International Journal of Advances in Engineering and Management (IJAEM)
Volume 7, Issue 03 Mar. 2025, pp: 371-383 www.ijaem.net ISSN: 2395-5252

Development and Evaluation of Artificial Neural Network Techniques for Predictive Maintenance of a Reciprocating Compressor

A. M. Callistus[1] and E. G. Saturday[2]
Mechanical Engineering Department, University of Port-Harcourt, Nigeria

Date of Submission: 10-03-2025 Date of Acceptance: 20-03-2025

ABSTRACT
This work focused on the development and evaluation of ANN-based techniques for the predictive maintenance of a reciprocating compressor. Predictive maintenance, PdM entails minimizing unplanned downtime while extending the life of the equipment through smart monitoring and analysis to predict impending failure. The predictive power of ANN has been exploited in this study to shift from reactive approaches towards maintenance to proactive ones, with improved operational efficiency cost-effectively. Operations data over three consecutive months were retrieved and utilized to train the model. The model iterated through the steps of forward propagation, computation of the loss, and back-propagation to optimize the weights and biases using the training data. From this, it was observed that the ANN model was very effective in identifying the failure conditions at which compressors will fail, with an average of 98% on all predicted cases. The model's robustness is underlined by the high value of precision, recall, and F1-scores it returns; hence, it can be relied upon for failure prediction. Among those that were advisable in boosting the model were regularization, cross-validation, optimization of hyper-parameter, augmentation of data, and simplification of the model. The new direction that this study gives to failure prediction in compressors, especially the reciprocating ones, has performed impressively by showing how much machine learning and neural networks can contribute to industrial predictive maintenance.

Key words: Reciprocating compressor, Predictive Maintenance, Artificial Neural Network,Machine Learning.

I. INTRODUCTION

The downtime of equipment presents a major issue for the manufacturing industry. When machinery breaks down, it remains non-operational until repairs are completed, leading to a decline in productivity. Reducing downtime is essential for improving overall efficiency. By analyzing historical data to detect patterns of wear and tear, machine learning techniques can predict potential failures and facilitate proactive maintenance scheduling. This method helps lower maintenance costs, maintain productivity levels, and optimize the use of resources [1].

The competitiveness of a company is more crucial than ever in the current economic panorama, being immensely influential to the company's ability to provide quality products at low prices. Machine maintenance, with its direct impact in machine downtime and production costs, is directly related to a manufacturing companies' ability to be competitive in terms of cost, quality and performance [2]. Modern maintenance approaches intend to lower failure rates and improve production times but are not widely applied yet. These modern techniques reflect a transition from corrective maintenance practices to more proactive ones: proactive maintenance has the advantage of fixing problems before they come into place, replacing parts after a certain level of deterioration has been identified, as opposed to fixing the fault after the fact [3] /

Proactive maintenance includes preventive maintenance and predictive maintenance. Preventive maintenance consists in performing periodic inspections and other operations according to a predetermined schedule, usually based on time in service. However, this type of maintenance is imperfect, unreliable and costly [4-5]. To achieve a

International Journal of Advances in Engineering and Management (IJAEM)
Volume 7, Issue 03 Mar. 2025, pp: 371-383 www.ijaem.net ISSN: 2395-5252

fully proactive approach, preventive maintenance must be complemented with predictive maintenance. Moreover, companies would benefit from using predictive maintenance throughout the equipment's life cycle to detect the onset of degradation and equipment failure. Predictive maintenance indicates the correct time to perform maintenance; as a result, machines spend less time offline and components are changed only and when needed. Predictive maintenance performs both prediction and diagnosis of an equipment's condition, providing information about the nature of the problem, where it is occurring and why, and when an equipment failure is likely to happen Predictive maintenance techniques can be implemented through the monitorization of equipment combined with intelligent decision methods. Machine Learning and Data Mining techniques can be used to draw insights from the data and accurately predict outcomes to support decision-making and help organizations improve their operations and competitiveness [6-8]. Machine Learning approaches commonly used for fault detection and diagnosis include Artificial Neural Networks Support Vector Machines, and Decision Trees among others [9-13]

Predictive maintenance (PdM) has been gaining prominence in multidisciplinary research groups, proposing the creation and integration of lines of research related to data acquisition, infrastructure, storage, distribution, security, and intelligence. The impact of maintenance represents a total of 15 to 60% of the total costs of operating of all manufacturing/ However, the companies do not measure correctly the amount spent related. Thereby, we can justify the need for studies on how to use new technologies that can change this scenario. In this review, the research focuses precisely on PdM growth and shows that it will be a differential in the implementation of Industry 4.0. Data collected from the multiple sensors in Industry 4.0 environments provide new opportunities for solutions of remaining life prediction of an asset. The idea that PdM can generate scheduling action based on equipment performance or conditions through time becomes exciting and even primordial for the future of the Industry [14]. One of the main requirements for effective PdM achievement is enough amount of data from all parts of the manufacturing process. As a result, it can diminish maintenance costs and downtime, and improve productivity and quality as well. The challenge of predicting the Remaining Useful Life (RUL) of an asset is common in engineering, mechanics, and automation applications. This concept is a part of Prognostics and Systems Health Management (PHM), a complete industry management cycle. In PHM, there are three main axes: observation, analysis, and action. In this context, research related to the PdM is directly linked to the observation axis of PHM and using intelligent methods to predict the failure [15].

Artificial Neural Networks, or ANNs were utilized in the current study. Artificial Neural Networks refer to the computing systems that are modelled similarly to the human brain. ANNs consist of units known as neurons which interact with each other and are organized into layers [16].Artificial neural networks (ANN), has rapidly surged in acceptance as successful computing architectures and models for tasks such as classification, clustering, pattern recognition and even forecasting in various sectors in the recent past.It is necessary to build a predictive maintenance model that is easier, quicker, and cheaper as it helps to prevent breakdowns, improve safety and increase the operational life of the compressor. The failure of a reciprocating compressor was modelled using an Artificial Neural Network.

