Received March 3, 2020, accepted March 14, 2020, date of publication March 19, 2020, date of current version March 31, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2981872

A Case Study for Software Quality Evaluation Based on SCT Model With BP Neural Network

BEN YAN[1], HUA-PING YAO[1], MASAHIDE NAKAMURA[2], (Member, IEEE), ZHI-FENG LI[1], (Student Member, IEEE), AND DONG WANG[3]
[1]Luoyang Institute of Science and Technology, Luoyang 471023, China
[2]Graduate School of System Informatics, Kobe University, Kobe 657-8501, Japan
[3]Nara Institute of Science and Technology, Ikoma 630-0101, Japan

Corresponding author: Hua-Ping Yao (yanbenjp@gmail.com)

This work was supported in part by the National Natural Science Foundation of China under Grant U1604149, and in part by the Henan Province Education Department Cultivation Young Key Teachers in University under Grant 2016GGJS-157.

ABSTRACT With the increasing for function, scale, hierarchy and complexity of software project, the software life cycle and development stage show a trend of cross-cutting and fuzzy boundary. The non-technical factors, such as poor management and control during the implementation of software projects, are the major reason for causing the low success rate of software projects recently. Therefore, the software quality evaluation under complex environment should take the cross-influence between different stages of software life cycle and different quality evaluation standards into consideration. Our research is to construct a new software quality evaluation model by using the influence relationship and the influence intensity index between project management domain and project quality evaluation criteria including scope, cost, and time. First, we came up with the definition of software project management domain in the process of software project development and management. Second, we proposed a mathematical method for extracting the direct or indirect influence relation between them, and give a definition for the quantitative evaluation index and its calculation formula. At last we proposed to construct a neural network training model which includes evaluation model logic relationships and software quality quantitative evaluation index. Through study and training by simulated software project management data, we can discover some key data, such as normal threshold range of influence, factor weights, etc. Therefore, a complete evaluation system is built, and the scientific nature and accuracy of the proposal evaluation system will be improved.

INDEX TERMS Software quality evaluation model, back propagation neural network, project management domain, project sub-management domain, SCT, CMMI.

I. INTRODUCTION

According to the Standish group's 2015 report, about 70% of large software projects in the world are unsuccessful, either planned to be delayed, or over budgeted, or lacking in functionality etc. [1]. There are many factors for causing the low success rate of software projects, especially the non-technical factors, such as poor management and control during the implementation of software projects, which are the major reason for causing the low success rate of software projects recently [4]. This is related to the traditional quality control theory and evaluation system of software project, which focus on the local evaluation of different development stages of software life cycle. However, the cross-influence between different stages of software life cycle and different quality evaluation standards is not taken into consideration. Especially with the increasing for function, scale, hierarchy and complexity of software project, the software life cycle and development stage show a trend of cross-cutting and fuzzy boundary, which makes the analysis factors, the cross relationship of these intersection factors in the process of software project management and quality evaluation became more and more complex, and the negative impact of the traditional software project quality control evaluation system becomes more prominent.

The associate editor coordinating the review of this manuscript and approving it for publication was Porfirio Tramontana.

The theory and method of traditional software engineering have solved some quality issues in project development and management. However, the problem of how to improve the success rate of software projects is not properly solved. In order to solve the issues for software quality evaluation under complex environment of software project, we propose a set of theories and methods based on the software project management data.

In our study, we first proposed and redefined the software project management domain based on the research results of actual projects, then we analyzed and summarized the direct influence relationship between project management domain, sub management domain and SCT (scope, cost, time). After this, the relation topological diagram is established and apply the mathematical method of incidence matrix to further analyze and excavate the indirect influence relation among them.

Based on the proposal above, by analyzing the direct and indirect influence relationship among the project management domain, sub-management domain and the quality evaluation standard SCT for a specific project, qualitative analysis and evaluation can be realized for the completion quality of the project, but the disadvantages of this proposal are also obvious, such as the qualitative analysis is easy to be affected by subjective factors, lack of analysis accuracy, inaccurate comparison etc.

In order to further complete the evaluation index quantitative evaluation based on the above theory, we came up with quantitative evaluation index (direct impact contribution rate, indirect impact contribution rate, and cumulative impact contribution rate) and its calculation formula, which basically solve the problem of quantitative evaluation method based on software project management domain as well as project quality evaluation standard SCT.

