International Journal of Research in Engineering and Science (IJRES)
ISSN (Online): 2320-9364, ISSN (Print): 2320-9356
www.ijres.org Volume 14 Issue 11 ǁ November 2026 ǁ PP. 57-68

Design of a Computer-Based Condition Monitoring System for Gas Turbine

Chukwuemeka Osmund Azodo[1], Barinaadaa Thaddeus Lebele-Alawa[2], Rex Kemkom Chima Amadi[3]

[1,2,3]Department of Mechanical Engineering, Rivers State University, Port Harcourt, Nigeria
**Corresponding Author: Email: azodoemeka@gmail.com, ORCID: 0009-0003-1720-6435*

Abstract
Nigeria's power sector faces persistent electricity shortages despite substantial installed gas turbine capacity, primarily due to reactive maintenance strategies that fail to detect incipient faults early. This study presents the design, development, and techno-economic validation of a Computer-Based Condition Monitoring System (CBCMS) for a GE Frame-5 (MS5001) gas turbine operating at a Nigerian power generation facility. The system integrates industrial data acquisition via Modbus TCP and MQTT protocols, workflow orchestration with Node-RED, lightweight machine learning analytics using ML5.js, and real-time visualization with P5.js. A feedforward artificial neural network regression (ANN) model was trained on 50,698 preprocessed operational records to establish a dynamic "healthy baseline" for performance prediction. A hybrid outlier detection method combining 3-sigma and quartile-based limits was applied during preprocessing. The final multivariate ANN model, incorporating compressor discharge pressure, compressor inlet temperature, inlet guide vane angle, fuel valve position, fuel flow rate, and fuel lower heating value, achieved a coefficient of determination ($R^2 = 0.99$). Techno-economic analysis across three effectiveness scenarios (5%, 10%, and 20% reduction in downtime and breakdown costs) yielded net present values ranging from ₦797.5 million to ₦3.35 billion, with payback periods as short as 3–10 weeks. The work demonstrates a scalable, cost-effective pathway for transitioning from reactive to predictive maintenance, with applicability to gas turbines, compressors, pumps, and other rotating equipment, offering substantial economic returns suitable for resource-constrained energy and industrial sectors in emerging economies.

Keywords: *Gas Turbine, Condition Monitoring, Machine Learning, Artificial Intelligence ,Predictive Maintenance, Techno-Economic Analysis, Artificial Neural Network, Nigeria, Power Generation Plant, Turbomachinery, Rotating Equipment,*

Date of Submission: 28-10-2026 Date of acceptance: 07-11-2026

I. INTRODUCTION

Nigeria's electricity sector continues to underperform relative to its potential, with the national grid distributing only 4,000–5,000 MW against an installed capacity exceeding 13,000 MW [1,2]. This persistent gap stems from infrastructure decay, gas supply constraints, outdated transmission networks, sector illiquidity, and high technical and commercial losses [2]. Gas turbines, which form a critical backbone of grid stability and are central to Nigeria's Energy Transition Plan [1], are particularly vulnerable. Inadequate maintenance regimes and exposure to grid instability frequently precipitate unplanned outages, elevated forced outage rates, and prohibitively high maintenance expenditures that erode plant economics [1,2].

Conventional reactive maintenance practices, which intervene only after fault manifestation, are fundamentally inadequate for detecting incipient degradation in complex turbomachinery [3]. Such approaches result in extended downtime, costly emergency repairs, secondary damage to downstream components, and lost generation revenue [4]. There is therefore an urgent need for a proactive, datadriven condition monitoring framework capable of establishing a dynamic healthy operating baseline, detecting subtle performance deviations in real time, and enabling condition-based maintenance interventions before catastrophic failure occurs [3-5].

Recent literature demonstrates growing interest in data-driven approaches for gas turbine health monitoring. De Castro-Cros et al. [6] developed an autoencoder-based soft sensor for compressor condition assessment that successfully tracked long-term performance drift over five years of operation. De Giorgi et al. [7] reported a diagnostics platform combining artificial neural networks (ANN) and support vector machines (SVM) that achieved 99.4% diagnostic success, with SVM exhibiting superior generalization on unseen data. Liu and Karimi [8] constructed high-fidelity surrogate models using ANN and high-dimensional model representation (HDMR) techniques, attaining average prediction errors below 1% while generating gas turbine correction curves.

Singh et al. [9] applied linear discriminant analysis (LDA) to laboratory-scale turbine fault diagnosis, reporting 98.6% classification accuracy. Seo et al. [10] developed a hybrid SVM-ANN method for defect diagnosis in gas turbine engines under off-design conditions, achieving RMS error rates below 5% for triple defect scenarios. Kiakojoori and Khorasani [11] explored dynamic neural networks (NARX and Elman networks) for gas turbine degradation prediction, demonstrating that Elman networks achieved high accuracy even under severe degradation conditions.

Despite these advances [6-11], several practical limitations persist. Many studies have focused primarily on algorithmic performance in controlled or simulated environments, with insufficient attention to the engineering realities of industrial deployment: seamless integration with existing distributed control systems (DCS) via standard protocols [12], real-time execution on resource-constrained hardware, scalability across multiple units, cybersecurity considerations, and the techno-economic justification required to secure capital investment from plant operators in emerging economies [13]. There remains a clear need for an integrated, end-to-end condition monitoring framework that is simultaneously technically rigorous, lightweight enough for practical deployment, and demonstrably cost-effective under realistic operating conditions.

