2     0.396528

Designated epoch number reached. ANFIS training completed at epoch 2.

Minimal training RMSE = 0.396528
>>
```

Fig. 6: Training error over 100 Epoch, ANFIS model development phase



# of inputs: 2
# of outputs: 1
# of input mfs:
3 3

Load data — Type: ○ Training ● Testing ○ Checking ○ Demo; From: ○ file ● worksp.

Generate FIS — ○ Load from file ○ Load from worksp. ● Grid partition ○ Sub. clustering

Train FIS — Optim. Method: hybrid; Error Tolerance: 0; Epochs: 1000

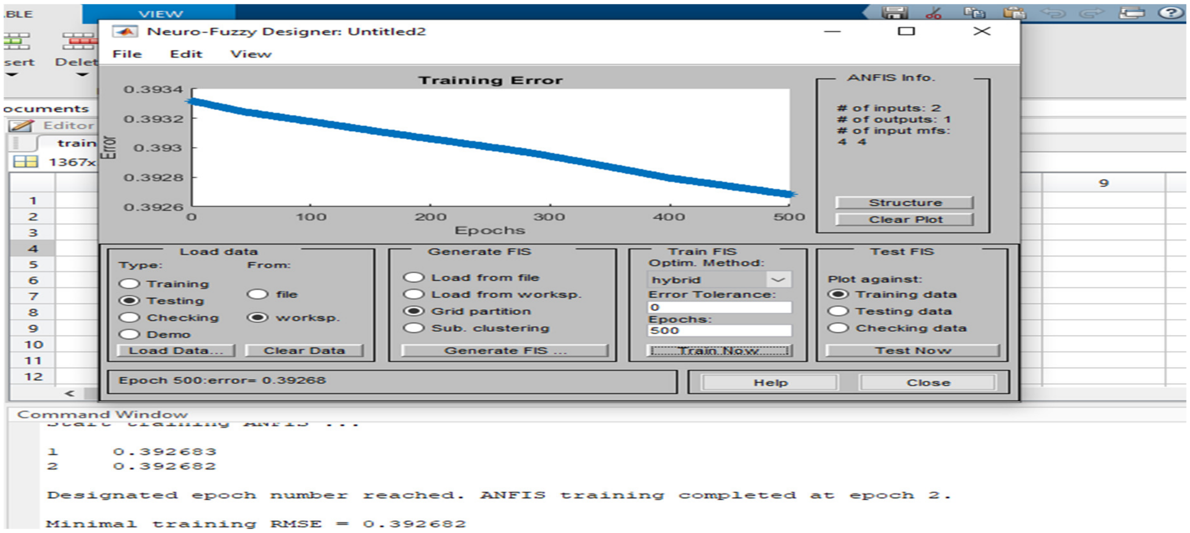
Test FIS — Plot against: ● Training data ○ Testing data ○ Checking data

Epoch 1000:error= 0.39536

```
Start training ANFIS ...

1     0.395357
2     0.395356

Designated epoch number reached. ANFIS training completed at epoch 2.

Minimal training RMSE = 0.395356
```

Fig. 7: Training error over 1000 Epoch, ANFIS model development phase

The hyperparameters were tuned to see if a lower value of RMSE could be obtained. The Neuro Fuzzy system was fitted with the training data and used to train the model with three (3) membership functions and a number of epochs set this time to 1000. The results of the training are shown in Figure 7. The minimum RMSE after the 1000th epoch was 0.393356.



ANFIS Info.
# of inputs: 2
# of outputs: 1
# of input mfs:
4 4

Type: ○ Training ● Testing ○ Checking ○ Demo
From: ○ file ● worksp.

○ Load from file
○ Load from worksp.
● Grid partition
○ Sub. clustering

Optim. Method: hybrid
Error Tolerance: 0
Epochs: 500

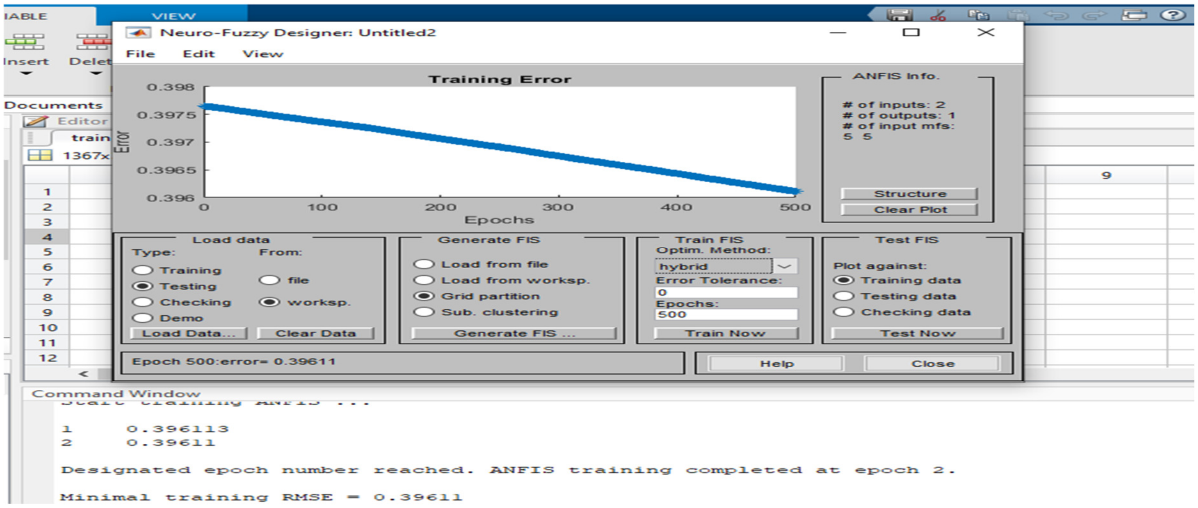
Plot against: ● Training data ○ Testing data ○ Checking data

Epoch 500:error= 0.39268

