Model and simulation of the fuzzy reliability assessment of the electric vehicle motor

Xuesheng Zhou, Jun Yang, Jian Wang and Jingwei Li

Abstract
The energy crisis has spurred the development of electric vehicles (EVs). As one of key components of EVs, the reliability of motor drive system directly affects the wide application of EVs. Due to the bad working environment and complex working mode, the reliability of the motor is more prominent. However, at present, there are few researches on the reliability of drive motor of EVs. Therefore, research on the reliability of electric vehicle motor drive system is of great significance. In this work, the reliability theory and prior motor reliability work were applied to the validity research of driving motor of EVs. Correlation studies were performed by using a permanent magnet synchronous motor. The failure modes of the driving motor and factors influencing reliability were examined considering the environment and motor to establish a fault tree model and analyze reliability weaknesses, including the winding insulation, bearings, and permanent magnets. The reliability of the motor was then modeled via MATLAB using a hybrid intelligent algorithm based on stochastic simulation and a neural network. The results showed that the model and the algorithm were effective. In addition, this work is helpful for electric vehicle designers to improve the reliability of electric vehicle motors.

Keywords
Back propagation neural network, driving motor, MATLAB, reliability, genetic algorithm

Date received: 19 March 2022; accepted: 5 July 2022
Handling Editor: Chenhui Liang

Introduction
As a whole, the automobile industry is facing large financial, energetic, and environmental challenges. Research efforts have thus been focused on developing electric vehicles to realize the electrification of the automobile power system and promote the strategic transformation of the current automobile industry.

Electric vehicles have one more electric drive link than traditional automobiles that includes the drive motor system and an energy storage unit. The drive motor system is an important part of electric and hybrid electric vehicles.1,2 Because of the complexity of the motor working environment, it has a great impact on the normal operation of the drive motor, so the reliability of the drive motor of the electric vehicle has become very important, and the reliability design and operation of the drive motor have become one of the focuses of attention.

The study of mechanical reliability began in the United States in the early 1960s at NASA as a result of many accidents caused by mechanical failures in space,3 which included using overload testing to examine the reliability of mechanical products. The reliability of mechanical structures and parts was studied under...
random dynamic loads. The preset reliability target value is directly applied to the design of mechanical parts whose stress distribution and strength distribution change with time. The Reliability theory of mechanical products in the United States has been mainly studied the reliability probability design method of mechanical parts. Great progress has been made toward mechanical failure prevention and detection via the collection and analysis of failure data. Some reliability design manuals and guides and reliability data manuals have been compiled.

In Japan, the Japan Science and Technology Alliance has promoted the design of practical reliability testing. The development of mechanical reliability testing depends on the inherent technology, through the reliability experiment and the use of information feedback, continuous improvement, to achieve reliability growth. Reliability theory should be applied to the design part with problems. In addition, due to the application of a large number of new technologies, new materials, new processes, and the lack of practical experience, the reliability technology was required to help designers to analyze and evaluate. In the aspect of reliability engineering application, more attention should be paid to the research and application of reliability experiment, fault diagnosis and life prediction technology, as well as the collection and feedback of product failure analysis and field operation data.

Recently, researchers have investigated reliability models, reliability analysis, and reliability test methods of motors. A reliability analysis model based on a fault tree has been proposed, as were several insulation life estimation models.

Although researchers have investigated the reliability of various motors types, little work has been done on reliability of the drive motor of an electric vehicle. The widespread use of electric vehicles will require more thorough reliability investigations of this drive motor. This work therefore aims to develop a fuzzy evaluation model on the reliability of the electric vehicle’s drive motor.

**Reliability theory of electrical machines**

The reliability of a product refers to its ability of a product to perform a specified function without failure within a specified time and under specified conditions. An instrument that works when needed is reliable, whereas one that sometimes works and sometimes does not is unreliable. In technical terms, the more reliable a product is, the longer it can work without failure. As such, reliability can be evaluated by a product’s failure rate and average failure-free interval.

Reliability theory is the basis of fault tree analysis. As such, several key technical indicators are described as follows:

**Reliability (R)**

The reliability of an equipment $R(t)$ is defined as the probability of the equipment that completes the specified function within the specified time $t$ under the specified conditions:

$$R(t) = P(T > t)$$

where $R(0) = 1$ and $R(+ \infty) = 0$.

**Cumulative failure probability**

The cumulative failure probability, or unreliability, is represented by the distribution of product life, $F(t)$, and provides the probability that a product will fail under specified conditions within a specified period of time. The cumulative failure probability can be expressed as:

$$F(t) = P(T < t)$$

OR

$$F(t) = 1 - R(t)$$

**Failure probability density**

The failure probability density $f(t)$ is the derivative of the cumulative failure probability with respect to time $t$, and represents the probability that the product will fail over time $t$. It can be expressed as:

$$f(t) = \frac{dF(t)}{dt} = F'(t)$$

OR

$$F(t) = \int_0^t f(t)dt$$

The cumulative failure probability and failure probability density are closely related to the product reliability, which determines the distribution type of product life. In the study of motor reliability, different parts have different life distribution types.

**Failure rate ($\lambda$)**

The failure rate $\lambda(t)$ is the probability of failure of an equipment in the next unit time $(0,t)$ without failure in the time. It can be calculated as:

$$\lambda(t) = \lim_{\Delta t \to 0} \frac{P(t < T \leq t + \Delta t | T > t)}{\Delta t}$$

The relationship among failure rate, reliability, and failure rate distribution can then be calculated:
Establishment of a reliability model of a motor in a fuzzy environment

For a variety of reasons, data collected under real-world and experimental conditions includes uncertainties, such as randomness, fuzziness, and roughness. These uncertainties may exist in the constraints of the problem and in the objective function to be optimized. As such, practical problems were studied via the traditional deterministic optimization, which contained too much uncertainty. To more accurately address these uncertainties, researchers have introduced methods of solving practical optimization problems. The theory of uncertainty has been applied to these practical problems more and more, the actual life of the motor is determined by the various components, and the running time of these components cannot be determined completely in advance. In order to study the reliability problems more closely and truly, the uncertainty theory is introduced into these problems to help us study these problems.