II. MATERIALS AND METHODS

2.1 System description

This section outlines the methodology used to train a model for the prediction a reciprocating compressor. Three months operational data was obtained from a reciprocating compressor at shell Nigeria. The data obtained was trained on Python software. This chapter also provides a detailed description of the approach, methods, and techniques employed to achieve the objectives of this study. The methodology involves several key steps to build, evaluate, and utilize a neural network model for predicting compressor failures. The breakdown are as shown in the flow chart in Figure 1:

International Journal of Advances in Engineering and Management (IJAEM)
Volume 7, Issue 03 Mar. 2025, pp: 371-383 www.ijaem.net ISSN: 2395-5252

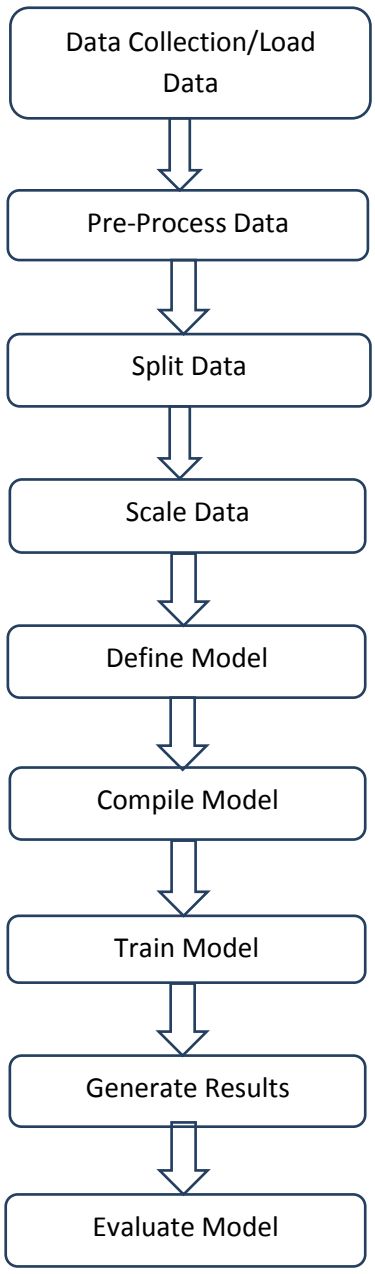

Figure 1: System Flow Chart

This flowchart represents a standard workflow for developing and implementing an ANN for predictive maintenance, focusing on the systematic processing of data and iterative model improvement. Each step is crucial to ensure that the model is both accurate and reliable for predicting compressor failures. The breakdown of each component is as follows:

2.2Data collection/Load Data

The first step involves importing the dataset, which contains historical data related to the compressor. This data include various operational parameters like coolant temperatures, oil pressure, engine speed, water jacket levels, lube tank level, engine oil level, to provide the model with the necessary inputs for training and evaluation. Threemonths operational data were collected from

International Journal of Advances in Engineering and Management (IJAEM)
Volume 7, Issue 03 Mar. 2025, pp: 371-383 www.ijaem.net ISSN: 2395-5252

sensors in the compressor. Historical records of compressor failures, including the date and nature of the failure were taken.

2.3Pre-Process Data

This step involves cleaning and preparing the data. It also includes handling missing values, removing outliers (noise), and converting categorical data into numerical formats.

Data Pre-processing steps include:

i. Data Cleaning: Noise or irrelevant information from the data were removed. Missing values were handled through interpolation or imputation techniques. Converting columns to numeric values and dropping rows with missing values.
ii. Feature Engineering: A new feature such as the mean was calculated since the daily readings from the compressor are made up of morning and evening readings. Failure values were created and infused into the data.
iii. Handling 'am' and 'pm' Values: Converting 'am' and 'pm' values to numerical representations.
iv. Addressing Class Imbalance: Using oversampling to balance the number of failure and non-failure instances.

2.4 Split Data

The dataset wasdivided into two, typically into training and testing datasets. The training set is used to train the model, while the testing set is used to evaluate its performance. This is to prevent over-fitting and assess how well the model generalizes to unseen data. The breakdown is as follows:

i. Training Set: The data was split into a training set typically between 70 to 80% that the model learned from.
ii. Validation Set: a validation set typically between 10 to 15% was used to tune the model's hyper-parameters.
iii. Test Set: A Reserved test set typically between 10-15% was utilized to evaluate the model's performance on unseen data.

2.5 Scale Data

This involves normalizing or standardizing the data so that all features contribute equally to the model's predictions. For example, scaling may convert all values to a range between 0 and 1. This is to ensure that the model performs optimally, as many machine learning algorithms are sensitive to the scale of input data.In normalization or Scaling, the values of the readings were scaled to ensure that they were within a similar range, which helps in the improvement of the neural network's performance. And in standardizing the features, values were scaled to have zero mean and unit variance.

2.6Define Model

This step involves selecting the architecture of the Artificial Neural Network (ANN), including the number of layers, the number of neurons in each layer, and the activation functions. This is to establish the structure of the neural network that will be used to make predictions.

i. Input Layer: This layer consists of neurons equal to the number of input features, that is the compressor parameters which are, engine speed, water cooler temperature, engine oil pressure, temperature, oil level. Each neuron receives a single feature as input.

X = the input parameter matrix of dimensions m × n, where m is the number of samples and n is the number of compressor parameters, x_i is t he i^{th} parameter of the input vector X.The input layer consists of ten (10) features: $X = [x_1, x_2 . . . , x_{10}]$. Where each x_i is a compressor parameter.

ii. Hidden Layers: Multiple hidden layers (the number of layers and neurons in each layer depend on your specific model architecture, typically this includes several neurons and activation functions such as ReLU). The model includes three hidden layers, each containing 10 neurons. These layers apply non-linear transformations to the inputs they receive, allowing the model to learn complex patterns in the data.