```
Start training ANFIS ...

1      0.392683
2      0.392682

Designated epoch number reached. ANFIS training completed at epoch 2.

Minimal training RMSE = 0.392682
```

Fig. 8: Training error over 500 Epoch, ANFIS model development phase

With four (4) membership functions and 500 epochs, the Neuro Fuzzy system was given the training data and used to teach the model. The results of the training are shown in Figure 8. The minimum RMSE after the 500th epoch was 0.392682.



ANFIS Info.
# of inputs: 2
# of outputs: 1
# of input mfs:
5 5

Type: ○ Training ● Testing ○ Checking ○ Demo
From: ○ file ● worksp.

○ Load from file
○ Load from worksp.
● Grid partition
○ Sub. clustering

Optim. Method: hybrid
Error Tolerance: 0
Epochs: 500

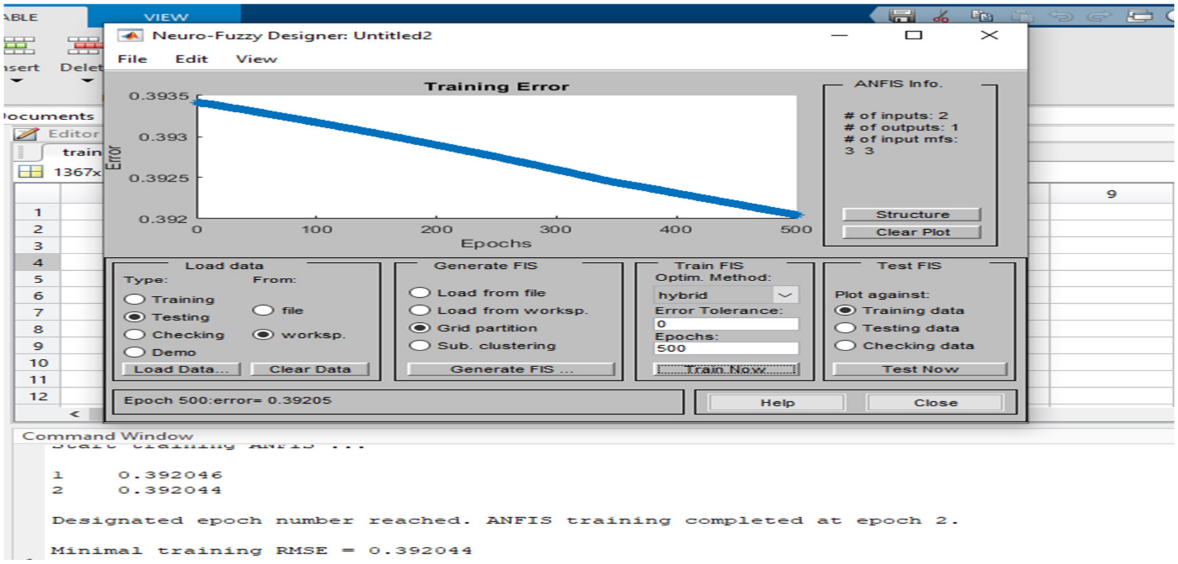
Plot against: ● Training data ○ Testing data ○ Checking data

Epoch 500:error= 0.39611

```
Start training ANFIS ...

1      0.396113
2      0.39611

Designated epoch number reached. ANFIS training completed at epoch 2.

