Failure-Rate Prediction for De Havilland Dash-8 Tires Employing Neural-Network Technique

Ahmed Z. Al-Garni,* Ahmad Jamal,† Abid M. Ahmad,‡ Abdullah M. Al-Garni,§ and Mueyyet Tozan¶
King Fahd University of Petroleum and Minerals, Dhahran 31261, Saudi Arabia

An artificial neural-network model for predicting the failure rate of De Havilland Dash-8 airplane tires utilizing the two-layered feedforward back-propagation algorithm as a learning rule is developed. The inputs to the neural network are independent variables, and the output is the failure rate of the tires. Six years of data are used for model building and validation. Model validation, which reflects the suitability of the model for future prediction, is performed by comparing the predictions of the model with that of the Weibull regression model. The results show that the failure rate predicted by the artificial neural network more closely agrees with the actual data than the failure rate predicted by the Weibull model.

Nomenclature

C	=	intercept
d	=	integer, $1 \leq d \leq m$
$F(t)$	=	failure rate at time t
$f\,(\text{net})$	=	log-sigmoid function
i	=	integer, $0 \leq i \leq N'$
j	=	integer, $1 \leq j \leq k$
k	=	integer, $m \leq k \leq N + n$
k'	=	number, $0.65 < k' < 1$
l	=	number of landings
m	=	number of inputs to the neural network
m'	=	slope of a straight line
N	=	number of neurons in neural network
N'	=	number of observations
n	=	number of outputs to the neural network
O_s	=	outputs from the neural network, s varies from 1 to n
$O(t)$	=	$O_s(t)$
P_{hg}	=	hth tire of gth airplane
$R(t)$	=	reliability, $1 - F(t)$
$T(t)$	=	time beyond a given time, $T > t$
t	=	flight operational time
t_i	=	flight operational time, at the observation
$t_{\min}$	=	minimum time t
t_o	=	minimum guaranteed life of the tire
t_r	=	cumulative contact time on the runway
W	=	weight matrix for the hth tire of the gth airplane; h varies from 1 to 4
W_{kj}	=	element of the weight matrix
w	=	independent variable in regression
X_d	=	input to the neural network, d varies from 1 to m
x_j	=	normalized X_d
x_k	=	activation level of the neurons
Z	=	dependent variable in regression
β, η	=	parameters of Weibull model
$\lambda(t)$	=	instantaneous failure rate of the tires

Introduction

TIRES of airplanes, like tires of automobiles, are subjected to a number of wear-out processes, for example, uniform wear, accelerated wear at certain spots, microchipping, localized tire deformation, and so forth. In the case of airplanes, when the tires are in contact with the runway on landing, the conditions of wear are far more severe than the corresponding conditions in automobiles on highways. In the case of airplanes, the loads are not as uniform, there is a variety of shock loads, and a severe load spectrum is generated that can cause accelerated wear. Because tires are important aircraft components and the safety of an aircraft greatly depends on the reliability of its tires, their periodic monitoring and preventive maintenance are essential measures to increase aircraft reliability and are crucial for safe takeoff and landing. Tire life is defined by the wear limits set by controlling aviation agencies. When the tire damage due to wear-out processes reaches this critical limit, the tire is considered to have failed. The time to reach this critical manifestation of wear can be measured either by associated flight time or in terms of number of landings. It can also be written as $t \propto t_r$ and $t \propto l$ where t is the flight operational time, t_r is the time that the airplane tires are in contact with runway, and l is the number of landings. The tire life is not a fixed value but rather a random quantity, which is determined by t, bounded by $t_o < t < \infty$, where t_o is the minimum expected life.

Accurately modeling the failure rate of airplane tires is of prime importance. This model should accurately predict the time of failure to avoid crashes during landing or takeoff. Various conventional regression models can be developed to model this failure rate. However, much interest has recently been focused on the application of artificial neural-network (ANN) modeling,[1−7] and it has been shown that the ANN performs better than the regression models.

The ability of ANN to model multivariate problems without making complex dependency assumptions among the input variables is an advantage over the statistical models. Moreover, ANN extracts the implicit nonlinear relationships among the input variables through a learning process from the training data set. These features make neural networks good alternatives to conventional regression techniques. The objective of the present paper is to build an ANN model that predicts the failure rate of De Havilland Dash-8 airplane tires and to compare it with the Weibull regression model. The rest of the paper is organized as follows: in the second section, an introduction to neural networks and feedforward back-propagation (BP) networks is presented; in the third section, the failure time data for the tires is presented; in the fourth section, a regression model (the Weibull model) and the neural-network model are developed;

a comparison of the results obtained from the Weibull and neural-network models with the real data is presented in the fifth section; and the sixth section concludes the paper.

Artificial Neural Network (ANN)

Introduction to ANN

An ANN is an information-processing system that has certain performance characteristics in common with biological neural networks. ANNs are computational systems that mimic the biological neural networks of the mammalian brain. The human brain contains about 100 billion neurons (neuron cells), interconnected in a complex manner via synapses (junctions between axons and dendrites), thus constituting a network. An ANN is a collection of neurons that are arranged in specific formations. Neurons are grouped into layers. A multilayer network usually consists of an input layer, one or more hidden layers, and an output layer. The number of neurons in the input layer corresponds to the number of parameters that are presented to the network as inputs. The same is true for the output layer. Neural-network analysis is not limited to a single output, and neural nets can be trained to build neuron models with multiple outputs. The neurons in the hidden layer or layers are responsible primarily for feature extraction. They provide increased dimensionality and accommodate such tasks as classification and prediction.