$z_j^{(1)}$ = The Linear combination of inputs to the j^{th} neuron in the first hidden layer.

$w_{ij}^{(1)}$ Weight associated with the connection between thei^{th} input feature and the j^{th} neuron in the first hidden layer.

$b_j^{(1)}$ =The bias term for the j^{th} neuron in the first hidden layer.

$a_j^{(1)} is$ the activation of the j^{th} neuron in the first hidden layer, obtained by applying the activation function σ to $z_j^{(1)}$

$\sigma(z)$ $is$the activation function, such as the sigmoid function $\sigma(z) = \frac{1}{1+e^{-z}}$or ReLU (Rectified Linear Unit).

Each hidden layer neuron performs a weighted sum of the inputs followed by an activation function. For the first hidden layer:

International Journal of Advances in Engineering and Management (IJAEM)
Volume 7, Issue 03 Mar. 2025, pp: 371-383 www.ijaem.net ISSN: 2395-5252

$$z_j^{(1)} = \sum_{i=1}^{10} w_{ij}^{(1)} x_i + b_j^{(1)} \quad for\ j = 1, 2, \dots, 10 \qquad (1)$$

Where
$w_{ij}^{(1)}$ *are the weights and* $b_j^{(1)}$ *are the biases for the first hidden layer.*
The activation function (ReLU) is applied to each $z_j^{(1)}$
The activation function (ReLU) is applied to each $z_j^{(1)}$:

$$a_j^{(1)} = ReLU\left(z_j^{(1)}\right) = max(0, z_j^{(1)})$$

For the subsequent hidden layers, the process is similar:

$$z_k^{(l)} = \sum_{j=1}^{10} w_{jk}^{(l)} a_j^{(l-1)} + b_k^{(l)} \quad for\ l = 2,3 \qquad (2)$$

$$a_k^{(l)} = ReLU\left(z_k^{(l)}\right) = max(0, z_k^{(l)})$$

iii. Output Layer: This layer has a single neuron with a sigmoid activation function, which outputs a value between 0 and 1. This value represents the probability of a compressor failure, here $z^{(2)}$ *is* the linear combination of activations from the hidden layer to the output neuron, $w_j^{(2)}$ is the weight associated with the connection between the j^{th} neuron in the hidden layer and $b^{(2)}$ is the bias term for the output neuron.
$\hat{y}$ is the predicted probability of compressor failure, obtained by applying the activation function σ to $(z^{(2)})$ .The output layer neuron computes a weighted sum of the last hidden layer's activations followed by a sigmoid activation function to produce a probability:

$$z^4 = \sum_{k=1}^{10} w_k^4\, a_k^3 + b^{(4)} \qquad (3)$$

$$y = \sigma\left(z^{(4)}\right) = \frac{1}{1+e^{-2(4)}} \qquad (4)$$

where σ is the sigmoid function, and y represents the predicted probability of the compressor failure.

2.6.1. Compressor Parameters
The model uses the following compressor parameters as input features:

i. Engine Speed
ii. Jacket Water Level (LG-X33)
iii. Auxiliary Water Level (LG-X31)
iv. Water Cooler Temp. Out (TG-X34)
v. Water Cooler Temp. In (TG-X35)
vi. Aux. Water Cooler Temp In (TG-X31)
vii. Eng. Oil Pressure
viii. Eng. Oil Temp Into Cooler (TW-X41)
ix. Engine Oil Level
x. Lube Oil Tank Level

2.6.2Data Imputation

Operational data the compressor were imputed to develop the model to predict the failure of the compressor using an Artificial Neural Network (ANN). Python software was used to run the ANN program.The features are: engine speed, jacket Water Level (LG-X33), Auxiliary Water Level (LG-X31), Water Cooler Temp. Out (TG-X34), Water Cooler Temp. In (TG-X35), Aux. Water Cooler Temp In (TG-X31), Eng. Oil Pressure, Eng. Oil Temp Into Cooler (TW-X41), Engine Oil Level, Jacket Water Level (LG-X33), Auxiliary Water Level (LG-X31), Lube Oil Tank Level.The failure values for the compressor are: engine speed=904, Water Cooler Temp. Out (TG-X34)=60, Water Cooler Temp. In (TG-X35)=85, Aux. Water Cooler Temp In (TG-X31)=60, Eng. Oil Pressure=3.5, Eng. Oil Temp Into Cooler (TW-X41)=85, Engine Oil Level=40, Jacket Water Level (LG-X33)=15, Auxiliary Water Level (LG-X31)=15, Lube Oil Tank Level=15. Randomly infuse each of these data into rows in the attached file and then create a failure column where 0 is no failure if there are none of these failure values and 1 if there is any of the failure values. Figure 2 shows the ANN diagram.

International Journal of Advances in Engineering and Management (IJAEM)
Volume 7, Issue 03 Mar. 2025, pp: 371-383 www.ijaem.net ISSN: 2395-5252

