An Intelligent FMEA System Implemented with a Hierarchy of Back-Propagation Neural Networks

Chiang Ku, Yun-Shiow Chen and Yun-Kung Chung
Department of Industrial Engineering and Management Yuan Ze University
Chung Li, Taiwan
ieychung@saturn.yzu.edu.tw

***Abstract*—This paper has used a series of back-progation neural networks (BPNs) to form a hierarchical framework adequate for the implementation of an intelligent FMEA (failure modes and effects analysis) system. Its aim is to apply this novel system as a tool to assist the reliability design required for preventing failures occurred in the operating periods of a system The hierarchical structure upgrades the classical statistic off-line FMEA performance. From the simulated experiments of the proposed BPN-based FMEA system (N-FMEA), it has found that the accuracy of the failure modes classification and the reliability calculation are knowledgeable and potential for performing pragmatic preventive maintenance activities. As a result, this paper conducts an effective FMEA process and contributes to help FMEA working teams to reduce their working loading, shorten design time and ensure system operating success.**

***Keywords*—back-propagation neural networks, failure modes and effects analysis, preventive maintenance, reliability design.**

I. INTRODUCTION

FMEA (Failure Modes, Effects and Analysis) has been addressed over the past four decades. Its analysis procedure is based on FMEA guidelines described in QS-9000 standard regulated by teams working under the Automotive Division of the American Society for Quality Control (ASQC) and the Automotive Industry Action Group (AIAG) located at Chrysler, Ford, and General Motors. The QS-9000 standard requires suppliers to the automotive industry to conduct the FMEAs of product design and manufacturing process in an effort to eliminate failures (defects) before they happen. Thus, it needs a systematically proactive approach for performing the identification and prevention of the occurrences of system failure problems, further to advance safety, extend warranty, and then increase customer satisfaction. Also, FMEA is conducted, in principle, in the product (reliability) design or process (quality) development stages, but conducting an FMEA on an existing system (product or process) may create great benefits [1, 2].

Brainstorming used in an FMEA team is one of such a proactive way to identify potential failure modes that could affect a system's process or its quality; then the affecting severity degree, RPN (risk priority number), is calculated and classified for each failure mode to determine its corresponding causalities and select failure corrective actions. These associative tasks involving with failure corrections and reliability improvement make FMEA be a mechanism to know where, what and why a system's functions may fail, and how they can be corrected reliably, before its related failures occur. Since its failure preventions can be done before a system works, FMEA can not only continuously improve product and/or process reliability and quality, but also reduce cost and shorten time required for the research and development (R&D) of a system, e.g., product, equipment or process [3]. Additionally, for the CMMI (Capability Maturity Model Integration) quality management needed in the process of a software product development, FMEA is also creditable for preventing those problems of the software's malfunctions prior to the software's release. The software quality accreditation process may be affected with a certain degree if a failure (bug) hidden in the software could not be found in advance [4, 5].

Yet, at present, it has become clear that the functional requirements of operating all factors or parameters in the reliability and quality design, such as on-line failure modes identification and real-time computation of reliability parameters, might not be simultaneously taken into account with a classical statistic FMEA technique, thus a better approach is required to make ascent in the ability of performing FMEA tasks. Fortunately, the approaches based on artificial neural networks (ANNs) for the fault identification, classification and computation have been put forward and proved effectively [6, 7], and these ideas inspired this paper to develop an alternative FMEA system with ANNs, called back-propagation neural networks (BPNs), which has been used widely in various applications [8, 9, 10]. This paper will present the performance and the stable BPN learning behavior applied to FMEA situation considered, and from the implementation experiences, the suitability and superiority of the proposed hierarchical framework composed of BPNs for performing FMEA tasks are striking.

In what follows, the outline of this paper is described. Section II gives an introduction to FMEA. For readers who are not proficient with FMEA, this section serves as a foundation of the subject. Then, the background in BPN is reviewed, including its mechanism used to perform the optimal learning process and further applied to FMEA. Subsequently, some of the previous published work done on the application of ANN to fault identification and classification, which will be extended to develop the intelligent FMEA system proposed in this paper, is briefly reviewed. In Section V, the details of the newly hierarchical structure for performing FMEA tasks are proposed, and the solution accuracy of its simulated numerical experiments are conduced in the penultimate section for evaluating the performance of the proposed hierarchical BPN network. The final section concludes the findings of this paper. The contributions of this paper will be described and some of the ways that this paper can be extended in the future are pinpointed in this section.