Minimal training RMSE = 0.39611
```

Fig. 9: Training error over 5 membership function and 500 epoch, ANFIS development phase

The Neuro Fuzzy system was fitted with the training data and used to train the model with five (5) membership functions and a number of epochs set to 500. The results of the training are shown in Figure 9. The minimum RMSE after the 500th epoch was 0.39611.



ANFIS Info.: # of inputs: 2 # of outputs: 1 # of input mfs: 3 3

Load data — Type: ○ Training ● Testing ○ Checking ○ Demo; From: ○ file ● worksp.

Generate FIS: ○ Load from file ○ Load from worksp. ● Grid partition ○ Sub. clustering

Train FIS — Optim. Method: hybrid; Error Tolerance: 0; Epochs: 500

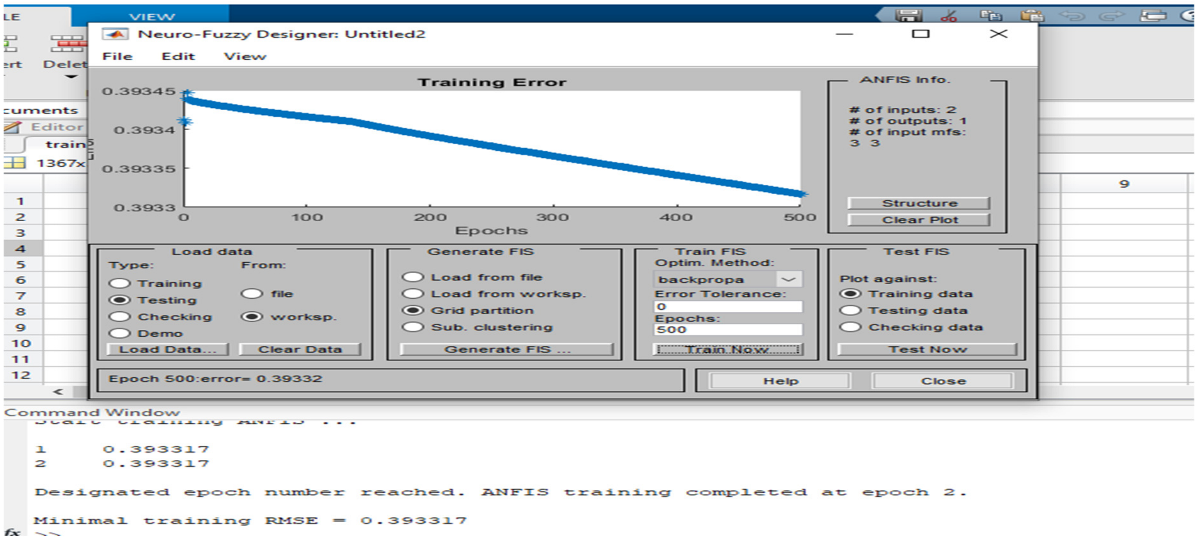
Test FIS — Plot against: ● Training data ○ Testing data ○ Checking data

Epoch 500:error= 0.39205

```
Start training ANFIS ...

1      0.392046
2      0.392044

Designated epoch number reached. ANFIS training completed at epoch 2.

Minimal training RMSE = 0.392044
```

Fig. 10: Training error over 3 membership function and 500 epoch, ANFIS development phase

The Neuro Fuzzy system was fitted with the training data and used to train the model with three (3) membership functions and a number of epochs set to 500. The results of the training are shown in Figure 10. The minimum RMSE after the 500th epoch was 0.392044.



ANFIS Info.: # of inputs: 2 # of outputs: 1 # of input mfs: 3 3

Load data — Type: ○ Training ● Testing ○ Checking ○ Demo; From: ○ file ● worksp.

Generate FIS: ○ Load from file ○ Load from worksp. ● Grid partition ○ Sub. clustering

Train FIS — Optim. Method: backpropa; Error Tolerance: 0; Epochs: 500

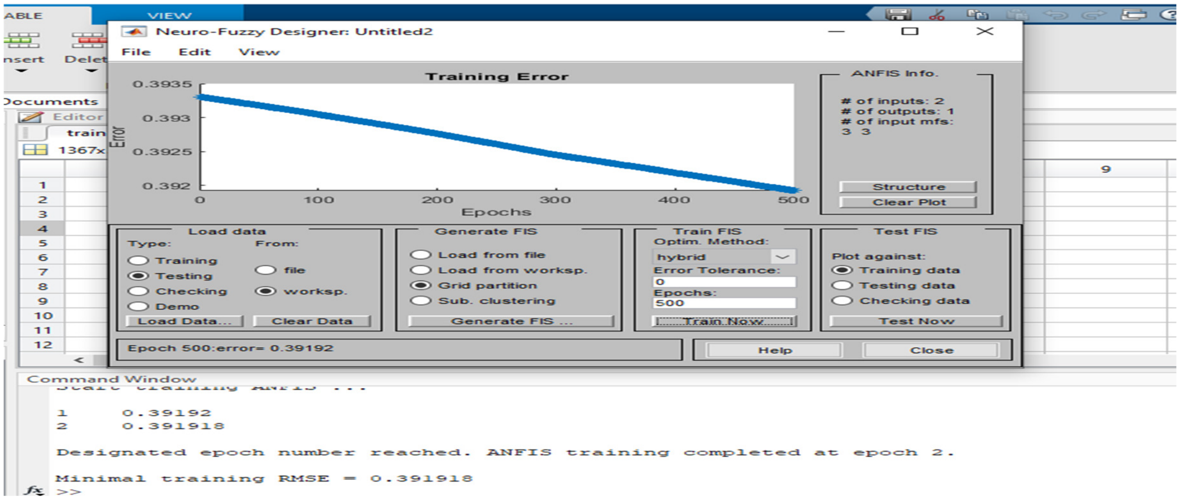
Test FIS — Plot against: ● Training data ○ Testing data ○ Checking data

Epoch 500:error= 0.39332

```
Start training ANFIS ...

