European Journal of Artificial Intelligence and Machine Learning
www.ej-ai.org

RESEARCH ARTICLE

Fault Diagnosis for Momentum Wheels of Communication Satellite Based on Artificial Neural Network

Ajah C. Ogbonnaya, Emmanuel Eronu, Farouq E. Shaibu, Ikechukwu N. Amalu, and Bara'u G. Najashi

ABSTRACT

With the advent of several fault detection techniques in modern control systems design, this paper adopted the Artificial Neural Network (ANN) Fault Detection scheme for the Fault Detection of the Attitude Control System for a Communication Satellite. In satellite applications, telemetry data can be very large, and ANN is best suited for network modeling involving large sets of data. The availability of real satellite data from Nigcomsat-1R communication satellite provided a practical platform to assess the fault detection algorithm. Results obtained showed a good correlation between raw satellite telemetry data and Neural Network model-generated results for subsequent fault detection. The fault detection models were able to detect faults, log them and provide a notification to enhance subsequent isolation and rectification. Momentum Wheel Speed and Torque were used to investigate the performance of the wheels while the Momentum Wheel Voltage and Current helped to monitor the wheel's health state. A fault is detected if the absolute difference between original output (MW Torque) and the NN Torque output is greater than 0.012. With this, an accuracy of 100% and mean squared error of 9.8489e-6 were achieved.

Keywords: artificial neural network, communication satellite, fault detection, momentum wheel attitude control system.

Published Online: March 5, 2023

ISSN: 2796-0072

DOI: 10.24018/ejai.2023.2.2.18

A. C. Ogbonnaya*
Nigerian Communications Satellite Limited, Nigeria
(e-mail: ajah.c.ogbonnaya@gmail.com)
E. Eronu
University of Abuja, Abuja, Nigeria
(e-mail: eronu.majiyebo@uniabuja.edu.ng)
F. E. Shaibu
University of Abuja, Abuja, Nigeria
(e-mail: farouq.shaibu@uniabuja.edu.ng)
I. N. Amalu
Nigerian Communications Satellite Limited, Nigeria
(e-mail: iykeamalu@yahoo.co.uk)
B. G. Najashi
Baze University, Abuja, Nigeria
(e-mail: najashi.gafai@bazeuniversity.edu.ng)

****Corresponding Author***

I. Introduction

The working state of a machine has been of uttermost concern to man since the birth of machines. Early engineers rely solely on physical observation and the use of biological senses to determine the fault in a machine. The advent of measuring instruments or sensors provides more data to detect faults but their reliability and sometimes demand recalibration provide more concerns for engineers [1]. The need for Fault Detection and Isolation (FDI) has increased as the system gets more complex. The computation demands the Satellite Attitude Control System makes this task even more challenging [2].

Due to the high pointing accuracy (especially in geostationary and deep space missions) needed to ensure that a spacecraft points to the desired direction at all times during its lifespan, the need to continually monitor its attitude control system has become imperative [3]. To avert performance degradation, the attitude control system of a satellite needs to be continually checked for fault detection in order to provide quick diagnosis and subsequent solutions [4].

Non-statistical methods include fuzzy logic, neural networks, artificial immune systems, and deep learning. This last category can be especially useful for highly complex systems, high noise environments, and benefit from large data sets. Especially deep learning eliminates the need for feature extraction and is highly self-learning and adaptive. Owing to the learning abilities, robustness to noise, and ability to represent complex systems, artificial intelligence may be a good solution for fault detection and isolation. It is identified that k-nearest neighbors (k-NN), Naive Bayes classifiers, Support Vector Machines (SVM), and Artificial Neural Networks (ANN) have been applied most, with deep learning starting to be applied [5].

One area for fault detection and isolation currently being explored in depth is the knowledge-based, data-driven methods [7]. These methods make use of the growing amounts of telemetry data and increasing onboard processing power and can generally be divided into statistical and non-statistical methods [8]–[9]. Statistical methods include principal component analysis, partial least squares, and support vector machines [10].

The problem of fault detection and identification in Satellite Attitude and Orbit Control Systems (AOCS) has received a great deal of attention during the past years [11]. The improvements in the accuracy and reliability of AOCS contribute directly to the success and reliability of satellites in space. Hence, an exceptional level of autonomy is required. Fault detection and identification is an essential component of an autonomous system. This led to research in developing new methods for supervision, fault detection, fault isolation, and fault recovery. The inherent nonlinearity of satellite dynamics however makes the accurate and efficient fault detection of AOCS a challenging problem [11].

The massive amounts of telemetry data transmitted by an in-orbit satellite are the sole observational basis of the satellite's operation. Through the analysis of telemetry data, ground telemetry, track, and command (TT&C) stations can determine the satellite's operational state and detect possible faults in a timely fashion, assisting the normal operation of the in-orbit satellite. Currently, the fixed threshold method is the main method for fault detection in telemetry data in engineering [11], [12]. With a simple mechanism and rapid detection speed, this method can be used to detect anomalies beyond the threshold in a timely manner.

However, due to the influence of complex noise in the actual telemetry data, the fixed threshold method is prone to producing false alarms in the detection. In addition, the method cannot detect anomalies within the threshold.

According to [8], the dynamic equation of a satellite is described by:

$$H\dot{\omega} = -S(\omega)H\omega + \tau \tag{1}$$

Where:
$\tau \in R^3$ – represents the control input vector action on the satellite in x, y, and z axes directions respectively as shown in equation (2).
ω – the angular velocity vector of the satellite, expressed in the satellite body frame given by equation (3),
H – the symmetric positive definite inertia matrix of the satellite and
$S(\omega)$ – the cross-product matrix given by equation (4).

$$\tau = \begin{bmatrix} \tau_x & \tau_y & \tau_z \end{bmatrix}^T \tag{2}$$

$$\omega = \begin{bmatrix} \omega_1 \omega_2 \omega_3 \end{bmatrix}^T \tag{3}$$

$$S(\omega) = \begin{bmatrix} 0 & \omega_3 & -\omega_2 \\ -\omega_3 & 0 & \omega_1 \\ \omega_2 & -\omega_1 & 0 \end{bmatrix} \tag{4}$$

If the kinetic equation based on the choice of attitude representation using Euler angles is:

$$\begin{bmatrix} \dot{\psi} \\ \dot{\theta} \\ \dot{\phi} \end{bmatrix} = \Pi(\psi,\theta,\phi) \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix} \tag{5}$$

$$\Pi(\psi,\theta,\phi) = \begin{bmatrix} 0 & s\phi/c\theta & c\phi/c\theta \\ 0 & c\phi & -s\phi \\ 1 & s\phi\tan\theta & c\phi\tan\theta \end{bmatrix} \tag{6}$$

where:
ψ, Θ, ϕ, s and c – roll, pitch, yaw, sine, and cosine respectively.