Fuzzy definition

As defining motor reliability in an uncertain environment requires functions containing uncertain parameters that are not easily compared, decision criteria must be first set according to the optimization requirements and are discussed in this section.

In fuzzy theory, \( Pos\{A\} \) describes the possibility of event A. Assuming \( \Theta \) is a non-empty set, \( p(\Theta) \) then indicates a power set of \( \Theta \) and is as:

1. \( Pos\{\Theta\} = 1 \)
2. \( Pos\{\emptyset\} = 0 \)
3. For any collection \( \{A_i\} \) in \( p(\Theta) \), \( Pos\{U_iA_i\} = sup Pos\{A_i\} \)

\( Pos \) is a probability measure and a triple \( (\Theta, p(\Theta), Pos) \) is called a probability space. The necessity measure of set \( A \) is defined as the impossibility of opposing sets \( A^c \), or \( Nec\{A\} = 1 - Pos\{A^c\} \).

Thus,

1. \( Nec\{\Theta\} = 1 \)
2. \( Nec\{\emptyset\} = 0 \)
3. \( If Pos\{A\} < 1, Nec\{A\} = 0 \)
4. \( If A \subset B, Nec\{A\} \leq Nec\{B\} \)
5. \( For any A \in p(\Theta), Nec\{A\} + Pos\{A^c\} = 1 \).

The credibility of an event is defined as the average of the probability and necessity, and plays an important role in defining expectations. The credibility of event \( A \) can be defined as:

\[
\lambda(t) = \lim_{\Delta t \to 0} \frac{P(t < T \leq t + \Delta t | T > t)}{\Delta t} = \lim_{\Delta t \to 0} \frac{P(t < T \leq t + \Delta t)}{\Delta t} = \frac{f(t)}{R(t)} = \frac{F'(t)}{R(t)} \]

Thus

\[
\lambda(t) = \frac{F'(t)}{1 - F(t)} = \frac{f(t)}{1 - F(t)} = \frac{f(t)}{R(t)} = \frac{R'(t)}{R(t)}
\]

Integrating the differential equation yields:

\[
\int_0^t \lambda(t) dt = - \ln R(t)
\]

According to the change of failure rate, the working process of a product can be divided into early failure, accidental failure, and wear and tear failure periods, the three of which can be mapped into a bathtub curve.

**Mean time between failure (MTBF)**

The mean time between failure (MTBF) is the expected value of the reliable operation time of the equipment, and can be expressed as:

\[
MTBF = E(T) = \int_0^{+\infty} f(t) dt = \int_0^{+\infty} tdF(t)
\]

\[
= \int_0^{+\infty} \int_0^{+\infty} dF(t) = \int_0^{+\infty} R(u) du
\]

**Mean time to repair (MTTR)**

The mean time to repair (MTTR) is determined by the repair probability density and as follows:

\[
MTTR = \int_0^{+\infty} g(t) dt = \int_0^{+\infty} tM'(t) dt = \int_0^{+\infty} t(1 - e^{-ut}) dt = \int_0^{+\infty} tue^{-ut} dt = 1/u
\]

A motor is a complex electromechanical energy conversion device made of many parts, each with varied failure mechanisms and life cycles, all of which impact the overall motor failure timelines. For example, the shaft, casing, and end cover of the motor are still in the accidental failure period while other parts, such as the winding insulation or bearings, are already in the loss failure period. Accurately studying the reliability of the motor therefore requires separately analyzing the failure of each part and then studying their impact on the overall failure rate of the motor.
\[ Cr\{A\} = \frac{1}{2} (\text{Pos}\{A\} + \text{Nec}\{A\}) \]  

(9)

Its nature is summarized as follows:

1. \( Cr\{\Theta\} = 1 \)
2. \( Cr\{\emptyset\} = 0 \)
3. If \( A \subseteq B \), \( Cr\{A\} \leq Cr\{B\} \)
4. \( Cr \) is self-dual, that is, for any \( A \in P(\Theta) \), \( Cr\{A\} + Cr\{A^c\} = 1 \)
5. \( Cr \) is subadditive, that is, for any \( A, B \in P(\Theta) \), \( Cr\{A \cup B\} \leq Cr\{A\} + Cr\{B\} \)

Thus, if the probability of a fuzzy event is 1, the event may not be realized. Similarly, when the necessity of the event is 0, the event may be realized. However, if the credibility of the event is 1, then it must occur, and if the credibility is 0, then it does not occur. Thus, credibility, rather than possibility, provides the measure of probability in fuzzy theory and is therefore used to express the chance of fuzzy events.