Recent Developments

The basic idea of the neural network was initiated by MuCullock and Pittis.[8] They studied the ability of a model neuron to interconnect several basic components. Later, Rosenblatt[9] coined the name "perceptron" and devised an architecture that received much attention. However, a rigorous analysis of the perceptron, made by Minsky and Papert,[10] demonstrated that it had certain limitations. This almost brought research in this area to a halt, but later the work of Hopfield[11] revived the interest in neural networks. Since then, a variety of ANN algorithms have been proposed and used in recent years. Presently, research on ANNs is being performed in a great number of disciplines, ranging from neurobiology psychology to engineering sciences.

ANN Working Methodology

A typical ANN operation starts with the training stage. This stage is conducted using various training data sets that include the respective inputs and the corresponding desired outputs. The initial network connection weights are set to equal small random numbers. After the network is properly trained, the recall stage starts. In this stage, a set of test data is applied to the network. Afterward, the performance of the network is analyzed. This performance depends on various factors such as the statistical soundness of the training data set, the structure and size of the network, the initial network weights, the learning strategy, and input variables.

Back-Propagation Algorithm

Most of the currently used ANNs for process estimation or prediction problems are layered feedforward neural networks (FNNs), also called multilayer perceptrons. The BP algorithm is the most widely used learning procedure for neural networks.[12,13] It is in fact a gradient descent-error-correcting algorithm. Before beginning training, some small random numbers are usually used to initialize each weight on each connection. BP requires preexisting training patterns, and involves a forward-propagation step followed by a BP step. The forward-propagation step begins by sending the input signals through the nodes of each layer. A nonlinear activation function, called the sigmoidal function, is usually used at each node for the transformation of the incoming signals to an output signal. This process repeats until the signals reach the output layer and an output value is calculated. The BP step calculates the error by comparing the calculated and target outputs. New sets of weights are iteratively calculated by modifying the existing weights based on these error values until a minimum overall error, or global error, is obtained. The mean-square error (MSE) is usually used as a measure of the

global error.[14] The following logic is assumed in BP[13]:

$$x_j = \text{normalized } X_d \qquad 1 < d \leq M \qquad (1)$$

$$\text{net}_k = \sum_{j=1}^{k-1} W_{kj} x_j \qquad m \leq k \leq N+n \qquad (2)$$

$$x_k = f(\text{net}_k) \qquad m < k \leq N+n \qquad (3)$$

$$O_s = x_{N+s} \qquad 1 \leq s \leq n \qquad (4)$$

$$f(\text{net}) = \frac{1}{1 + e^{-\text{net}_k}} \qquad (5)$$

where m is the number of inputs to the network, n is the number of outputs of the neural network, and X_d represents the actual inputs to the network (which have to be normalized and then initially stored in x_i). The nonlinear activation function $f(\text{net})$ in Eq. (5) is a log-sigmoid function, and it depends on the desired output data range. N is a constant, which represents the number of intermediate neurons in the neural network. It can be any integer as long as it is not less than m. The value of $N+m$ determines how many neurons are there in the network (if we include the inputs as neurons). W is the weight matrix in each layer whose size depends on the number of neurons in the corresponding adjacent layers of neural network. The term x_k is called the "activation level" of the neuron, and O_s is the output from the neural network. The input and output to the neuron is given in Fig. 1. The significance of these equations is illustrated in Fig. 2, which shows the connection in the network.

There are $N+n$ circles, representing all the neurons in the network, including the input neurons. The first m circles are copies of the inputs $X_1, X_2, \ldots, X_m$; they are included as a part of the vector $\boldsymbol{x}$ only as a way of simplifying the notation. Every other neuron is the network such as neuron number k, which calculates net_k and x_k, takes input from every cell that precedes it in the network. Even the last output cell, which generates O_s, takes input from other output cells, such as the one whose output is O_{s-1}.
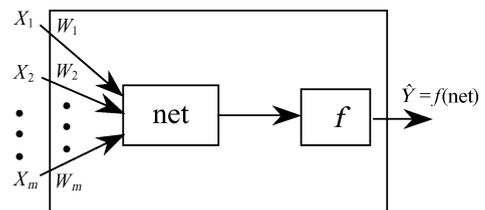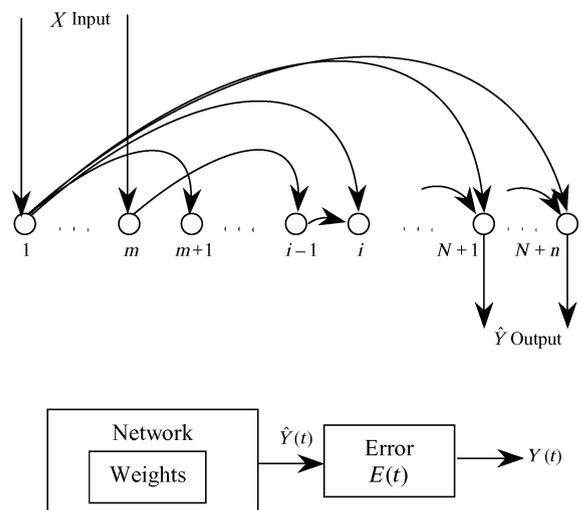

Fig. 1   Artificial neuron with activation function.
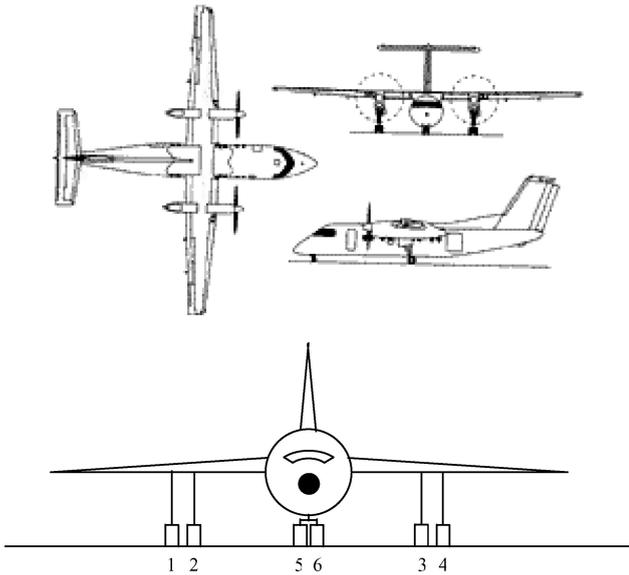

Fig. 2   Network design for BP.