1      0.393317
2      0.393317

Designated epoch number reached. ANFIS training completed at epoch 2.

Minimal training RMSE = 0.393317
```

Fig. 11: Training error over 3 membership function and 500 epoch – backpropagation optimizer, ANFIS development phase

The Neuro Fuzzy system was fitted with the training data and used to train the model with three (3) membership functions and a number of epochs set to 500. But this time using the backpropagation algorithm optimization algorithm rather than the hybrid algorithm, which was used for the others. The results of the training are shown in Figure 11. The minimum RMSE after the 500th epoch was obtained as 0.393317.



ANFIS Info.
# of inputs: 2
# of outputs: 1
# of input mfs:
3 3

Load data Type: ● Testing; From: ● worksp.

Generate FIS: ● Grid partition

Optim. Method: hybrid
Error Tolerance: 0
Epochs: 500

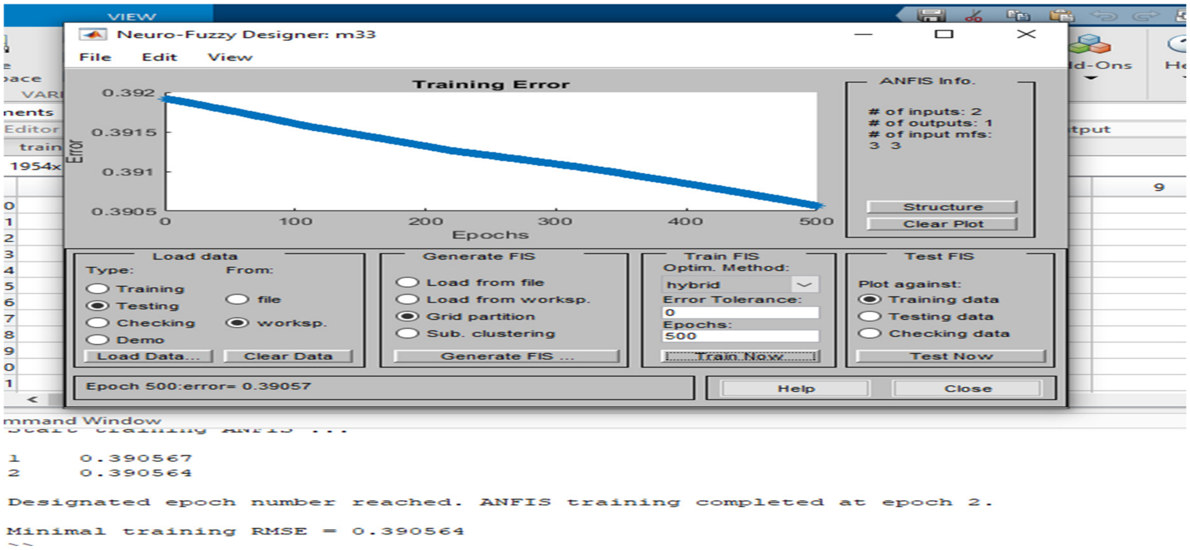
Plot against: ● Training data

Epoch 500:error= 0.39192

```
Start training ANFIS ...

1      0.39192
2      0.391918

Designated epoch number reached. ANFIS training completed at epoch 2.