An actuator fault can be considered by adding an extra torque component τ_F to the dynamic equation such that:

$$H\dot{\omega} = -S(\omega)H\omega + \tau + \tau_F \tag{7}$$

where:
τ_F – an unknown fault. If we define new variables *x* and *u* where:

$$x = \begin{bmatrix} \Omega \\ \omega \end{bmatrix} \text{ and } u = \tau,\ \Omega = [\psi\ \theta\ \phi]^T$$

then we can further deduce that:

$$\begin{gathered} \dot{x} = f(x,u) + T_F \\ f(x,u) = \begin{bmatrix} \Pi(\psi,\theta,\phi) \\ H^{-1}(\tau - S(\omega)H\omega) \end{bmatrix} \\ T_F = \begin{bmatrix} 0 \\ H^{-1}\tau_F \end{bmatrix} \end{gathered} \tag{8}$$

x can further be rewritten in the form:

$$\dot{x} = Ax + g(x,u) + T_F \tag{9}$$

where:
*g(x,u) = f(x,u) – Ax and m*atrix *A* is a Hurwitz matrix.

Parameterizing the mapping g by feed-forward neural network architecture a recurrent network model based on the above equation is constructed. The following model is considered for the observer design.

$$\dot{\hat{x}} = Ax + g(x,u) + \hat{T}_F(x,w) \tag{10}$$

where:

$$\hat{T}_F(x,w) = \begin{bmatrix} 0 \\ H^{-1}\hat{\tau}_F(x,w) \end{bmatrix} \tag{11}$$

The neural network weight is updated based on a back-propagation (BP) scheme. Backpropagation is one of the gradient descent algorithms used to reduce the performance function by updating of the neural network weights by moving them along the negative of the performance function gradient [13].

II. Procedure

A. Data Acquisition

For this work, two months of real satellite data were downloaded from the Nigcomsat-1R satellite ground control station in Abuja, Nigeria. The outlook of some of the Momentum Wheel hourly telemetry data is shown in Fig.1 through Fig.6.

Both MW1 and MW2 speeds show some repetitive trends every two weeks but their corresponding torques are more random with unusual spikes that may be due to surges or bit errors.

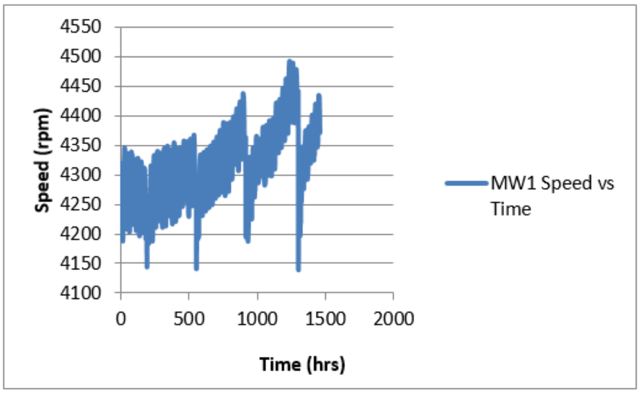


Fig. 1. MW1 Spee Plot.

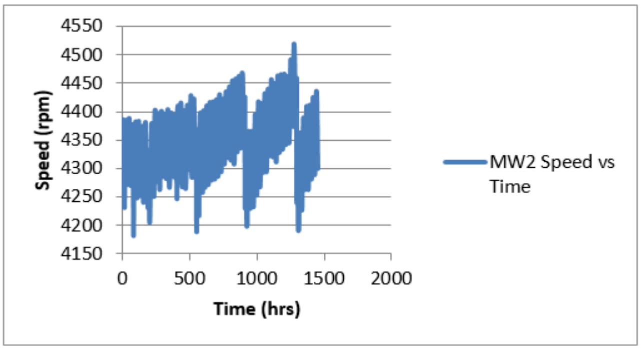


Fig. 2. MW2 Speed vs. time.

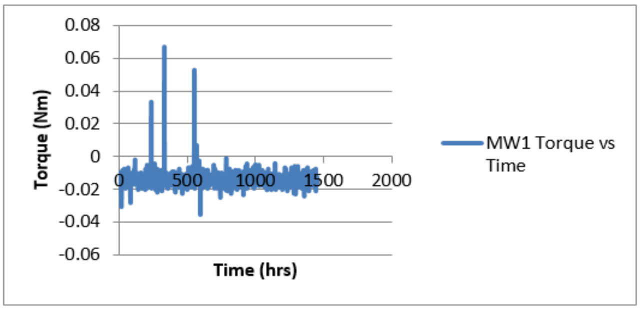


Fig. 3. MW1 Torque plot.

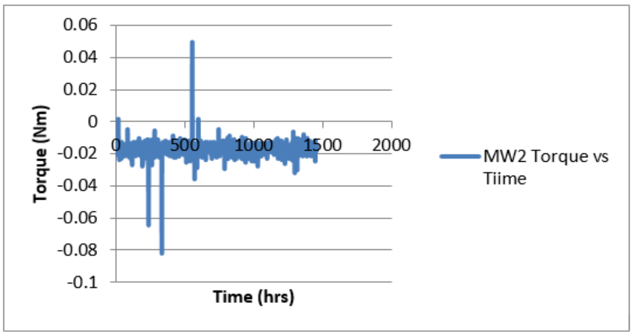


Fig. 4. MW2 Torque plot.