**Fuzzy variable**

In the development of fuzzy mathematics, some scholars have given some definitions of fuzzy variables. We use the following definitions of fuzzy variables:

Assuming \( \xi \) is a function from the possibility space \((\Theta, p(\Theta), \text{Pos})\) to the real line \(\mathbb{R}\), \( \xi \) is a fuzzy variable. Assuming \( \xi \) is a fuzzy variable on the possibility space \((\Theta, p(\Theta), \text{Pos})\), its membership function can be derived from the probability measure \(\text{Pos}\) and is shown as follows:

\[ \mu(x) = \text{Pos}\{\theta \in \Theta | \xi(\theta) = x\}, x \in \mathbb{R} \]  

(10)

If a quaternion \((r_1, r_2, r_3, r_4)\), where \( r_1 < r_2 \leq r_3 < r_4 \), is defined as a trapezoidal fuzzy variable, its membership function is shown as follows:

\[ \mu(x) = \begin{cases} \frac{x - r_1}{r_2 - r_1}, & r_1 \leq x \leq r_2 \\ 1, & r_2 \leq x \leq r_3 \\ \frac{x - r_3}{r_4 - r_3}, & r_3 \leq x \leq r_4 \\ 0, & \text{else} \end{cases} \]

When \( r_2 = r_3 \), trapezoidal fuzzy variables are triangular fuzzy variables. The possibility, necessity, and credibility of trapezoidal fuzzy variables were thus considered according to the definitions of possibility, necessity, and credibility, and were shown as follows:

\[ \text{Pos}\{\xi \leq 0\} = \begin{cases} 1, & r_2 \leq 0 \\ \frac{r_2 - r_1}{r_2 - r_1}, & r_1 \leq 0 \leq r_2 \\ 0, & \text{else} \end{cases} \]

\[ \text{Nec}\{\xi \leq 0\} = \begin{cases} 1, & r_4 \leq 0 \\ \frac{r_4 - r_3}{r_4 - r_3}, & r_3 \leq 0 \leq r_4 \\ 0, & \text{else} \end{cases} \]

\[ Cr\{\xi \leq 0\} = \begin{cases} 1, & r_4 \leq 0 \\ \frac{r_2 - r_1}{r_2 - r_1}, & r_1 \leq 0 \leq r_2 \\ \frac{r_2 - r_1}{r_2 - r_1}, & r_1 \leq 0 \leq r_2 \\ 0, & \text{else} \end{cases} \]

Here, the relevant variables of motor reliability are triangular fuzzy variables. If function \( y \in [0, 1] \), \( \xi \) is a fuzzy variable and satisfies \( y(x) = Cr\{\theta \in \Theta | \xi(\theta) \leq x\} \), then \( y \) represents the credibility distribution of the fuzzy variable \( \xi \). Generally speaking, the credibility distribution \( y(x) \) is the credibility of fuzzy variable \( \xi \) with a value of less than or equal to \( x \). The distribution of credibility is neither left nor right continuous.

Assuming a triangular fuzzy variable is \( \xi = (r_1, r_2, r_3) \), the credibility distribution can be obtained as:

\[ y(x) = \begin{cases} 0, & x \leq r_1 \\ \frac{x - r_1}{r_2 - r_1}, & r_1 \leq x \leq r_2 \\ \frac{x - r_2}{r_3 - r_2}, & r_2 \leq x \leq r_3 \\ 1, & r_3 < x \end{cases} \]

Its dependability density function is:

\[ y(x) = \begin{cases} \frac{1}{2(r_2 - r_1)} - \frac{1}{2}, & r_1 \leq x \leq r_2 \\ \frac{1}{2(r_3 - r_2)} - \frac{1}{2}, & r_2 \leq x \leq r_3 \\ 0, & \text{else} \end{cases} \]

Compared with random variables, different indicators are used to compare fuzzy variables. Optimism and pessimism are good indicators.

Assuming \( \xi \) represents a fuzzy variable and \( \alpha \in (0, 1] \), then the optimistic value of \( \alpha \) is:

\[ \xi_{\text{sup}}(\alpha) = \sup\{r | Cr\{\xi \geq r\} \geq \alpha\} \]  

(11)

Assuming \( \xi \) represents a fuzzy variable and \( \alpha \in (0, 1] \), then the pessimistic value of \( \alpha \) is:

\[ \xi_{\text{inf}}(\alpha) = \inf\{r | Cr\{\xi \leq r\} \geq \alpha\} \]  

(12)

The expected value of a fuzzy variable can thus be defined as:

Assuming \( \xi \) represents a fuzzy variable, its expected value can be found via the following equation if at least one of the integrals is finite.

\[ E[\xi] = \int_{0}^{+\infty} Cr\{\xi \geq r\} dr - \int_{-\infty}^{0} Cr\{\xi \leq r\} dr \]  

(13)

Therefore, the expected value of the triangular fuzzy variable \( \xi = (r_1, r_2, r_3) \) is \( E[\xi] = \frac{1}{3} (r_1 + 2r_2 + r_3) \), and the expected value of trapezoidal fuzzy variable \( \xi = (r_1, r_2, r_3, r_4) \) is \( E[\xi] = \frac{1}{4} (r_1 + r_2 + r_3 + r_4) \).

If, for two fuzzy variables \( \xi \) and \( \eta \), \( \xi(\theta) > \eta(\theta) \) for any \( \theta \in \Theta \), then \( \xi > \eta \), of which \( \Theta \) is a fuzzy set. In fact,
Petri nets are mathematical representations of discrete parallel systems that were first introduced by Carl A. Petri. Petri nets developed in 1960 was used to describe asynchronous and concurrent computer system models. Petri nets are known for their strict mathematical expression, intuitive graphical expression, rich system descriptions, and system behavior analysis technology, and have the ability to provide a solid conceptual basis for computer science.\textsuperscript{14–17} Classical Petri nets are simple process models consisting of place and transition nodes, connections, and tokens, and as shown in Figure 1.

where a circular node represents a place, square nodes represent transitions, a directed arc represents a connection between a place and a transition, and a token is a dynamic object in a place that can be moved from one place to another. Based on the directed bipartite graph of the place and the transition node, the token distribution representing the state information is added, and the event-driven state evolution is made according to the initiation rules, so as to reflect the dynamic operation process of the system.

A six-tuple $N = (P, T, I, O, M)$ is a Petri net if the following conditions are met.