Fig. 3 Airplane and its tires.

Tire-Failure Time Data

The data were collected from a local aviation facility in Saudi Arabia. The data represents the time-to-failure of tires for the De Havilland Dash-8 series over a period of 6 years for a fleet of three airplanes. These three airplanes have the registration numbers N724A, N725A, and N759A. Data was collected for all six tires of each airplane. In this type of aircraft (De Havilland Dash-8 series), there are six tires, two on the left, two on the right, and two in the front near the nose of the airplane. For convenience, we have named the three airplanes in serial order so that airplane N724A is A, N725A is B, and N759A is C. Tires are also numbered as 1 and 2 to the left, 3 and 4 to the right, and 5 and 6 in the front, as shown in Fig. 3. A tire of any of the three airplanes can be represented by P_{hg}; for example, P_{3A} refers to the third tire on the right of the airplane A, that is, N724A. Failure is defined whenever, at the inspection time, it is observed that the tire needs to be replaced according to the aviation standards being followed. The data, which is obtained from the log book of each airplane, are recorded in two forms: as flying time in hours between the replacements and as number of landings between the replacements. However, in the present study, flying time is used as an indicator of life of the tires.

Tire-Failure Prediction Models

Regression Model

Reliability of Tires in Terms of Flight Time

The reliability $R(t)$ of a tire characterizes the probability of its survival beyond a given time t; that is, $R(t) = P(T > t)$, and in general terms, it can be defined as[15,16]

$$R(t) = \exp\left[-\int_0^t \lambda(t)dt\right] \qquad (6)$$

where $\lambda(t)$ is the instantaneous failure rate of the tires and t is proportional to t_r, which in turn is proportional to l. Tires are subjected to an increasing failure rate as the operational time, that is, the number of landings, increases. Thus, the most suitable characterization on instantaneous tire failure rate will be described by a power-law function of time, so that

$$\lambda(t) = [\beta/(\eta - t_0)]\left[(t - t_0)/(\eta - t_0)^{\beta - 1}\right] \qquad (7)$$

where η is a scale parameter that expresses the characteristic life and β is a shape parameter of the model that determines the severity of the wear-out process. Using this power-law failure-rate model,

Eqs. (6) and (7) represent a well-known three-parameter Weibull reliability model, which can be written as follows:

$$R(t) = \exp\left\{-[(t - t_0)/(\eta - t_0)]^\beta\right\}, \qquad t > t_o \qquad (8)$$

where t is the random variable characterizing the life of the tire, $t_o < t < \infty$.

Fitting the Weibull Model Failure Data

To fit the data, the complementary function to the reliability function $R(t)$ is often used, which is also known as the cumulative function $F(t) = 1 - R(t)$ and defines $P(T > t)$. Thus, using Eq. (8), one can write

$$F(t) = 1 - \exp\left\{-[(t - t_0)/(\eta - t_0)]^\beta\right\}, \qquad t > t_o \qquad (9)$$

among various approaches used in fitting the Weibull model to the failure data. A procedure used by Ref. 16 is the most lucid, and it is easy to implement. This method linearizes the equation $F(t)$ as follows:

$$\ell n[1 - F(t)] = -[(t - t_0)/(\eta - t_0)]^\beta$$

$$\ell n(\ell n\{1/[1 - F(t)]\}) = \beta \ell n(t - t_0) - \beta \ell n(\eta - t_0) \qquad (10)$$

Now let

$$Z = \ell n(\ell n\{1/[1 - F(t)]\})$$

$$w = \ell n(t - t_0), \qquad m' = \beta, \qquad C = -\beta \ell n(\eta - t_0)$$

Equation (10) may now be put in the form

$$Z = m'w + C \qquad (11)$$

After arranging the failure data in ascending order, the probability distribution function can be substituted by its estimate, using the median rank formula[15]:

$$F(t_i) = i/(N' + 1), \qquad 0 \le i \le N' \qquad (12)$$

Linearized Eq. (11) can be fitted to the experimental data $F(t_i)$ vs $(t_i - t_0)$ for $i = 1, 1, 2, \ldots, N'$. By performing the linear-regression analysis using linearly transformed Eq. (11), the parameters β and η can be determined. This approach implies that t_0 is known. The value of t_0 is less than $t_0 = k't_{min}$, where $0.65 < k' < 1$. A starting point can be taken as $t_0 = 0.6 t_{min}$. If a straight-line fit is poor, then this value can be adjusted between $0.65 t_{min} - 0.99 t_{min}$ until a good fit is obtained.

A spreadsheet (MS Excel) was used to perform this analysis on the tires of all the three airplanes. Table 1 gives the complete analysis for P_{5A}. The regression output for this analysis is presented in Table 2, which gives the values of the parameters of the Weibull model.

Neural-Network Model

In this section, an ANN is developed to model the failure rate of the tires. The input to the neural network is time in hours, and the output of the neural network is the failure rate corresponding to that time. Because the present study represents a dynamic system, which is one whose state varies with time, an autoregressive model that uses inputs corresponding to previous points in time can be used.[14] For such purpose, three cases are studied:

1) One input $m = 1$, one output $n = 1$, and six intermediate neurons $N = 6$.