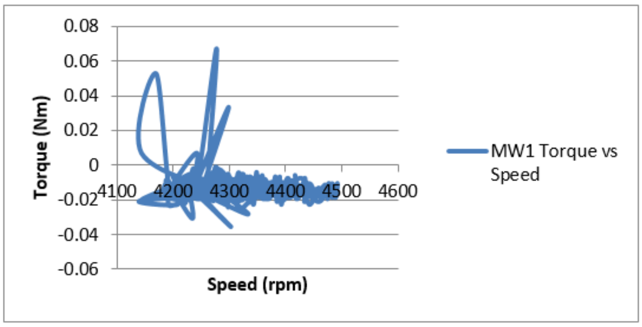


Fig. 5. MW1 Torque-speed distribution.

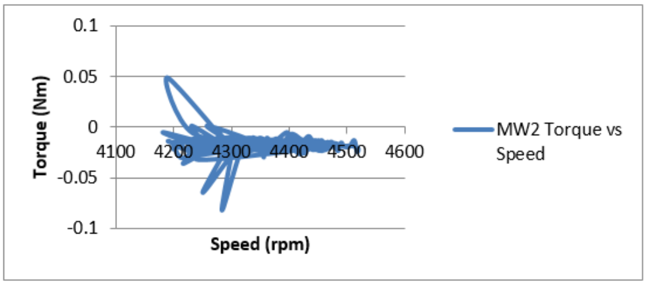


Fig. 6. MW2 Torque-speed distribution.

Similar data available for the wheel currents and voltages will not be shown here but the result of the telemetry data compared with the Neural Network results will be provided later.

B. Data Mining

The raw satellite telemetry data are non-linear with bad correlation. MATLAB analytical and curve fitting tools were used to process the data to a much more usable state while still retaining essentially the main properties of the initial data. Some unwanted data errors sometimes caused by bit errors in digital transmission due to noise, interference, distortion, or bit synchronizations must be removed from the raw data to get the user data. The first level of mined data obtained for MW1 is shown in Fig.7.

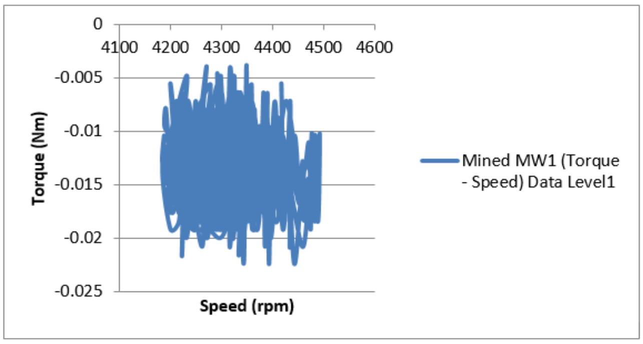


Fig. 7. Mined MW1 Torque-speed data level 1.

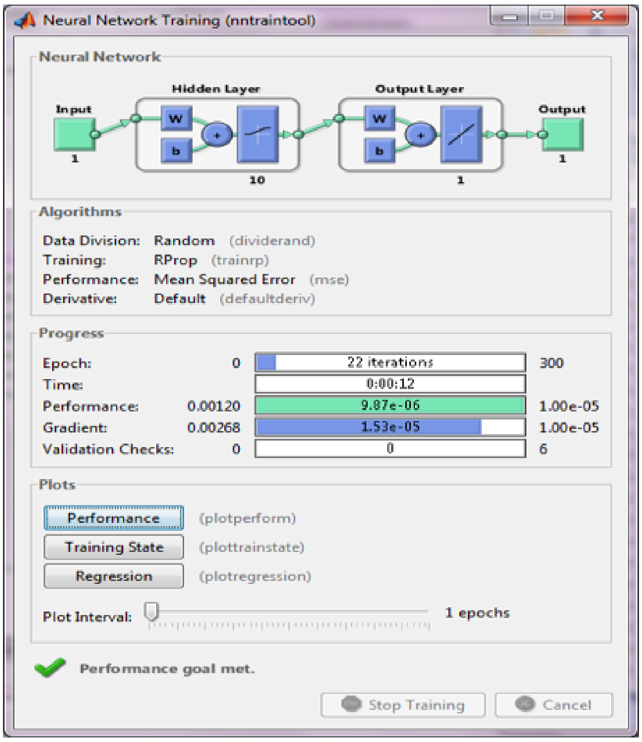
C. Network Modelling and Optimization

The choice of using ANN for this research work is based on its efficiency in network modeling involving the handling of large data samples such as telemetry data as applicable here. Neural Network modeling involving Resilient Propagation (RPROP) and Levenberg-Marquardt (LM) algorithms were compared. Although the Levenberg-Marquardt algorithm has been found to be faster and has better performance than the other algorithm in training, the

Resilient Backpropagation algorithm has the best accuracy in the testing period [14]. The relative performance of these algorithms depends on the given task, but RPROP produces better performance in terms of convergence speed, stability, and generalization properties [14]. The Resilient Propagation algorithm was adopted for this paper. It is worth reiterating here that the nature of the randomness of the telemetry data made the network design very difficult. To achieve any reasonable form of network modeling, some sets of telemetry data obtained were carefully mined before use.

Matlab M-files call the text files containing the MW telemetry data and extract the relevant input and output data for subsequent Neural Network modeling. The MW speed and corresponding MW torques were first used to model a NN that checks the momentum wheel's performance onboard the spacecraft. For this design, a single input, ten elements at the hidden layer, and one element at the output layer were adopted.

After several iterations and optimization, the *logsig* and *purelin* transfer functions were chosen for use in the hidden and output layer respectively. The *trainer* network training function was used to update weight and bias values according to the resilient backpropagation (RPROP) algorithm. A training rate of 0.05, a maximum epoch of 300, and a goal of 1e-5 were set for the network. The Matlab *gensim()* command was used to generate a Simulink model of the M-file Neural Network for ease in dynamic simulation using Matlab Simulink. Fig. 8 shows the NN MLP architecture, algorithm, and simulation progress.