The present study directly addresses this gap by designing and validating a complete CBCMS architecture that couples robust data acquisition (Modbus TCP/MQTT) [12], workflow automation (Node-RED) [14], lightweight on-browser machine learning (ML5.js) [15], and interactive visualization (P5.js) [16]. The system was developed specifically for the operational constraints of a Nigerian power plant and subjected to rigorous statistical validation and full techno-economic appraisal [13,17], thereby bridging the divide between research prototype and actionable industrial solution.

This study aimed to design, develop, and validate a Computer-Based Condition Monitoring System (CBCMS) employing machine learning techniques to predict gas turbine performance and detect anomalies [4]. The specific objectives were: (i) to analyze and select key performance parameters (KPPs) most indicative of machine health [3]; (ii) to design a robust data acquisition, cleaning, and normalization framework [12]; (iii) to formulate, train, and optimize a machine learning regression model capable of establishing a dynamic healthy baseline [8,15]; (iv) to validate the model and develop an anomaly detection methodology based on residual analysis [3,4]; and (v) to design and implement a functional prototype software application for real-time health visualization and operator alerts [14-16].

II. METHODOLOGY

2.1 Materials

The primary asset for this study was a GE MS5001 (Frame-5) gas turbine at a Nigerian power generation facility, with standard thermodynamic references used for validation. The integrated dataset comprised real-time operational and gas quality parameters (e.g., temperatures, pressures, load, fuel gas composition, and Lower Heating Value), historical performance and maintenance logs, and site ambient condition data (temperature, humidity, pressure). The monitoring system was developed using the Modbus TCP protocol for communication, Node-RED for data orchestration, and the ML5.js and P5.js libraries for machine learning and visualization.

2.2 Data Acquisition

Real-time operational data was acquired from the GE MS5001 gas turbine via Modbus TCP communication protocol interfacing with the plant's Distributed Control System (DCS). The dataset comprised 50,698 time-stamped records of key parameters including temperatures, pressures, fuel flow, and generator load. Ambient condition data (temperature, humidity, pressure) was continuously monitored using on-site meteorological instruments.

2.3 Key Performance Parameter Selection and Normalization

Key performance parameters (KPPs) were identified through a combination of thermodynamic firstprinciples analysis and Pearson correlation screening against generator load as the primary output variable [3,18]. The selected parameters—generator load, fuel stroke reference, fuel gas flows (energy, mass, and volumetric), compressor discharge pressure, and average exhaust temperature— exhibited the strongest relationships with overall machine performance [6,8]. Standard-day corrections were applied to normalize all performance data to reference ambient conditions, thereby isolating machine-specific degradation signals from environmental variability [18].

The Pearson product-moment correlation coefficient was used to quantify the strength and direction of linear relationships between variables:

$$r_{xy} = \frac{\sum_{i=1}^{N}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{N}(x_i - \bar{x})^2 . \sum_{i=1}^{N}(y_i - \bar{y})^2}} \quad (1)$$

2.4 Data Preprocessing and Quality Assurance

Raw operational data invariably contains sensor noise, transmission errors, and occasional outliers that can degrade model performance [19]. A hybrid outlier detection and removal strategy was implemented, combining statistical (3-sigma) limits with quartile-based (interquartile range) boundaries [19]. For each load bin, upper and lower bounds were computed using a hybrid approach that balances statistical rigor with empirical adaptability. The hybrid approach defines upper and lower bounds as:

$$Upper\ Bound\ =\ min\{(m\ \times\ DWATT_Rounded\ +\ b)\ +\ 3\sigma, P_{99}\} \quad (2)$$

$$Lower\ Bound\ =\ max\{(m\ \times\ DWATT_Rounded\ +\ b)\ -\ 3\sigma, P_{1}\} \quad (3)$$

where m represents the slope of the regression line, b represents the intercept, σ is the standard deviation, and P_{99} and P_1 are the 99th and 1st percentiles respectively. Following outlier removal, data were normalized using z-score standardization:

$$X' = \frac{X - \mu}{\sigma} \quad (4)$$

where X' is the standardized value, μ is the mean of the feature, and σ is its standard deviation.

2.5 Machine Learning Model Architecture and Training

A feedforward artificial neural network (ANN) was selected for its proven capability to capture the complex, non-linear mappings between multiple input parameters and gas turbine output [7,8,10]. The network architecture consisted of an input layer, one or more hidden layers with non-linear activation functions, and a single-output regression layer predicting generator load (MW). The key mathematical relationships are as follows:

Forward pass for a neuron:

$$Y_i\ = f\left(\sum_{j=1}^{n} w_{ij}.x_j\ +\ b_i\right) \quad (5)$$

Mean squared error (MSE) loss function:

$$\boldsymbol{J(\mathbf{w}, b) = \frac{1}{m}\sum_{i=1}^{m}(y_i - \hat{y}_i)^2} \quad (\mathbf{6})$$

The iterative training cycle comprised: (i) random initialization of weights and biases; (ii) forward propagation of input batches through the network; (iii) computation of the MSE loss against groundtruth load values; (iv) backward propagation of the loss gradient through all layers; (v) parameter update using the Adam optimizer [20]; and (vi) repetition across multiple epochs until convergence criteria were satisfied [20].