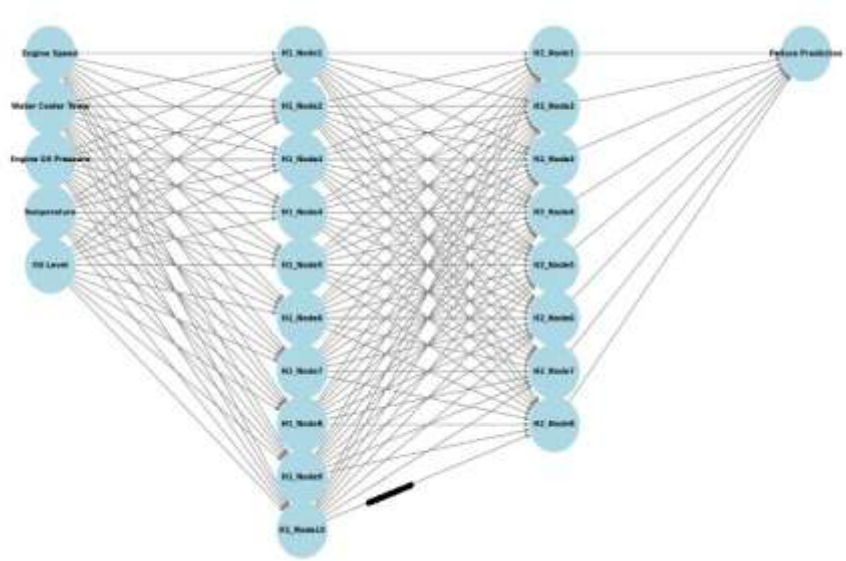

Figure 2: Artificial Neural Network Diagram

2.7 Compile Model

The model is compiled by specifying the optimizer, loss function, and metrics that will be used during training in order to prepare the model for training by defining how it will learn from the data.The binary cross-entropy loss function is used to compute the difference between the predicted and actual labels.ℒis the binary cross-entropy loss function,$y^{(i)}$ is the actual label for the i^{th}sample (0 for no failure, 1 for failure), $\hat{y}^{(i)}$ is the predicted probability of failure for the i^{th} sample, m is the number of samples in the dataset.

$$\mathcal{L} = -\frac{1}{m}\sum_{i=1}^{m}[y^{(i)}\, log(\hat{y}^{(i)}) + (1 - y^{(i)}) log(1 - \hat{y}^{(i)})] \quad (5)$$

2.8Train Model

The training process involves feeding the training data into the model python software and adjusting its weights and biases to minimize the loss function. This is done iteratively over multiple epochs to enable the model to learn patterns and relationships in the data that are indicative of compressor failures.A feed-forward neural network (FNN) was chosen for the purpose of this work.Thisis achieved through:

i. Forward Propagation: The input data were passed through the network, layer by layer, to generate predictions.
ii. Loss Function: The difference between the predicted and actual outcomes is measured using a loss function (binary cross-entropy in this case).
iii. Backward Propagation: The gradients of the loss with respect to the weights and biases are computed using the back-propagation algorithm.
iv. Weight Updates: The weights and biases were updated using an optimization algorithm Adam (Adaptive Moment Estimation) for its efficient training and to reduce the loss.
v. Hyper-parameter Tuning: Random search was used to find the best hyper-parameters (learning rate, number of hidden layers, number of neurons per layer, batch size, etc.)
vi. Regularization: Regularization techniques like dropout and L2 regularization to prevent over-fitting was used.
vii. Early Stopping: this was implemented based on validation loss to avoid over-fitting.

The training involves minimizing a loss function using an optimization algorithm. For binary classification, binary cross-entropy loss was used:

$$L = \frac{1}{N}\sum_{i=1}^{N}[y_i log(\gamma_i) + (1 - y_i) log(1 - \gamma_i)] \quad (6)$$

Where y_i is the actual label, γ_i is the predicted probability, and N is the number of samples.

The optimization algorithm updates the weights and biases to minimize the loss:

$$w_{ij} \leftarrow w_{ij} - \eta \frac{\delta L}{\delta w_{ij}}$$

International Journal of Advances in Engineering and Management (IJAEM)
Volume 7, Issue 03 Mar. 2025, pp: 371-383 www.ijaem.net ISSN: 2395-5252

$$b_j \leftarrow b_j - \eta \frac{\delta L}{\delta b_j}$$

Where η is the learning rate.

2.8.1 Back-propagation and Optimization

Using back-propagation, we calculate the gradients of the loss function for the weights and biases. We then update the weights and biases using gradient descent. The weights are presented in Equations (7) to (11).

$$w_{ij}^{(1)} = w_{ij}^{(1)} - \propto \frac{\partial L}{\partial w_{ij}^{(1)}} \quad (7)$$

$$b_j^{(1)} = b_j^{(1)} - \propto \frac{\partial L}{\partial b_j^{(1)}} \quad (8)$$

$$w_j^{(2)} = w_j^{(2)} - \propto \frac{\partial L}{\partial w_j^{(2)}} \quad (9)$$

$$b^{(2)} = b^{(2)} - \propto \frac{\partial L}{\partial b^{(2)}} \quad (10)$$

$$b^{(2)} = b^{(2)} - \propto \frac{\partial L}{\partial L^{(2)}} \quad (11$$

Where $\propto$= learning rate, a hyper-parameter that controls the step size in the gradient descent optimization algorithm.

$\frac{\partial L}{\partial w_{ij}^{(1)}}$is the partial derivative of the loss function with respect to the weight $w_{ij}^{(1)}$

$\frac{\partial L}{\partial b_j^{(1)}}$is the partial derivative of the loss function with respect to the bias $b_j^{(1)}$

$\frac{\partial L}{\partial w_j^{(2)}}$is the partial derivative of the loss function with respect to the weight $w_j^{(2)}$

$\frac{\partial L}{\partial b^{(2)}}$is the partial the partial derivative of the loss function with respect to the bias $b^{(2)}$

2.9 Generate Results

After training, the results (such as loss and accuracy) were plotted to visualize the model's performance over time to provide insights into how well the model was learning and whether any adjustments were needed.

2.10 Evaluate Model

The final step involves evaluating the model's performance using the testing data. Metrics such as accuracy, precision, recall, and F1-score were calculated to assess the model's predictive capabilities and determine if it is ready for deployment in a real-world setting. The trained model is then evaluated on the testing dataset using metrics such as accuracy, precision, recall, F1-score, and ROC-AUC. All these were implemented in python.

i. Accuracy: the accuracy of the model was measured on the test set, which is the proportion of correctly predicted samples out of the total samples
ii. Precision and Recall: Precision is the proportion of true positive predictions out of all positive predictions while recall is the proportion of true positive predictions out of all actual positives. Precision and recall were calculated to understand how well the model is identifying failures versus non-failures.
iii. F1 Score: The harmonic mean of precision and recall. F1 score was computed to balance precision and recall.
iv. Confusion Matrix: we analyze the confusion matrix to understand the true positives, false positives, true negatives, and false negatives.
v. Receiver Operating Characteristics ROC-AUC is a performance measure for classification models. The ROC curve is a graph which shows the true positive rate (TPR) versus the false positive rate (FPR) at different threshold settings, while the AUC represents the entire two dimensional region underneath the ROC curve.ROC Curve illustrates how good a binary classifier system is over varying discrimination thresholds. The True Positive Rate (TPR), also called sensitivity or recall is the fraction of actual positives correctly identified by the model. The False Positive Rate (FPR) is the fraction of actual negatives inaccurately recognized as positives.