Data Division: Random (dividerand)
Training: RProp (trainrp)
Performance: Mean Squared Error (mse)
Derivative: Default (defaultderiv)

Epoch:	0	22 iterations	300
Time:		0:00:12	
Performance:	0.00120	9.87e-06	1.00e-05
Gradient:	0.00268	1.53e-05	1.00e-05
Validation Checks:	0	0	6

(plotperform)
(plottrainstate)
(plotregression)

Plot Interval: 1 epochs

✓ Performance goal met.

Fig. 8. NN MLP architecture, algorithm, and simulation progress.

Fig. 9 shows the Simulink model of this NN. The input takes in a constant (that is Telemetry data for MW speed and Torque at a given time) and generates the corresponding output (MW Torque).

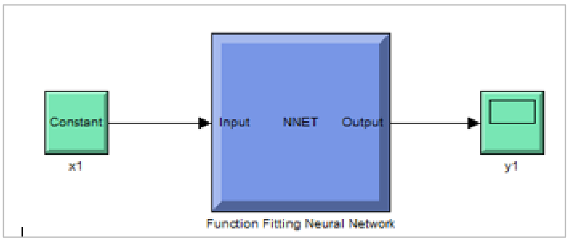
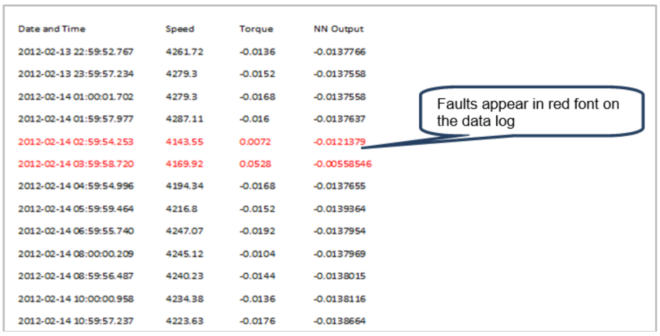


Fig. 9. Simulink model of the Neural Network.

III. Results and Discussion

The Neural Network design model's output and input and output data are saved in a text file to ease computational comparison and analysis. An extract from the text file for MW1 is shown in Fig. 10.

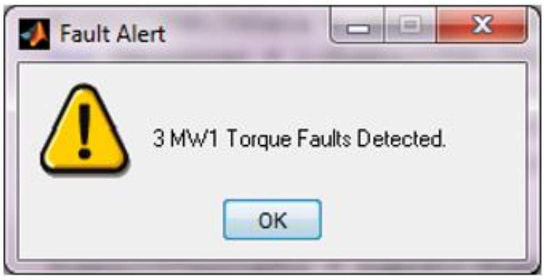


Date and Time	Speed	Torque	NN Output
2012-02-13 22:59:52.767	4261.72	-0.0136	-0.0137766
2012-02-13 23:59:57.234	4279.3	-0.0152	-0.0137558
2012-02-14 01:00:01.702	4279.3	-0.0168	-0.0137558
2012-02-14 01:59:57.977	4287.11	-0.016	-0.0137637
2012-02-14 02:59:54.253	4143.55	0.0072	-0.0121379
2012-02-14 03:59:58.720	4169.92	0.0528	-0.00558546
2012-02-14 04:59:54.996	4194.34	-0.0168	-0.0137655
2012-02-14 05:59:59.464	4216.8	-0.0152	-0.0139364
2012-02-14 06:59:55.740	4247.07	-0.0192	-0.0137954
2012-02-14 08:00:00.209	4245.12	-0.0104	-0.0137969
2012-02-14 08:59:56.487	4240.23	-0.0144	-0.0138015
2012-02-14 10:00:00.958	4234.38	-0.0136	-0.0138116
2012-02-14 10:59:57.237	4223.63	-0.0176	-0.0138664

Faults appear in red font on the data log

Fig. 10. MW1 speed and torque data compared with NN output.

The NN design ensures that the associated MW speed and the corresponding torque are logged in a Fault-Log file in the event of a fault to ease subsequent identification and isolation. A message box incorporated in the design pops up to indicate the number of faults detected as logged in the text file. A fault is detected if the absolute difference between the original output (MW1 Torque) and the NN Torque output is>0.012. This value was chosen after carefully studying the maximum difference between the original output data and the neural network-trained outputs from several results. A preview of the pop-up window and the Fault-Log file is shown in Fig. 11 and Table I respectively.



3 MW1 Torque Faults Detected.

Fig. 11. MW1 Torque Fault Alert.

The best performance of the Neural Network Training, Validation, and Test within the set goals is shown in Fig. 12. The set goal, the number of epochs, and the training rate influence the overall training performance of the network. This, however, depends on the linearity of the data provided and the network algorithm is chosen.

DOI: http://dx.doi.org/10.24018/ejai.2023.2.2.18

TABLE I: MW1 SPEED, TORQUE, AND TIME DATA FAULT-LOG

Date and Time	Speed (rpm)	Torque (Nm)
2012-01-22 20:59:59.820	4234.38	-0.0304
2012-01-25 15:59:56.154	4333.01	-0.028
2012-01-27 00:59:56.172	4236.33	-0.0016
2012-01-30 02:59:59.206	4145.51	-0.02
2012-01-30 14:59:55.503	4183.59	-0.0144
2012-01-31 23:59:55.523	4298.83	0.0336
2012-02-04 23:59:58.574	4277.34	0.0672
2012-02-14 02:59:54.253	4143.55	0.0072
2012-02-14 03:59:58.720	4169.92	0.0528
2012-02-14 21:00:00.977	4242.19	0.0072
2012-02-15 23:59:58.757	4302.73	-0.0352

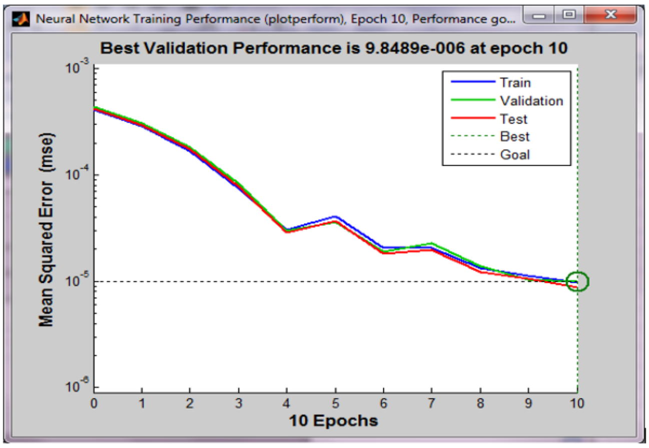

Neural Network Training Performance (plotperform), Epoch 10, Performance go...
Best Validation Performance is 9.8489e-006 at epoch 10
Mean Squared Error (mse)
10 Epochs
Train
Validation
Test
Best
Goal
10^{-3}
10^{-4}
10^{-5}
10^{-6}
0
1
2
3
4
5
6
7
8
9
10

Fig. 12. MW1 Speed-Torque Neural Network Training Performance.