Minimal training RMSE = 0.391918
>>
```

Fig. 12: Training error over 3 membership function and 500 epoch – hybrid optimizer, ANFIS development phase)

The Neuro Fuzzy system was fitted with the training data and used to train the model with three (3) membership functions and a number of epochs set to 500, using the hybrid optimization algorithm, which was used for the others, to see if there would be a change to the RMSE. The results of the training are shown in Figure 12. The minimal RMSE obtained after the 500th epoch was 0.391918.



ANFIS Info.
# of inputs: 2
# of outputs: 1
# of input mfs:
3 3

Load data Type: ● Testing; From: ● worksp.

Generate FIS: ● Grid partition

Optim. Method: hybrid
Error Tolerance: 0
Epochs: 500

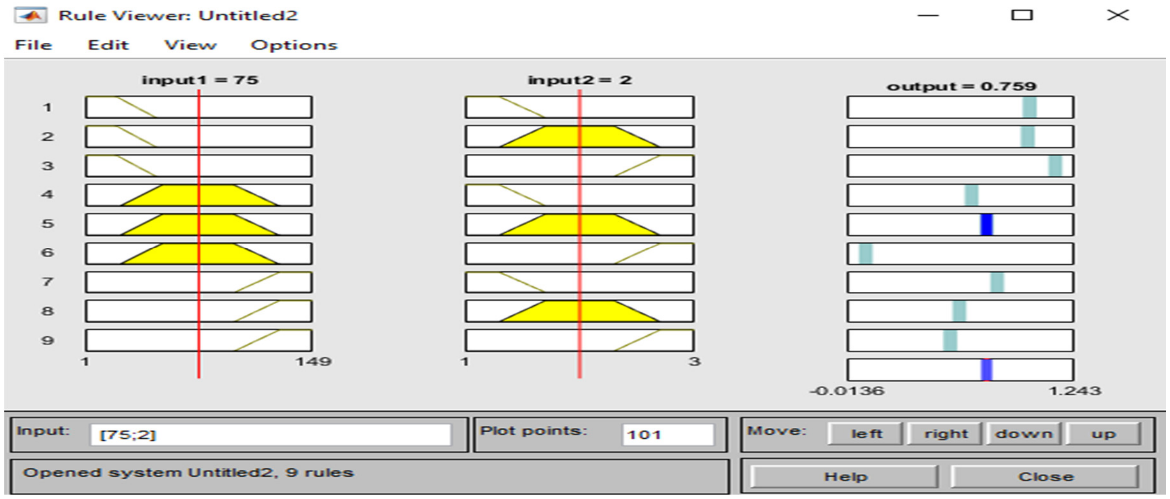
Plot against: ● Training data

Epoch 500:error= 0.39057

```
Start training ANFIS ...

1      0.390567
2      0.390564

Designated epoch number reached. ANFIS training completed at epoch 2.

Minimal training RMSE = 0.390564
```

Fig. 13: Effect of multiple iteration over 3 membership function and 500 epoch – backpropagation optimizer, ANFIS development phase

The Neuro Fuzzy system was fitted with the training data and used to train the model with three (3) membership functions and a number of epochs set to 500. Using the hybrid optimization algorithm once again to see the effect of multiple iterations. The results of the training as shown in Figure 13. The minimum RMSE after the 500th epoch was obtained as 0.390564.



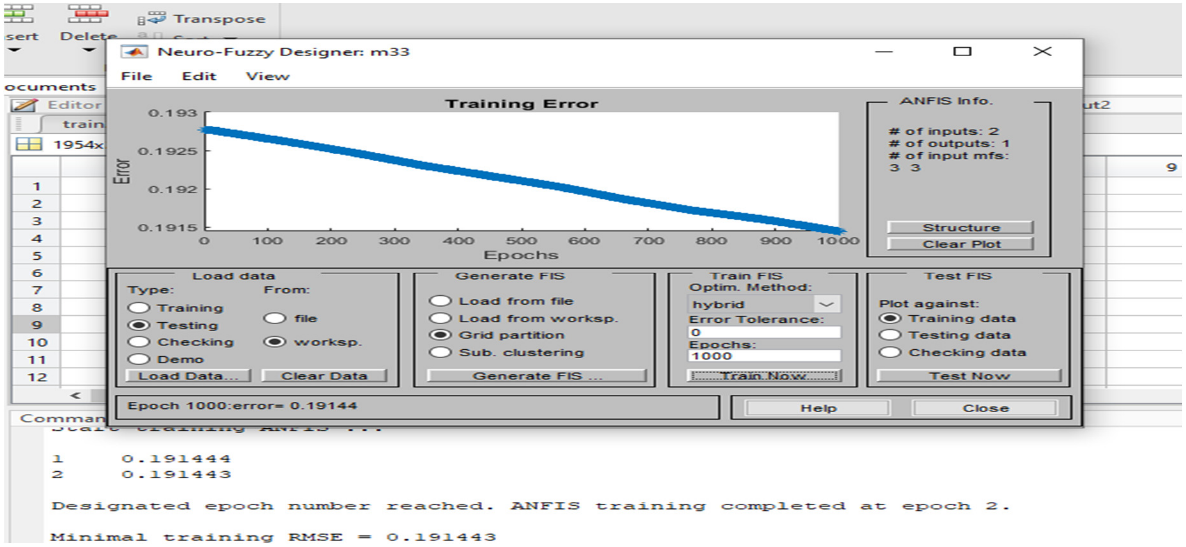
Input: [75;2]

Plot points: 101

Opened system Untitled2, 9 rules

Fig. 14: 9 Inference Rule Based Viewer for ANFIS model

The Rule Viewer for the ANFIS model is shown in Figure 14; it shows the nine (9) inference rules and shows how the fuzzy logic system makes decisions based on the inputs. It was observed that the ANFIS model did not give a very acceptable RMSE as it was not so close to zero. In order to solve this problem, the class data was decategorized and converted to a float, and retraining was undertaken.



# of inputs: 2
# of outputs: 1
# of input mfs:
3 3

Type: ○ Training ● Testing ○ Checking ○ Demo

From: ○ file ● worksp.

○ Load from file
○ Load from worksp.
● Grid partition
○ Sub. clustering

Optim. Method: hybrid

Error Tolerance: 0

Epochs: 1000

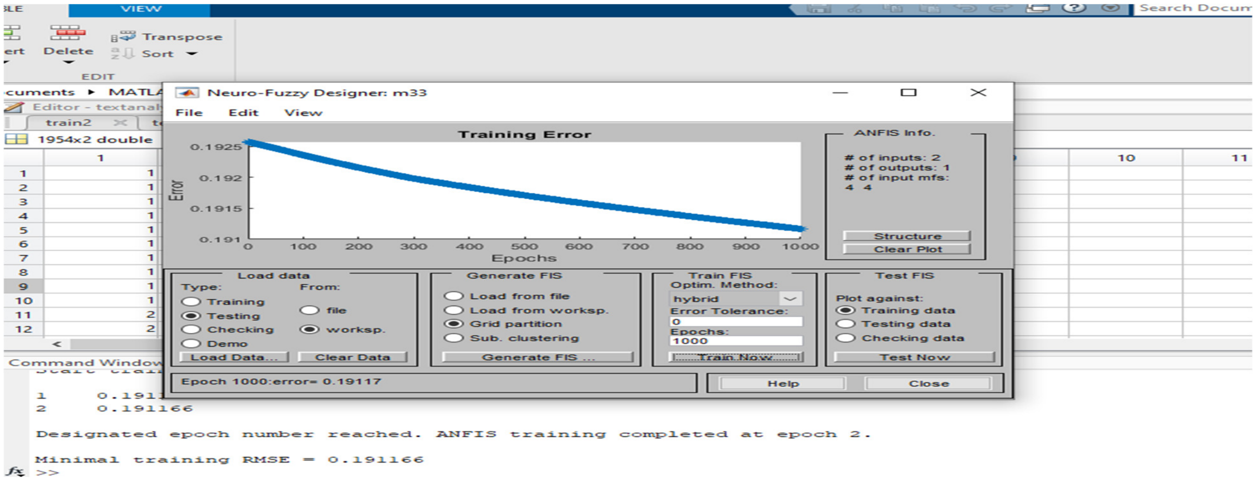
Plot against: ● Training data ○ Testing data ○ Checking data

Epoch 1000:error= 0.19144