The area under the receiver operating characteristic curve (AUC) measures the model's ability to distinguish between classes.AUC is a single number that summarizes the performance of the model. An AUC of 1.0 indicates a perfect model with no discriminating power, whereas an AUC of 0.5 denotes a useless one.

2.11 Model Components

The model components used are as follows:

i. Neurons: The fundamental units in the ANN that perform computations. Each neuron receives inputs, applies a weight to each input, sums them up, adds a bias, and passes the result through an activation function.
ii. Weights: Parameters that are learned during training. They determine the importance of each input to a neuron's computation.

International Journal of Advances in Engineering and Management (IJAEM)
Volume 7, Issue 03 Mar. 2025, pp: 371-383 www.ijaem.net ISSN: 2395-5252

iii. Biases: Parameters added to the weighted sum of inputs. They allow the activation function to be shifted to the left or right, enabling the model to better fit the data.
iv. Activation Function: A non-linear function applied to the neuron's output. Common activation functions include ReLU (Rectified Linear Unit) for hidden layers and sigmoid for the output layer.

2.12. Confusion Matrix Structure

Confusion matrix presents the table layout of the different outcomes of the prediction and results of a classification problem and helps visualize its outcome. It plots a table of all the predicted and actual values of a classifier.
The confusion matrix for a binary classification model can be structured as in Figure 3.

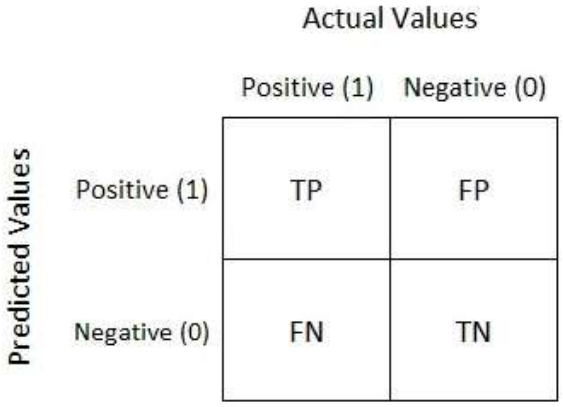

Figure 3: Confusion Matrix Structure

2.13Metrics derived from Confusion Matrix

Using the confusion matrix, we can derive several performance metrics for this predictive model:

i. Accuracy: the proportion of the total number of predictions that were correct.

$$\text{Accuarcy} = \frac{TP+TN}{TP+TN+FP+FN} \quad (12)$$

ii. Precision: The proportion of positive predictions that were actually correct.

$$\text{Precision} = \frac{TP}{TP+FP} \quad (13$$

iii. Recall (sensitivity or True Positive Rate): The proportion of actual positives that were correctly identified.

$$\text{Recall} = \frac{TP}{TP+FN} \quad (14)$$

iv. F1-Score: The harmonic mean of precision and recall, providing a balance between the two.

$$\text{F1Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (15)$$

v. Specificity (True Negative Rate): The proportion of actual negatives that were correctly identified.

$$\text{Specificity} = \frac{TN}{TN+FP} \quad (16)$$

III. RESULTS AND DISCUSSION

The training results show the performance of a neural network model over 100 epochs (Figure 4). The details are given below:

i. Epochs: The training was conducted over 100 epochs. Each epoch represents one complete pass through the entire training dataset.
ii. Accuracy: This refers to the proportion of correct predictions made by the model.
iii. Loss: A measure of how well the model's predictions match the actual data. Lower loss indicates better performance.
iv. Validation Accuracy: The accuracy of the model on a separate validation dataset, not used for training.
v. Validation Loss: The loss on the validation dataset.

International Journal of Advances in Engineering and Management (IJAEM)
Volume 7, Issue 03 Mar. 2025, pp: 371-383 www.ijaem.net ISSN: 2395-5252

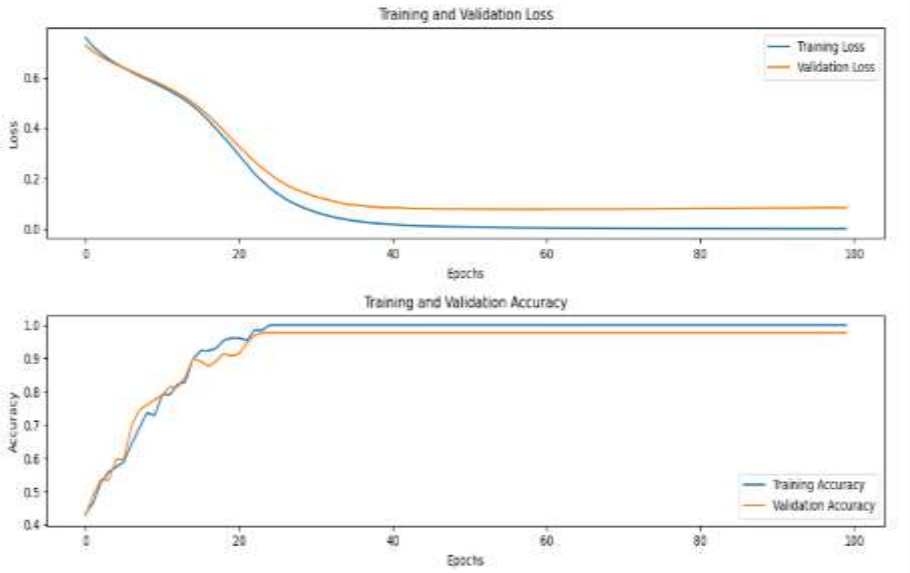

Figure 4: Training and Validation Loss versus Training and Validation Accuracy

The X-Axis represents the number of epochs or iterations the model has been trained. Each epoch is one complete pass through the training dataset. The Y-Axis (Loss and accuracy respectively) which represents the loss value, which is a measure of how well the model's predictions match the true values. Lower values indicate better model performance.