- $P = \{p_1, p_2, \cdots, p_n\}$ is a finite set of places, where $n$ is the number of places, $n > 0$;
- $T = \{t_1, t_2, \cdots, t_s\}$ is a finite set of transitions, where $s$ is the number of transitions, $s > 0$,
  $P \cap T = \emptyset$;
- $I : P \times T \rightarrow N$ is the input function that defines the set of directed arc repeats or weights from $P$ to $T$, where $N = \{0, 1, 2, \cdots\}$ is the set of non-negative integers;
- $O : T \times P \rightarrow N$ is an input function that defines a set of directed arc repeats or weights from $T$ to $P$;
- $M : P \rightarrow N$ is the identification distribution in each place; and
- $M_0 : P \rightarrow N$ is the initial identity distribution in each place.

Its element composition diagram is shown in Figure 2. The application of Petri net to analyze system faults is to take the events that the system does not want to happen as the top database, and find out all possible factors leading to this event step by step as the intermediate place and the bottom place. The set of logical relationships can then be expressed by the different connections of the basic elements of Petri net to establish a

\begin{figure}
\centering
\includegraphics[width=0.5\textwidth]{Figure1.png}
\caption{The structure of classical petri nets.}
\end{figure}

\begin{figure}
\centering
\includegraphics[width=0.5\textwidth]{Figure2.png}
\caption{Element composition of a petri net.}
\end{figure}
fault tree model that can then be diagnosed by using graph theory. The top, middle, and bottom events in the fault tree are represented by places, transitions, and connections in the Petri net, and as shown in Figure 3.

**Reliability model of motor**

The motor of an electric vehicle was shown schematically in Figure 4, which consists of a stator, rotor, and shaft.

**Numerical calculation**

The failure causes of each part in the reliability prediction of the motor of electric vehicles were listed in Table 1.

In the associated sub-project run schedule, as shown in Table 2, the subproject runtime was assumed to be a fuzzy variable and the execution time of each subproject was assumed to be a triangular fuzzy variable, expressed in triples \((r_1, r_2, r_3)\), of which \(r_1 < r_2 < r_3\). Different optimal solutions were obtained by assigning different values to several key parameters of genetic algorithm, thus proving the effectiveness of the algorithm by comparison. In the hybrid intelligent algorithm for the next numerical example, there are several pre-given parameters: population size \(size_{pop}\), crossover probability \(P_c\), and mutation probability \(P_m\).

As there are seven events at the bottom of the petri net model of the electric vehicle motor, the number of

| Fault tree model | Petri net model | Cause of failure |
|------------------|----------------|-----------------|
| B1               | p1             | Stator core failure |
| B2               | p2             | Stator winding failure |
| B3               | p3             | Rotor core failure |
| B4               | p4             | Permanent magnet fault |
| B5               | p5             | Bearing failure |
| B6               | p6             | Axis bending |
| B7               | p7             | External spline failure |
| M4               | p8             | Shaft failure |
| M1               | p9             | Stator failure |
| M2               | p10            | Rotor fault |
| M3               | p11            | On-shaft failure |
| T                | p12            | Motor failure |

| Petri net model | Reason of failure | Lifetime          |
|-----------------|-------------------|-------------------|
| t1              | Stator failure caused by stator core failure | \((18000,20000,22000)\) |
| t2              | Stator failure caused by stator winding failure | \((8000,9000,10000)\) |
| t3              | Rotor failure caused by rotor core failure | \((16000,18000,20000)\) |
| t4              | Rotor fault caused by permanent magnet fault | \((9000,10000,11000)\) |
| t5              | On-shaft failure due to bearing failure | \((8000,10000,12000)\) |
| t6              | Shaft failure caused by shaft bending | \((16000,18000,20000)\) |
| t7              | Shaft failure caused by external spline failure | \((14000,16000,18000)\) |
| t8              | Shaft failure due to fault of shaft assembly | ——— |
| t9              | Motor failure due to stator failure | ——— |
| t10             | Motor failure due to rotor failure | ——— |
| t11             | Motor failure due to on-shaft failure | ——— |
input and hidden layer neurons was set as $u = 7$ and $b = 12$, respectively. The value of the given error term was 0.000000004. Based on random simulation, 4000 input and output data points were generated and divided into two groups: 3000 points for training and 1000 points for testing. Note that the number of training data points is set to 4000 for two reasons. First, because there are too few data points, the neural network cannot be fully trained, resulting in inaccurate results. Second, too much data increases the training time and the response time of the solution model, which makes the neural network fitting difficult and affects the work progress. It should be noted that the neural network training data selection should be appropriate, and more data does not necessarily mean the acquisition a higher training accuracy.

In the analysis of fault mode and reliability, the fault mode, influencing factors, causes and harm degree of the motor are analyzed in depth, but there is no clear logic for the motor fault mode. By using the fault tree to analyze the reliability of the motor, we can clearly see the logical relationship between various fault modes and motor faults, and can analyze the probability of faults. The fault tree model, developed according to the components and their logical relationships, is shown in Figure 5.

A permanent magnet synchronous motor (PMSM) is mainly composed of stator, rotor, permanent magnet, shaft, and frame cover. From each part of the motor, the logical relationship between the fault mode of each component and the motor fault is analyzed. Motor fault is the most undesirable fault, so motor fault is the top event, and the fault mode of each component is the bottom event.\(^{18-20}\) In this work, the causes of various faults are analyzed and a fault tree of a PMSM for an electric vehicle driving motor is established according to the classification of motor faults for electric vehicles in the national standard named “Classification and Judgment of Drive Motor System Faults for Electric Vehicle” and is as shown in the following figure.

Combined with the Petri net concept, the Petri model of the electric vehicle motor is shown in Figure 6.