```
Start training ANFIS ...

1      0.191444
2      0.191443

Designated epoch number reached. ANFIS training completed at epoch 2.

Minimal training RMSE = 0.191443
```

Fig. 15: Training error over 3 membership function and 1000 epoch – hybrid optimizer, ANFIS development phase

The Neuro Fuzzy system was fitted with the new training data and used to train the model with three (3) membership functions and the number of epochs set to 1000. Using the hybrid optimization algorithm once again to see the effect of multiple iterations. The results of the training are shown in Figure 15. The minimum RMSE after the 1000th epoch was obtained as 0.191443.



1954x2 double

ANFIS Info.
# of inputs: 2
# of outputs: 1
# of input mfs:
4 4

Load data
Type: ○ Training ● Testing ○ Checking ○ Demo
From: ○ file ● worksp.

Generate FIS
○ Load from file
○ Load from worksp.
● Grid partition
○ Sub. clustering

Train FIS
Optim. Method: hybrid
Error Tolerance: 0
Epochs: 1000

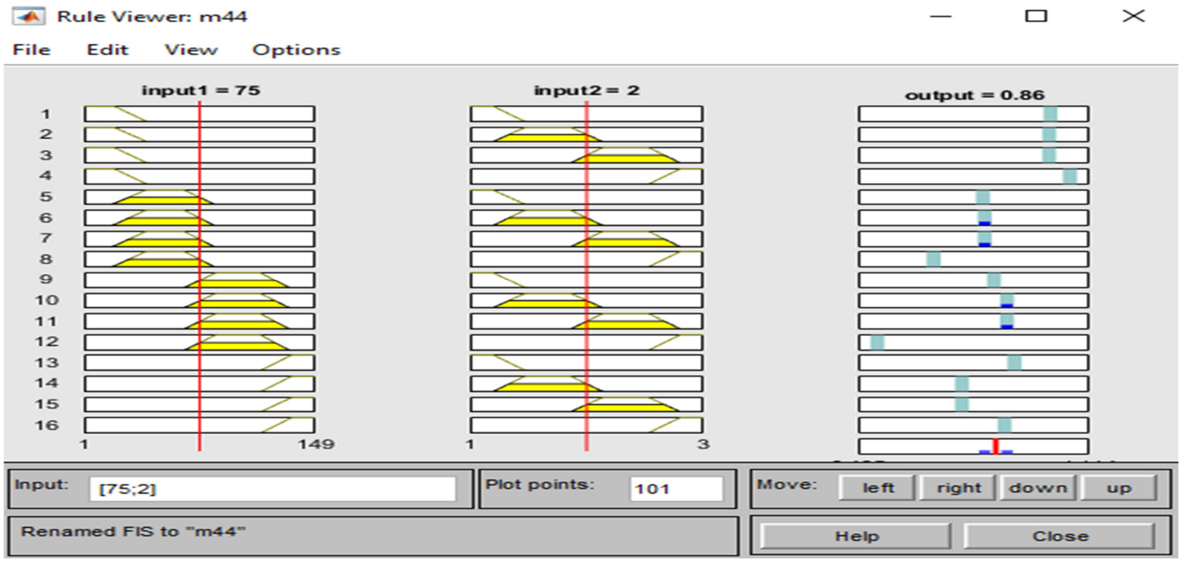
Test FIS
Plot against:
● Training data
○ Testing data
○ Checking data

Epoch 1000:error= 0.19117

