06.30.2024 JOURNAL OF

African Sustainable Development (JASD) Vol. 4 No.2

A NEW MACHINE LEARNING MODEL FOR PREDICTING RELIABILITY OF CRANES

NADU EMMANUEL OBI; HAROLD UGOCHUKWU NWOSU; & MATHEW, U SHADRACK

Department of Mechanical Engineering, University of Port Harcourt.
obinadu@gmail.com

ABSTRACT

In this study, new machine learning models were developed for the assessment of reliability and availability for machinery maintenance using pattern recognition artificial neural network. The choice of a classification model stemmed from the nature of available data. Hence, the input data variables for cranes were obtained from Hyster RS45-27 CH and Konecranes Liftace TFC 45 97-2002. The artificial neural network model was developed using MATLAB. Trial and error was initially used to arrive at the neural network architecture that gave the lowest mean square error. The architecture that was finally selected consists of input layer, three hidden layers and an output layer. The first and the last hidden layers had a total of 10 neurons each, while the second hidden layer had a total of 20 neurons. However, for the cranes, the highest prediction accuracy went to PRN-LMA with an accuracy of 87%, followed by PRN-CGF with an overall prediction accuracy of 86.3%. Next were PRN-OSS and PRN-BFG with prediction accuracies of 84.9% and 84.6% respectively, while PRN-BR was the least

Introduction

Most production industries require complex systems that are available for an extended period, and some of the indices used to verify the quality of any operating system are the system's reliability and accessibility. This is because it has become increasingly difficult to keep operating systems in excellent functioning order due to the increase in innovation these days. As the technologies of operating systems advance, there is a high necessity to pay great attention to the machine's maintainability, availability, and reliability (Soualhi *et al.*, 2020). The reliability of a component or system is the likelihood that the component or system would operate satisfactorily for a defined period under specified operating conditions. The

E-ISSN 3026-8575 P-ISSN 3027-1266

accurate, with an accuracy of 76.6%. Generally, the PRN-LMA models gave the highest prediction accuracy, while the Bayesian regularization models (PRN-BR) gave the least prediction accuracy. Particularly, PRN-CGF model, followed by PRN-LMA model predicted the highest number of failure days, while both models gave the highest prediction accuracy for failure days.

Keywords: Reliability, Machine Learning, Pattern Recognition, Cranes.

Possibility that an item or system will work satisfactorily as required when used under certain predefined conditions is known as its availability. Maintainability refers to the likelihood that an item, component or equipment could be restored to its satisfactory functional conditions in a specific range and time frame under specified conditions by individuals with the required skillsets, resources, and techniques (Odeyar *et al.*, 2022).

Currently, research on asset health and life span prediction has significantly increased in engineering asset management discipline reflecting the awareness of engineers today about the importance of machines' operational state. Maintenance has often been the traditional approach applied to increase machine availability. However, machine availability does not always imply machine usage because the ripple effects of breakdowns result in idling losses and subsequent productivity losses. Breakdowns and other failures often lead to machines underutilization and to mitigate this and other setbacks (Serey *et al.*, 2023).

Furthermore, degradation affects the equipment's life span at distinct periods, lowering the system's reliability (Ezendiokwere *et al.*, 2021). With better understanding and research on maintainability, availability, and reliability assessment, the health and life span of machinery can be significantly improved. In general, proper reliability and availability analysis can improve the performance of plant equipment. This is because the plant's efficiency suffers when the operating system is unavailable and unreliable. Poor availability and reliability lead to failures in production units.

Although there are different maintenance methods that can be adopted under different conditions, yet, the maintenance management's key role includes mostly predictive maintenance (PdM), preventive maintenance (PM), and condition-based maintenance. The maintenance team of the various departments in different establishments most times carries out daily/routine plant maintenance checks. In addition to the equipment daily checks, the maintenance section and the vendors carry out scheduled maintenance and overhauling after a certain period, for instance, after about 176 hours. A monthly

diagnosis of the operational conditions like vibration, speed, throughput, and state of the lubrication oil is evaluated by the maintenance department (Fontes & Pereira, 2016).
Every reliability analysis effort involves searching the state space of the system for those states that represent the event of interest, typically failure of the given system. This essentially translates into a search procedure to efficiently identify states to be examined and then using a mechanism to evaluate these states. This makes reliability and availability studies suitable for application of machine learning techniques like artificial neural networks (Payette & Abdul-Nour, 2023).

Materials and Methods

The work flow for the general artificial neural network design process has important key steps (Fontes & Pereira, 2016). These include data collection and preparation, network creation and configuration, initialization of weights and biases, network training, validation (post-training analysis) and demonstration**.** The failure data used for developing the artificial neural network machine learning model in this study was from the Port Harcourt wharf/seaport. The raw failure data obtained were tabulated according to month of operation, with each table having date, start and stop times, daily operating hours and frequency of machine failures. In addition, time between failure (TBF) data was equally generated from the raw data.
These input variables were first divided into a training, test and validation data sets in the ratio 60:20:20. The whole data sets were later fed into a pattern recognition classification feed forward neural network. Pattern recognition is the process of classifying input data into objects, classes, or categories using computer algorithms based on key features or regularities (Serey *et al.*, 2023). Pattern recognition networks (PRN) are feed forward neural networks that can be trained to classify inputs according to target classes. It has applications in areas like computer vision, image segmentation, object detection, radar processing, speech recognition and text classification, but rarely used for reliability and availability studies (Odeyar *et al.*, 2022).
However, Soualhi *et al.* (2020) suggested that this method presents many advantages for fault detection and diagnostics given its reliable results and facility for manipulating artificial intelligence tools like machine leaning. The target data for pattern recognition networks usually consists of vectors of all zero values except for a 1 in element i, where i is the class they are to represent. The choice of a classification model stemmed from the nature of available data. The artificial neural network model was developed using MATLAB.

Trial and error was initially used to arrive at the neural network architecture that gave the lowest mean square error. The architecture that was finally selected consists of input layer, three hidden layers and an output layer. The first and the last hidden layers had a total of 10 neurons each, while the second hidden layer had a total of 20 neurons. The final model had easy start status (which records whether the machine was easily started or not), daily operating hours and time between failures as input variables, while the output variable was the machine failure potential. The MATLAB model results were subsequently analyzed.

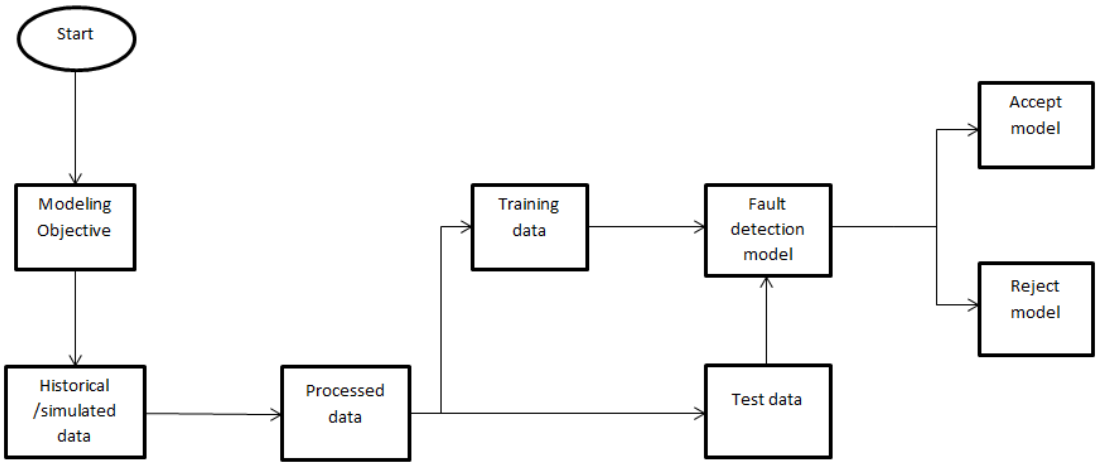

Figure 1 Workflow for developing data-driven machine learning model for fault detection (Odeyar *et al.*, 2022)

Optimization Algorithms

A number of optimization algorithms were utilized in training the pattern recognition neural network model. These algorithms include the following.

Levenberg–Marquardt Algorithm

Levenberg–Marquardt algorithm is an optimization technique for nonlinear least-square minimization problems. LMA is a local optimization algorithm; hence, the delivered solution is often a local minimum. LMA is characterized by the ability to find a final solution although the starting point can be far from this solution. This algorithm is nearby to Newton's method, but unlike the latter, the so-called Hessian matrix is not calculated in LMA. Instead, it is approximated, and a new parameter called damping parameter (regularization), whose the role is to stabilize the algorithm and avoid

singular matrix cases during the optimization process, is introduced. The following equations highlight, respectively, the Hessian matrix approximation and the gradient matrix utilized in LMA (Chu *et al.,* 2017):

$$H = J^T J \quad (1)$$

$$g = J^T e \quad (2)$$

where T stands for matrix transpose operator, J is the Jacobian matrix and e is the error vector. In addition to using the above-mentioned Hessian matrix approximation, the introduced damping parameter is employed in the Newton's formula to obtain the updating equation shown: in which, t is the iteration subscript, and the coefficient λ denotes the damping parameter that is decreased iteratively if there is enhancements in the objective function, or increased otherwise. For the case of MLP weights optimization, the x term in (3) stands for weights.

$$x_{i+1} = x_i - (H - \lambda I)^{-1} \times g \quad (3)$$

Being highly efficient and stable with a high convergence speed, LMA has hitherto received great interest in neural network weights optimization (Amar *et al.*, 2022).

Bayesian Regularization (BR) Algorithm

Bayesian regularization (BR) algorithm is a well-formulated algorithm for neural network training phase. Indeed, the principle of BR algorithm in optimizing the weights and bias of neural networks consists in applying LMA concept after formulating a minimization problem that includes an objective function with a summation of weighted two error terms: squared network weights (E_w) and squared errors (E_D). The following equation shows this objective function:

$$F(\omega) = \alpha E_w + \beta E_D \quad (4)$$

where α and β point out the objective function parameters. These two parameters are gained from Bayes' theorem. The optimization process of BR consists firstly in selecting the training set and guessing the weight vector, and this by means of Gaussian distribution. Accordingly, by applying some algebraic operations to the outcomes of the initialization step, the α and β optimum values can be achieved. Then, LMA is used to minimize $F(\omega)$, and according to its results, the weights are updated. These steps are reiterated until a stopping criterion is verified (Amar *et al.*, 2022).

Conjugate Gradient Algorithm

All the conjugate gradient algorithms start out by searching in the steepest descent direction (negative of the gradient) on the first iteration.

$$\boldsymbol{p_o} = -\boldsymbol{g_o} \quad (5)$$

A line search is then performed to determine the optimal distance to move along the current search direction:

$$\boldsymbol{X_{i+1}} = \boldsymbol{X_i}\alpha_i\boldsymbol{p_i} \quad (6)$$

Then the next search direction is determined so that it is conjugate to previous search directions. The general procedure for determining the new search direction is to combine the new steepest descent with the previous search direction:

$$\boldsymbol{p_i} = -\boldsymbol{g_i} + \beta_i\boldsymbol{p_{i-1}} \quad (7)$$

The various versions of the conjugate gradient algorithm are distinguished by the manner in which the constant β_i is computed. For the Fletcher-Reeves update, the procedure is:

$$\beta_i = \frac{\boldsymbol{g_i^T g_i}}{\boldsymbol{g_{i-1}^T g_{i-1}}} \quad (8)$$

The above equation represents the ratio of the norm squared of the current gradient to the norm squared of the previous gradient. The conjugate gradient algorithms only a little more storage than the simpler algorithms. Therefore, these algorithms are good for networks with a large number of weights.