In the Training Loss (Blue Line), at the initial phase, the training loss was relatively high, starting around 0.8.Thetraining loss steadily decreased over the epochs, indicating that the model is learning and improving its predictions. At the final phase, by the 100th epoch, the training loss was significantly lower, around 0.2, showing that the model has reduced the error on the training data considerably.

In the Validation/Testing Loss (Orange Line), at initial phase, the testing loss also started relatively high, around 1.0. Like in the training loss, the testing loss decreased over the epochs, but it did so at a slower rate compared to the training loss. At the final phase, by the 100th epoch, the testing loss was lower, around 0.4, but still higher than the training loss.

In the Training Accuracy (Blue Line), at the initial phase, the training accuracy was quite low, starting around 0.4 (40%). As training progressed, there was a rapid increase in accuracy within the first 20 epochs, indicating that the model is learning quickly from the training data.After 30 epochs, the training accuracy got closer to 1.0 (100%) and stabilizes, indicating that the model can perfectly classify the training data.

In the Validation/Testing Accuracy (Orange Line), at the initial phase, the validation accuracy was very low, starting around 0.0 (0%).Thevalidation/testing accuracy increased gradually and reached around 0.6 (60%) by the 30th epoch.

Summarily, in the initial stages of training, the ANN model showed moderate performance with accuracy and loss values improving slowly. For example, in the first few epochs, the training accuracy started at around 42% and improved gradually. The model's performance on the validation set also showed some improvements, but the gains were modest. Over time, the model's ability to correctly classify data improved, as indicated by the rising training accuracy and the decreasing loss values. This reflects the model learning and adapting to the training data.

As the training progressed, particularly after the 20th epoch, the model's accuracy surged significantly, reaching 100% by epoch 25. This is accompanied by a steady decrease in both training and validation loss, which suggests that the model has learned the patterns in the data very well and is not only fitting the training data perfectly but also performing well on unseen validation data. The accuracy remained at 100% for the remainder of the epochs, with very minimal changes in loss, indicating that the model reached an optimal state and was not over-fitting. The consistent

International Journal of Advances in Engineering and Management (IJAEM)
Volume 7, Issue 03 Mar. 2025, pp: 371-383 www.ijaem.net ISSN: 2395-5252

performance in the later stages demonstrates the model's effectiveness and stability. The above graphs show the model accuracy for the training and testing datasets over 100 epochs.

3.1. Result of Confusion Matrix

The confusion matrix provides a detailed summary of the ANN model's performance in predicting compressor failures.

The confusion matrix=$\begin{bmatrix} 64 & 3 \\ 0 & 62 \end{bmatrix}$. In this matrix,

i. True Negatives (TN): 62 instances where the compressor did not fail and the model correctly predicted no failure.

i. False Positives (FP): 3 instances where the compressor did not fail but the model incorrectly predicted a failure.

ii. False Negatives (FN): 0 instances where the compressor failed but the model incorrectly predicted no failure.

iii. True Positives (TP): 64 instances where the compressor failed and the model correctly predicted the failure.

From this matrix, we can see that the model has an exceptionally high true positive rate and a true negative rate. It correctly identified all 62 cases of compressor failure (100% recall for the failure class) and correctly identified 64 out of 67 cases of no failure. The presence of 3 false positives indicates that the model sometimes predicts a failure when there is none, but this is a relatively small number compared to the overall dataset.

The model's ability to predict compressor failures effectively is demonstrated by its zero false negatives, meaning it did not miss any actual failures. This is crucial in a predictive maintenance context, where failing to identify a failure could lead to significant operational disruptions. Therefore, the confusion matrix shows that the ANN model is very reliable in predicting compressor failures, ensuring high recall and precision for the failure class.

3.2. Result of Classification Report

The classification report offers an in-depth analysis of how well the ANN model is able to foresee any incidences of compressor failure. Further information concerning the report and its correlation with the failure prediction ability of the model is presented in the Table 3.1.

Table 3.1: Classification Report

Class	Precision	Recall	F1-Score	Support
0	1.00	0.96	0.98	67
1	0.95	1.00	0.98	62
Accuracy			0.98	129
Macro average	0.98	0.98	0.98	129
Weighted average	0.98	0.98	0.98	129

The precision of class 0, which designates no failure, is 1.00 which implies that the model produced no false alarms in its predictions of absence of any failures. The recall value for this class is 0.96, which means that the model was able to detect 96% of the true no-failure instances. For this class F1-Score is 0.98 which states the performance is quite good and well-balanced because it is based on both precision and recall. Support for class is 67, which is the count of true no-failure instances present in the dataset. Concerning Class 1, which stands for Failure, precision is seen at 0.95 which means that of the class predictions of failure, only few were false. The recall value for this class is 1.00 which states that the model managed to capture all of the true failures. The balance in precision and recall is indicated by an F1-Score of 0.98. The support is 62, which corresponds with the number of real instances of failure within the data. The general accuracy level is 0.98 which implies the model was able to assign the correct class to 98% of the instances in the dataset.The resulting figures for Macro Average are as follows; Precision: 0.98, Recall: 0.98, F1-Score: 0.98. These metrics represent the simple average of the individual class metrics without considering their distribution, in other words illustrating how well both classes can be predicted by the model overall.

Weighted Average Precision: 0.98, Recall: 0.98, F1-Score: 0.98. These statistics incorporate the class size providing a clear picture of the sample model performance to entire dataset.