Hyperparameters—including number of epochs, learning rate, number of hidden neurons, and batch size—were systematically tuned [21]. The optimal configuration was selected using a weighted scoring metric that balanced prediction accuracy (R^2 and MSE) against computational training time.

Multiple model configurations were evaluated: univariate and bivariate regressions using individual or paired predictors (compressor discharge pressure, exhaust temperature) with ambient temperature; and a final comprehensive multivariate model incorporating compressor discharge pressure (bar), compressor inlet temperature (°C), IGV angle (°), fuel valve position (%), fuel mass flow (kg/h), and fuel lower heating value (MJ/m^3) [8].

All models were assessed using the coefficient of determination:

$$R^2 = 1 - \frac{\sum(y_i - \hat{y}_i)^2}{\sum(y_i - \bar{y})^2} \quad (7)$$

and mean squared error [20]:

$$\text{MSE}\ =\ \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \quad (8)$$

2.6 Techno-Economic Modelling Framework

A comprehensive techno-economic model was developed to quantify the financial viability of CBCMS deployment [13]. The framework follows a net present value (NPV) approach over a 15-year project lifetime [17]. The key governing equations are:

Real discount rate:

$$i = \frac{i' - i_f}{1 + i_f} \quad (9)$$

where i' is the nominal discount rate and i_f is the inflation rate.

Capital recovery factor (CRF):

$$CRF\left(i, R_{proj}\right) = \frac{i(1+i)^{R_{proj}}}{(1+i)^{R_{proj-1}}} \quad (10)$$

Avoided downtime and breakdown cost savings:

$$\Delta C_downtime = C_downtime \times \eta_downtime \quad (11)$$

$$\Delta C_breakdown = C_breakdown \times \eta_breakdown \quad (12)$$

Net annual savings:

$$\Delta C_{net} = \Delta C_{downtime} + \Delta C_{breakdown} - C_{ann} \quad (13)$$

Net Present Cost:

The Net Present Cost (C_{NPC}) represents the total cost of implementing and operating the CBCMS, discounted to present value. As proposed by Dalton et.al (2008), C_{NPC} is calculated using:

$$C_{NPC} = C_{ann}\, X\, CRF\left(i, R_{proj}\right) \quad (14)$$

Where C_{ann} is the annualized cost of the CBCMS, $CRF\left(i, R_{proj}\right)$ is the Capital Recovery Factor, i is the annual real discount rate, and R_{proj} is the project lifetime.

Net present value:

$$NPV = (\Delta C_{net} / CRF) - C_cap \quad (15)$$

Payback period:

$$\boldsymbol{P}BP = C_cap / \Delta C_{net} \quad (16)$$

2.7 Software Prototype Design and Architecture

A functional prototype software application was designed to operationalize the machine learning model for real-time condition monitoring, addressing Objective (v) of this study. The system architecture follows a modular, event-driven approach comprising four integrated layers:

2.7.1 Data Acquisition Layer: Real-time operational data is acquired from the plant's Distributed Control System (DCS) via Modbus TCP protocol [12]. The system polls critical parameters including temperatures, pressures, fuel flows, and generator load at intervals ranging from one to thirty seconds.

2.7.2 Data Processing and Orchestration Layer: Node-RED [14], a flow-based development tool, serves as the central workflow orchestrator. It handles data cleansing, normalization, and formatting before routing data to the machine learning model for inference. Node-RED also manages alert generation and notification delivery via SMTP (email), SMS, and messaging platforms.

2.7.3 Predictive Analytics Layer: The trained ANN regression model, implemented using ML5.js [15] (a lightweight JavaScript library built on TensorFlow.js), performs real-time inference directly in the browser. This eliminates the need for server-side computation, reducing latency and infrastructure costs. The model predicts expected generator load based on current input parameters, and residuals (differences between actual and predicted values) are continuously monitored for anomaly detection.

2.7.4 Visualization and User Interface Layer: An interactive dashboard was developed using P5.js [16], a JavaScript library for creative coding and data visualization. The dashboard displays real-time operational parameters, predicted values, prognostic metrics (failure probability, remaining useful life), and alert notifications. Two distinct interfaces were implemented:

i. TrainLab (TLab): A graphical user interface for configuring, training, and exporting machine learning models without command-line interaction.

ii. GT Smart Health Monitor (GT-SHM): A real-time monitoring dashboard for plant operators, featuring dynamic performance visualization and multi-channel alerts.

The CBCMS is designed for flexible deployment on workstation CPUs, edge devices (Raspberry Pi/mini-PCs), or cloud platforms, depending on resource availability.