From the Neural Network result log file, we can make a comparison of the given output data and the corresponding output data from the trained network given the same set of initial inputs. The result obtained for mined data (levels 1 and 2) for MW1 Torque given its speed as input is shown in Fig. 13 and 14 respectively. In each of the two figures, a comparison is made between the given output and the NN output over time with respect to the available hourly telemetry data.

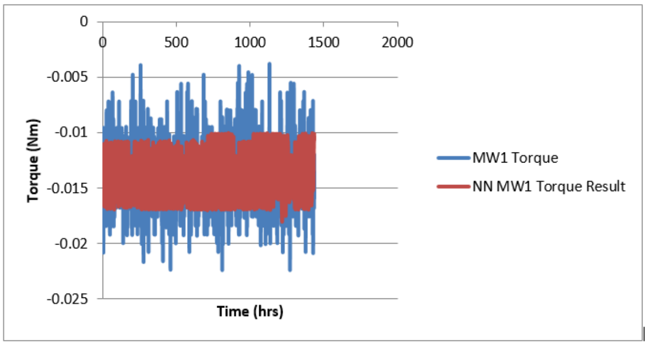


Fig. 13. MW1 Torque Telemetry Level 1 vs. Neural Network Training Result.

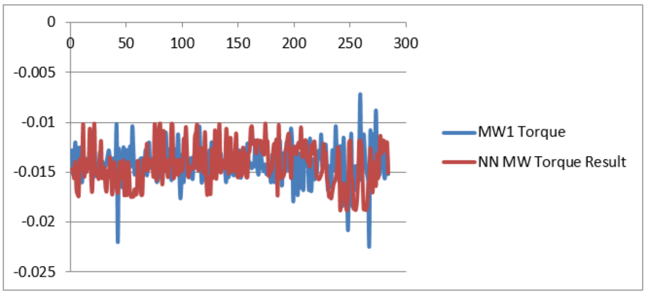


Fig. 14. MW1 Torque Telemetry Level 2 vs. Neural Network Training Result.

As shown in Fig. 15, there is a large disparity (error) between the NN output, and the given output (MW1 torque). We can therefore deduce that there is more correlation between the Neural Network output and the initial output for the level 2 data set. Although the number of useful sample data has been significantly reduced. With further data mining and modification of the training algorithm, we can almost replicate the initial output data sets with minimal error.

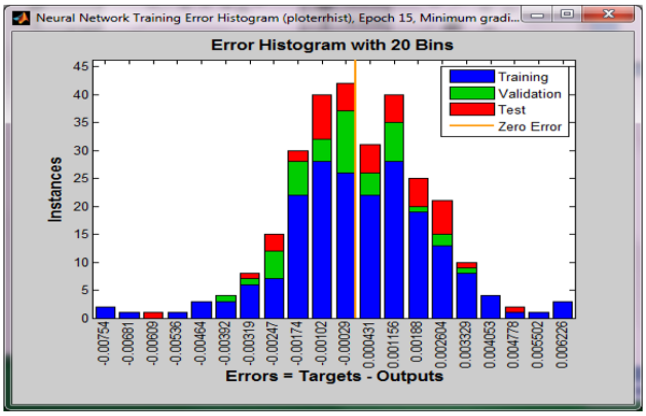


Fig. 15. MW1 Torque Error Histogram Level 2

The Neural Network Regression plot is shown in Fig. 16.

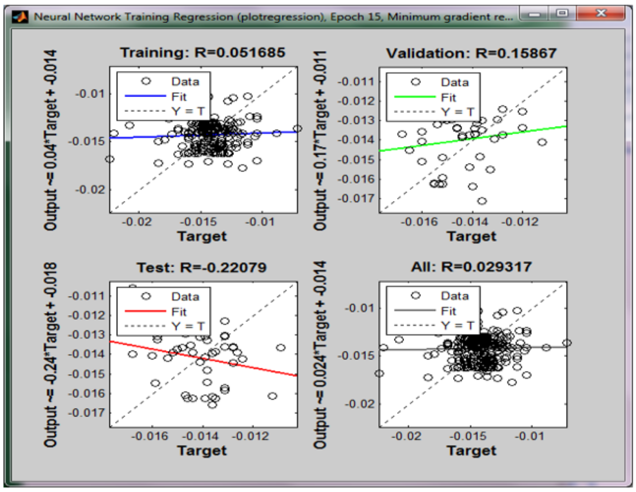

Neural Network Training Regression (plotregression), Epoch 15, Minimum gradient re...
Training: R=0.051685
Output ~= 0.04*Target + -0.014
Validation: R=0.15867
Output ~= 0.17*Target + -0.011
Test: R=-0.22079
Output ~= -0.24*Target + -0.018
All: R=0.029317
Output ~= 0.024*Target + -0.014
Data
Fit
Y = T
Target

Fig. 16. MW1 Torque NN Training Regression Level 2.

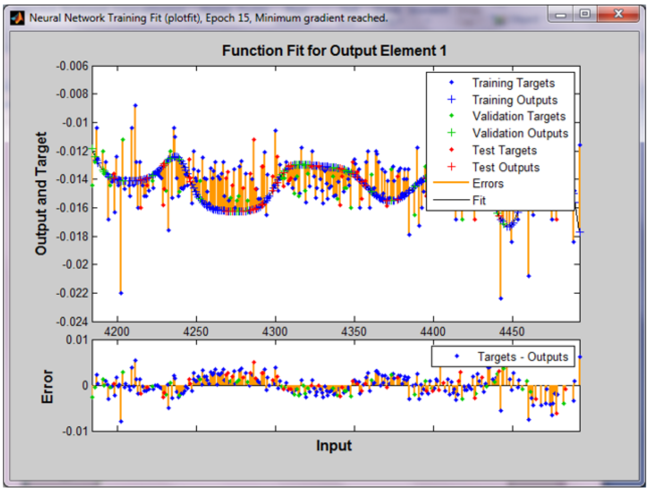


Fig. 17. MW1 Torque NN Training Fit Level 2.