```
1     0.191
2     0.191166

Designated epoch number reached. ANFIS training completed at epoch 2.

Minimal training RMSE = 0.191166
>>
```

Fig. 16: ANFIS Model Development (9)

The Neuro Fuzzy system was fitted with the new training data and used to train the model, but this time with four (4) membership functions and a number of epochs set to 1000. Using the hybrid optimization algorithm once again to see the effect of multiple iterations. The results of the training are shown in Figure 16. The minimum RMSE after the 1000th epoch was obtained as 0.191166. This represents the most minimal RMSE obtained from all the results presented.



Input: [75;2]

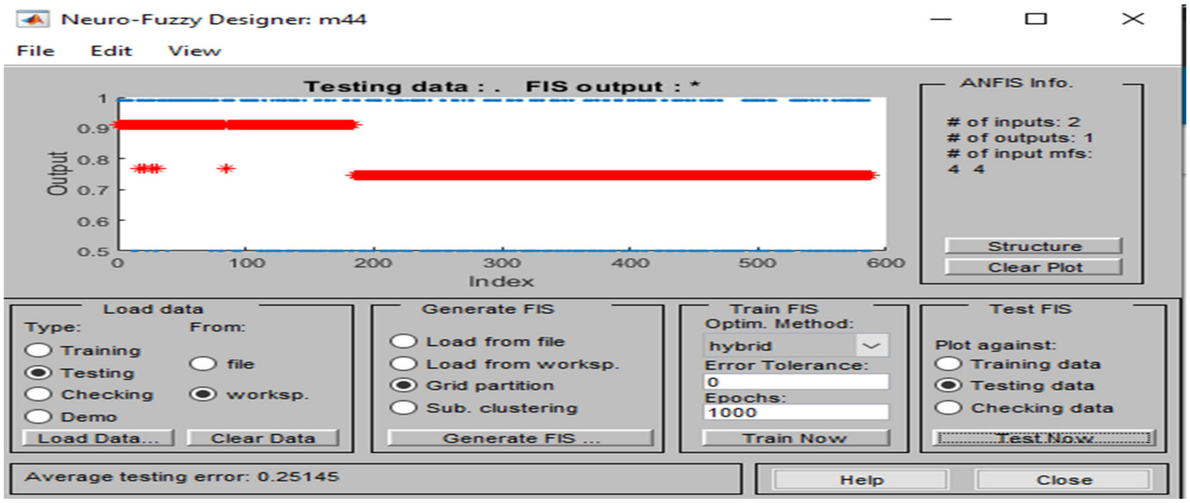
Plot points: 101

Renamed FIS to "m44"

Fig. 17: 16 Inference Rule Based Viewer for new ANFIS model

The Rule Viewer for the new ANFIS model is shown in Figure 17; it shows sixteen (16) inference rules and shows how the fuzzy logic system makes decisions based on the inputs.

It was observed that the new ANFIS model gave a very acceptable RMSE as it was closer to zero than any of the other models.



ANFIS Info.: # of inputs: 2 / # of outputs: 1 / # of input mfs: 4 4

Load data — Type: ○ Training ● Testing ○ Checking ○ Demo; From: ○ file ● worksp.

Generate FIS — ○ Load from file ○ Load from worksp. ● Grid partition ○ Sub. clustering

Train FIS — Optim. Method: hybrid; Error Tolerance: 0; Epochs: 1000

Test FIS — Plot against: ○ Training data ● Testing data ○ Checking data

Average testing error: 0.25145

Figure 18: Fitting over 4 membership function and 1000 epoch -hybrid optimizer, ANFIS development phase

The Neuro Fuzzy system was fitted with the new training data and used on the test data in order to measure the prediction accuracy of our model against unseen data. Constant factors include the original parameters, with four (4) membership functions and the number of epochs set to 1000. Using the hybrid optimization algorithm. The results of the accuracy are shown in Figure 18. The training error after the 1000th epoch was obtained as 0.25145. This represents an accuracy of 74.856%.

DISCUSSION

The data was saved as a feature_select.csv file and split into training and testing datasets. The ANFIS toolbox was used to create a fuzzy inference system, with inputs subdivided into two categories, word and count frequency. The ANFIS system was trained with the dataset 70%, and the RMSE was measured after each set.

We started with 3 membership functions and a hybrid optimization algorithm in the range of 100, 500, and 1000 epochs, and we obtained a minimal RMSE of 0.392044 with three membership functions and 500 iterations. The ANFIS model was found to provide an RMSE that was not very acceptable because it was not near zero. The class data was decategorized, transformed to a float, and retrained in order to address this.

The minimum RMSE obtained occurred after the 1000th epoch as 0.191166 with four membership functions. This represents a training accuracy of 80.88%. After testing the model with the previously unseen test data (30%), we obtained a training error of 0.25145. This represents a prediction accuracy of 74.86%. This shows an improvement of previous works carried out by other authors, including Solanki and Bagde (2024), who pointed out that Naïve Bayes and Random Forest models for the detection of spam social engineering attacks are 64% and 65%, respectively.

CONCLUSION