II. FMEA INTRODUCTION

As aforementioned, FMEA can identify the potential failures of each of the constituent functions performing the whole correct operation of a reliability system (a process or a product), and then associate those identified failures' types (modes) with the causes and the effects which may influence on the whole process (product); also, it is frequently used as a primary tool for system reliability improvement and maintenance planning before they are functioned and operated. By means of its performing FMEAs, a company is benefited both with significant cost and time saving in its product development and process maintenance and with the reduction of potential risk of costly liability that does not perform as promised. Thereafter, FMEA often is considered as the first step of designing the reliability and quality required in a system study. Figure 1 shows a progress of FMEA performed in the course of a process (product) development [11]. As shown in the figure, each of the five phases has its own FMEA tasks, from the initial conceptual prototype design to the final maintenance management through the design of interface with the prototyped system, the upgrade system design for more practices and the verification of the upgraded system. The successful verification makes the system useful and this usage has been kept by correct maintenance until its usage life reaches.

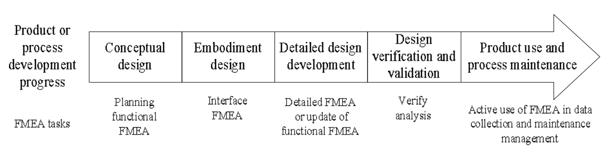

Figure 1 A general progress of FMEA tasks for a system life cycle

TABLE 1. AN FMEA FORM

FMEA Type (Design or process) :									Project Name /Description :					Date (Orig.) :		
Responsibility :									Prepared By :					Date (Rev.)		
Core Term :																
Item	Potential failure mode	Potential effects	Severity (S)	Class	Potential Causes	Occurrence (O)	Current actions	Detectivity (D)	RPN	Recommended actions	Responsibility & target Completion date	Actions taken	Severity (S)	Occurrence (O)	Detectivity (D)	RPN

Table 1 is a form of FMEA process used to record the status of system usage during its life cycle. The heading of each column can be modified or more information can be added into the form for the necessity of practical FMEA tasks. It should be noticed that one of the main FMEA tasks is to calculate the risk number RPN for each failure mode. The higher the risk number is, the more serious the failure could be, and the more cautious the failure mode is. For a detailed description of the FMEA process, see [12, 13]. It is usually advisable to determine the RPN value of each failure mode before completing the last columns of the table because in this way the corrective actions required against each item can be judged in the light of he ranked severity and the resources available. Technically, an FMEA process contains the following tasks [14]:

- Assign a label to each process's component.
- Describe the functions of each component.
- Identify potential failures for each function.
- Assess the effects of the failures.
- Find the causes of the failures.
- Estimate the probability of failure occurrence.
- Calculate the servility of the failure.
- Rank a priority for a severity.
- Address the highest severities.
- Determine the likelihood of detecting the failure
- Update the FMEA as corrected and improved actions are taken.
- Document historical FMEAs for future reference to aid in analysis of field failures and deliberation of process design changes.

These FMEA tasks will be logically and hierarchically mapped onto the intelligent N-FMEA architecture proposed in this paper. Generally, a traditional procedure used to perform the above FMEA tasks can be summarized as follows [13, 15].

Step1. Review systematically the component or constituent functions to ensure that any failure produces less damage to the entire system.

Step2. List all possible failure modes of each constituent reviewed via brainstorm process.

Step3. Observe the effects that each mode of failure would have on other constituents in the system and their functions.

Step4. Find all the possible causes of each failure mode.

Step5. Scale numerically the failure modes on a range of 1 to 10. Expert experience and reliability data should be used together to determine [16, 17]:

Step5-1. occurrence (O) of each failure mode, i.e., frequency of failure mode happens, (1 = very rare, 10 = very often).