2) Two inputs $m = 2$, one output $n = 1$, and six intermediate neurons $N = 6$.

3) Three inputs $m = 3$, one output $n = 1$, and six intermediate neurons $N = 6$.

For the second and third cases, one and two previous time inputs are taken, respectively, for each time input. The activation function (log-sigmoid function) takes the input and squashes the output into the range 0 to 1 as shown in Fig. 4. This function is commonly used

Table 1   Regression analysis of the failure data (h) of P_{5A} for Dash-8

i	t_i, h	$X_d = (t_i - t_0)$	$\ell n(t - t_0)$	$F(t_i) = i/(N' + 1)$	$\ell n(\ell n\{1/[1 - F(t_i)]\})$	Regression
1	57	22.8	3.1268	0.0357	−3.3141	−4.3211
2	80	45.8	3.8243	0.0714	−2.6022	−2.3957
3	91	56.8	4.0395	0.1071	−2.1775	−1.8015
4	98	63.8	4.1558	0.1429	−1.8698	−1.4807
5	102	67.8	4.2166	0.1786	−1.6260	−1.3128
6	108	73.8	4.3014	0.2143	−1.4223	−1.0787
7	116	81.8	4.4043	0.2500	−1.2459	−0.7946
8	117	82.8	4.4164	0.2857	−1.0892	−0.7611
9	122	87.8	4.4751	0.3214	−0.9474	−0.5992
10	122	87.8	4.4751	0.3571	−0.8168	−0.5992
11	123	88.8	4.4864	0.3929	−0.6952	−0.5680
12	125	90.8	4.5087	0.4286	−0.5805	−0.5065
13	128	93.8	4.5412	0.4643	−0.4714	−0.4168
14	131	96.8	4.5726	0.5000	−0.3665	−0.3299
15	134	99.8	4.6032	0.5357	−0.2649	−0.2456
16	134	99.8	4.6032	0.5714	−0.1657	−0.2456
17	136	101.8	4.6230	0.6071	−0.0679	−0.1908
18	137	102.8	4.6328	0.6429	0.0292	−0.1639
19	137	102.8	4.6328	0.6786	0.1266	−0.1639
20	148	113.8	4.7344	0.7143	0.2254	0.1168
21	155	120.8	4.7941	0.7500	0.3266	0.2815
22	155	120.8	4.7941	0.7857	0.4321	0.2815
23	167	132.8	4.8888	0.8214	0.5439	0.5430
24	168	133.8	4.8963	0.8571	0.6657	0.5637
25	168	133.8	4.8963	0.8929	0.8036	0.5637
26	169	134.8	4.9038	0.9286	0.9704	0.5842
27	172	137.8	4.9258	0.9643	1.2036	0.6450

Table 2   Regression output for failure data, h, for P_{3A}

Parameter	Value
Constant C	−12.9523
Std. error	0.3335
R squared	0.9150
No. of observations N'	27
Degree of freedom	25
Std. error of coefficient	0.1683
β	2.7604
η	143.2873
t_0	34.2

Fig. 4   Log-sigmoid function.

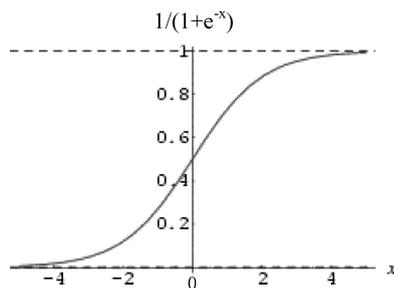**Fig. 5   Comparison of failure rate $F(t)$ against time, predicted by using one-, two-, and three- input neural networks.**

in multilayer networks that are trained using the BP algorithm, and also this function is differentiable. The predicted failure rate can be found by using the forward-pass calculation, Eqs. (1)–(4). The training of the neural network is carried out using the BP technique.[14] The objective is to minimize the sum squared error given by

$$\text{error} = \sum [F(t) - O(t)]^2$$

where $F(t)$ is the actual failure rate, and $O(t) = O_s(t)$ is the final output that is calculated from the neural-network model. The number of passes is usually set to a high number. The initial error is high because the initial weights were assigned randomly. As the network is trained, the error decreases and converges to a minimum value.
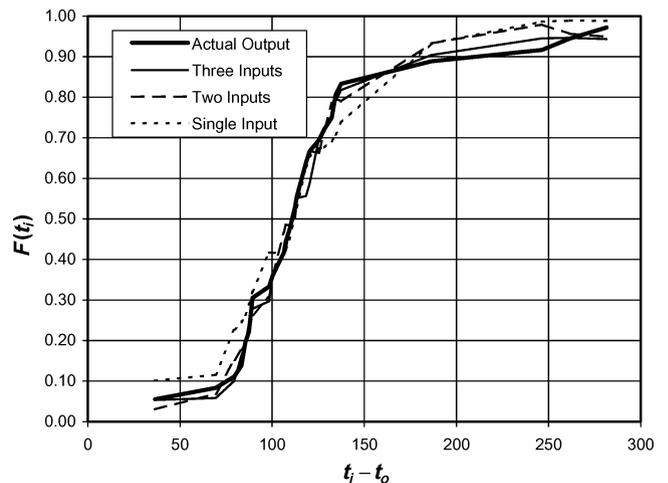
The comparison of all three cases is presented in Fig. 5. The average percentage difference of the failure rate with that of the actual data is found to be 19.84%, 8.56%, and 5.66% for neural networks having one, two, and three inputs, respectively. Moreover, with informal visual inspection of Fig. 5, it can be observed that the neural-network model with three inputs is closer to the actual data than other cases (i.e., one and two inputs). Therefore, the three-input neural-network model has been adopted for the present study.