Adaptive Neuro-Fuzzy Inference System (ANFIS) is an effective technique for spam classification that combines the strengths of neural networks and fuzzy logic. Our system for spam classification leverages fuzzy logic and neural networks to accurately detect spam emails based on their content and metadata features. Its ability to handle nonlinearity and provide interpretable results makes it an effective choice for spam detection systems. The development of the system using the MATLAB toolbox provided a comprehensive detail of how ANFIS can effectively classify emails into spam and ham. The Adaptive Neuro-Fuzzy Inference System demonstrated superior performance, robustness, adaptability, and interpretability compared to other spam classification methods based on the research studies provided.

REFERENCES

Abdelrahim, A. A., Ahmed, A., El-hadi, and Hamza, I. (2000). Feature Selection and Similarity Coefficient Based Method Email Spam Filtering. *International Conference on Computing, Electrical and Electronic Engineering,* 16(21): 245–250.

Awad, M. and Foqaha, M. (2016). Email spam classification using hybrid approach of RBF neural network and particle swarm optimization. *Int. J. Netw. Secur. Appl.*, 8(4):

Barreno, M., Nelson, B., Sears, R. and Joseph, J. D. Tygar, J. D. (2006). Can machine learning be secure? in: *Proceedings of the 2006 ACM Symposium on Information Computer and Communications Security, Taipei, Taiwan. 16–25.*

Beaman, C., Canadian Institute for Cybersecurity, Isah, H., & Canadian Institute for Cybersecurity. (2022). Anomaly Detection in Emails using Machine Learning and Header Information. In *Canadian Institute for Cybersecurity* [Journal-article].

Bhowmick, A. and Hazarika, S. M. (2016). Machine Learning for E-Mail Spam Filtering: Review, Techniques and Trends, arXiv:1606.01042v1.1–27.

Dada, G. E., Bassi, J. S., Chiroma, H., Abdulhamid, M., Adetumbi, A. O., and Ajibuwa, O. E. (2019). Machine learning for email spam filtering: review, approaches and open research problems. 5(6), e01802

Fonseca, D. M., Fazzion, O. H., Cunha, E., Las-Casas, I., Guedes, P. D., Meira, W. and Chaves, M. (2016). Measuring characterizing and avoiding spam traffic costs. *IEEE Int. Comp*. 99 (2016).

Karthika, R. and Visalakshi, P. (2015). A hybrid ACO based feature selection method for email spam classification, *WSEAS Trans. Comput*. 14: 171–177.

Kaspersky Lab Spam Report (2023). https://securelist.com/spam-phishing-report-2023/112015/

Kaur, A. and Sarangal, M. (2009). A Hybrid Approach for Enhancing the Capability of Spam Filter. *International Journal of Computer Applications Technology and Research*, 2(6). ISSN 759 – 762.

Lughofer, E. (2022). Evolving fuzzy and neuro-fuzzy systems:

Fundamentals, stability, explainability, useability, and applications. In Handbook on Computer Learning and Intelligence: Volume 2: Deep Learning, Intelligent Control and Evolutionary Computation (pp. 133-234).

Makhsoos, N. T., Ebrahimpour, R. and Hajiany, A. (2009). Face Recognition Based on Neuro-Fuzzy System, 9(4): 319–326.

Nelson, B., Barreno, M., Chi, F.J., Joseph, A. D., Rubinstein, B. I. P., Saini, U., Sutton, C., Tygar, J. D. and Xia, K. (2008). Exploiting Machine Learning to subvert your spam filter, in: Proceedings of the 1st USENIX Workshop on Large-Scale Exploits and Emergent Threats, San Francisco, California. 1–9.

Solanki, R. and Bagde, P. (2024). Employing ANFIS model for detection of Spam Social Engineering Attack, *International Journal of Engineering Research and Technology* (IJERT) 13(04). 1-6.

Zhang, K., Hao, W., Yu, X. and Shao, T. (2023). A Symmetrical Fuzzy Neural Network Regression Method Coordinating Structure and Parameter Identifications for Regression. *Symmetry*, 15(9): 1711.