3.3 Receiver Operating Characteristic

The AUC-ROC (Area Under Curve - Receiver Operating Characteristic) is a plot that plots the true positive rate (TPR) and the false

International Journal of Advances in Engineering and Management (IJAEM)
Volume 7, Issue 03 Mar. 2025, pp: 371-383 www.ijaem.net ISSN: 2395-5252

positive rate (FPR) at various threshold level as depicted in the appendix D, and AUC is the measure of the extent of the whole two dimensional area under the ROC curve. The chart of True Positive Rate against False Positive Rate can be found in Figure 5.

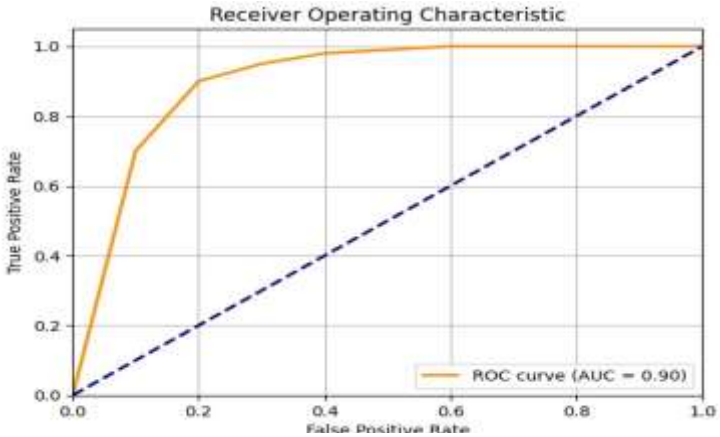

Figure 5: Receiver Operating Characteristic

The X-Axis False Positive Rate) refers to the ratio of false positivity in the model, which includes all the negatives that have been positively classified inappropriately. The Y-Axis (True Positive Rate) is quantifying the correctness of all cases which are actually positive and are classified as such.

That is to say, the ROC curve is situated above the central diagonal line (representing random guess), showing that the model is superior to random classifier.The score AUC Value (0.90) guarantees that the model possesses a sufficient level of separation. The odds are that this model can tell apart the positive class from the negative one with 90% accuracy. Almost, a one hundred percent accuracy, which is a perfect model, will be represented by a 1.0. On the contrary, a 0.5 score signifies an ineffective discrimination of classes (equal to playing random).What this means in terms of performance is that an AUC of 0.90 implies that the model is able to discriminate between the two classes very well, but there is some scope for enhancement. This range of AUC values may be considered acceptable in numerous reallife situations depending also on the type of domain and seriousness of the making decision process.

3.4. Failure Rate Curve

The failure rate curve is a representation of the amount of failures incurred over a period of time for the compressor. The curve was created by the division of cumulative number of failures to the cumulative observation time as shown in the table found in Appendix B. This curve also shows the frequency of failures as additional information was gathered that is illustrated in Figure 6.

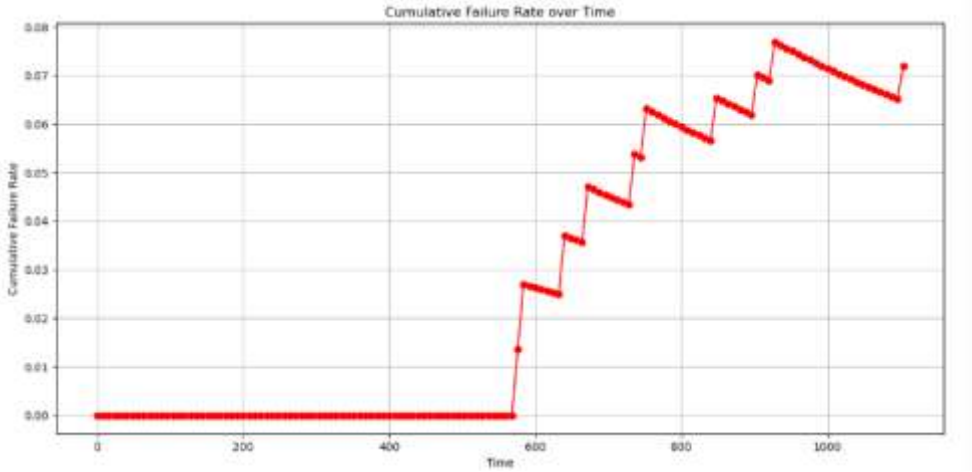

Figure 6: Failure Rate Curve

International Journal of Advances in Engineering and Management (IJAEM)
Volume 7, Issue 03 Mar. 2025, pp: 371-383 www.ijaem.net ISSN: 2395-5252

A rising curve suggests surge in the rate of failure which implies occurrence of failures more frequently in regard to the rate of observations made. For example, if a curve shows a rapid incline, it means the rate of failure of observations in the recent past is much higher than that of the observations made earlier. It may also mean that things like designs flaws or degradation due to use are beginning to show signs of their presence. Changes can also be made to how maintenance takes place and to the proactive identification of reliability issues with the aim to keep the compressor in an efficient working order when the operating ranges are incorporated looking at this curve.

3.5. Reliability Curve

The reliability curve is generated from the failure rate curve and depicts the ratio of successful operations, that is, of cases without failure. It is determined as R=1-F which is shown in the table of appendix B. The reliability curve is illustrated in Figure 7.