Broyden, Fletcher, Goldfarb and Shanno (BFGS) Quasi-Newton

Newton's is an alternative to the conjugate gradient methods for fast optimization. The basic step of Newton's method is:

$$\boldsymbol{X_{i+1}} = \boldsymbol{X_i} - \boldsymbol{A_i^{-1}}\boldsymbol{g_i} \quad (9)$$

Where A_i^{-1} is the Hessian matrix (second derivatives) of the performance index at the current values of the weights and biases.

Newton's method often converges faster than conjugate gradient methods. Unfortunately, it can complex and expensive to compute the Hessian matrix for feed-forward neural networks. There is a class of algorithms that is based on Newton's method, but which does not require calculation of second derivatives. These are called quasi-Newton (or secant) methods. They update an approximate Hessian matrix at each

iteration of the algorithm. The quasi-Newton method that has been most successful in published studies is the Broyden, Fletcher, Goldfarb and Shanno (BFGS) update.

This algorithm requires more computation in each iteration and more storage than the conjugate gradient methods, although it generally converges in fewer iterations. The approximate Hessian matrix must be stored, and its dimension is of the order of the number of weights and biases in the network. BFGS quasi-Newton method can train any network as long as its weight, net input and transfer functions have derivative functions. Back-propagation is usually utilized in calculating derivatives of performance with respect to the weight and bias variables X. Each variable is adjusted according to the following:

$$X_{i+1} = X_i + adX \tag{10}$$

Where dX is the search direction. The parameter a is selected to minimize the performance along the search direction. The line search function is used to locate the minimum point. The first search direction is the negative of the gradient of performance. Subsequently, the search direction is computed according to the following formula:

$$dX = \frac{-H}{gX} \tag{11}$$

Where gX is the gradient and H is the approximate Hessian matrix.

One Step Scant Method

Because the BFGS algorithm requires more storage and computation in each iteration than the conjugate gradient algorithms, there was a need to for a secant approximation with smaller storage and computation requirements. The one step secant method is an attempt to bridge the gap between conjugate gradient algorithms and the quasi-Newton algorithms. The one step secant method does not store the complete Hessian matrix, but assumes that at each iteration, the previous Hessian was the identity matrix. This has the advantage that the new search direction can be computed without computing a matrix inverse. The algorithm requires less storage and computation per epoch than the BFGS algorithm. However, it requires slightly more storage and computation per epoch than the conjugate gradient algorithms. Each variable is adjusted according to the following:

$$X_{i+1} = X_i + adX \tag{12}$$

Where dX is the search direction. The parameter a is selected to minimize the performance along the search direction. The line search function is used to locate the minimum point. The first search direction is the negative of the gradient of performance. Subsequently, the search direction is computed from the new gradient and the previous steps and gradients according to the following formula:

$$dX = -gX + Ac * X_{-step} + Bc * dgX \quad (13)$$

Where gX is the gradient, X_{-step} is the change in the weights on the previous iteration and dgX is the change in gradient from the last iteration.

Results and Discussion

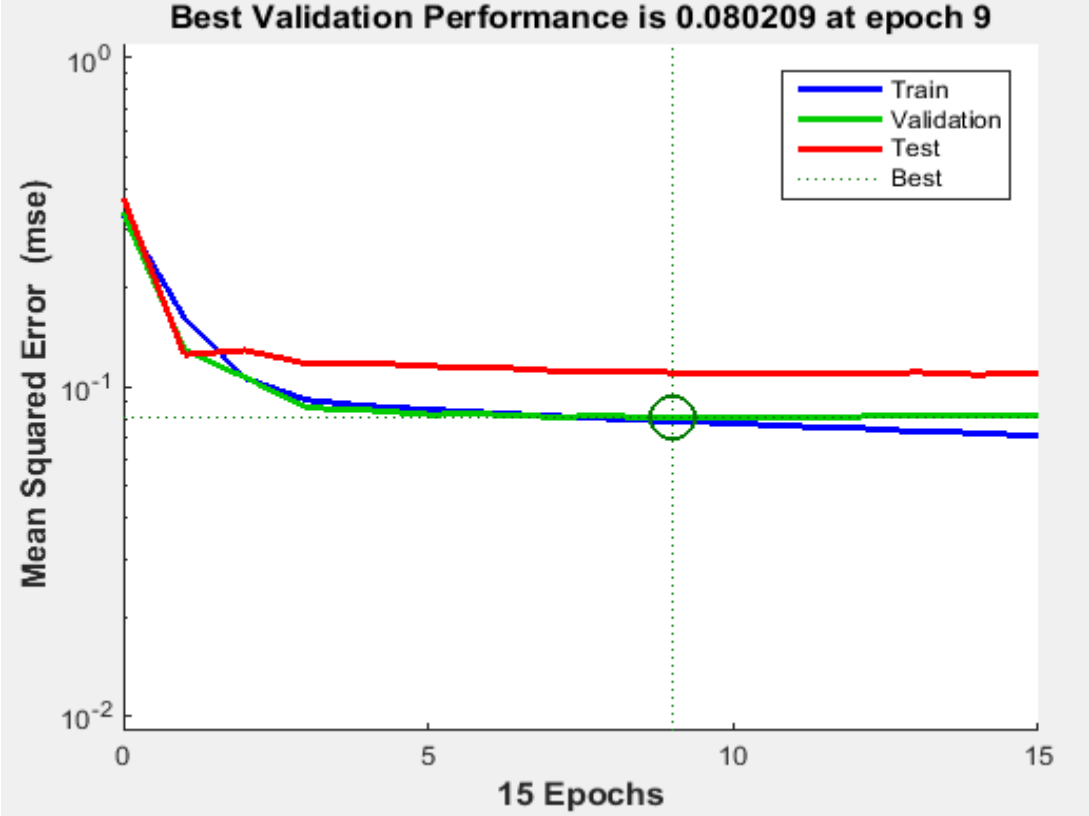

Figure 2 Training, validation, and testing curves for pattern recognition Levenberg-Marquardt (PRN-LMA) model

Figure 2 shows training, validation, and testing curves for pattern recognition Levenberg-Marquardt (PRN-LMA) model of cranes. From the graph, it can be seen that all errors including training, validations, and test errors consistently reduced until after

the third epoch. The combined errors continued to reduce such that it recorded the lowest combined training, validation and test error values occurred during the ninth epoch. After this period, both validation and training errors started to increase, while the training error decreased further as the number of epochs increased. The similarity in trend between the training, validations, and test curves depicts a successful training of the pattern recognition Levenberg-Marquardt (PRN-LMA) model and its potential use in assessing the failure potential of cranes.

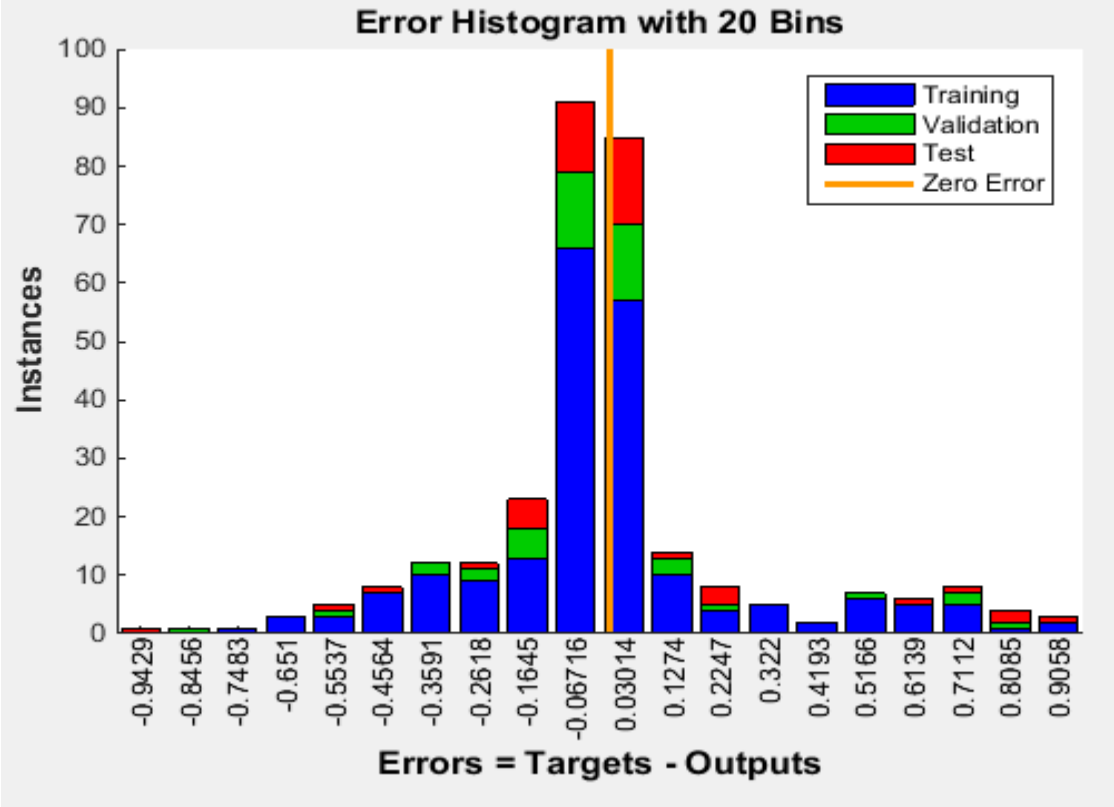

Figure 3 Prediction error histogram for pattern recognition Levenberg-Marquardt (PRN-LMA) model

Figure 3 shows the prediction error histogram for pattern recognition Levenberg-Marquardt (PRN-LMA) model for cranes. Here, the error is measured as the difference between the target variable values and the corresponding predicted outcome values. A good model should have errors congregated around the zero error mark, such that the highest histogram bars would be found around the zero error line. From the histogram, it can be deduced that a majority of the error instances were located around the zero error line, while the error instances recorded further away from the zero error line were

actually few. This can be confirmed by the smaller number of error instances recorded for errors further from the zero error line. Since the majority of the error instances were congregated around the zero error line, it demonstrates the ability of the pattern recognition Levenberg-Marquardt (PRN-LMA) model to predict failure of cranes. This result is obviously useful in assessing the failure potential of cranes.