Step 5-2. severity (S) of a failure, i.e., how serious is a failure occurrence impact (effect) on the system function. (1 = the least effect, 10 = the highest effect).

Step 5-3. detectability (D) of a failure occurrence, i.e., the difficulty of detecting a failure before the system is used. (1 = very easy, 10 = very difficult).

Step6. Compute RPN = O $\times$ S $\times$ D, for each failure mode. RPN indicates the relative priority of each mode in the failure prevention activities.

Step7. Rank RPN for each failure mode

Step8. Select the corrective actions required and record expected reliability and quality data.

Step9. Drive continuous risk assessment to reduce RPN of each failure mode for increasing the system usage safety.

Table 2 readies these steps of carrying out an FMEA process.

TABLE 2. A GENERAL PRROCEDURE PERFORMING FMEA

Step	FMEA Tasks
1	review the constituent functions
2	brainstorm possible failure modes
3	observe possible effects on the failure modes
4	find possible causes to the effects
5	scale risk parameters (1 to 10)
5 - 1	determine the frequency of failure occurrence
5 - 2	determine the severity of each failure mode
5 – 3	determine detectability of each failure mode
6	compute risk priority number (RPN)
7	rank RPN for each failure mode
8	take corrective actions to reduce RPN
9	drive continuous risk assessment (go back to step 1)

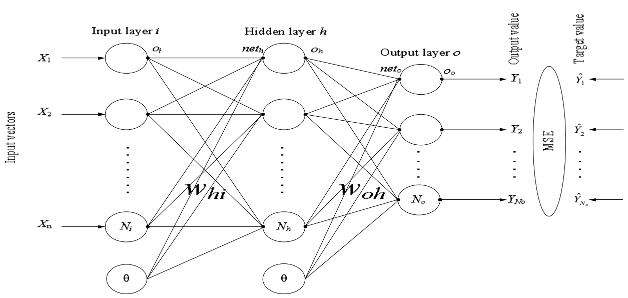

Figure 2. A standard BPN structure

III. ANN INTRODUCTION

An ANN structure is variously defined by several layers of artificial neurons, interconnections between layer neurons and an activation function of each neuron. A typical ANN has three sequentially combined layers, input, hidden and output layers, in each of them a neuron has its own summed up inputs that result in a different activation level for itself. The layered neurons may be fully or partially interconnected. Figure 2 draws a standard ANN structure with the three layers. The training method used the ANN is called supervised training because the target values are the supervisor that functions as to correct the errors between them and output values. Another type of an ANN training is unsupervised. The un-supervision means that there are no target values, and its training process is performed by self-organizing its own weights. A supervised ANN training algorithm, called back-propagation network (BPN) algorithm, used in Figure 2 is briefly addressed below, which was taken from the authors' previous paper [18]. Detailed introduction to ANN can be found in [9, 10, 19, 20].

A. Standard BPN

A typical BPN consists of a number of neurons organized into three layers. The first layer is called an input layer with one neuron for each variable or feature X_i of a pattern or a vector ($X_1, X_2, \ldots, X_n$) in a training data set in the form of pairs of ($X_i, \hat{Y}_{N_o}$); $\hat{Y}_{N_o}$ is the external known feature used as a target value to supervise the BPN output response Y_{No} for monitoring whether the BPN learning process is optimal. Similarly, the third layer functions as an output layer consisting of neurons in which each Y_{No} represents for a specified feature meaning to be analyzed. In between, there is a hidden layer also formed by a number of neurons accountable for activating and learning the input features. Neurons in one layer are fully connected to neurons of a succeeding layer. The BPN propagates its input vectors, which can be any multivariate data series, to its output layer through the hidden layer, so that each connection with quantified with a real number called a weight ($\boldsymbol{w}$) can be adapted (trained) to be optimal value by means of minimizing the MSE between Y_{No} and $\hat{Y}_{N_o}$; thus, the weights are thought of as the BPN's memorization or connection strength of the input vectors.