DOI: http://dx.doi.org/10.24018/ejai.2023.2.2.18

From the Error Histogram, NN Training Regression, and the Output Element Function Fit (all are mined level 2 data results), we can easily see that the output error is occasionally high. This is not a result of the bad Neural Network design but rather the non-linearity of the Training Output (MW1 Torque) which has a large disparity as shown earlier in Fig. 3 making it extremely difficult for the training network to minimize errors.

All the results for other MW telemetry data will not be repeated here independently because their Neural Network designs are similar. However, some error histograms, Neural Network Training Performance, and Function Fit for Output Elements of the Neural Network model for MW1 and MW2 Speed-Torque, as shown in Fig. 17, Fig. 18, Fig. 19, Fig. 20, Fig. 21, and Fig. 22. The Voltage-Current Telemetry data (not mined) and error histograms for MW1 and MW2 are presented in Fig. 23, Fig. 24, Fig. 25, Fig. 26, and Fig. 27 accordingly.

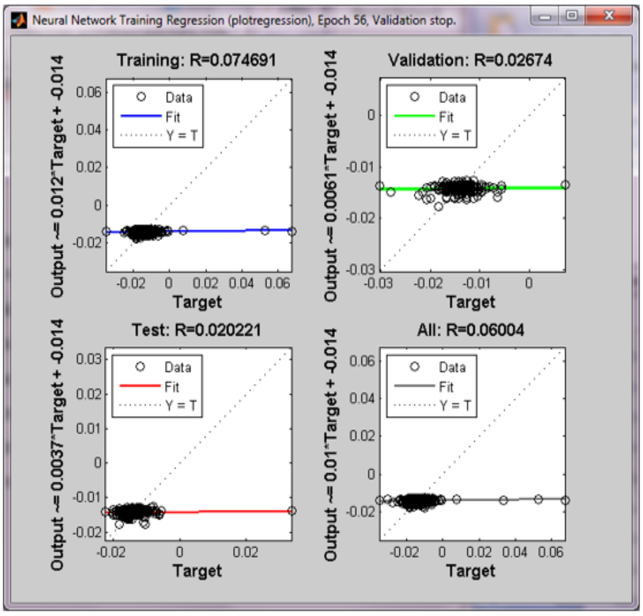


Fig. 18. MW1 Torque Neural Network Training Regression.

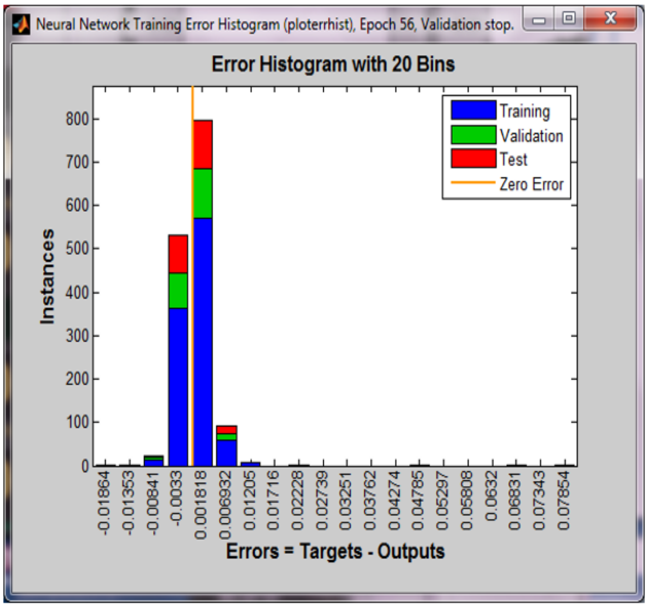


Fig. 19. MW1 Torque Error Histogram.

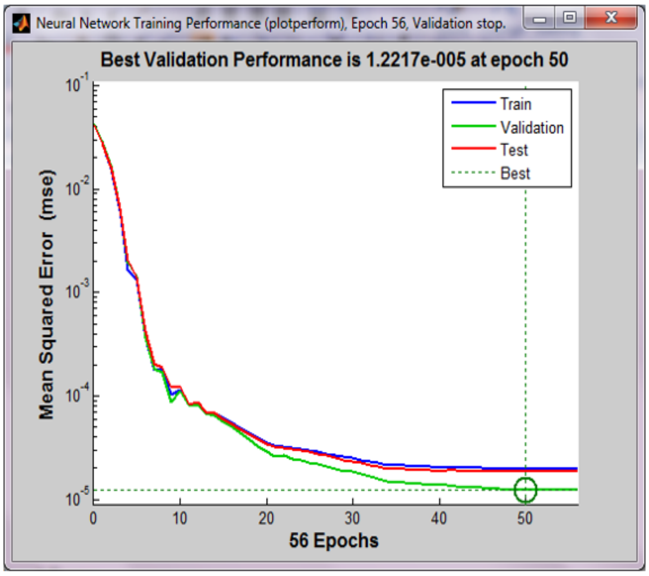


Fig. 20. MW1 Torque NN Training Performance.