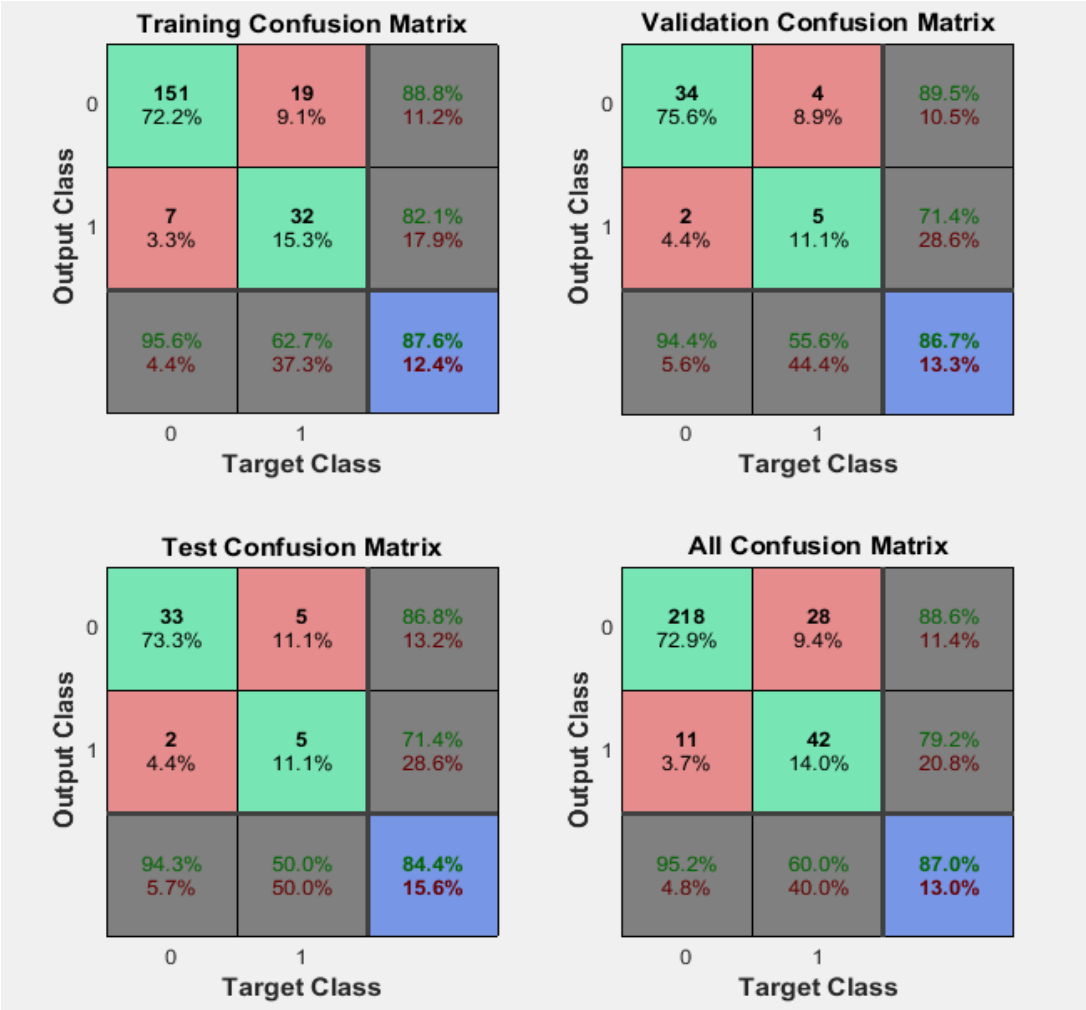

Figure 4 Confusion matrix for pattern recognition Levenberg-Marquardt (PRN-LMA) model

Figure 4 shows a representation of the confusion matrix for pattern recognition Levenberg-Marquardt (PRN-LMA) model. From the figure, it can be noticed that the confusion matrices of the test, validation and test data were all presented. In addition, a combined matrix of all the data was equally presented. The figure shows that after successfully training the Levenberg-Marquardt (PRN-LMA) model, the training result

gave 87.6% correct classifications against 12.4% wrong classification. When the trained model was validated using validation data, the validation results gave 86.7% correct predictions, with 13.3% wrong classifications. The trained model was equally tested and the test results showed that 84.4% of the test data were correctly classified, while 15.6% were wrongly classified. However, the combined confusion matrix showed that for the whole data set, 87.0% of the data set was accurately classified using the available input variables, but 13.0% of the data set was wrongly classified. The wrong classifications include both false positive and false negative classifications.

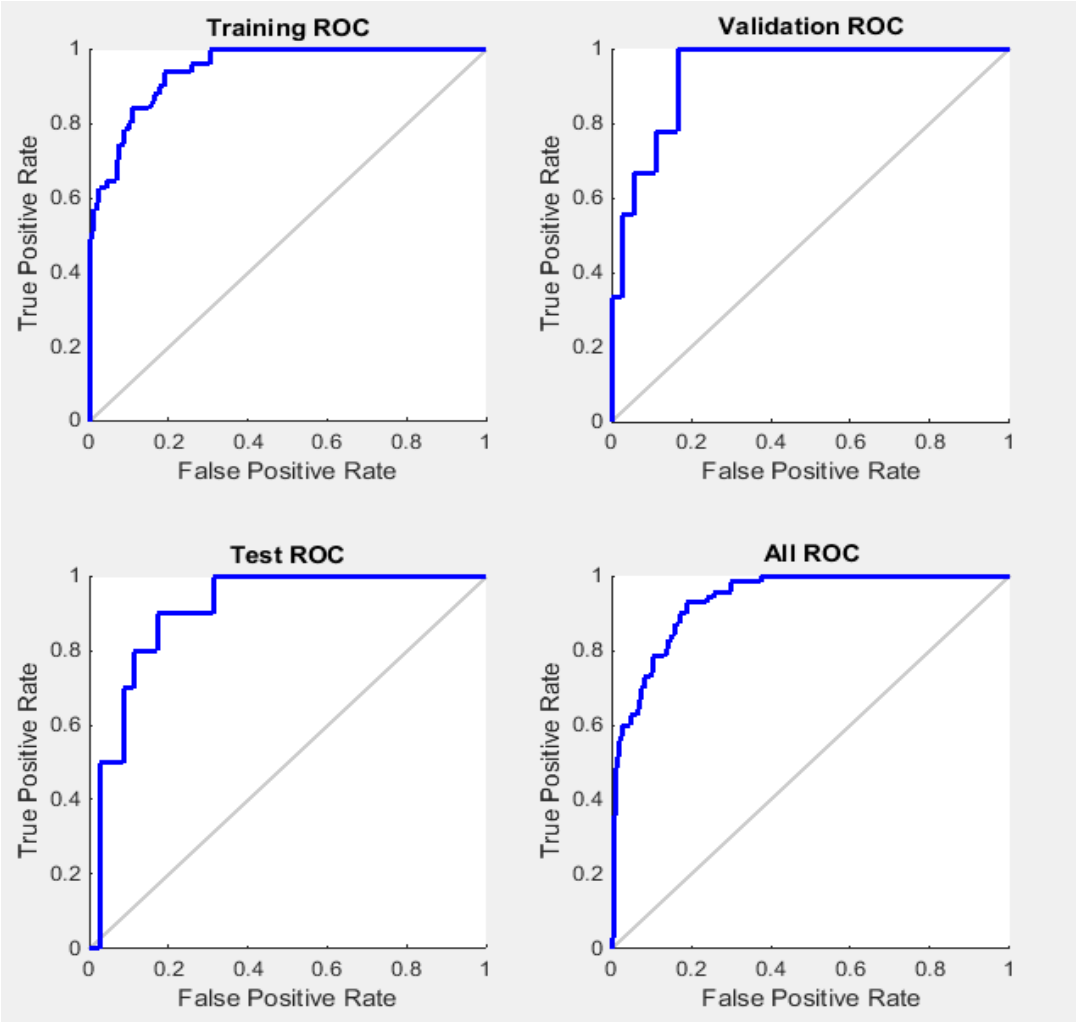

Figure 5 Receiver operating characteristic plot for pattern recognition Levenberg-Marquardt (PRN-LMA) model

Figure 5 shows the receiver operating characteristic plot for pattern recognition Levenberg-Marquardt (PRN-LMA) model. The figure depicts a plot of true positive rate

against false positive rate. Plots like this are equally helpful in gauging the accuracy of machine learning classification models. Ideally, the receiver operating characteristic curve for a good model would normally line the outer edges of the upper triangle formed by counterdiagonal that divides the square of the plot into two. Such that less accurate models have receiver operating characteristic curves that somewhat deviates from the aforementioned ideal curve. The more the receiver operating characteristic curve deviates from the egdes, the less accurate the classification model. From the figure, it can be deduced that the receiver operating characteristic curves for training, validation, and test data all had slight deviations from the edges, which confirms the substantial accuracy of the characteristic plot for pattern recognition Levenberg-Marquardt (PRN-LMA) model for accurately classifying the failure potential of cranes.

Another important metric derivable from the receiver operating characteristic plot is the area under the curve (AUC). The area under the curve of an accurate classification model would amount to unity, given that the square that makes up the receiver operating characteristic plot always has a unit length. But a completely inaccurate model would have a receiver operating characteristic plot that lines the counterdiagonal, which gives an area under the curve value of 0.5. Hence, most classification models will have area under the curve values that lie between 0.5 and 1. Since the receiver operating characteristic plots shown above are considerably far from the counterdiagonal, the Levenberg-Marquardt (PRN-LMA) model can be described as satisfactorily accurate.

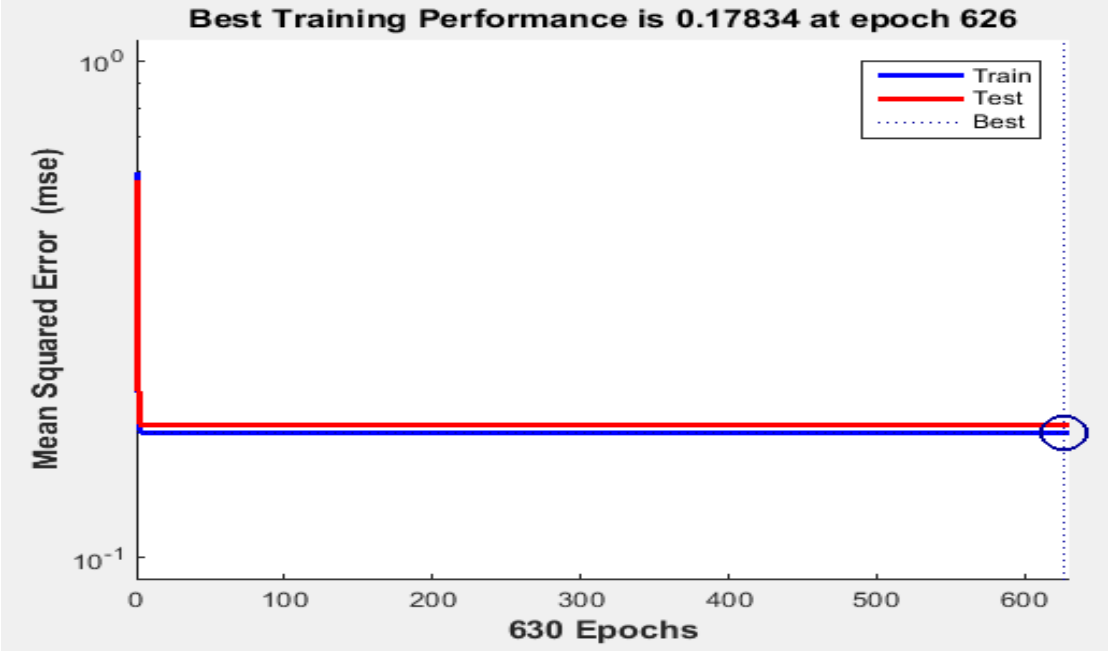

Figure 6 Training, validation, and testing curves for pattern recognition Bayesian regularization (PRN-BR) model

Figure 6 shows training, validation, and testing curves for pattern recognition Bayesian regularization (PRN-BR) model of cranes. From the graph, it can be seen that all errors

including training, and test errors consistently reduced shortly after the training was initiated. The combined errors continued to reduce such that it recorded the lowest combined training, and test error values occurred during the six hundred and twenty sixth (626th) epoch. The longer time it took the Bayesian regularization (PRN-BR) model to achieve optimality in terms of training and testing errors can be explained by the fact the Bayesian regularization optimization algorithm is a more computationally demanding routine to undertake compared to other optimization algorithms.