The working flow chart of the entire neural-network analysis is shown in Fig. 6, and the neural-network architecture employed is shown in Fig. 7. The sizes of the weight matrices W_1, W_2, and W_3 are 2×4, 4×3 and 1×5, respectively. Training the BP network requires the following:

1) Select the training pair from the training set; apply the input vector to the network input terminal.

2) Calculate the output of the network [using Eqs. (1)–(4), forward pass].

3) Calculate the error (the difference between the network output and desired output).

4) Adjust the weights of the network in a way that minimizes the error. It would quicken the process if the weights not being used are zeroed out.

5) Repeat steps 1 through 4 for each vector in the training set until the error for the entire set is acceptably low. Steps 1 and 2 constitute the forward, and steps 3 and 4 are the reverse passes.

These steps can easily be understood by the flow chart shown in Fig. 8.

Model Adequacy and Comparison

Evaluating the model adequacy is an important part of any model-building problem. The idea is to examine whether the fitted model is in agreement with the observed data. An informal visual assessment method has been adopted. Figure 9a shows a comparison between the actual and the predicted failure rate for P_{1A}, using a neural network and the Weibull model. For the performance evaluation of the neural-network model and the regression model, a predictive accuracy of the two models for the given tire data has been compared. Figures 9a through 9f show the actual failure rate, the predicted failure rate from the neural-network model, and the predicted failure

rate from the Weibull regression model for the six tires of the first airplane, that is, 724A. In general, it is observed from the results in Figs. 9a–9f, 10a–10c, and 11a–11c that the neural-network model predicts the failure rate better than the Weibull regression model. The results can be considered in two groups (groups A and B). Group A is when the rate of $F(t)$, with respect to $(t_i - t_0)$, is large at the earlier stage or becomes large after a short time, and/or if there is no major change in the rate of $F(t)$ that takes place and remains that way for a longer time, for example, Fig. 9a for the first tire of the first airplane, P_{1A}. Group B is when the rate of $F(t)$, with respect to $(t_i - t_0)$, at the earlier stage is small and remains small for a long time, and/or if there is a major change in the rate of $F(t)$ that takes place and remains that way for a long time, for example, Fig. 10b for the fifth tire of the second airplane, P_{5B}.

Group A can be considered as ten tires: P_{1A}, P_{3A}, P_{4A}, P_{1B}, P_{3B}, P_{4B}, P_{5B}, P_{6B}, P_{1C}, and P_{5C}. Group B can be considered as eight tires: P_{2A}, P_{5A}, P_{6A}, P_{2B}, P_{2C}, P_{3C}, P_{4C}, and P_{6C}. The first airplane is taken as a typical case as shown in Figs. 9a through 9f for tires P_{1A}, P_{2A}, P_{3A}, P_{4A}, P_{5A}, and P_{6A}, respectively. For the other airplanes,
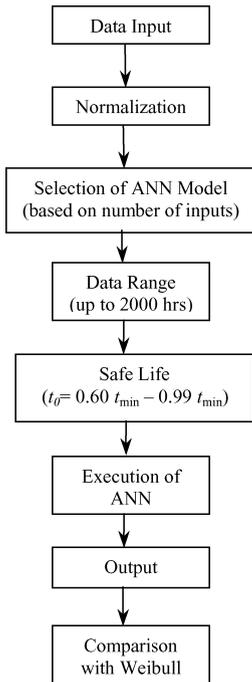

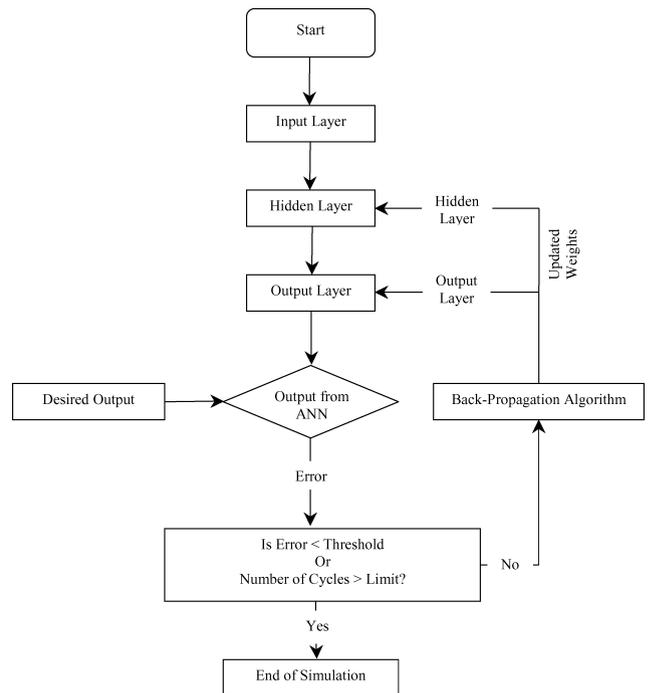Fig. 6 Working flow chart of neural-network analysis.