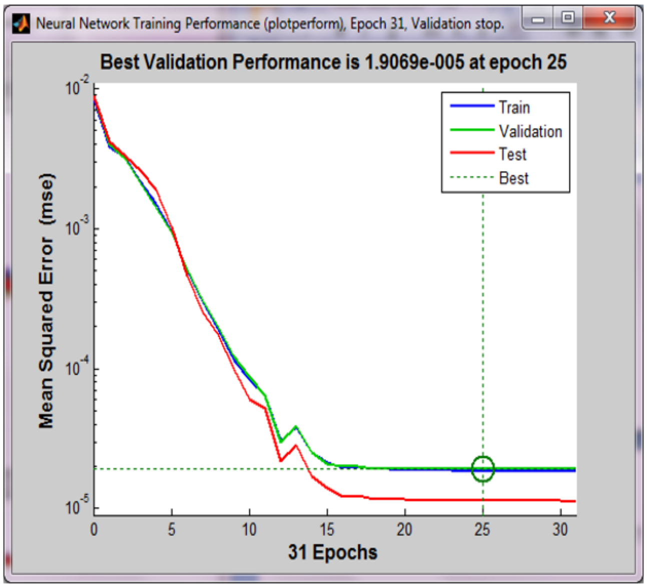


Fig. 21. MW2 Torque NN Training Performance.

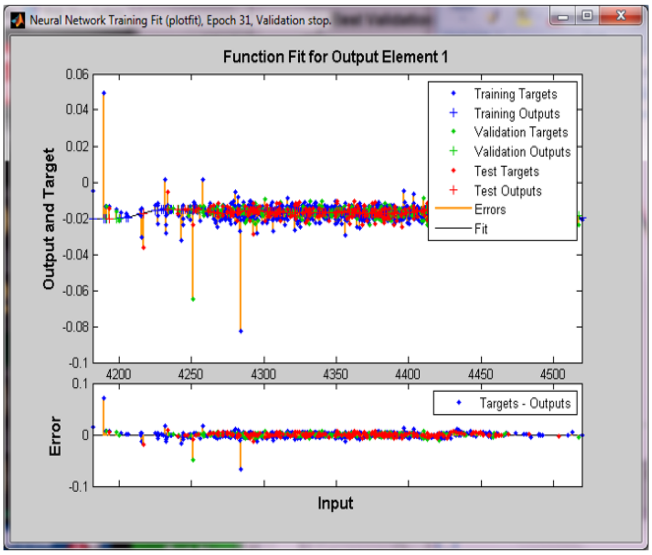


Fig. 22. MW2 Torque NN Training Fit.

DOI: http://dx.doi.org/10.24018/ejai.2023.2.2.18

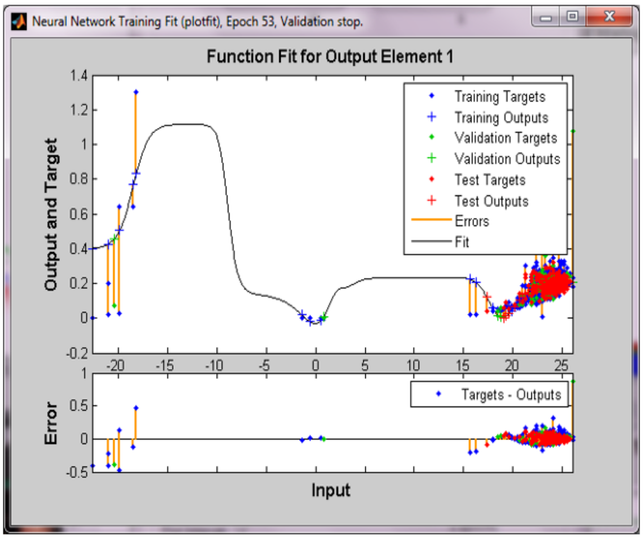


Fig. 23. MW1 Current NN Training Fit.

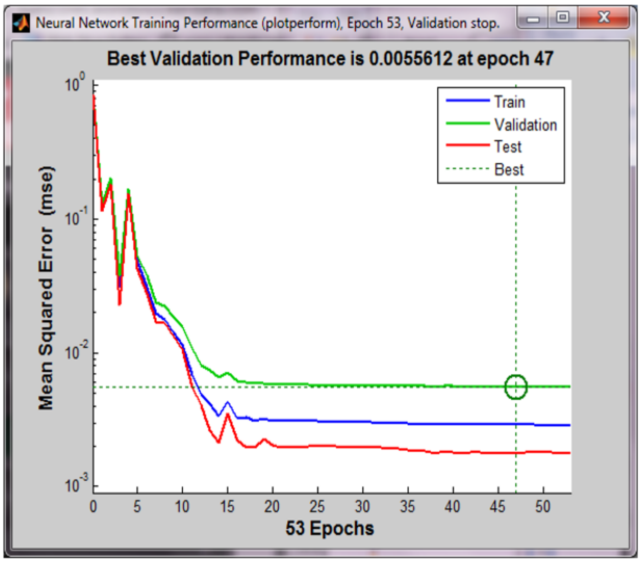


Fig. 24. MW1 Current NN Training Performance.

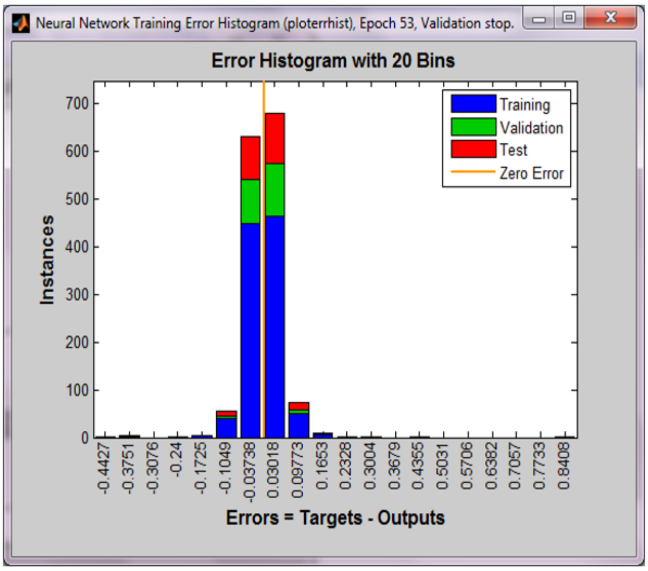


Fig. 25. MW1 Current Error Histogram.