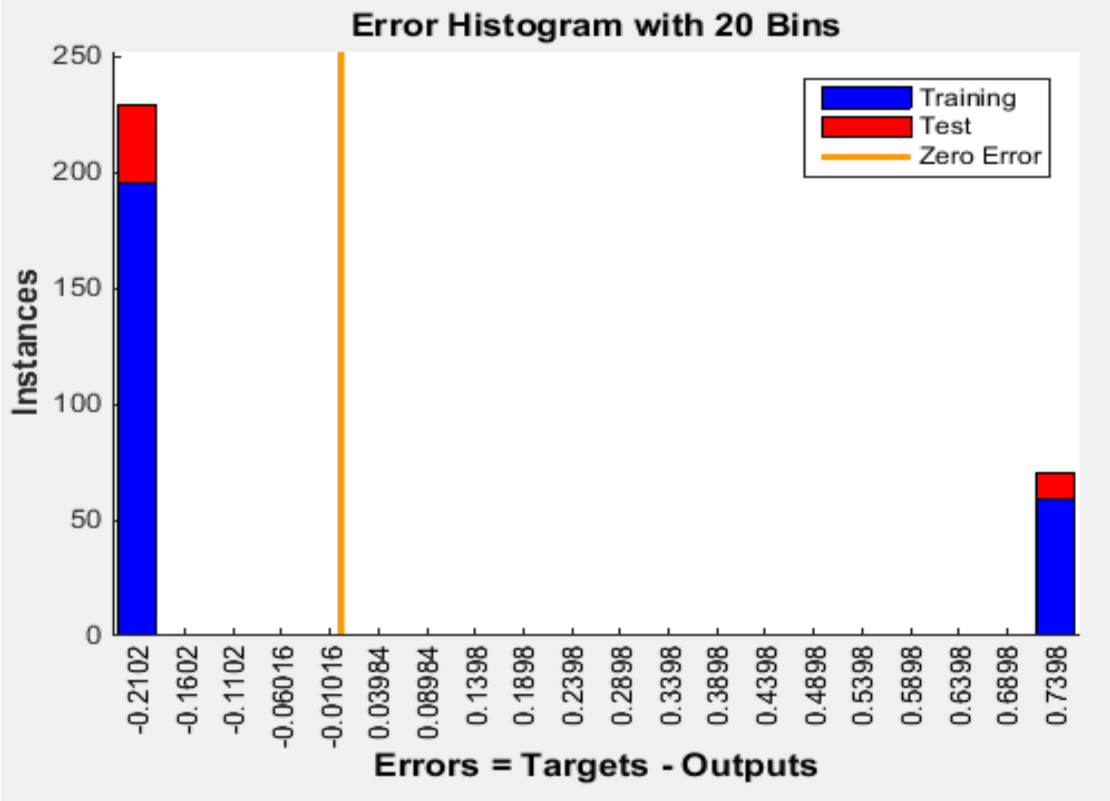

Figure 7 Prediction error histogram for pattern recognition Bayesian regularization (PRN-BR) model

Figure 7 shows the prediction error histogram for pattern recognition Bayesian regularization (PRN-BR) model for cranes. From the histogram, it can be deduced that all the error instances were located far from the zero error line. Since all the error instances were far away from the zero error line, it demonstrates the inability of the pattern recognition Bayesian regularization (PRN-BR) model to predict failure of cranes. This result has obviously implications when assessing the failure potential of cranes. It

shows that the Bayesian regularization (PRN-BR) model cannot be used with confidence in assessing the reliability and availability of cranes.

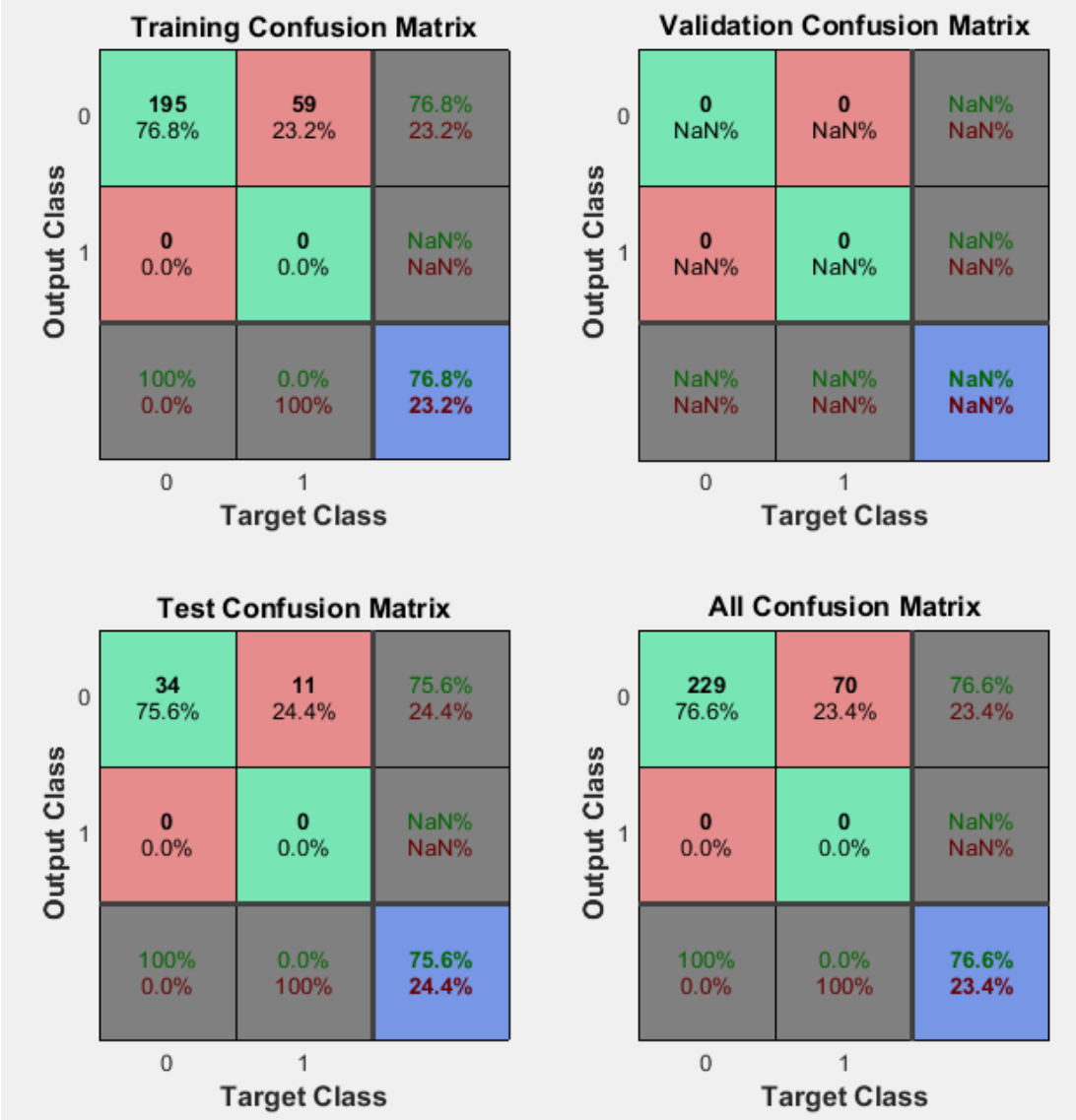

Figure 8 Confusion matrix for pattern recognition Bayesian regularization (PRN-BR) model

Figure 8 shows a representation of the confusion matrix for pattern recognition Bayesian regularization (PRN-BR) model. From the figure, it can be noticed that the confusion matrices of the test, validation and test data were all presented. In addition, a combined matrix of all the data was equally presented. The figure shows that after successfully training the Bayesian regularization (PRN-BR) model model, the training result gave a

comparatively low 76.8% correct classifications against an equally high 23.2% wrong classification. The trained model was equally tested and the test results showed that a comparatively low 75.6% of the test data were correctly classified, while an equally high 24.4% were wrongly classified. The model does not have entries for validation confusion matrix because the Bayesian regularization optimization algorithm does not utilize validation data. However, the combined confusion matrix for both training and test data showed that for the whole data set, a meager 76.6% of the data set was accurately classified using the available input variables, and a considerably high 23.4% of the data set was wrongly classified. This results corroborates the error histogram results obtained in Figure 7 above.

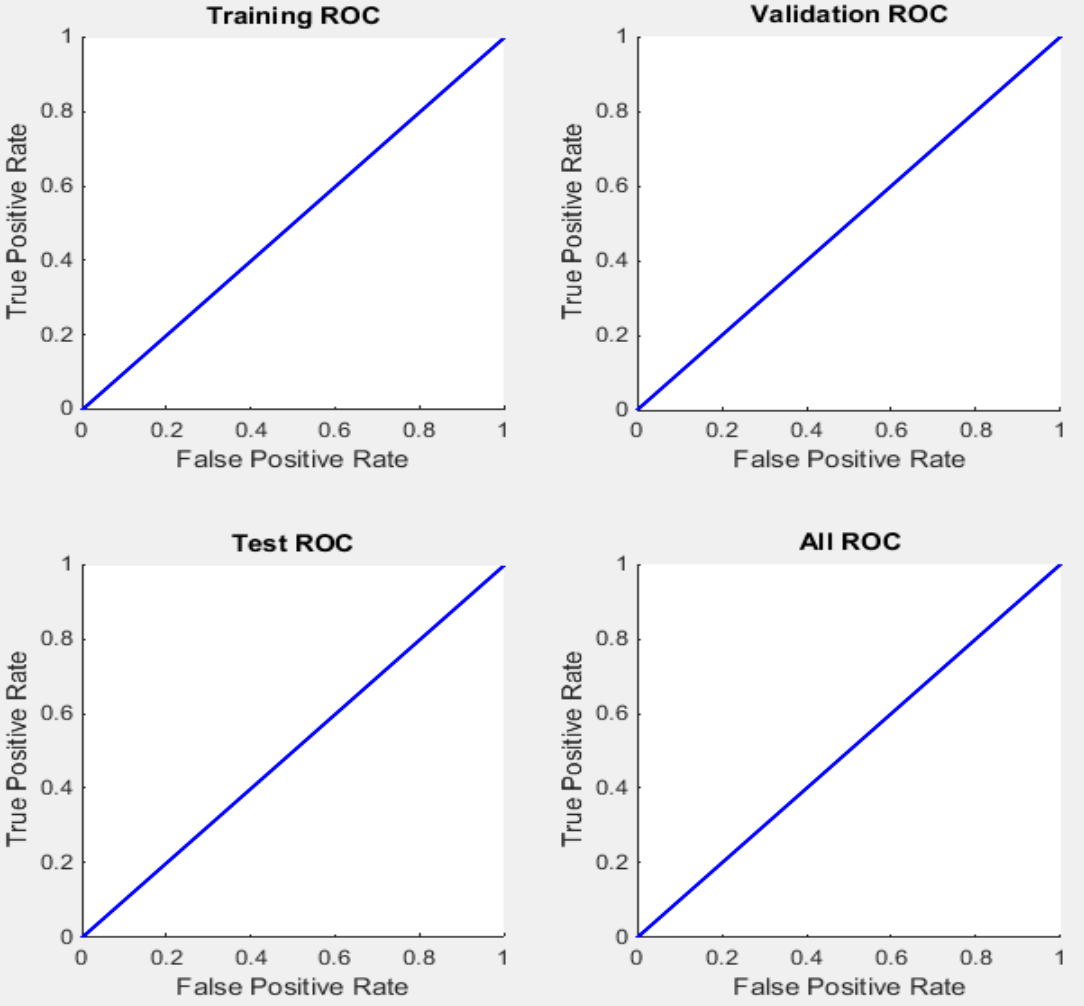

Figure 9 Receiver operating characteristic plot for pattern recognition Bayesian regularization (PRN-BR) model

Figure 9 shows the receiver operating characteristic plot for pattern recognition Bayesian regularization (PRN-BR) model. From the figure, it can be deduced that the receiver operating characteristic curves for training, validation, and test data all lied on the counterdiagonal of the square representing the receiver operating characteristic plot. This confirms the substantial inaccuracy of the characteristic plot for pattern recognition Bayesian regularization (PRN-BR) model for determining the failure potential of cranes. In addition, since the receiver operating characteristic plots shown above are actually on the counterdiagonal, it gives an area under the curve value of 0.5 for all the plots. This result further confirms that the Bayesian regularization (PRN-BR) model cannot be described as satisfactorily accurate.