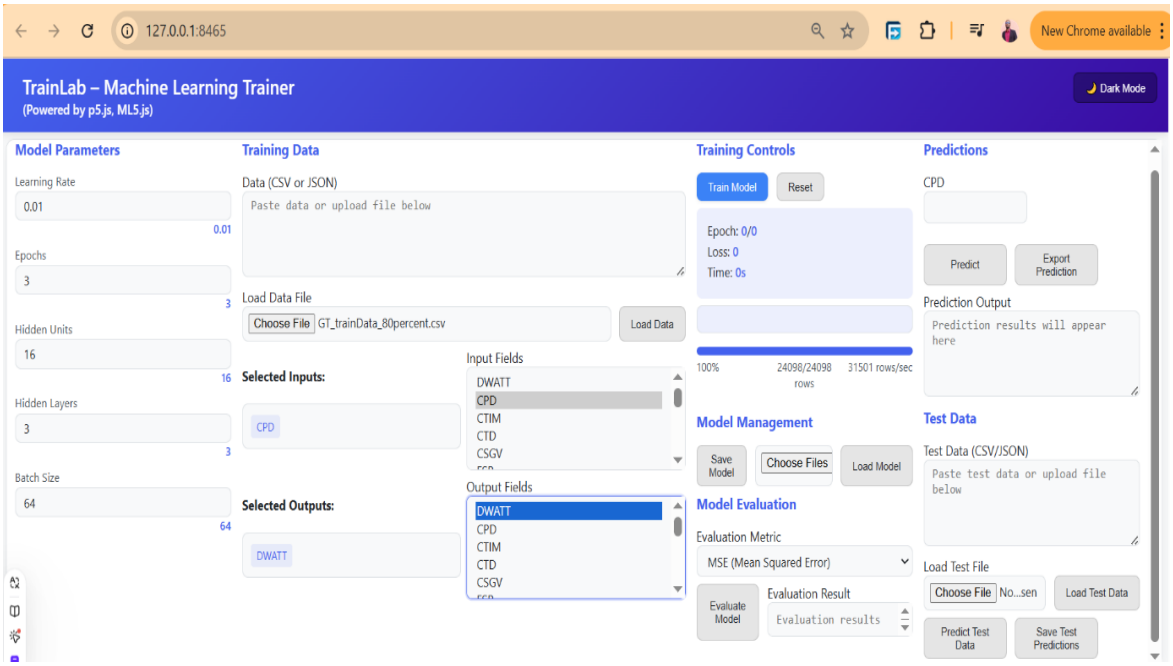


Figure 1: TrainLab GUI for Gas Turbine Machine Learning Training

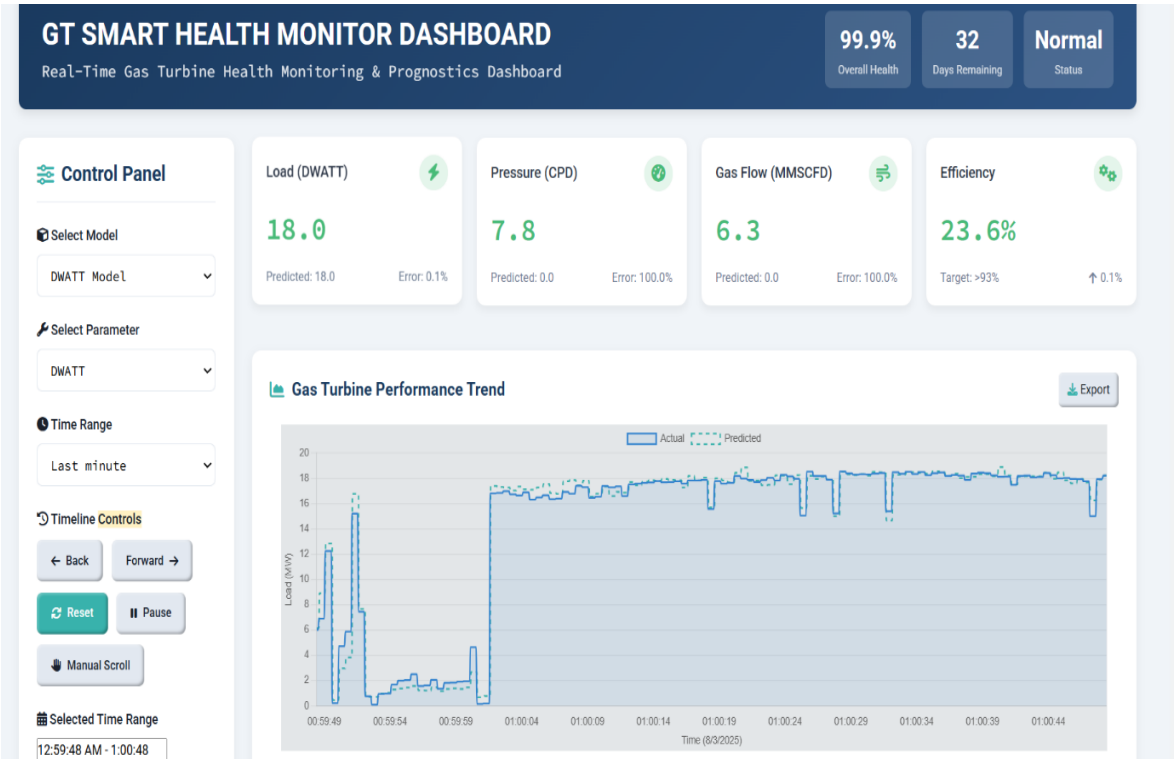


Figure 2: GT-Smart Health Dashboard – Real-Time Monitoring Interface

III. RESULTS AND DISCUSSION

3.1 Ambient Temperature Trends and Normalization Necessity

Analysis of site ambient temperature distributions across three consecutive half-year periods (July–December 2023, January–June 2024, and July–December 2024) revealed a clear progressive warming trend [1]. The histogram mode increased from 27–28 °C to 31–32 °C, with the mean rising approximately 3 °C and the upper tail extending to 37 °C [1]. This temporal shift underscores the critical importance of ambient normalization in any performance monitoring model [18]. Without correction, apparent performance degradation could be misattributed to machine health when in fact it reflects changing environmental boundary conditions [18].