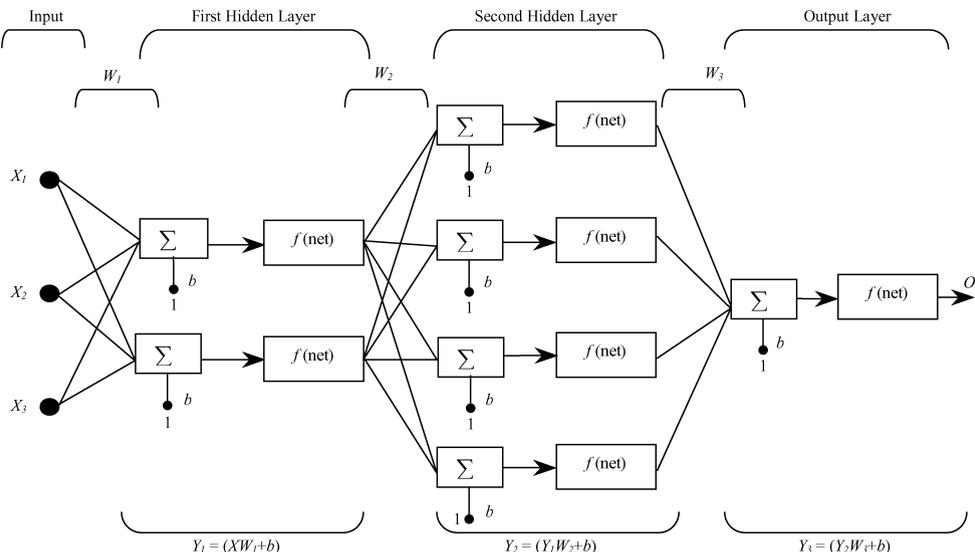Fig. 8 Flow chart of neural-network algorithm.
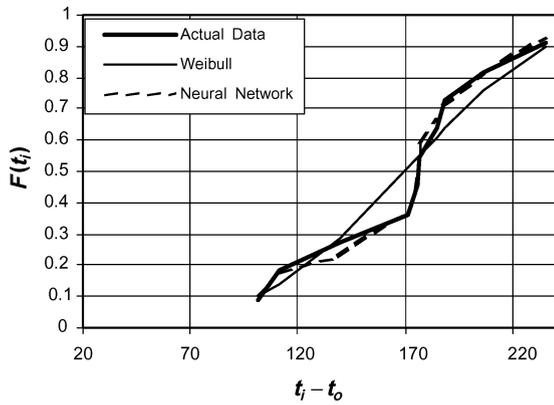

Fig. 7 Neural-network architecture.

Fig. 9a   Failure $F(t)$ for the Dash-8 tire P_{1A} vs failure data, h.
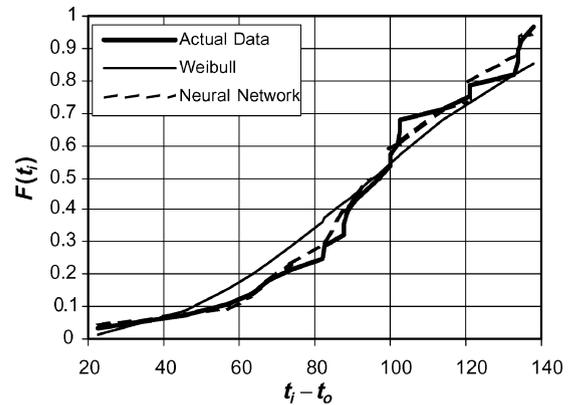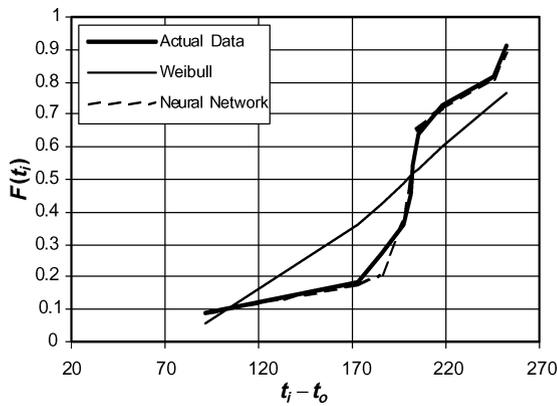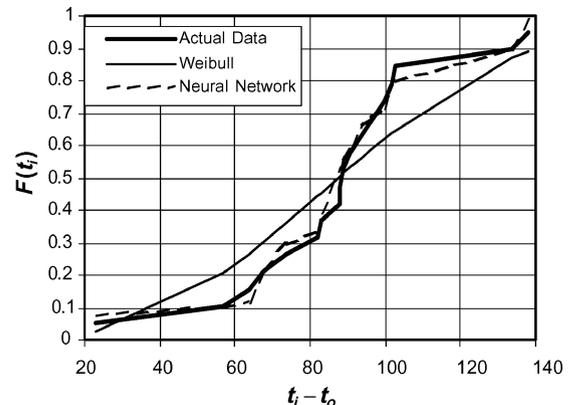

Fig. 9b   Failure $F(t)$ for the Dash-8 tire P_{2A} vs failure data, h.

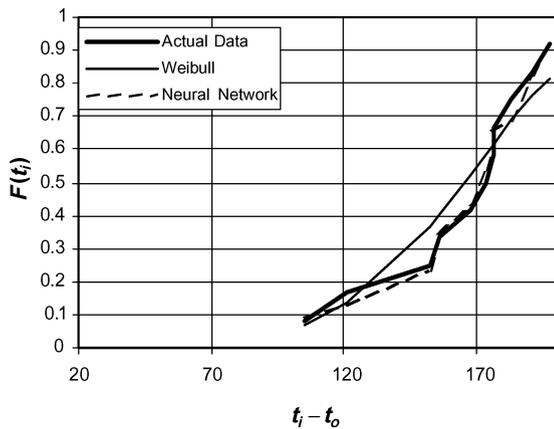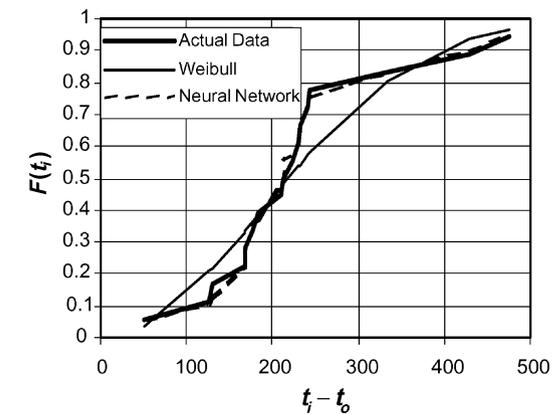**Fig. 9c   Failure $F(t)$ for the Dash-8 tire P_{3A} vs failure data, h.**