Mathematically, net_h , the input to neuron h is a collected signal come from each input neuron i, is given by:

$$net_h = \sum_{i=1}^{N_i} w_{hi} o_i \quad , h = 1, 2, \ldots, \quad Nh$$

(1)

where w_{hi} is the weight associated with the connection from input neuron i to hidden neuron h; o_i is the input-layer's output value which is the same as its own input feature X_i. Unlike o_i, o_h, the neuron output of the hidden layer, is determined by both the activation function f and net_h, the collected signal of neuron h. One common f is the sigmoid as follows:

$$o_h = f(net_h) = \frac{1}{1 + \exp[-(net_h + \theta)]}$$

(2)

where θ is a bias term or threshold value of neuron h responsible for accommodating non-zero offsets in the data. The computation of the input values net_o and output values o_o of the neurons in the output layer are similar with (1) and (2), respectively.

Once the output value o_o (=Y_o) estimated by the BPN has been obtained, the MSE can be calculated by:

$$\mathrm{MSE}(\boldsymbol{w}) = \sum_{o=1}^{No} \left(Y_o - \hat{Y}_o\right)^2$$

(3)

Equation (3), the objective function of the training procedure, is used to find a set of optimal weights (and biases) that permit the BPN to estimate its output values ($\boldsymbol{Y}$) as close as possible to the external known targets ($\hat{\boldsymbol{Y}}$), in order to minimize their error function MSE($\boldsymbol{w}$). The adaptation of the weights completes when the MSE($\boldsymbol{w}$) satisfies the previously regulated convergence criteria. At this point, the BPN is considered trained, and then the trained BPN may be used to evaluate its generalization or solution capability (solvability) by using other un-training sets, known as the validation set.

B. *Why ANN for FMEA*

There are many classical and standard FMEA systems available to correlate the causality and severity among system features (parameters). They all have their own advantages and disadvantages but an obvious weakness is that they cannot tackle an unknown or unforeseen failure event, which means they can only identify what has already saved in their data base. This significant weakness causes a classical FMEA system not to

(1) easily integrate with other reliability design systems,
(2) promptly associate failure modes, effects, causes with corrections,
(3) appropriately interpret a defect process,
(4) necessarily memorize the newly occurred failures, and
(5) competently predict future failures.

These disadvantages, fortunately, can be made up by a modern machine learning technique, such as ANN.

IV. LITERATURE REVIEW

It has been not a novel idea of taking advantage of ANN to fault identification and classification. Watanabe,. Hou and Himmelblau [7] presented a hierarchical ANN system to recognize the faults of a chemical reactor. The system diagnosed faults and classified based on the known operating data. Its hierarchy divided a large number of data patterns into several smaller subsets so that the classification could be carried out more accurately. One of its advantages was that multiple faults can be detected in unseen data even if the system was trained with data representing single faults. Kurd and Kelly [21] applied a fuzzy ANN to a FMEA problem. A fuzzy self-organizing map was developed to deal with the qualitative and quantitative features. The fuzzy ANN map enabled the association of the features for mitigation of potential failure modes and was used for approximating the value of criticality. Failure modes associated with functional properties were systematically identified using HAZOP guide words. This approach ensured that a wide range of potential hazardous failures for specific applications were discovered. Sharma et al. [22] applied fuzzy methodology (FM) combined with Petri net, as an approximate reasoning tool to deal with the imprecise, uncertain and subjective information related to system performance. Various reliability parameters, e.g., repair time, failure rate, mean time between failures, availability and expected number of failures, are computed to quantify the uncertain behavior of system; further, the FM was demonstrated to rank RPN values of a traditional FMEA where the uncertain parameters cannot be treated with. Tian and Noore [23] took ANN trained by genetic algorithm to predict the cumulative failure time of software, the predictability was increased because of the non-differential optimization method.

Su and Huang [24] also demonstrated that a probability of software reliability of a system can be predicted for estimating software reliability growth trend. The last two papers imply that ANN can be used to estimate the probability of a failure mode occurrence in an FMEA. Boyd et al. [6] addressed an expert system (ES) method, and supported with NASA Ames Research Center, to build up a computerized diagnosis system for real-time systems that was based on the integration of a knowledge base derived from design-phase FMEA with the inference engine embedded in a real-time data parameter monitoring program. An example system from the commercial automotive industry was demonstrated the ES method. This real-time ES for dealing with FMEA problems was one of the inspirations on developing the N-FMEA system described below.