With the advancement of our research, we also find that the impact contribution rate and cumulative impact contribution rate will be affected by the different characteristics of software projects, such as code volume, technical framework, and even team culture etc. Thus, it is important to accurately define the weight of the above indicators in different projects in the whole evaluation system.

For further improving the scientific nature and accuracy of the proposal evaluation system, this article proposed to construct a three-layer neural network training model based on BP artificial neural network technology, which includes the logical relationship of the evaluation model as well as the quantitative evaluation index of SCT. Furthermore, we also proposed to define the sub management domains as the input layer, the management domains as the hidden layer, and the SCT as the output layer. The series weights between the input layer and the hidden layer represent the impact contribution rate of the sub-management domain to the management domain (including direct and indirect contribution rate), the series weights between the hidden layer and the output layer represent the cumulative impact contribution rate of the management layer to SCT.

At last, we conducted an experiment for learning, training and testing with the simulated software project management data, and discover some key data including the normal threshold range of influence factor weights, evaluation criteria and SCT evaluation criteria. According to the experimental results, the proposal in this article is proved effective and scientific.

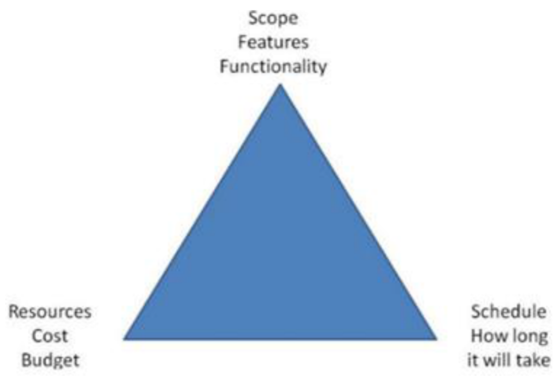


FIGURE 1. SCT triangle.

II. PRELIMINARIES

A. THE SOFTWARE EVALUATION CRITERION SCT

SCT is the abbreviation of words scope, cost and time [8]. Project management is the planning, monitoring and control of all aspects of a project and the motivation of all those involved to achieve the project objectives on time (T) and to the specified cost (C), quality and performance (S). If you change one of the three variables, the laws of project management say that one of the others has to change too. Figure 1 is a diagram of SCT triangle

For example, when the scope of a project added, the time or the cost has to go up as well. When a project wants to deliver in less time, the budget has to increase for increasing human resources or reduce the scope of this project. When the cost of a project has reduced, it means that the scope or the time of this project has to reduced too.

In this paper, we consider SCT as a criterion for evaluating the success of software projects. For a software project A, if it is completed within the established budget and established function on time, we judge that A is a success project. Otherwise, we judge that A is a failed project.

B. THE THEORETICAL BASIS OF SOFTWARE PROJECT MANAGEMENT DOMAIN

In the field of software engineering, CMMI is a relatively mature mode, which is suitable for software product development process management in large and medium-sized enterprises. Its core idea is to divide the process of software development into several both related and independent subprocesses. Through monitoring and researching each subprocess, we can control the whole process of software product development effectively, so as to ensure the quality of software products which are developed.

In the CMMI model, five levels are defined to evaluate the maturity of software product development (completion level, management level, definition level, quantitative management level, and optimization level), and each level is the basis for the next level. The evaluation system of CMMI also divides the process of software development into 22 process domains. It defines the CMMI management level of the company by evaluating the completion quality of each

TABLE 1. Defination of process area in CMMI model.

No.	Process Area	Level				
		Initial	Repeatable	Defined	Managed	Optimizing
1	Requirements Management		√	√	√	√
2	Project Planning		√	√	√	√
3	Project Monitoring and Control		√	√	√	√
4	Supplier Agreement Management		√	√	√	√
5	Measurement and Analysis		√	√	√	√
6	Process and Product Quality Assurance		√	√	√	√
7	Configuration Management		√	√	√	√
8	Requirements Development			√	√	√
9	Technical Solution			√	√	√
10	Product Integration			√	√	√
11	Verification			√	√	√
12	Validation			√	√	√
13	Organizational Process Focus			√	√	√
14	Organizational Process Definition			√	√	√
15	Organizational Training			√	√	√
16	Integrated Project Management			√	√	√
17	Risk Management			√	√	√
18	Decision Analysis and Resolution			√	√	√
19	Organizational Process Performance				√	√
20	Quantitative Project Management				√	√
21	Organizational Innovation and Deployment					√
22	Causal Analysis and Resolution					√

process domain and the quantity of achieved process domains (TABLE 1) [47], [48].