The minimum expected time model can then be established:
Motor reliability model

A BP neural network optimization algorithm based on a genetic algorithm was then used to solve the developed reliability statement of an electric vehicle motor.

Algorithm flow

A genetic algorithm was used to optimize the BP neural network algorithm flow, which was as shown in Figure 7:

The optimization of the BP neural network can be divided into three parts: The determination of the neural network structure, the optimization of the genetic algorithm, and the prediction of the BP neural network. The BP neural network structure is determined according to the number of input and output parameters of the fitting function, which then is used to determine the individual length of the genetic algorithm. The genetic algorithm then is used to optimize the weights and thresholds of the BP neural network. Each individual in the population contains a network ownership value and threshold, and has a fitness value, which is calculated by a fitness function. The genetic algorithm is used to find the optimal individual fitness through selection, crossover, and mutation. The prediction of BP neural network is to use the genetic algorithm to get the initial weight and threshold value assigned by the optimal individual to the network, and the network is trained to predict the output of the function.

The developed model requires seven input parameters and delivers two output parameters, so the BP neural network structure is $7 - 12 - 2$, that is, to say that there are two nodes in the input layer, 12 nodes in the hidden layer and 2 nodes in the output layer, there are $7 \times 12 + 12 \times 2 = 108$ weights and $12 + 2 = 14$ thresholds in common, so the individual coding length of the genetic algorithm is $108 + 14 = 122$. From the obtained 4000 sets of input and output data, 3000 sets were randomly selected for network training; the remaining 1000 were used as test data. Taking the absolute sum of the prediction error of the training data as the individual fitness value, the smaller the individual fitness value is, the better the individual is.
A BP network is a multi-layer feedforward neural network that uses a backpropagation error learning algorithm. BP algorithm is Rumelhart because of its simple structure, adjustable parameters, training algorithms, and good operability, and have thus been widely used.23,24 According to statistics, 80%–90% of the neural network models are using BP network or its deformation, making it the most widely used algorithm in artificial neural network design. BP network is the core of forward network and the most essential and perfect part of neural network. Although BP neural network is the most widely used algorithm in artificial neural network, it also has some defects, such as slow learning convergence speed, hindering convergence to the global minimum, and a complex network structure that is difficult to determine.

A BP neural network is used to first determine the number of neurons in the input, hidden, and output layers, which are as follows:

The number of input-layer neurons of a neural network is based on the number of bits at the bottom of the Petri model, that is the number of basic events in the fault tree model. Thus, the number of input neurons can be represented by a life distribution function of all components of the system that can cause system failures.

The number of output neurons is 1:1 of the system lifetime function. In the aspect of neural network structure, the main problem is to determine the optimal number of hidden layer neurons. Although the number of hidden layers can theoretically be infinite, too many increase the required training and response time, whereas too few make the neural network lack generalization. It can generally be represented by

\[ s = \sqrt{u + v + b} \]

where \( u \) and \( v \) denote the number of input and output layer neurons, respectively, and \( b \) is a constant between 1 and 10.

The back propagation method is then used to calculate the weights and bias conditions of the neural network model. In this method, all weights are adjusted according to the error terms calculated by using the gradient method. In the learning of the neural network, that is, the calculation of the weights of the connections, the errors between the output of several available training data points and the actual output are minimized. Here, the error term was controlled by the MATLAB function named net.trainParam.goal, which represents the square of the error between the output of the averaged neural network and the actual output of the available training data points.

BP neural networks are more mature than earlier neural network in both theory and performance and have the advantages of clear thinking, rigorous structure, and strong operability. The most outstanding is

---

**Figure 7.** Optimization of BP neural network process by GA.
BP neural network has strong nonlinear mapping ability and flexible network structure.\textsuperscript{25,26} However, because the core of BP learning algorithm is the optimal gradient descent method, which is mainly used to guide the adjustment direction of network connection weight and threshold, BP neural network inevitably has the following defects:

1. The convergence rate of network learning is too slow. Because the learning complexity of BP algorithm is an exponential function of the sample size, if the network size is large or the number of learning samples is large, it often takes a long time to learn, or even the learning cannot be completed. Even a relatively simple problem requires hundreds or even thousands of studies to converge. The main reason is that the learning rate of the network is too small, which can be improved by changing the learning rate or adaptive learning rate.

2. It is easy to fall into the local minimum problem. BP algorithm is a nonlinear optimization problem based on stochastic gradient descent method. It inevitably has local minimum problem and often gets local minimum point. This is because when the gradient descent method is used to train the network, the minimum value of the error is gradually reached along the slope of the error function from a certain starting point, but in fact, the local optimum is usually obtained.

3. The network structure is difficult to determine. It is very difficult to select the number of layers and hidden nodes in the hidden layer of the network, and there is no theoretical basis so far. How many hidden nodes are chosen will affect the performance of the network, usually according to the network environment and experience or experiment in the learning process of the network, so the network has a great deal of redundancy.

4. The learning and memory of the network are unstable. A well-trained network, on the basis of the original to provide it with a new model, will make the original connection right all disrupted, and it must be retrained to continue to use the network, which undoubtedly increases the difficulty of network learning.

In addition, the selection of network structure, initial connection weights and threshold has great influence on the network training, but it cannot be accurately obtained. In view of these characteristics, a genetic algorithm can be used to optimize the neural network.