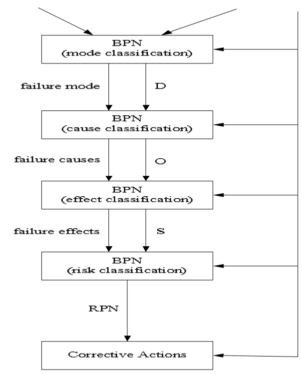

Figure 3. Conceptual N-FMEA hierarchy

V. IMPLEMENTATION OF INTELLIGENT N-FMEA

Based on the authors' practice experience of implementing real-case of an automatic statistic FMEA system [25] and inspired by the ideas described in the previous section of the literature review, this section will depict N-FMEA as an automatic FMEA system structured with a hierarchical BPN. Figure 3 conceptualizes the workflow of the N-FMEA functional performing. The first BPN block functions the classification of failure modes whose features or signals come from the simulated FMEA database being examined and tested. The FMEA database stored a lot of experiences and experimental information of reliability experts. It was set up with the results of the expert brainstorming colloquium. Both system features and FMEA experienced data are presented to N-FMEA for its training. The trained N-FMEA can be used to identify the unseen features for checking that their mode(s) is (are) normal or failure. The subsequent BPN block associates the identified failure modes with the corresponding causes that made the failure occurrences. The effects resulted from the cause occurrences are then respectively related each other. These respective relations are accomplished by the third BPN block. After the three risk parameters, D, O and S, were obtained from the previous three BPN blocks, the failure risk (RPN) of the whole system being tested can be calculated, and the corrective actions corresponding to the failure modes identified can be taken from the FMEA database.

Figure 4 details the hierarchy of N-FMEA. The hierarchy formed by BPNs is one sort of an effective problem-solving structure composed of several levels in which each level contains a number of BPNs used to do the FMEA classification tasks. Usually, it clarifies a sophisticate FMEA problem, which a single BPN difficultly solves, into a number of sub-problems or of specified problems and outclasses the result of a single BPN done.

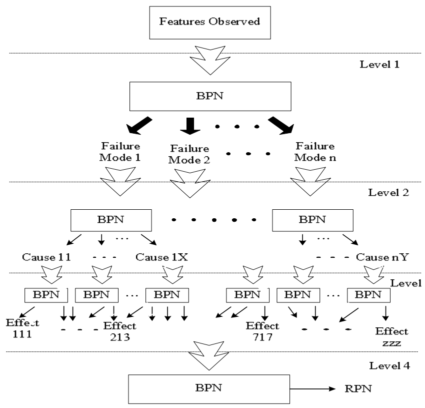

Figure 4. Detailed N-FMEA hierarchical structure

In the figure, the first level has only a single BPN used to discriminate failure modes from a normal mode in terms of testing signals generated from a simulated process. The FMEA data base was set up with the results of brainstorming colloquium. For doing the experiments of the proposing N-FMEA system, the FMEA database was [25] built up with Execl tool. The number of BPNs within the second level is the same as the number of failure modes defined in the first level. Each second-level BPN will be subsequently used to determine the cause(s) or factor(s) of each failure occurred in the previous level. After the failure cause(s) was determined, each cause's corresponding effect(s) is associated by the respective BPN in the third level. This hierarchical BPN-based FMEA system disaggregates a complex statistic FMEA procedure and correlates the super- and sub-ordination between the separated subsystems; that is, the top level is allowed to set targets to its subordinated level for associating causes with effects. BPN of the lowest level functions as a mechanism for the RPN computation. Detailed training and testing data will be shown in the next section for the demonstration of N-FMEA.