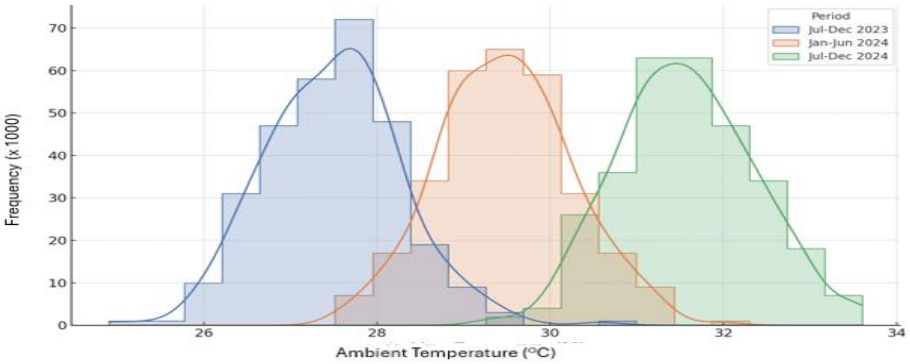


Figure 3: Temperature Distribution Trends for July 2023 to December 2024

3.2 Generator Load Profile Characteristics

The load profile of the studied GE MS5001 unit exhibited strong high-load dominance consistent with baseload power generation duty [6]. The majority of operating hours were concentrated near rated capacity, with relatively infrequent part-load excursions [6]. The hybrid outlier filtering approach was particularly effective in this context, preserving the dense high-load data cloud while cleanly removing sensor anomalies and transient excursions [19].

The raw, unfiltered dataset yielded an R^2 of 0.953. After applying the $\pm 1\sigma$ filter, the cleaned dataset achieved an R^2 of approximately 0.99, representing a 3.9% improvement in explained variance. This confirms that the outlier removal process successfully eliminated anomalous measurements while preserving the genuine thermodynamic relationship between power output and fuel consumption.

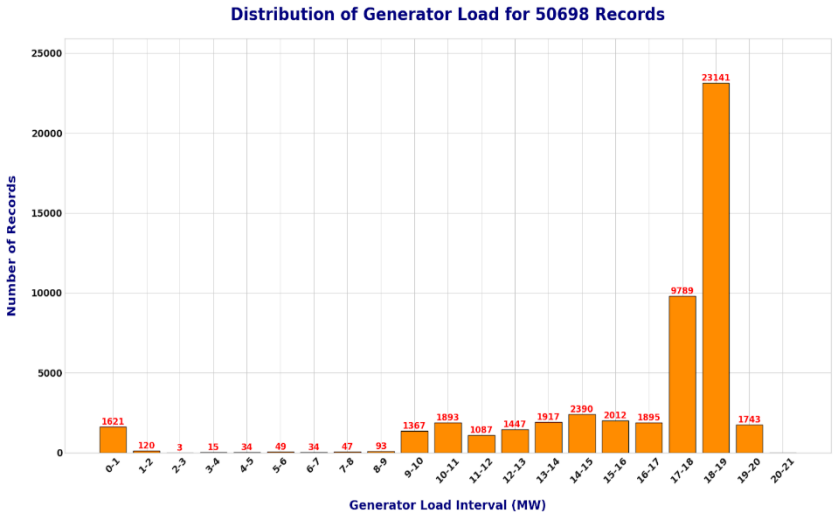


Figure 4: Generator Load Profile: High-Load Dominance

3.3 Machine Learning Model Performance

Progressive model development demonstrated clear performance gains with increasing input dimensionality and physical fidelity [8]. Compressor discharge pressure (CPD)-based models achieved R^2 values of 0.95 (univariate), rising to 0.96 with IGV angle, 0.97 with compressor inlet temperature (CTIM), and 0.99 when all three predictors were combined [6]. Exhaust gas temperature (TTXM)-based models similarly improved from $R^2 = 0.93$ (univariate) to 0.98 when augmented with CTIM and IGV [6].

The final comprehensive multivariate model, employing six physically meaningful inputs (CPD, CTIM, IGV angle, fuel valve position, fuel mass flow rate, and fuel LHV), attained $R^2 = 0.99$ on the held-out validation set [6]. This exceptional accuracy confirms that the selected parameters collectively capture the dominant thermodynamic relationships governing gas turbine output [8], enabling a highly reliable dynamic healthy baseline against which real-time performance can be benchmarked [4,6].