**Genetic algorithm**

A genetic algorithm is an evolutionary algorithm, the basic principle of which is to imitate the evolutionary theory of natural selection. Genetic algorithm is to encode the problem parameters as chromosomes, and then exchange the information of chromosomes in the population with the iterative crossover and mutation operations, and finally generate the chromosomes in line with the optimization goal.\textsuperscript{27–29} The simple genetic algorithm was proposed by Professor J. Holland of Michigan University in 1962 and published in the influential monograph named “Adaptation in Natural and Artificial Systems.” In optimization problems, if the objective function is multi-peak or the search space is irregular, it is likely that the algorithm will wander in the local optimal solution, and the advantage of genetic algorithm is just good at global search, and the analysis of mathematical problems does not need very deep, beginners will be very convenient to use.

In a genetic algorithm, chromosomes correspond to data or arrays, which are usually represented by one-dimensional string structure data. A genetic algorithm follows a basic procedure, in which $N$ initial chromosomes are first randomly generated to form a population. Then an evaluation function is used for evaluating the quality of each chromosome to get the adaptability of chromosomes to the environment, which is called adaptability.\textsuperscript{30,31} A selection process is then carried out to select the best chromosomes from the current chromosomes to make them into a new generation of chromosomes. The better the fitness, the greater the chance of being selected. Generally, we choose chromosomes in a way similar to rotating roulette wheel in the selection process. After a new population is obtained through the selection process, we perform crossover and mutation operations. The purpose of crossover is to dig out the diversity of individuals in the population and prevent the genetic algorithm from falling into the “trap” of local optimal solution. Through these operations, we can get a new generation of chromosomes, a new population, which we call progeny. Next, we repeat the selection, crossover and mutation operations on the offspring. After a certain number of iterations, the best chromosomes obtained in the above steps are used as the optimal solution of the optimization problem. Through the above steps, it is not difficult to find that genetic algorithm has many advantages over traditional algorithms. For example, each generation of genetic algorithm studies the population of more than one point, while the traditional algorithm usually studies a single point at a time, which makes the genetic algorithm have more application scope than the traditional algorithm. The running process of genetic algorithm is time-consuming, and sometimes it is not
suitable for the system that needs immediate feedback. In addition, the solution obtained by genetic algorithm has nothing to do with the selection of initial point, so it is more suitable to solve the problem of irregular space and multi-peak optimization, and is not easy to fall into the “trap” of local optimal solution. Of course, genetic algorithm has more advantages in computer applications, and is suitable for parallel computing.

Genetic algorithm has shown a very broad application prospect in solving global optimization problems, and has been successfully applied to practical problems such as optimal control, transportation problems, traveling salesman problems, mapping, equipment location, statistics, pattern recognition, vehicle scheduling, and network optimization.

Through the above analysis, the BP neural network optimized by genetic algorithm uses a genetic algorithm to optimize the initial weight and threshold of the BP neural network, so that the optimized BP neural network can better predict the function output. The elements of BP neural network optimized by genetic algorithm include population initialization, fitness function, selection operation, crossover operation, and mutation operation.

**Population initialization**

Each individual is a real string composed of four parts: the connection weight between the input layer and the hidden layer, the threshold value of the hidden layer, the connection weight between the hidden layer and the output layer, and the threshold value of the output layer. An individual contains all the weights and thresholds of the neural network. If the structure of the network is known, it can form a neural network whose structure, weights, and thresholds are determined.

\[
\begin{align*}
X''_1 &= \left( x_1^{(1)} \times u + x_1^{(2)} \times (1 - u), x_2^{(1)} \times u + x_2^{(2)} \times (1 - u), \ldots, x_n^{(1)} \times u + x_n^{(2)} \times (1 - u) \right) \\
X''_2 &= \left( x_1^{(2)} \times u + x_1^{(1)} \times (1 - u), x_2^{(2)} \times u + x_2^{(1)} \times (1 - u), \ldots, x_n^{(2)} \times u + x_n^{(1)} \times (1 - u) \right)
\end{align*}
\]

In this paper, the individual coding length is: 
\[7 \times 12 + 12 + 12 \times 2 + 2 = 122\]

**Fitness function**

According to the initial weights and thresholds of the BP neural network obtained by the individual, the output of the prediction system is trained with the training data. The individual fitness value \(F\) is calculated as the \(E\) of the absolute value of the error between the prediction output and the expected output:

\[F = \kappa \left( \sum_{i=1}^{n} \text{abs}(y_i - a_i) \right)\]

where \(n\) is the number of output nodes of the network, \(y_i\) is the expected output of the \(i\)-th node of BP neural network, \(a_i\) is the predicted output of the \(i\)-th node, and \(\kappa\) is the coefficient.

**Selection process**

Chromosomes were selected by rotating \(sizepop\) times to construct a new generation of chromosomes to update the chromosomes. First, the cumulative probability \(q_i\) was calculated for each chromosome \(x_i\), where \(q_0 = 0\), as:

\[q_i = \sum_{j=1}^{i} \text{Eval}(x_j), i = 1, 2, \ldots, sizepop\]

A random number \(s\) was then generated in the interval \((0, sizepop)\). The chromosome satisfying \(q_{i-1} < s \leq q_i\) was then selected as \(X_i\). This process was then repeated by the number of elements in \(sizepop\) to develop the new generation of chromosomes.