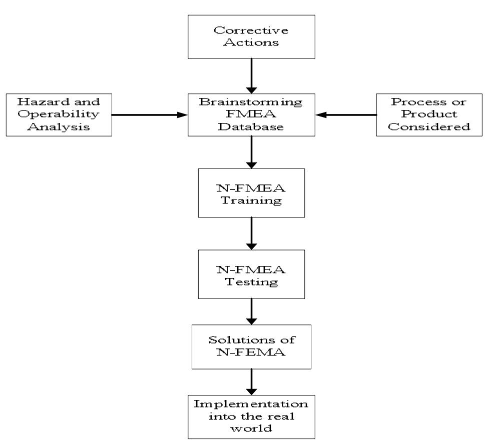

Figure 5. The N-FMEA training and testing procedure

VI. SIMULATED EXPERIMENTS OF N-FMEA

Figure 5 shows the workflow of the N-FMEA training and testing. After the FMEA database was brainstormed and established by the experienced reliability experts, the examples for the N-FMEA training can be performed, and then carries out the testing phase. The solution phase examines the system to get its FMEA information and N-FMEA can be implemented for real-case study. Partial portion of training and testing data are shown in Tables 3 to 8. Table 3 has two examples for training possible failure modes and their corresponding FMEA data collected from the brainstorming FMEA database. The entry values in the table can be obtained from Table 4 to Table 8 in which each table describes the data needed to do the FMEA tasks. These data were made hundreds of combinations or correlations for the simulated failure modes, causes, effects and their respective RPN values. The combinations were partitioned so that 85% of them were used to train N-FMEA, while the remaining 15% was reserved for the testing of validation. Training of the local BPNs was achieved using back-propagation of error (0.001). One combination is to a single failure mode and its relative FMEA data. Table 9 and Table 10 are the examples of such combinations. Lots of other various combinations can be found in [26], and more N-FMEA performance experiments can be available in it as well. There 30 BPN networks were trained to learn the simulated FMEA data. Table 11 is one of N-FMEA outputs for RPN computation. The error between the N-FMEA estimated and the manual RPNs are shown in the table.

TABLE 3. AN EXAMPLE USED TO TRAIN N-FMEA

Possible Failure Mode	Possible Failure Cause	Detect-ability (D)	Occurr-ence (O)	Seve-rity (S)	RPN	Rank
X	x1	2	3	7	42	5
	x2	6	4	5	120	1
	y1	3	5	3	45	4
Y	y2	9	2	6	108	2
	y3	1	8	9	72	3

TABLE 4. TRAINING DATA OF FAILURE OCCURRENCE

Degree	Score	Probability of Occurrence (O)
Very low	1	0
Low	2	1/10000
	3	1/5000
Moderate	4	1/2000
	5	1/1000
	6	1/500
High	7	1/200
	8	1/50
Very high	9	1/5
	10	1/2

TABLE 5. TRAINING DATA OF DETECTABILITY

Degree	Score	Detect-ability (D)
Very low	1	0-5
	2	6-15
Low	3	16-25
	4	26-35
Moderate	5	36-45
	6	46-55
	7	56-65
High	8	66-75
	9	76-85
Very high	10	86-100

TABLE 6. TRAINING DATA OF SEVERITY

Degree	Severity (S)
Very less problem	1
A minor problem	2 3
Dissatisfaction	4 5 6
High dissatisfaction	7 8
Serious consequences for safety	9 10

TABLE 7. TRAINING DATA OF D, O AND S

D	O	S	Degree
1	1	1	very low
2, 3	2, 3	2,	low
4, 5, 6	4, 5, 6	4, 5, 6	middle
7, 8	7, 8	7, 8	high
9, 10	9, 10	9, 10	very high

TABE 8. TRAINING DATA OF RPN

RPN (scale range)	Class score	Risk degree
1-50	25	very low
50-100	75	very low to low
100-150	125	low
150-250	200	low to middle
250-350	300	middle
350-450	400	middle to high
450-600	525	high
600-800	700	high to very high
800-1000	900	very high