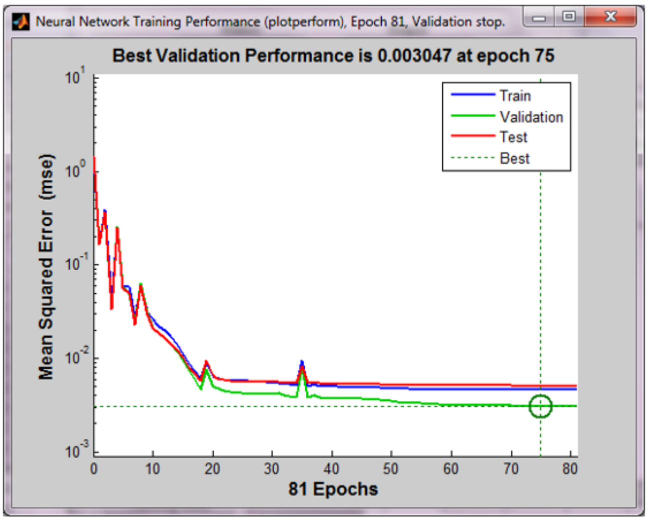


Fig. 26. MW2 Current NN Training Performance.

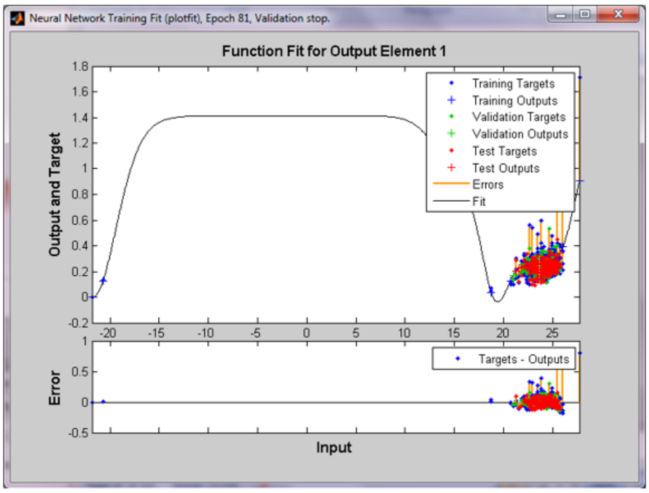


Fig. 27. MW2 Current NN Training Fit.

The results obtained above showed that considering the error margin, the mined data (as shown from the MW1 data) produced a better network design than the raw data.

IV. Conclusion

In this paper, detailed analysis and satisfactory results were obtained using Matlab/Simulink. The Mean Squared Error (MSE) for MW1 Speed-Torque Neural Network Training is 1.2217e-5 at epoch 50, 9.8489e-6 at epoch 10 for MW1 Torque NN Training and 1.9069e-5 at epoch for MW2 Torque NN Training. A fault is detected if the absolute difference between the original output (MW Torque) and the NN Torque output is greater than 0.012. This value was chosen after carefully studying the maximum difference between the original output data and the neural network-trained outputs from several results. The results obtained also showed that considering the error margin, the mined data produced a better network design than the raw data.

Acknowledgment

The authors appreciate the Nigerian Communications Satellite Limited (NigComSat) for providing data for this research.

Funding

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Conflict of Interest

The authors declare no conflict of interest.

References

[1] Z. Gao, C. Cecati, and S. X. Ding. A survey of Fault Diagnosis and Fault-Tolerant Techniques-part 1: Fault Diagnosis with Model-Based and Signal-Based Approaches, “in *IEEE Transactions on Industrial Electronics*. June 2015; 62(6): 3757–3767.

[2] H. A. Talebi and R. V. Patel, A. Neural network-based fault detection scheme for satellite attitude control systems. *Proceedings of the 2005 IEEE Conference on Control Applications, Toronto;* Canada, August 2005.

[3] M, Tipaldi and B. Bruenjes. Survey on fault detection, isolation, and recovery strategies in the space domain. *Journal of Aerospace Information Systems*. 2015; 12(2): 235–256.

[4] Q. Li, X. Zhou, P. Lin, and S. Li. Anomaly detection and fault diagnosis technology of spacecraft based on telemetry-mining. In *Proc. 3rd International Symposium on System & Control in Aeronautics & Astronautics, Harbin*; pp. 233-236; 2010.

[5] R. Liu, B. Yang, E. Zio, and X. Chen. Artificial intelligence for fault diagnosis of rotating machinery: A review. *Mechanical Systems and Signal Processing*. 2018; 108: 33–47.

[6] O. Carvajal-Godinez, J. Guo, and E. Gill. Agent-based algorithm for fault detection and recovery of gyroscope’s drift in small satellite missions. *Acta Astronautica*. 2017; 139: 181–188.

[7] K. R. Chowdhary. *Natural language processing*. *Fundamentals of Artificial Intelligence*. Springer, New Delhi, pp. 603–649, 2020.

[8] Ruonan Liu, Boyuan Yang, Enrico Zio, and Xuefeng Chen. *Artificial intelligence for fault diagnosis of rotating machinery: A review*, “in *Mechanical Systems and Signal Processing*, 2018; 108: 33–47.

[9] F. Li, R. Li, L. Tian, L. Chen and J. Liu. Data-driven time-frequency analysis method based on variational mode decomposition and its application to gear fault diagnosis in variable working conditions. *Mechanical Systems and Signal Processing*. 2019; 116:462–479.

[10] A. Taghizadeh-Alisaraei and A. Mahdavian. Fault detection of injectors in diesel engines using vibration time-frequency analysis. *Applied Acoustics*. 2019; 143: 48–58.

[11] A. Glowacz. Fault diagnosis of single-phase induction motor based on acoustic signals. *Mechanical Systems and Signal Processing*. 2019; 117:65–80.

[12] G. C. Brito, R. D. Machado and A. C. Neto. *Model-based vibration condition monitoring for fault detection and diagnostics in large hydrogenators,* “in *Proceedings of the 10th International Conference on Rotor Dynamics-IFToMM*, 2019; 2: 105–119.

[13] H. A. Talebi and R. V. Patel. A neural network-based fault detection scheme for satellite attitude control systems. *Proceedings of the 2005 IEEE Conference on Control Applications, Toronto*, Canada, August 2005.

[14] Ö. Kisi and E. Uncuoğlu. *Comparison of three back-propagation training algorithms for two case studies*. Indian Journal of Engineering and Materials Sciences, October 2005.