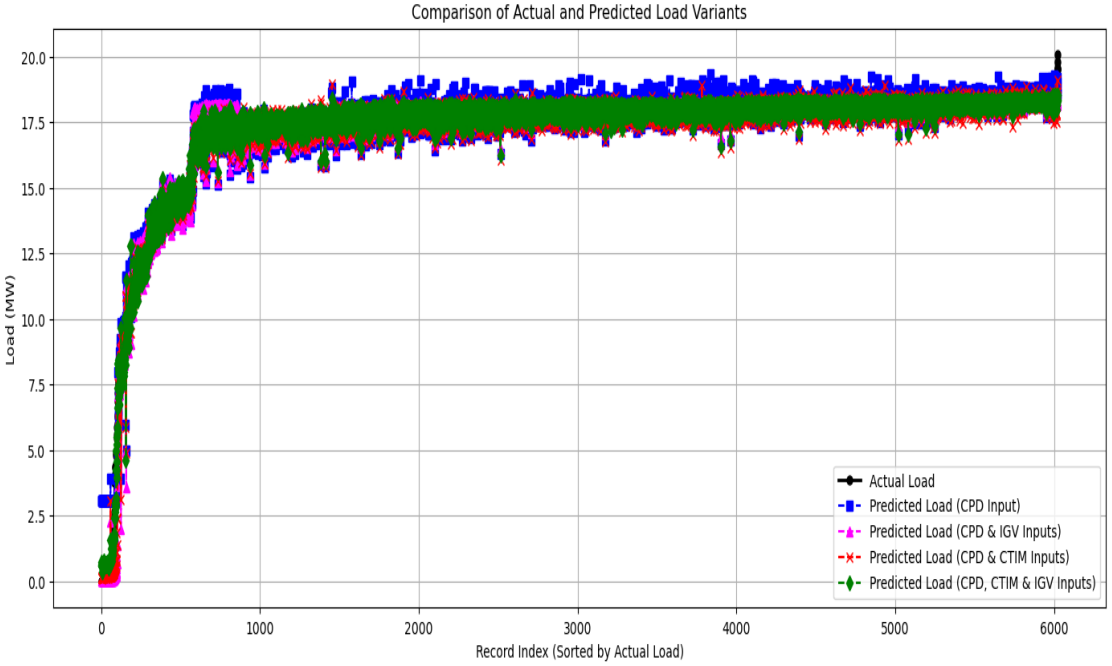


Figure 5: Comparison of Actual and Predicted Load with CPD, CTIM, & IGV Inputs

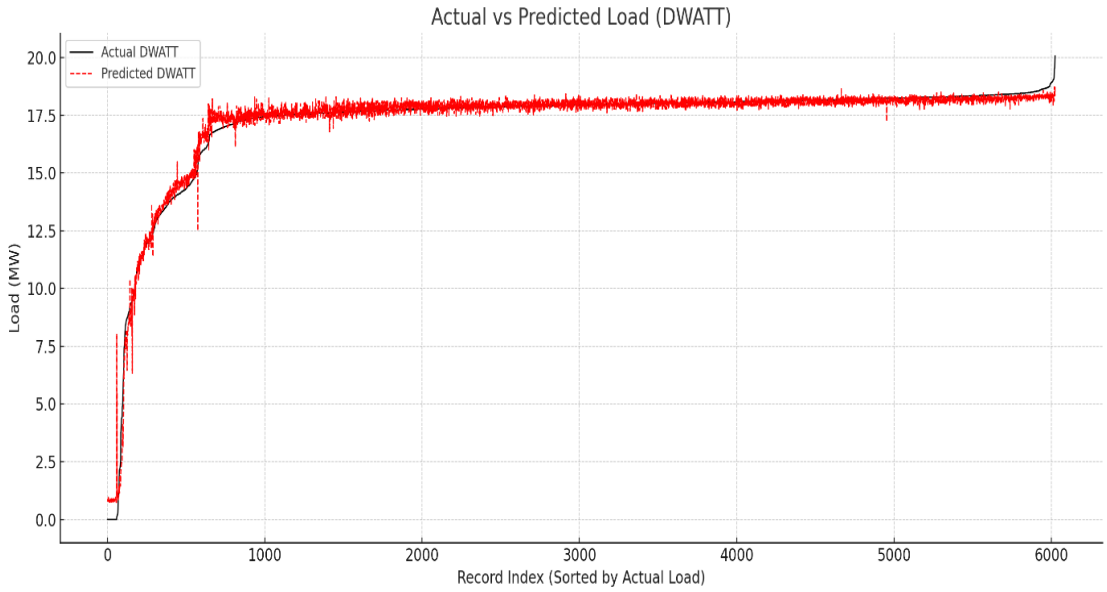


Figure 6: Figure Actual vs Predicted DWATT (Load) Using Final Model Configuration

3.4 Hyperparameter Tuning and Selection

A weighted scoring system that penalized both prediction loss and training time was used to evaluate various combinations of epochs, learning rates, hidden units, layers, and batch sizes. The optimal configuration was achieved with 2 epochs, a learning rate of 0.01, 3 hidden layers, 16 neurons per layer, and a batch size of 64. This configuration yielded a minimum weighted score of 4.90, balancing a low prediction loss ($R^2 = 0.99$) with a rapid training time (7 seconds).