TABLE 9. TRAINING SAMPLES FOR FAILURE MODES CLASSIFICATION

Sample	Feature Pattrens						Failure Modes	Sample	Feature Pattrens						Failure Modes
	X_1	X_2	X_3	X_4	X_5	X_6			X_1	X_2	X_3	X_4	X_5	X_6	
1	0.40	0.53	0.41	0.44	0.46	0.53	Normal	41	0.09	0.10	0.03	0.45	0.53	0.45	Mode 2
4	0.50	0.41	0.40	0.47	0.51	0.44	Normal	44	0.19	0.07	0.02	0.41	0.44	0.44	Mode 2
12	0.44	0.59	0.57	0.48	0.52	0.55	Normal	52	0.18	0.18	0.14	0.98	0.88	0.96	Mode 3
13	0.54	0.50	0.47	0.58	0.46	0.51	Normal	53	0.04	0.04	0.10	0.85	0.85	0.83	Mode 3
14	0.53	0.53	0.55	0.56	0.54	0.53	Normal	54	0.18	0.16	0.19	0.83	0.84	0.84	Mode 3
23	0.49	0.44	0.57	0.56	0.44	0.58	Normal	63	0.89	0.94	0.86	0.13	0.18	0.06	Mode 4
29	0.54	0.42	0.52	0.41	0.58	0.41	Normal	69	0.89	0.94	0.81	0.06	0.19	0.12	Mode 5

TABLE 10. TRAINING SAMPLES FOR FAILURE SEVERITY CLASSIFICATION

Sam-ple	F_1	F_2	F_3	F_4	F_5	C_E	Seve-rity	Sam-ple	F_1	F_2	F_3	F_4	F_5	C_E	Seve-rity
1	0.99	0.57	0.71	0.79	0.76	0.24	II	41	0.14	0.30	0.79	0.59	0.72	0.01	IV
2	0.92	0.59	0.73	0.75	0.78	0.23	II	42	0.12	0.37	0.72	0.55	0.75	0.01	IV
5	0.92	0.92	0.75	0.75	0.78	0.37	I	45	0.19	0.36	0.72	0.58	0.78	0.02	IV
6	0.99	0.19	0.97	0.73	0.72	0.10	III	46	0.39	0.39	0.79	0.58	0.74	0.05	IV
7	0.97	0.15	0.96	0.73	0.76	0.08	IV	47	0.37	0.35	0.77	0.59	0.72	0.04	IV
8	0.98	0.64	0.90	0.78	0.74	0.33	I	48	0.38	0.34	0.78	0.59	0.74	0.04	IV
9	0.94	0.60	0.92	0.74	0.80	0.31	I	49	0.34	0.32	0.74	0.54	0.75	0.03	IV

TABLE 11. TRAINING SAMPLES FOR RPN COMPUTATION

Sam-ple	RPN	error	Sam-ple	RPN	error	Sam-ple	RPN	error	Sam-ple	RPN	error
1	5.8751	-0.3752	21	1.6054	0.0097	41	0.7118	-0.0045	61	0.2378	-0.0358
2	0.0154	0.0001	22	0.0039	0.0002	42	0.7079	0.0093	62	0.9176	-0.1016
3	0.2712	0.0126	23	0.0564	-0.0018	43	0.0057	-0.0004	63	4.0122	-0.0930
4	3.3524	-0.5288	24	0.8219	0.0219	44	4.5512	-0.2697	64	0.9830	-0.0244
5	0.0065	-0.0003	25	65.713	-0.2853	45	0.0544	0.0039	65	0.0390	-0.0033

REFERENCES

[1] M.C. Eti, S.O.T. Ogaji and S.D. Probert, **“**Reducing the cost of preventive maintenance (PM) through adopting a proactive reliability-focused culture**,”** Applied Energy, vol. 83, no.11, pp.1235-1248, 2006.

[2] J-Z Tang and Q-F Wang, “Online fault diagnosis and prevention expert system for dredgers,” Expert Systems with Applications, vol.34, no.1, pp.511-521, 2008.

[3] L. P. Chao and K. Ishii, “Design Process Error Proofing: Failure Modes and Effects Analysis of the Design Process,” Journal of Mechanical Design, vol.129, no.5, pp.491-501, 2007.

[4] Goddard, P.L., “Software FMEA techniques**,”** Proceedings Annual Reliability and Maintainability Symposium, 2000, pp.118 – 123.

[5] B. Tekinerdogan, H. Sozer and M. Aksit, “Software architecture reliability analysis using failure scenarios,” Journal of Systems and Software, 2007, in press.