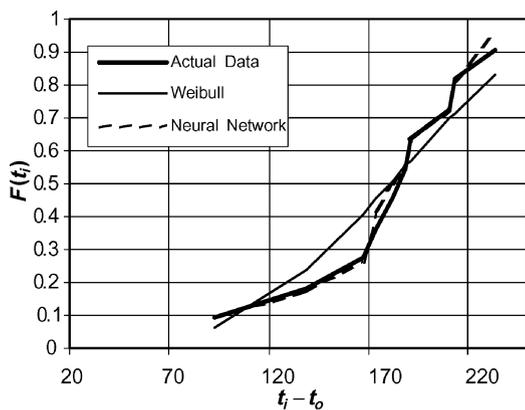**Fig. 9d   Failure $F(t)$ for the Dash-8 tire P_{4A} vs failure data, h.**

Fig. 9e   Failure $F(t)$ for the Dash-8 tire P_{5A} vs failure data, h.

Fig. 9f   Failure $F(t)$ for the Dash-8 tire P_{6A} vs failure data, h.

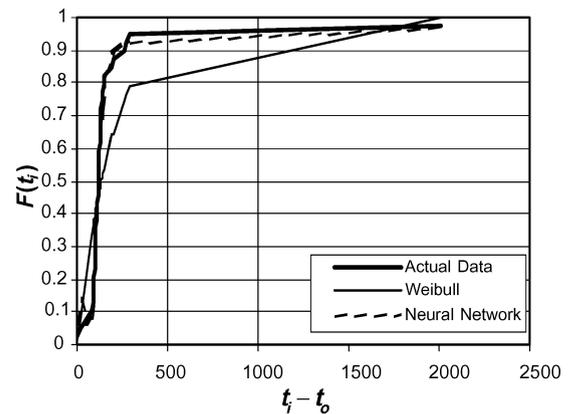**Fig. 10a   Failure $F(t)$ for the Dash-8 tire P_{1B} vs failure data, h.**

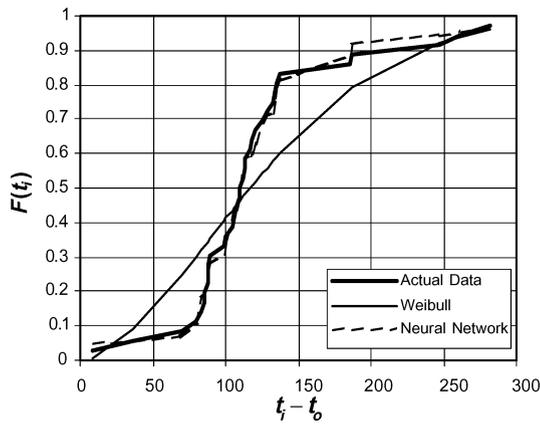**Fig. 10b   Failure $F(t)$ for the Dash-8 tire P_{5B} vs failure data, h.**

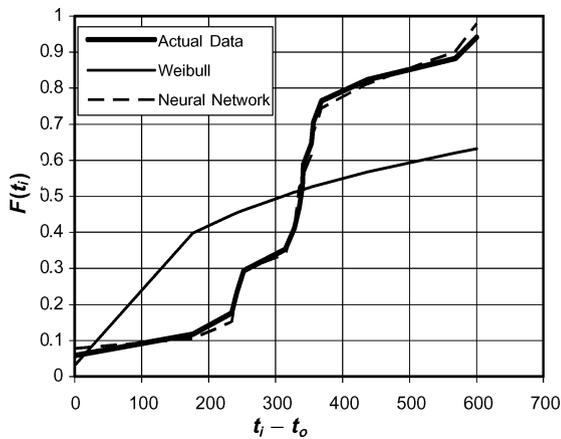**Fig. 10c Failure $F(t)$ for the Dash-8 tire P_{6B} vs failure data, h.**

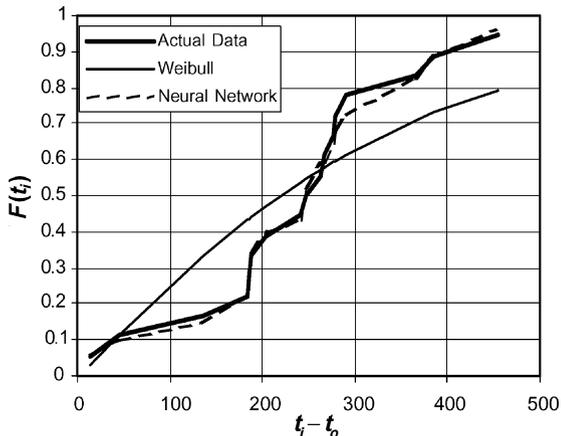**Fig. 11a Failure $F(t)$ for the Dash-8 tire P_{3C} vs failure data, h.**