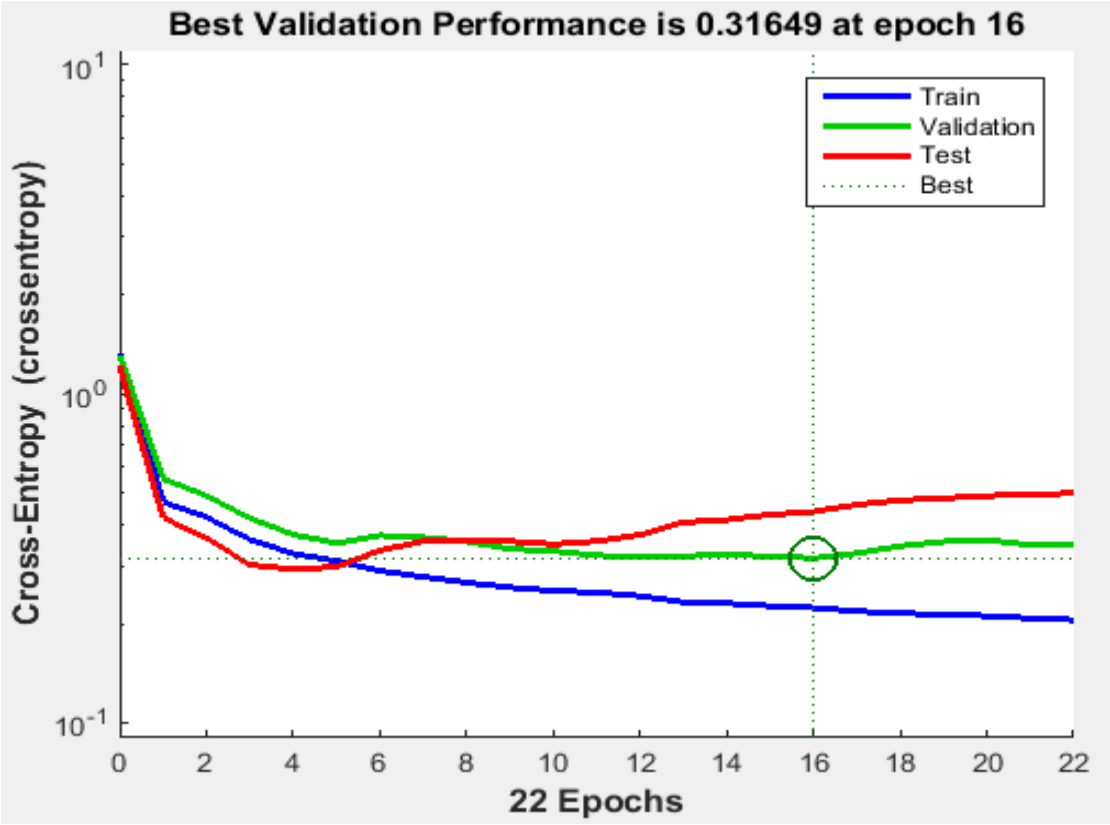

Figure 10 Training, validation, and testing curves for pattern recognition conjugate gradient (PRN-CGF) model

Figure 10 shows training, validation, and testing curves for pattern recognition conjugate gradient (PRN-CGF) model of cranes. From the graph, it can be seen that all errors including training, validations, and test errors consistently reduced until after the third epoch. The combined errors continued to reduce (rather slowly) such that it

recorded the combined training, validation and test error values at the sixteenth epoch. After this period, both validation and training errors started to increase, while the training error decreased further. The similarity in trend between the training, validations, and test curves depicts a successful training of the pattern recognition conjugate gradient (PRN-CGF) model and its potential use in assessing reliability and availability of cranes.

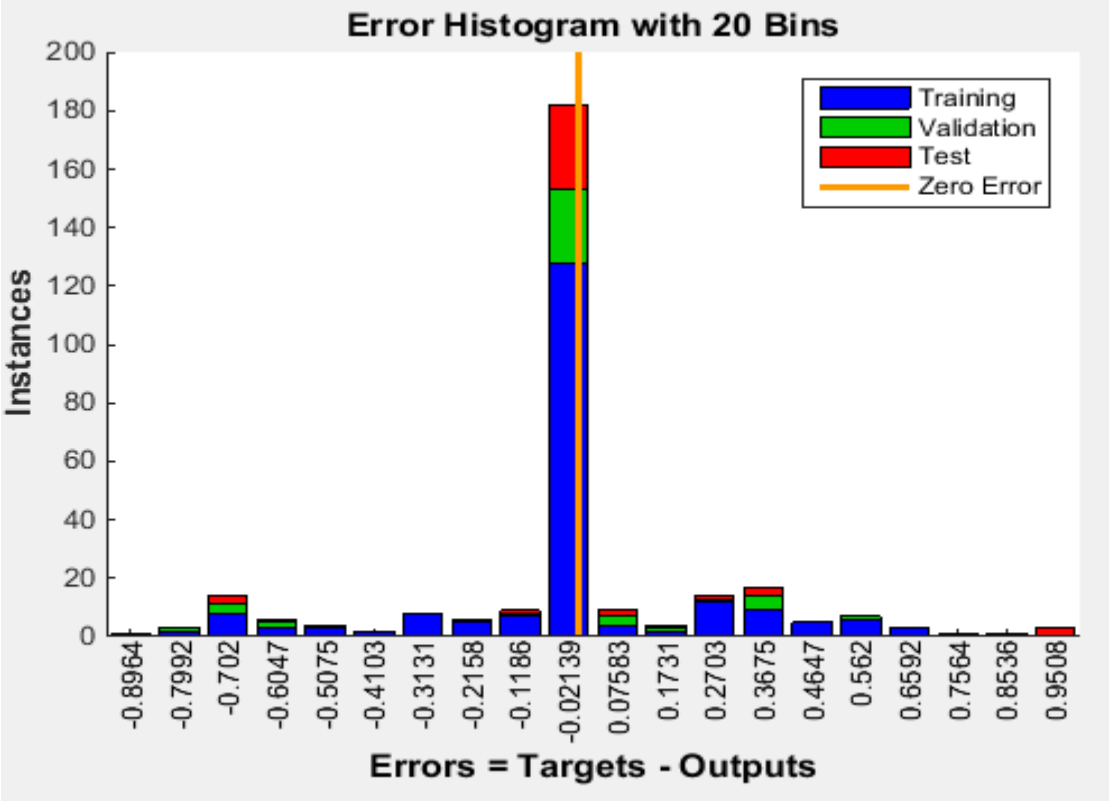

Figure 11 Prediction error histogram for pattern recognition conjugate gradient (PRN-CGF) model

Figure 11 shows the prediction error histogram for pattern recognition conjugate gradient (PRN-CGF) model for cranes. Here, the error is measured as the difference between the target variable values and the corresponding predicted outcome values. A good model should have errors congregated around the zero error mark, such that the highest histogram bars would be found around the zero error line. From the histogram, it can be deduced that a majority of the error instances were located around the zero error line, while the error instances recorded further away from the zero error line were actually few. This can be confirmed by the smaller number of error instances recorded

for errors further from the zero error line. Since the majority of the error instances were congregated around the zero error line, it demonstrates the ability of the pattern recognition conjugate gradient (PRN-CGF) model to predict failure of cranes. This result is obviously useful in assessing the reliability and availability of cranes.

Figure 12 Confusion matrix for pattern recognition conjugate gradient (PRN-CGF) model

Figure 12 shows a representation of the confusion matrix for pattern recognition conjugate gradient (PRN-CGF) model. From the figure, it can be noticed that the confusion matrices of the test, validation and test data were all presented. In addition, a combined matrix of all the data was equally presented. The figure shows that after successfully training the conjugate gradient (PRN-CGF) model, the training result gave

87.6% correct classifications against 12.4% wrong classification. When the trained model was validated using validation data, the validation results gave 82.2% correct predictions, with 17.8% wrong classifications. The trained model was equally tested and the test results showed that 84.4% of the test data were correctly classified, while 15.6% were wrongly classified. However, the combined confusion matrix showed that for the whole data set, 86.3% of the data set was accurately classified using the available input variables, but 13.7% of the data set was wrongly classified. The wrong classifications include both false positive and false negative classifications.

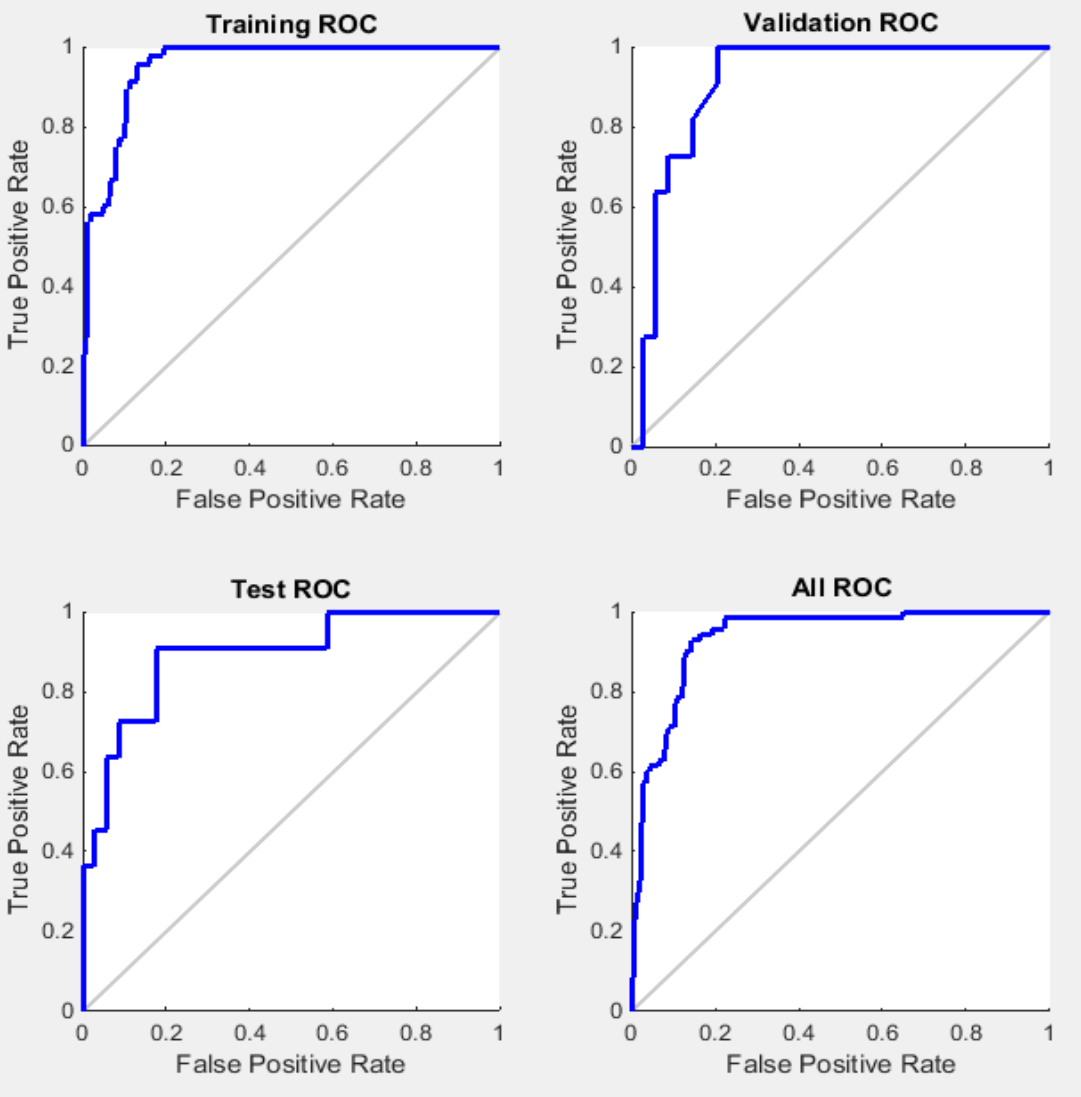

Figure 13 Receiver operating characteristic plot for pattern recognition conjugate gradient (PRN-CGF) model

Figure 13 shows the receiver operating characteristic plot for pattern recognition conjugate gradient (PRN-CGF) model. From the figure, it can be deduced that the receiver operating characteristic curves for training, validation, and test data all had slight deviations from the edges, which confirms the substantial accuracy of the characteristic plot for pattern recognition conjugate gradient (PRN-CGF) model for accurately classifying the failure potential of cranes. Another important metric derivable from the receiver operating characteristic plot is the area under the curve (AUC). Since the receiver operating characteristic plots shown above are considerably far from the counterdiagonal, the conjugate gradient (PRN-CGF) model can be described as satisfactorily accurate.