**Crossover operation**

The parent chromosomes were then selected by first randomly generating a number \(v\) from \([0, 1]\). If \(v \leq Pc\), where \(P_c\) represents the probability of crossover, chromosome \(X_i\) was selected as a parent chromosome. In general, we can probably get \(sizepop \times P_c\) parent chromosome, and remember \(X'_1, X'_2, \ldots\). These chromosomes were then cross-manipulated. For example, when cross-operating between \(X'_1 = (x_1^{(1)}, x_2^{(1)}, \ldots, x_n^{(1)})\) and \(X'_2 = (x_1^{(2)}, x_2^{(2)}, \ldots, x_n^{(2)})\), a number \(u\) is randomly generated from \([0, 1]\). The offspring are then calculated as:

\[
\begin{align*}
X''_1 &= \left( x_1^{(1)} \times u + x_1^{(2)} \times (1 - u), x_2^{(1)} \times u + x_2^{(2)} \times (1 - u), \ldots, x_n^{(1)} \times u + x_n^{(2)} \times (1 - u) \right) \\
X''_2 &= \left( x_1^{(2)} \times u + x_1^{(1)} \times (1 - u), x_2^{(2)} \times u + x_2^{(1)} \times (1 - u), \ldots, x_n^{(2)} \times u + x_n^{(1)} \times (1 - u) \right)
\end{align*}
\]

If the above two progeny chromosomes are feasible, they can then replace the two parent chromosomes. If one of the offspring chromosomes is feasible, the viable is retained, and the crossover operation is repeated to get another viable chromosome. Otherwise, crossover is repeated until two viable offspring chromosomes are generated.

**Mutation operation**

A parameter \(P_m\) beforehand was given as the probability of mutation operation. Similar to the crossover
selection process, \( sizepop \times P_m \) parent chromosome could be obtained. The procedure for mutation operation is as follows.

For a chromosome \( \mathbf{x} = \{x_1, x_2, \ldots, x_n\} \), a random number \( w \) from is generated from \([0, W]\), where \( W \) is a given positive number. A new vector can then be obtained:

\[
\mathbf{X}' = (x_1 + d_1 \times w, x_2 + d_2 \times w, \ldots, x_n + d_n \times w)
\]

where \( d_i \) is randomly generated in intervals of \([-1, 1]\), and \( i = 1, 2, \ldots, n \). If \( \mathbf{X}' \) is not, another randomly generated number \( w' \) from the interval \([0, w]\) is used to repeat the process until a viable chromosome is obtained to replace the selected parent chromosome.

### Hybrid intelligent algorithm

A hybrid intelligent algorithm was then developed to solve reliability problems. The steps are as follows:

1. **Step 1.** Initiate by producing \( sizepop \) chromosomes.
2. **Step 2.** Cross and mutate chromosomes.
3. **Step 3.** Calculate the objective function values of each chromosomes by stochastic simulation.
4. **Step 4.** Calculate the fitness of each chromosome according to the value of objective function.
5. **Step 5.** Select chromosomes by spinning the wheel.
6. **Step 6.** Repeat the times from Step 2 to Step 5
7. **Step 7.** Obtain the optimal solution as the chromosome with the highest fitness function.

### Results of training the BP neural network

The developed stochastic algorithm based on a neural network was then implemented based on the established fault tree and corresponding petri net model. The resulting average life of the electric vehicle motor, \( E(x) \), was 8895.5 h.

The errors between the predicted output and the desired output of the BP neural network are shown in Figures 8 and 9.

### Analysis of BP neural network optimized by genetic algorithm

The MATLAB software was used to establish the neural network model and input the corresponding training data, the training process was as shown in Figure 10.

The optimal weights and thresholds of BP neural network optimized by genetic algorithm were then assigned to the neural network. The life of the motor was predicted training the network 100 times with the training data. The prediction error was shown in Figure 11.

It can be seen from the Figure 11 that the prediction error of BP neural network optimized by genetic algorithm was more accurate, and the mean square error of
prediction of BP network optimized by genetic algorithm is $3.8919 \times 10^{-3}$, whereas the mean square error of prediction of unoptimized BP network is $4.5518 \times 10^{-4}$, and the mean square error of prediction is also greatly improved.

An ambiguous event whose probability is 1 may not be realized, and an ambiguous event whose necessity is 0 may be realized. If a fuzzy event has a credibility of 1, it must be implemented. If the trustworthiness of a fuzzy event is 0, it must not be implemented. Therefore, the role of the credibility of fuzzy events in fuzzy theory is equivalent to the role of probability in probability theory, and it is more scientific and realistic to use credibility measure to measure the probability of fuzzy event than possibility measure. The reliability results of electric vehicle motors are discussed in the next section.

**Resulting reliability $R(x)$ model of electric vehicle motors**

As shown in Table 3, it indicate, for example, the probability that the motor will not fail for 8200 h of safe operation is 0.884.

**Conclusion**

The reliability of motor operation is the key factor for the safe operation of electric vehicles. In order to dynamically describe the reliability elements and fault tree modeling process, in this paper, based on the concept of fault tree and Petri net, a reliability model of electric vehicle motor was firstly established by combining the definition and transformation rules of fault tree elements, which could provide a new method for reliability analysis of mechanical system. A hybrid intelligent algorithm was designed by combining a genetic algorithm with neural network to create a genetic-algorithm-optimized neural network simulation. Finally, the hybrid intelligent algorithm is applied to the reliability calculation of an electric vehicle motor, and the feasibility of the proposed method is verified. In conclusion, the research work in this paper is of positive significance to improve the development and design decision-making process of electric vehicle motors in the future.

**Declaration of conflicting interests**

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

**Funding**

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: The authors would like to thank University-level scientific research fund projects of ShanDong JiaoTong University (Nos. Z201636), the National Natural Science Foundation of China (Nos. 61803231) and Shandong Provincial Higher School Youth Innovation Technology Project of China (Nos. 2020KJB002) for the financial support of this research.