[6] Boyd, M.A.; Abou-Khalil, A.; Montgomery, T.A.; Gebrael, M., “Development of automated computer-aided diagnostic systems using FMECA-based knowledge capture methods,” Proceedings Reliability and Maintainability Symposium 1998, pp.285 – 291.

[7] K. Watanabe, S. Hirota, L. Hou and D. M. Himmelblau, “Diagnosis of multiple simultaneous fault via hierarchical artificial neural networks,” AIChE Journal, vol 40, no 5, 1994, pp.839-848.

[8] Jeng-Bin Li and Yun-Kung Chung, “A Novel Back-propagation Neural Network Training Algorithm Designed by an Ant Colony Optimization,” IEEE/PES Transmission and Distribution Conference and Exhibition: Asia and Pacific, 2005, pp.1 – 5.

[9] Samarasinghe, S., Neural Networks for Applied Sciences and Engineering: From Fundamentals to Complex Pattern Recognition, Taylor and Francis, 2006.

[10] Ripley, Brian D., Pattern Recognition and Neural Networks, Cambridge University Press**,** 2008.

[11] Bowles, J.B., “The new SAE FMECA standard,” Proceedings Annual Reliability and Maintainability Symposium, 1998, pp.48 – 53.

[12] R. E. McDermott, R. J. Mikulak and M. R. Beauregard, The Basics of FMEA, Productivity Press, 1996.

[13] B. Andersen and T. Fagerhaug, Root Cause Analysis: Simplified Tools and Techniques, 2nd ed., ASQ Quality Press, 2006.

[14] SAE ARP 5580, "Recommended Failure Mode and Effects Analysis (FMEA) practices for non-automobile applications," Society of Automotive Engineers, Aerospace Recommended Practice, ARP 5580, 2001.

[15] S. Simani, “Fault diagnosis of a simulated industrial gas turbine via identification approach,” International Journal of Adaptive Control and Signal Processing, vol.21, no.4, pp.326-353, 2007.

[16] Bowles, J.B., “An assessment of RPN prioritization in a failure modes effects and criticality analysis,” Proceedings Annual Reliability and Maintainability Symposium, 2003, pp. 380 – 386.

[17] J. K. Chen, “Utility Priority Number Evaluation for FMEA,” Journal of Failure Analysis and Prevention, vol.7, no.5, 2007, pp.321-328.

[18] Jeng-Bin Li and Yun-Kung Chung, “A Novel Back-propagation Neural Network Training Algorithm Designed by an Ant Colony Optimization,” IEEE/PES Transmission and Distribution Conference and Exhibition: Asia and Pacific, 2005, pp.1 – 5.

[19] Simon Haykin, *Neural Networks: A Comprehensive Foundation*, 2nd Ed., Macmillan College Publishing, New York, 2001.

[20] Marti, R. and El-Fallahi, Abdellah, “Multilayer neural networks: an experimental evaluation of on-line training methods,” *Computers and Operations Research*, vol. 31, pp. 1491-1513, August, 2004.

[21] Z. Kurd and T. P. Kelly, “Using fuzzy self-organizing maps for safety critical systems,” Reliability Engineering & System Safety, vol.92, no.1, 2007, pp.1563-1583.

[22] R. K. Sharma, D. Kumar and P. Kumar, “Predicting uncertain behavior of industrial system using FM—A practical case,” Applied Soft Computing, vol.8, no.1, 2008, pp.96-109.

[23] L. Tian and A. Noore, “Evolutionary neural network modeling for software cumulative failure time prediction,” Reliability Engineering & System Safety, vol.87, no.1, 2005, pp.45-51.

[24] Y-S Su and C-Y Huang, “Neural-network-based approaches for software reliability estimation using dynamic weighted combinational models,” Journal of Systems and Software, vol.80, no.4, 2007, pp.606-615.

[25] Chen, T.T., Yun-Shiow Chen and Yun-Kung Chung, “Implementation of an enhanced FMEA system for the PCBA design testing - a practical case study,” International Conference on Business of Electronic Product Reliability and Liability, 2004, pp.72 –76.

[26] Chiang Ku, An Intelligent FMEA System Constructed by a Hierarchical Artificial Neural Networks, Mater Thesis, Yuan Ze University, Taiwan, 2007. (in Chinese).