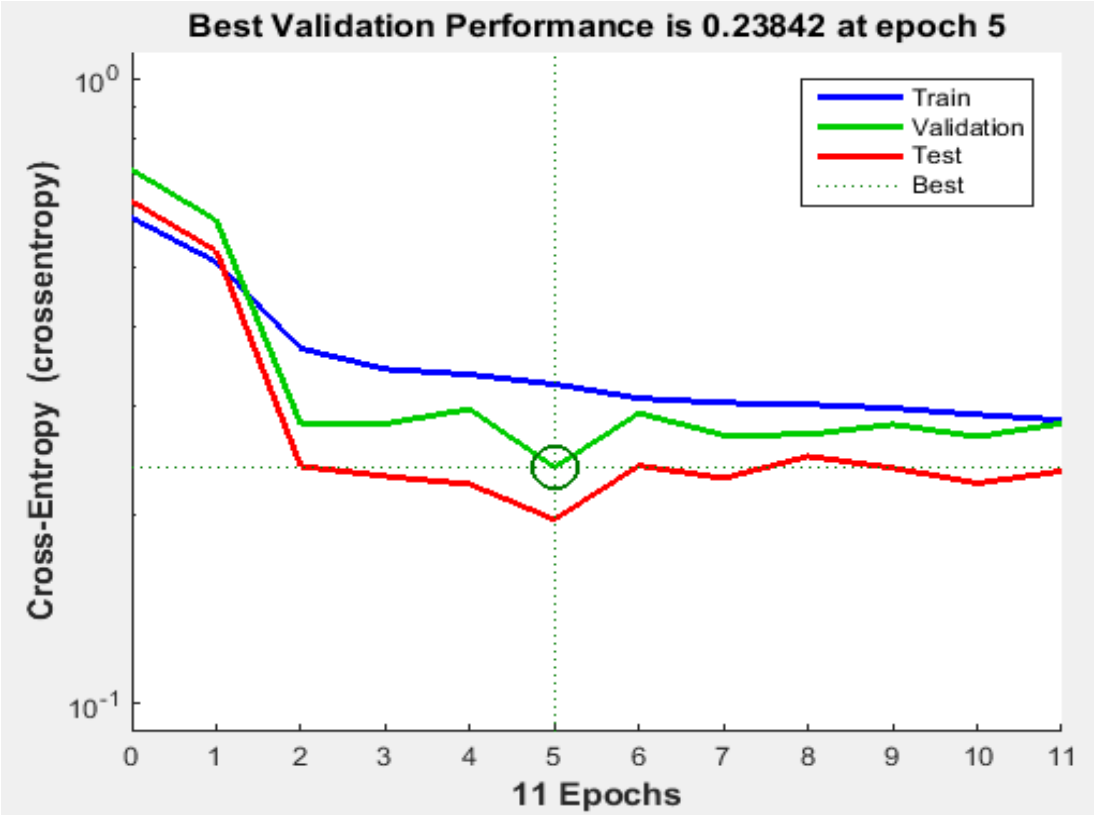

Figure 14 Training, validation, and testing curves for pattern recognition BFGS Quasi-Newton (PRN-BFG) model

Figure 14 shows training, validation, and testing curves for pattern recognition BFGS Quasi-Newton (PRN-BFG) model of cranes. From the graph, it can be seen that all errors including training, validations, and test errors consistently reduced until after the second epoch. The combined errors continued to reduce rather slowly such that it recorded the

lowest combined training, validation and test error values occurred during the fifth epoch. After this period, both validation and training errors started to increase, while the training error decreased further as the number of epochs increased. The similarity in trend between the training, validations, and test curves depicts a successful training of the pattern recognition BFGS Quasi-Newton (PRN-BFG) model and its potential use in assessing the failure potential of cranes.

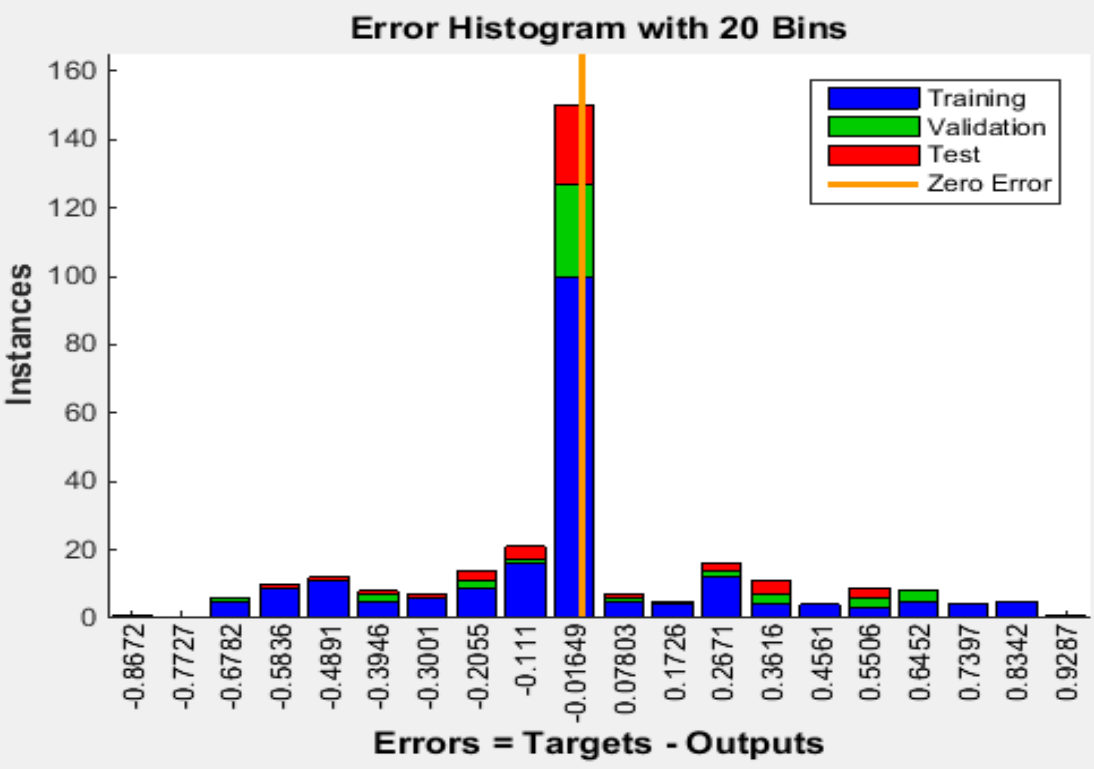

Figure 15 Prediction error histogram for pattern recognition BFGS Quasi-Newton (PRN-BFGS) model

Figure 15 shows the prediction error histogram for pattern recognition BFGS Quasi-Newton (PRN-BFGS) model for cranes. From the histogram, it can be deduced that a majority of the error instances were located around the zero error line, while the error instances recorded further away from the zero error line were actually few. This can be confirmed by the smaller number of error instances recorded for errors further from the zero error line. Since the majority of the error instances were congregated around the zero error line, it demonstrates the ability of the pattern recognition BFGS Quasi-Newton (PRN-BFGS) model to predict failure of cranes. This result is obviously useful in assessing the failure potential of cranes.

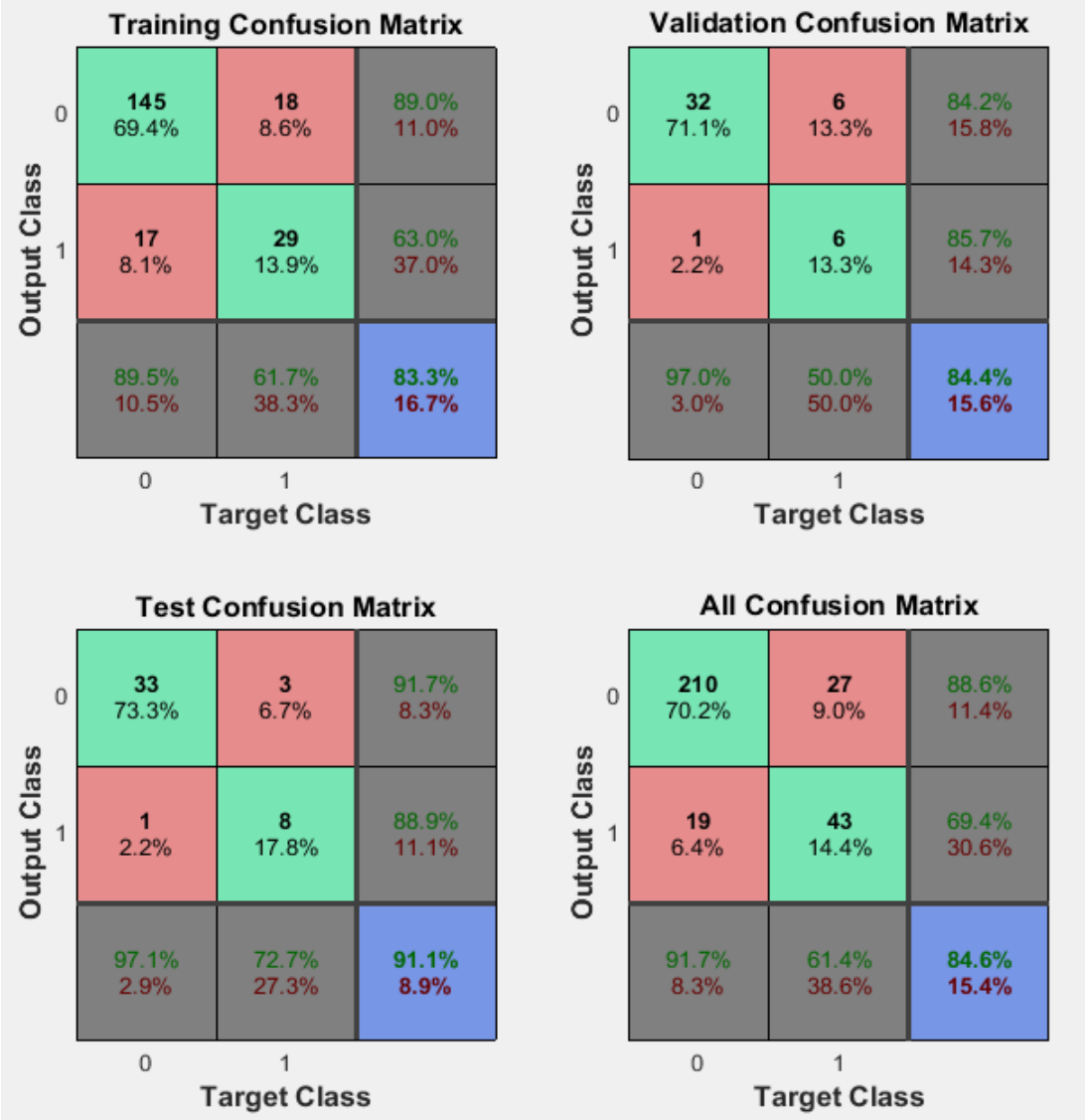

Figure 16 Confusion matrix for pattern recognition BFGS Quasi-Newton (PRN-BFGS) model

Figure 16 shows a representation of the confusion matrix for pattern recognition BFGS Quasi-Newton (PRN-BFGS) model. From the figure, it can be noticed that the confusion matrices of the test, validation and test data were all presented. In addition, a combined matrix of all the data was equally presented. The figure shows that after successfully training the BFGS Quasi-Newton (PRN-BFGS) model, the training result gave 83.3% correct classifications against 16.7% wrong classification. When the trained model was validated using validation data, the validation results gave 84.4% correct predictions,

with 15.6% wrong classifications. The trained model was equally tested and the test results showed that an impressive 91.1% of the test data were correctly classified, while only 8.9% were wrongly classified. However, the combined confusion matrix showed that for the whole data set, 84.6% of the data set was accurately classified using the available input variables, but 15.4% of the data set was wrongly classified. The wrong classifications include both false positive and false negative classifications.