Fig. 11b Failure $F(t)$ for the Dash-8 tire P_{4C} vs failure data, h.
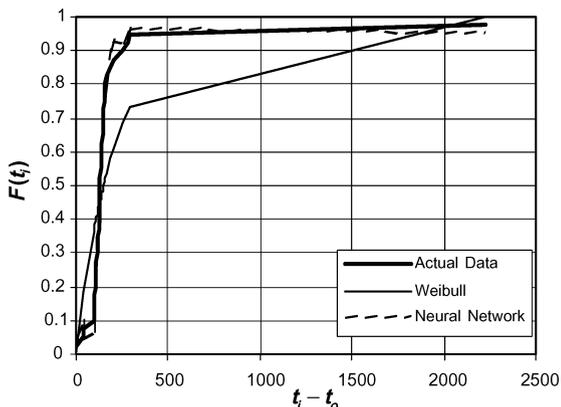

Fig. 11c Failure $F(t)$ for the Dash-8 tire P_{5C} vs failure data, h.

representative results are shown in Figs. 10a–10c and 11a–11c. For group A, the first, third, and fourth tires of first airplane (P_{1A}, P_{3A}, and P_{4A}); first, fifth, and sixth tires of second airplane (P_{1B}, P_{5B}, and P_{6B}), and fifth tire of third airplane (P_{5C}) are shown in Figs. 9a, 9c, and 9d; 10a–10c; and 11c, respectively. For group B, the second, fifth, and sixth tires of first airplane (P_{2A}, P_{5A}, and P_{6A}) and third and fourth tire of third airplane (P_{3C} and P_{4C}) are shown in Figs. 9b, 9e, and 9f and 11a and 11b, respectively.

Conclusions

In this study, the failure rates of the tires of three De Havilland Dash-8 airplanes are modeled using both a neural network and the Weibull regression model. A two-layered neural-network model is used. A comparative study shows that the three-input neural-network model performs much better with lesser percentage difference from the actual data than the two- and one-input models as verified by visual inspection. From the comparison between neural-network and Weibull regression models, it can be concluded that the neural network, in general, predicts better than the Weibull regression model, particularly when the rate of $F(t)$ with respect to $(t - t_0)$ at the earlier stage is small and remains small for a long time, and/or if there is a major change in the rate of $F(t)$ that takes place and remains that way for a long time.

Acknowledgments

The authors are grateful to the local aviation authority for supplying the data and to King Fahd University of Petroleum and Minerals for supporting this research.

References

[1]Wu, F. Y., and Yen, K. K., "Application of Neural Network in Regression Analysis," *IEEE Transaction on Power Systems*, Vol. 23, Nos. 1–4, 1992, pp. 93–95.
[2]Lu, C., Wu, H., and Vemmuri, S., "Neural Network Based Short-Term Load Forecasting," *IEEE Transaction on Power Systems*, Vol. 8, No. 1, 1993, pp. 336–342.
[3]Al-Garni, A. Z., "Neural Network–Based Failure Rate for Boeing-737 Tires," *Journal of Aircraft*, Vol. 34, No. 6, 1997, pp. 771–777.
[4]Al-Garni, A. Z., Ahmed, S. A., and Siddiqui, M., "Modeling Failure Rate for Fokker F-27 Tires Using Neural Network," *Transactions of the Japan Society for Aeronautical and Space Sciences*, Vol. 41, No. 131, 1998, pp. 29–37.
[5]Ganguli, R., Chopra, I., and Has, D., "Helicopter Rotor System Fault Detection Using Physics-Basd Model and Neural Network Analysis," *AIAA Journal*, Vol. 36, No. 6, 1998, pp. 1078–1086.
[6]Bailey, R. A., Pidaparti, R. M., Jayanti, S., and Palakal, M. J., "Corrosion Prediction in Aging Aircraft Materials Using Neural Networks," *AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference*, AIAA, Vol. 1, No. 3, 2000, pp. 2058–2067.
[7]Pidaparti, R. M., Jayanti, S., and Palakal, M. J., "Residual Strength and Corrosion Rate Predictions of Aging Aircraft Panels: Neural Network Study," *Journal of Aircraft*, Vol. 39, No. 1, 2002, pp. 175–180.
[8]MuCllock, M. C., and Pitts, W., "Logical Calculus of the Ideas Imminent in Nervous Activity," *Bulletin of Mathematical Biophysics*, Vol. 5, 1943, pp. 115–133.
[9]Rosenblatt, F., "The Perception: A Probabilistic Model for Information Storage and Organization in Brain," *Psychological Review*, Vol. 65, No. 6, 1958, pp. 386–408.
[10]Minsky, M. L., and Papert, S. A., *Perceptrons*, MIT Press, Cambridge, MA, 1969.
[11]Hopfield, J. J., "Neural Networks and Physical Systems with Emergent Collective Computational Abilities," *Proceedings of the National Academy of Science of the United States of America*, Vol. 79, No. 8, 1982, pp. 2554–2558.
[12]Rumelhart, D., Hinton, G., and Williams, R., *Parallel Distributed Processing*, MIT Press, Cambridge, MA, 1986.
[13]Werbos, P., "Back-Propagation Through Time: What It Does and How to Do It," *Proceedings of the IEEE*, Vol. 78, No. 10, 1990, pp. 1550-1560.
[14]Haykin, S., *Neural Networks, a Comprehensive Foundation*, Prentice-Hall, Englewood Cliffs, NJ, 1999.
[15]Kapur, K. C., and Lamberson, L. R., *Reliability in Engineering Design*, Wiley, New York, 1977.
[16]Al-Garni, A. Z., Sheikh, A. K., and Badar, M. A., "Failure Statistics of Airplane Tires and a Reliability-Based Forecasting Strategy," *Fourth Saudi Engineering Conference, KAAU, Jeddah, Saudi Arabia*, Vol. 4, 1995, pp. 463–469.