**ORCID iD**

Xuesheng Zhou https://orcid.org/0000-0003-1651-2468

**References**

1. Rezaee S, Farjah E and Khorramdel B. Probabilistic analysis of plug-in electric vehicles impact on electrical grid through homes and parking lots. *IEEE Trans Sustain Energy* 2013; 4: 1024–1033.
2. Fan J, Zhang C, Wang Z, et al. Thermal analysis of permanent magnet motor for the electric vehicle application considering driving duty cycle. *IEEE Trans Magn* 2010; 46: 2493–2496.
3. Si JS, Shi XJ, Yu XD, et al. Research on reliability and maintainability of mechanical equipment. In: Proceedings of 2011 international conference on electronic & mechanical engineering and information technology; 2011; 3. New York: IEEE.
4. Buschke HA. A practical approach to testing electronic equipment for susceptibility to AC line transients. *IEEE Trans Reliab* 1988; 37: 355–359.
5. Cotroneo D, Pietrantuono R and Russo S. Combining operational and debug testing for improving reliability. *IEEE Trans Reliab* 2013; 62: 408–423.
6. Turner MD. A practical application of quantitative accelerated life testing in power systems engineering. *IEEE Trans Reliab* 2010; 59: 91–101.

7. Ma K, Choi UM and Blaabjerg F. Prediction and validation of wear-out reliability metrics for power semiconductor devices with mission profiles in motor drive application. *IEEE Trans Power Electron* 2018; 33: 9843–9853.

8. Pippola J, Vaalasranta I, Marttila T, et al. Product level accelerated reliability testing of motor drives with input power interruptions. *IEEE Trans Power Electron* 2015; 30: 2614–2622.

9. Nakai H, Ohtani H, Satoh E, et al. Development and testing of the torque control for the permanent-magnet synchronous motor. *IEEE Trans Ind Electron* 2005; 52: 800–806.

10. Yan H, Xu Y, Cai F, et al. PWM-VSI fault diagnosis for a PMSM drive based on the fuzzy logic approach. *IEEE Trans Power Electron* 2019; 34: 759–768.

11. Rubaai A and Young P. Hardware/software implementation of fuzzy-neural-network self-learning control methods for brushless DC motor drives. *IEEE Trans Ind Appl* 2016; 52: 414–424.

12. Chen CS. TSK-type self-organizing recurrent-neural-fuzzy control of linear microstepping motor drives. *IEEE Trans Power Electron* 2010; 25: 2253–2265.

13. Rubaai A, Castro-Sitiriche MJ and Ofoli AR. Design and implementation of parallel fuzzy PID controller for high-performance brushless motor drives: an integrated environment for rapid control prototyping. *IEEE Trans Ind Appl* 2008; 44: 1090–1098.

14. Luan W, Qi L, Zhao Z, et al. Logic petri net synthesis for cooperative systems. *IEEE Access* 2019; 7: 161937–161948.

15. Ortiz L and Aguila A. Modeling and control of an automated system in surgical microscopy based on petri nets. *IEEE Lat Am Trans* 2016; 14: 96–101.

16. Tiplea FL and Leahu I. The reversible released form of petri nets and its applications to soundness of workflow nets. *IEEE Trans Syst Man Cybern: Syst* 2016; 46: 303–312.

17. Distefano S, Scarpa M and Puliafito A. From UML to petri nets: the PCM-based methodology. *IEEE Trans Softw Eng* 2011; 37: 65–79.

18. He C and Wu T. Analysis and design of surface permanent magnet synchronous motor and generator. *CES Trans Electr Mach Syst* 2019; 3: 94–100.

19. Onsal M, Demir Y and Aydin M. A new nine-phase permanent magnet synchronous motor with consequent pole rotor for high-power traction applications. *IEEE Trans Magn* 2017; 53: 1–6.

20. Aydin M and Gulec M. A new coreless axial flux interior permanent magnet synchronous motor with sinusoidal rotor segments. *IEEE Trans Magn* 2016; 52: 1–4.

21. Yang A, Zhuansun Y, Liu C, et al. Design of intrusion detection system for internet of things based on improved BP neural network. *IEEE Access* 2019; 7: 106043–106052.

22. Wu W, Feng G, Li Z, et al. Deterministic convergence of an online gradient method for BP neural networks. *IEEE Trans Neural Netw Netw* 2005; 16: 533–540.

23. Heravi AR and Abed Hodtani G. A novel correntropy-based gradient backpropagation algorithm for improving training in neural networks. *IEEE Trans Neural Networks Learn Syst* 2018; 29: 6252–6263.

24. Bershad NJ, Ibnkahla M, Blauwens G, et al. Fluctuation analysis of a two-layer backpropagation algorithm used for modeling nonlinear memoryless channels. *IEEE Trans Signal Process* 1999; 47: 1297–1303.

25. Pan X, Zhou W, Lu Y, et al. Prediction of network traffic of smart cities based on DE-BP neural network. *IEEE Access* 2019; 7: 55807–55816.

26. Palmes PP, Hayasaka T and Usui S. Mutation-based genetic neural network. *IEEE Trans Neural Networks Neural Netw* 2005; 16: 587–600.

27. Cheng YF, Shao W, Zhang SJ, et al. An improved multi-objective genetic algorithm for large planar array thinning. *IEEE Trans Magn* 2016; 52: 1–4.

28. Lu J, Chen Z, Yang Y, et al. Online estimation of state of power for lithium-ion batteries in electric vehicles using genetic algorithm. *IEEE Access* 2018; 6: 20868–20880.

29. Takshi H, Dogan G and Arslan H. Joint optimization of device to device resource and power allocation based on genetic algorithm. *IEEE Access* 2018; 6: 21173–21183.

30. Yamashita H, Cingoski V and Yokose Y. Genetic algorithms with assistant chromosomes for inverse shape optimization of electromagnetic devices. *IEEE Trans Magn* 2000; 36: 1052–1056.

31. Bansal A, Cannings C and Sheehan N. An evaluation of the application of the genetic algorithm to the problem of ordering genetic loci on human chromosomes using radiation hybrid data. *Math Med Biol* 1997; 14: 161–187.