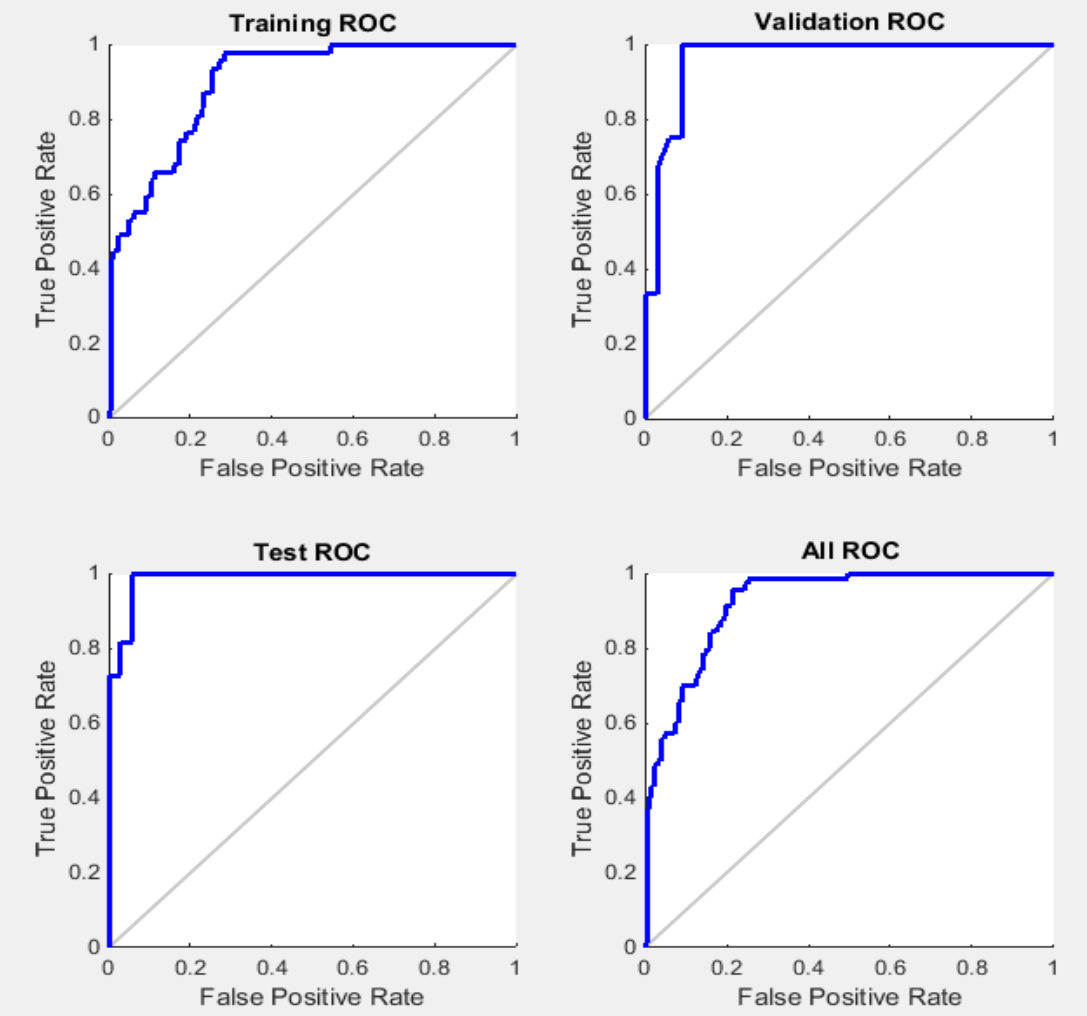

Figure 17 Receiver operating characteristic plot for pattern recognition BFGS Quasi-Newton (PRN-BFGS) model

Figure 17 shows the receiver operating characteristic plot for pattern recognition BFGS Quasi-Newton (PRN-BFGS) model. From the figure, it can be deduced that the receiver operating characteristic curves for training, validation, and test data all had slight

deviations from the edges, which confirms the substantial accuracy of the characteristic plot for pattern recognition BFGS Quasi-Newton (PRN-BFGS) model for accurately classifying the failure potential of cranes. Another important metric derivable from the receiver operating characteristic plot is the area under the curve (AUC). Since the receiver operating characteristic plots shown above are considerably far from the counterdiagonal, the BFGS Quasi-Newton (PRN-BFGS) model can be described as satisfactorily accurate.

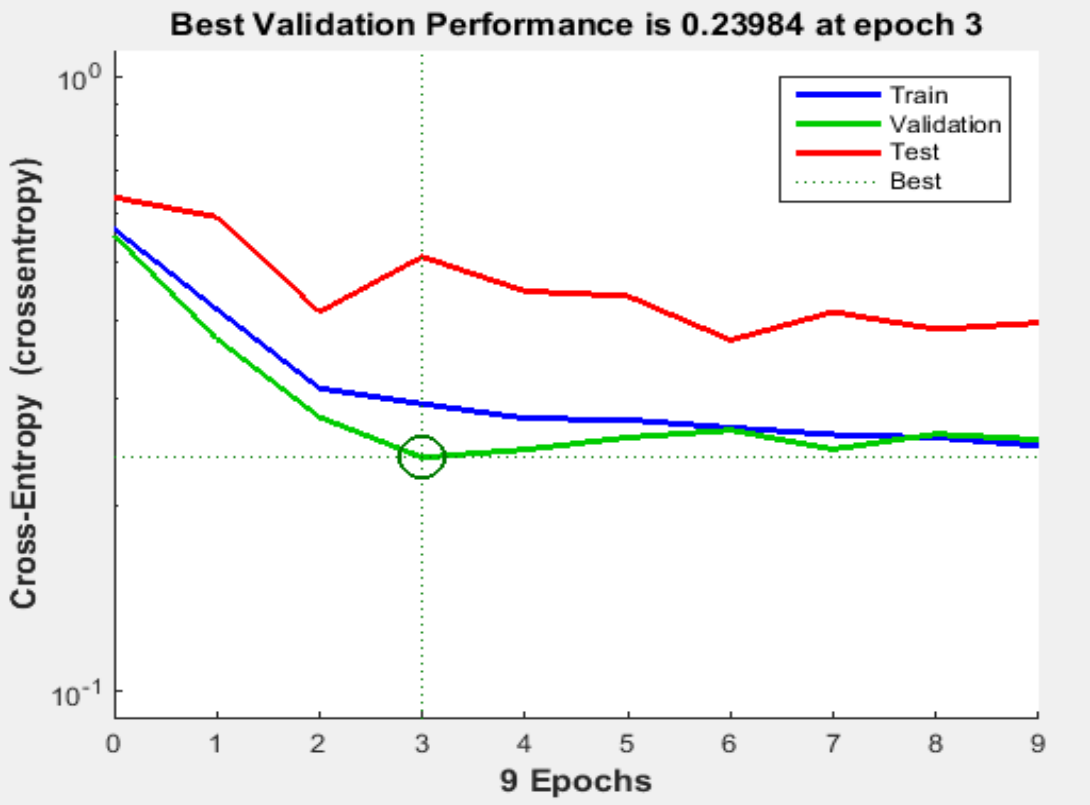

Figure 18 Training, validation, and testing curves for pattern recognition one step secant (PRN-OSS) model

Figure 18 shows training, validation, and testing curves for pattern recognition one step secant (PRN-OSS) model of cranes. From the graph, it can be seen that all errors including training, validations, and test errors consistently reduced until after the third epoch. The third epoch also serves as the epoch with the lowest combined errors, which represents an optimal model. After this period, both validation and training errors started to increase, while the training error decreased further. The similarity in trend between the training, validations, and test curves depicts a successful training of the

pattern recognition one step secant (PRN-OSS) model and its potential use in assessing reliability and availability of cranes.

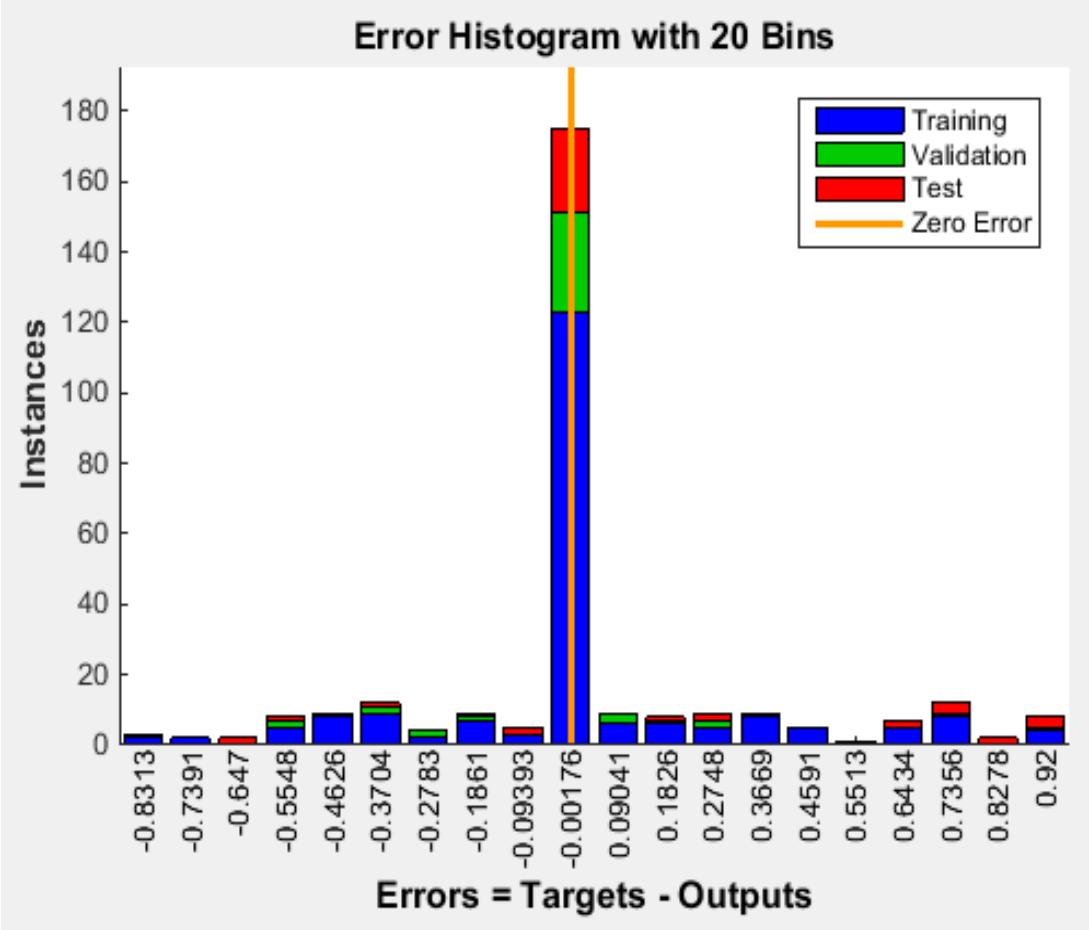

Figure 19 Prediction error histogram for pattern recognition one step secant (PRN-OSS) model

Figure 19 shows the prediction error histogram for pattern recognition one step secant (PRN-OSS) model for cranes. From the histogram, it can be deduced that a majority of the error instances were located around the zero error line, while the error instances recorded further away from the zero error line were actually few. This can be confirmed by the smaller number of error instances recorded for errors further from the zero error line. Since the majority of the error instances were congregated around the zero error line, it demonstrates the ability of the pattern recognition one step secant (PRN-OSS) model to predict failure of cranes. This result is obviously useful in assessing the reliability and availability of cranes.