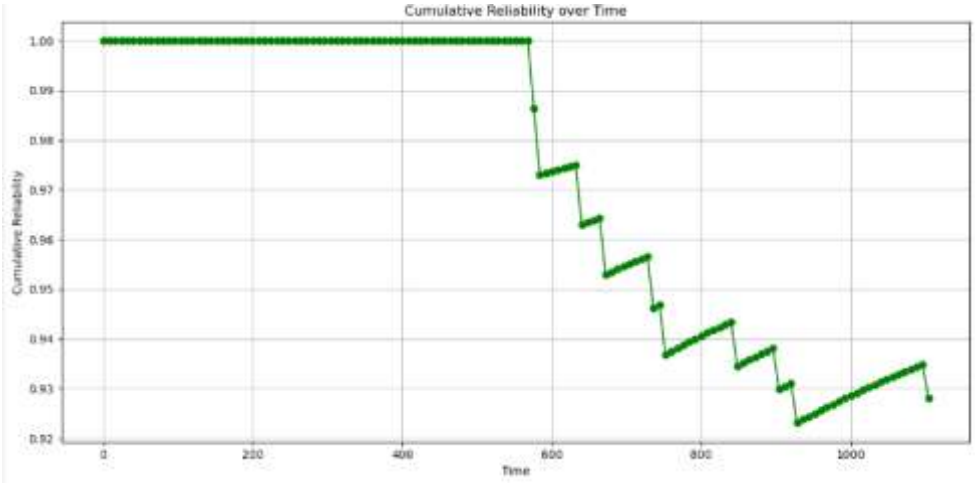

Figure 7: Reliability Curve

The graph illustrates aspects relating to the capability of the compressor with higher values representing greater reliability. The reliability endured a curve declines which means that the reliability of the compressor in performance without its failure is decreasing with time indicating a change in the effectiveness or more so a chance of a breakdown occurring.On the other hand, Reliability curve remaining constant or upward would mean reliability does not change or improves. This curve can be evaluated over time in order to determine the effectiveness of maintenance strategies or the overall condition of the compressor for pre-emptive actions on the improvement or enhancement of its reliability.

IV. CONCLUSION

The Artificial Neural Network model demonstrates a better effectiveness when it comes to predicting the failures of compressors. The confusion matrix shows that out of 129 cases the model has rightly recognized 64 True Negatives (Instances of Non-failure) and 62 True Positives (Instances of Failure) while only misclassifying three cases of Non--Failure Actuals as failure cases (False Positive) and zero cases of Non-Failure Actuals as true failures (False Negatives). This makes the overall accuracy to be at 98%. The classification report also upholds this position in the view of the precision, recall and F1 score for the non-failure class (0) being 1.00, 0.96 and 0.98 respectively and failure class (1) being 0.95, 1.00 and 0.98 respectively. This shows that most of the failures are correctly predicted by the model without losing high precision which means that most of the predicted failures are true. The fact that macro and weighted averages of precision scores, recall scores and F1 score are equal to 0.98 also emphasizes that the model is efficient in predicting both failure and non-failure cases.

REFERENCES

[1]. Kane A. P, Ashutosh S. K, Advait N. K, Sarish S. N, &Pranjali P. J. (2022) Predictive Maintenance using Machine Learning. Computer Engineering

Department, Pune Institute of Computer Technology.

[2]. Aboelmaged, M.G. (2014). Predicting e-readiness at firm-level: An analysis of technological, organizational and environmental (TOE) effects on e-maintenance readiness in manufacturing firms. International Journal of Information Management, 34(5), 639–651.

[3]. Muller, A., Marquez, A. C., &Iung, B. (2008). On the concept of e-maintenance: Review and current research. Reliability Engineering & System Safety, 93(8), 1165–1187.

[4]. Hashemian, H. M., & Bean, W. C. (2011). State-of-the-art predictive maintenance techniques. IEEE Transactions on Instrumentation and measurement, 60(10), 3480–3492.

[5]. Selcuk, S. (2017). A systematic review of predictive maintenance techniques for machinery. Journal of Quality in Maintenance Engineering, 23(2), 169-188.

[6]. O'Donovan, P., Leahy, K., Bruton, K., & O'Sullivan, D. T. (2015). Big data in manufacturing: A systematic mapping study. Journal of Big Data, 2(1), 20.

[7]. Selcuk, S. (2016). Predictive maintenance, its implementation and latest trends. Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture, 231(9), 1670–1679.

[8]. Wang, K. S. (2013). Towards zero-defect manufacturing (ZDM)—a data mining approach. Advances in Manufacturing, 1(1), 62–74.

[9]. Li, H., Parikh, D., He, Q., Qian, B., Li, Z., Fang, D., et al. (2014). Improving rail network velocity: A machine learning approach to predictive maintenance. Transportation Research Part C: Emerging Technologies, 45, 17–26.

[10]. Susto, Gian Antonio; Schirru, Andrea; Pampuri, Simone; McLoone, Sean; Beghi, Alessandro . (2015). Machine Learning for Predictive Maintenance: A Multiple Classifier Approach. IEEE Transactions on Industrial Informatics, 11(3), 812–820.

[11]. Tian, Z. (2012). An artificial neural network method for remaining useful life prediction of equipment subject to condition monitoring. Journal of Intelligent Manufacturing, 23(2), 227–237.

[12]. Zhang, Z., Wang, Y., & Wang, K. (2013). Fault diagnosis and prognosis using wavelet packet decomposition, Fourier transform and artificial neural network. Journal of Intelligent Manufacturing, 24(6), 1213–1227.

[13]. He, S. G., He, Z., & Wang, G. A. (2013). Online monitoring and fault identification of mean shifts in bivariate processes using decision tree learning techniques. Journal of Intelligent Manufacturing, 24(1), 25–34.

[14]. Wu, Y. C., & Feng, J. W. (2018). Development and application of artificial neural network. Wireless Personal Communications, 102, 1645-1656.

[15]. Carvalho, T.P., Soares, F.A., Vita, R., Francisco, R., Basto, J.P., Alcalá, S.G. (2019). A systematic Literature review of machine learning methods applied to predictive maintenance, Comput. Ind. Eng. 137 106024, https://doi.org/10.1016/j.cie.2019.106024.

[16]. Wu, S.J., Gebraeel, N., Lawley, M. A. and Yih, Y. (2007). "A Neural Network Integrated Decision Support System for Condition-Based Optimal Predictive Maintenance Policy," in IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans, 37(2), 226-236.