Table 1: Comparative Evaluation of Neural Network Hyperparameters

Hyper Parameter	Epoch	Learning Rate	Number of Hidden Units per Layer	Number of Hidden Layers	Batch Size	Loss (x10-4)	Training Time (secs)	Weighted Average of Loss & Training Time (With Penalty of 8:2)
Epoch	1	0.01	16	6	32	42.0	14	38.10
	(2)	0.01	16	6	32	4.5	17	**(8.59)**
	3	0.01	16	6	32	4.4	23	11.01
	10	0.01	16	6	32	4.3	67	30.21
Learning Rate	2	0.0001	16	3	32	221.6	14	198.26
	2	0.001	16	3	32	22.3	14	20.91
	2	**(0.01)**	16	3	32	4.7	14	**(7.52)**
	2	0.1	16	3	32	5.8	13	7.79
Hidden Units	2	0.01	4	3	32	4.4	14	7.39
	2	0.01	8	3	32	4.4	15	7.78
	2	0.01	**(16)**	3	32	4.4	14	**(7.38)**
	2	0.01	32	3	32	4.5	15	7.82
Batch Size	2	0.01	16	3	16	4.5	35	16.16
	2	0.01	16	3	32	4.4	16	8.17
	2	0.01	16	3	**(64)**	4.3	7	**(4.95)**
	2	0.01	16	3	128	5.5	4	5.27
	2	0.01	16	3	256	49.9	4	44.70
	2	0.01	16	3	512	134.5	1	120.34
Hidden Layer	2	0.01	16	1	64	101.0	8	90.41
	2	0.01	16	2	64	18.2	7	16.56
	2	0.01	16	**(3)**	64	4.2	7	**(4.90)**
	2	0.01	16	4	64	4.3	7	4.99
Optimal Parameter	**2**	**0.01**	**16**	**3**	**64**			

3.5 Techno-Economic Analysis and Financial Viability

The techno-economic appraisal, performed using economic parameters representative of baseload gas turbine operation in the Nigerian power sector [1,13,17], delivered exceptionally strong results across all evaluated scenarios. With an initial capital outlay of ₦14 million and annual operating costs of ₦3.5 million, the system generates substantial net annual savings even under the most conservative 5% effectiveness assumption (reduction in downtime and breakdown costs) [13,17].

At 5% effectiveness, net annual cash flow reaches ₦73.0 million, yielding an NPV of ₦797.5 million and a payback period of approximately 10 weeks [13,17]. At the baseline 10% effectiveness, NPV rises to ₦1.65 billion with payback in 5 weeks [13,17]. Under the optimistic 20% scenario, NPV reaches ₦3.35 billion with payback in just 3 weeks [13,17]. These results illustrate the substantial financial returns achievable through CBCMS deployment under realistic Nigerian operating conditions [1,13,17].

Table 2: Financial Scenario Comparison for CBCMS Implementation Effectiveness

Financial Metric	5% Scenario	10% Scenario	20% Scenario
Net Annual Cash Flow (₦)	72,990,568	149,481,136	302,462,272
Simple Payback Period (Weeks)	10	5	3
Net Present Value (₦)	797,537,414	1,647,989,184	3,348,892,724

IV. CONCLUSION

This study successfully achieved its aim of designing, developing, and validating a Computer-Based Condition Monitoring System for gas turbine applications. All stated objectives were met: critical KPPs were identified through thermodynamic and correlation analysis; a robust data acquisition and hybrid preprocessing pipeline was implemented; a high-accuracy ANN regression model ($R^2 = 0.99$) was trained and optimized to serve as a dynamic healthy baseline; a residual-based anomaly detection methodology was

established; and a functional prototype integrating real-time data streams, machine learning inference, and operator visualization was delivered using an accessible, open-source technology stack.

The techno-economic analysis provides compelling evidence that investment in predictive monitoring capability is not merely technically sound but financially transformative. Even under conservative assumptions of only 5% effectiveness in reducing downtime and breakdown costs, the project delivers an NPV approaching ₦800 million with payback in under three months. Higher effectiveness scenarios, which are realistically achievable given the model's demonstrated predictive fidelity, yield NPVs exceeding ₦3 billion. These returns directly support Nigeria's Energy Transition Plan by improving the reliability and availability of existing gas turbine assets without requiring massive new capital expenditure on generation capacity.

By demonstrating that lightweight, protocol-native, browser-executable artificial intelligence can deliver world-class predictive performance at a fraction of the cost of conventional proprietary systems, this work offers a replicable template for turbomachinery including gas turbines, compressors, pumps, and other rotating equipment, and for energy operators across resource-constrained environments in emerging economies.

NOMENCLATURE

Symbol	Description
ADAM	Adaptive Moment Estimation (optimizer)
AI	Artificial Intelligence
ANN	Artificial Neural Network
b	Intercept of regression line (outlier detection) [MW]
C_ann	Annual operating cost of the CBCMS [₦ yr^{-1}]
C_cap	Initial capital outlay [₦]
C_NPC	Net Present Cost [₦]
CBCM	Computer-Based Condition Monitoring
CBCMS	Computer-Based Condition Monitoring System
CBM	Condition-Based Maintenance
CMMS	Computerized Maintenance Management System
CPD	Compressor Discharge Pressure [bar]
CRF	Capital Recovery Factor [--]
CTIM	Compressor Inlet Temperature [°C]
DCS	Distributed Control System
DWATT_Rounded	Rounded generator load used in outlier bounds [MW]
IGV	Inlet Guide Vane angle [°]
i	Real discount rate [yr^{-1}]
i'	Nominal discount rate [yr^{-1}]
i_f	Inflation rate [yr^{-1}]
J(w,b)	Loss function (mean squared error) [$(MW)^2$]
KPP	Key Performance Parameter
LHV	Lower Heating Value of fuel [MJ m^{-3}]
LSTM	Long Short-Term Memory
m	Slope of regression line (outlier detection) or number of samples
ML	Machine Learning
MSE	Mean Squared Error
MQTT	Message Queuing Telemetry Transport