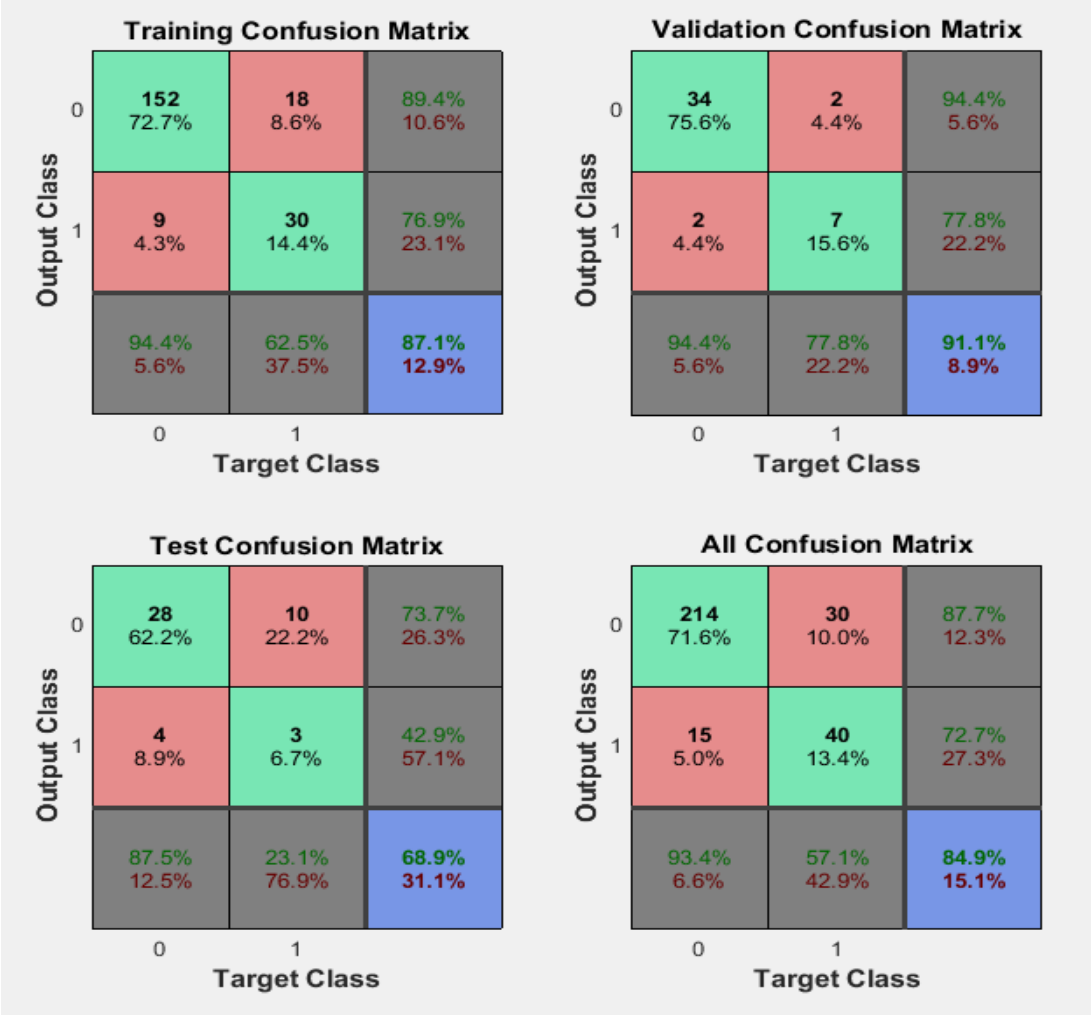

Figure 20 Confusion matrix for pattern recognition one step secant (PRN-OSS) model

Figure 20 shows a representation of the confusion matrix for pattern recognition one step secant (PRN-OSS) model. From the figure, it can be noticed that the confusion matrices of the test, validation and test data were all presented. In addition, a combined matrix of all the data was equally presented. The figure shows that after successfully training the one step secant (PRN-OSS) model, the training result gave 87.1% correct classifications against 12.9% wrong classifications. When the trained model was validated using validation data, the validation results gave an impressive 91.1% correct predictions, with a considerably low 8.9% wrong classifications. The trained model was equally tested and the test results showed that a rather small 68.9% of the test data were correctly classified, while a considerably high 31.3% were wrongly classified. However, the combined confusion matrix showed that for the whole data set, 84.9% of the data set

was accurately classified using the available input variables, but 15.1% of the data set was wrongly classified. The wrong classifications include both false positive and false negative classifications.

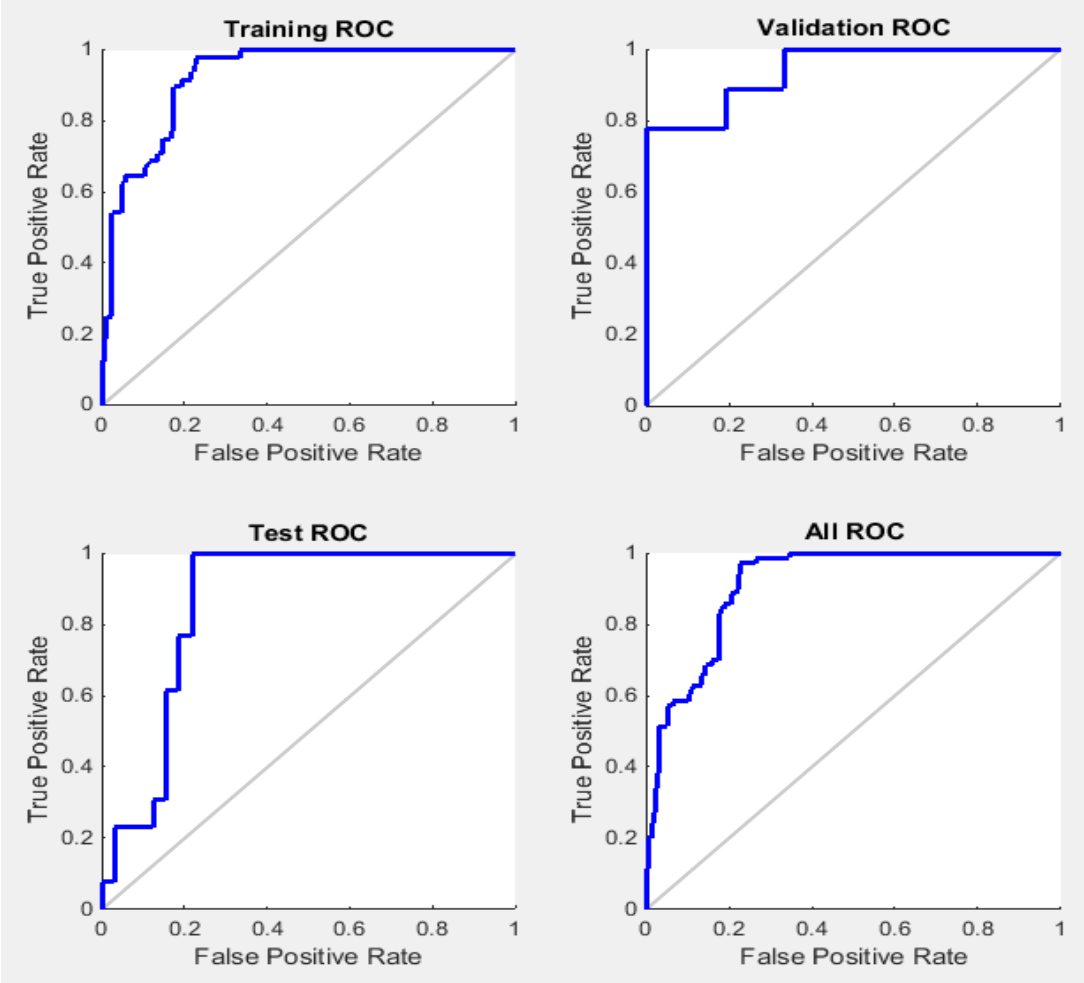

Figure 21 Receiver operating characteristic plot for pattern recognition one step secant (PRN-OSS) model

Figure 21 shows the receiver operating characteristic plot for pattern recognition one step secant (PRN-OSS) model. From the figure, it can be deduced that the receiver operating characteristic curves for training, validation, and test data all had slight deviations from the edges, which confirms the substantial accuracy of the characteristic plot for pattern recognition one step secant (PRN-OSS) model for accurately classifying the failure potential of cranes. Another important metric derivable from the receiver operating characteristic plot is the area under the curve (AUC). Since the receiver

operating characteristic plots shown above are considerably far from the counterdiagonal, the one step secant (PRN-OSS) model can be described as satisfactorily accurate.

Conclusion

New artificial neural network models were developed for assessing the reliability and availability of cranes. The machines utilized for data acquisition were from a nearby seaport. From the obtained results, the PRN-LMA models were the models that gave the highest prediction accuracy. In addition, the Bayesian regularization models (PRN-BR) gave the least prediction accuracy. Subsequently, PRN-CGF model, followed by PRN-LMA model predicted the highest number of failure days for the cranes, while both models gave the highest prediction accuracy for failure days. Obviously, the Bayesian regularization models gave the highest functional days predictions, but they could not correctly predict any of the failure days.

References

Ahmad, S., Badwelan, A., Ghaleb, A. M., Qamhan, A., Sharaf, M., Alatefi, M., & Moohialdin, A. (2018). Analyzing critical failures in a production process: Is industrial iot the solution? *Wireless Communications and Mobile Computing*, *2018*.

Amar, M. N., Ghahfarokhi, A.J., & Ng, C.S.W. (2022). Predicting wax deposition using robust machine learning techniques. *Petroleum, 8*(1), 167-173

Anandh, S., Bogeshwaran, K., Manikandan, G. N., Jamuna, P., & Sandhya, S. (2014). Failure analysis - watanabe model. *International Journal of ChemTech Research*, *6*(9), 4333–4336.

Chu, Z-Q., Sasanipour, J., Saeedi, M., Baghban A., & Mansoori, H. (2017). Modeling of wax deposition produced in the pipelines using PSO-ANFIS approach. *Petroleum Science and Technology, 35*(20), 1974-1981

Ezendiokwere N.E., Aimikhe V.J., Dosunmu A., & Joel O.F. (2021). Influence of depth on induced geo-mechanical, chemical, and thermal poromechanical effects. *Journal of Petroleum Exploration and Production Technology*, 11, 2917–2930

Fontes, C.H., & Pereira, O. (2016). Pattern recognition in multivariate time series: A case study applied to fault detection in a gas turbine. *Engineering Application of Artificial Intelligence, 49*, 10-18

Odeyar, P., Apel, O.B., Hall, R., Zon, B., Skrzypkowski, K. (2022). A review of reliability and fault analysis methods for heavy equipment and their components used in mining. *Energies, 15,* 6263.

Payette, M., & Abdul-Nour, G. (2023). Machine learning applications for reliability engineering: A review. *Sustainability, 15*(7), 6270

Serey, J., Alfaro, M., Fuertes, G., Vargas, M., Duran, C., Ternero, R., Rivera, R., & Sabattin, J. (2023). Pattern recognition and deep learning technologies, enablers of industry 4.0, and their role in engineering research. *Symmetry, 15*(2), 535

Soualhi, M., Nguyen, K.T.P., & Medjaher, K. (2020). Pattern recognition method of fault diagnostics based on a new health indicator for smart manufacturing. *Mechanical Systems and Signal Processing, 142*, 106680