N	Number of data samples [--]
NPV	Net Present Value [₦]
P_1	1st percentile value [unit of variable]
P_{99}	99th percentile value [unit of variable]
PBP	Payback Period [weeks]
R_proj	Project lifetime [years]
ΔC_net	Annual net savings / benefit [₦ yr^{-1}]
R^2	Coefficient of Determination [--]
r_xy	Pearson product-moment correlation coefficient [--]
SCADA	Supervisory Control and Data Acquisition
SVM	Support Vector Machine
TTXM	Mean Exhaust Gas Temperature [°C]
X	Original (raw) feature value [various]
X'	Z-score standardized feature value [--]
Y_i	Output of the i-th neuron
y_i	Actual generator load (target) [MW]
ŷ_i	Predicted generator load [MW]
ȳ	Mean of actual generator loads [MW]

Greek Symbols

Symbol	Description
μ	Mean of the feature [same unit as feature]
σ	Standard deviation of the feature [same unit as feature]
η_downtime	Effectiveness factor for downtime reduction [--]
η_breakdown	Effectiveness factor for breakdown cost reduction [--]

Subscripts

i	Data sample / observation / neuron index
j	Input feature or previous-layer neuron index
proj	Project lifetime qualifier (e.g. R_proj)
ann	Annualized quantity (e.g. C_ann)
cap	Capital / initial investment (e.g. C_cap)
NPC	Net Present Cost (e.g. C_NPC)
downtime	Downtime-related savings or effectiveness
breakdown	Breakdown-related savings or effectiveness
1	Annual savings (R_1) or 1st percentile (P_1)
99	99th percentile (P_{99})
Rounded	Rounded / binned generator load (DWATT_Rounded)
f	Inflation rate (i_f)
' (prime)	Nominal rate (i')

References

[1] Federal Government of Nigeria. (2022) Nigeria's Energy Transition Plan. Federal Ministry of Environment, Abuja.
[2] National Electricity Regulatory Commission (NERC). (2007) Power sector reform in Nigeria: A roadmap. NERC Publications.
[3] Nakra, B. C. (2006) Condition monitoring and predictive maintenance. Tata McGraw-Hill.
[4] Clifton, D. A. (2006) Novelty detection for engine condition monitoring. Ph.D. Thesis, University of Oxford.
[5] Jardine, A. K. S., Lin, D., & Banjevic, D. (2006) A review on machinery diagnostics and prognostics implementing condition-based maintenance. Mechanical Systems and Signal Processing, 20(7), 1483-1511.
[6] De Castro-Cros, M., et al. (2021) Condition assessment of industrial gas turbine compressor using a drift soft sensor based in autoencoder. Sensors, 21(21), 7125.
[7] De Giorgi, M. G., et al. (2018) A diagnostics platform for gas turbine engine health monitoring and performance analysis. Journal of Engineering for Gas Turbines and Power, 140(9), 091201.
[8] Liu, Z., & Karimi, I. A. (2020) Gas turbine performance prediction via machine learning. Energy, 192, 116627.
[9] Singh, R., Maity, A., Somani, B., & Nataraj, P. (2022) On-board fault diagnosis of a laboratory mini SR-30 gas turbine engine. IFAC-PapersOnLine, 55(22), 153-158.
[10] Seo, D.-H., Roh, T.-S., & Choi, D.-W. (2009) Defect diagnostics of gas turbine engine using hybrid SVMANN with module system in off-design condition. Journal of Mechanical Science and Technology, 23(5), 677685.
[11] Kiakojoori, S., & Khorasani, K. (2016) Dynamic neural networks for gas turbine engine degradation prediction, health monitoring, and prognosis. Neural Computing & Applications, 27(8), 2157-2192.
[12] Zhang, X., & Yu, L. (2017) Modbus TCP-based industrial automation network and its application in control systems. IEEE Transactions on Industrial Informatics, 13(2), 875-883.
[13] Smith, D., Brown, L., & Kim, H. (2016) Financial assessment models for energy infrastructure. Energy Policy, 98, 346-359.
[14] Node-RED. (2022) Node-RED: Low-code programming for event-driven applications. https://nodered.org/
[15] ml5.js. (2022) Friendly machine learning for the web. https://ml5js.org/
[16] McCarthy, L. (2015) Getting started with p5.js: Making interactive graphics in JavaScript and processing. Maker Media, Inc.
[17] Li, J., Yang, X., & Zhang, Y. (2013) Integrated energy systems: Optimization and performance evaluation. Springer.
[18] Volponi, A. J. (2020) Gas turbine parameter corrections. In Gas Turbine Condition Monitoring and Prognostics (pp. 45-78). American Institute of Aeronautics and Astronautics.
[19] Montgomery, D. C. (2019) Design and analysis of experiments (9th ed.). John Wiley & Sons.
[20] James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013) An introduction to statistical learning: With applications in R. Springer.
[21] Bergstra, J., & Bengio, Y. (2012) Random search for hyper-parameter optimization. Journal of Machine Learning Research, 13(1), 281-305.