Recently, more and more software companies have obtained CMMI certification. However, there are few companies which manage the actual development process according to the requirements of CMMI model, and there are two important reasons for it: The first one is that the CMMI system shows only ''what to do, but do not explain how to do it'', and the second one is that it ''ignores the importance of person's maturity in implementation'' [45].

In our laboratory, we have another research team for conducting some study to solve the above problems. The goal is to propose a method to improve their management level of software product development at a lower cost in a relatively short period of time, and finally approach or meet the management requirements of CMMI model. The core idea of the proposal is to enable the person who is familiar with traditional software development processes (solving the problem of people maturity) to achieve the above goals by using process domain of modified traditional software development process management (solving the problem of how to do it).

For this, based on referring to the 9 major knowledge systems of project management, and discussing with several project managers, the research team proposed to divide the scope of management which affects software quality into two levels: call the project management domain and project sub-management domain.

The project sub-management domains define more specific and detailed contents during software project management process, and the contents defined by each sub-management domain have a certain mapping relationship with the process area defined in CMMI model. On the other hand, project management domain refers to a collection of

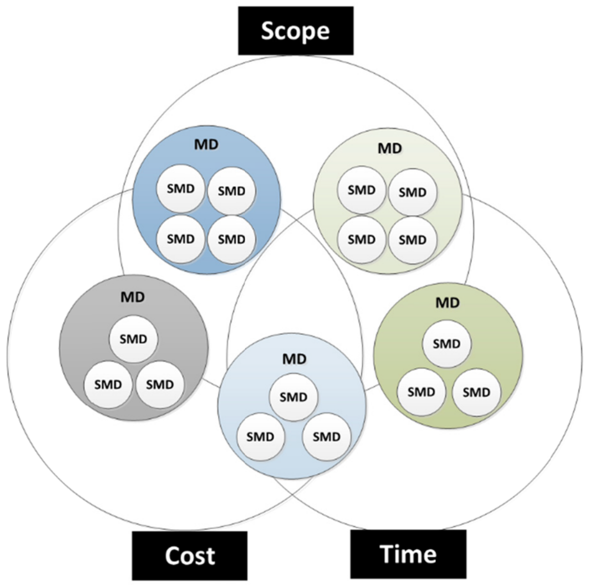


FIGURE 2. The relation between management domains and SCT.

some project sub-management domains with certain similar attributes, and it is also a management perspective classification that cannot be ignored in the development process of any software project. The above definition is in line with most project managers' current understanding of software project management priorities, and it is also the specific management content that project managers with certain experience are familiar with. Following such a proposal for software project management can largely solve the two main problems in implementing CMMI model mentioned above.

Figure 2 is a schematic diagram of the relationship among project sub-management domain, project management domain and SCT. Among them, MD is the abbreviation of management domain, which represents a certain management domain. SMD is the abbreviation of sub-management domain, representing a certain sub-management domain. As can be seen from the figure, a certain management domain is a collection of some sub-management domains. The completion degree of a certain sub-management domain will directly affect the completion degree of the management domain, and indirectly affect the quality of one or more SCT indicators.

The concept of management domain and sub-management domain mentioned in the article refers to a classification of a modified software development process domain based on the above research. The detail of the definition for project management domain and sub-management domain shows in chapter IV.

C. THE BP NEURAL NETWORK MODEL

BP neural network is a kind of multi-layer front feed neural network trained by the error back-propagation algorithm. It generally consists of three layers, the input layer, the middle layer (hidden layer) and the output layer [11]. The main characteristic is forward propagation of signal as well as back propagation of error. In the process of forward transmission, the input signal is processed layer by layer and transmitted down to the output layer. If the output layer does not get the desired output, the back propagation will be carried out and the weight threshold is adjusted layer by layer according to the feedback error until it returns to the input layer. It will be trained repeatedly until it reaches a preset error or number of sessions. It is similar to a nonlinear function, and its network input value and the predicted value are the independent and dependent variables of the function respectively [13], [22].

For example, when the number of input nodes is n and the number of output nodes is m, the BP neural network reflects the functional mapping from n independent variables to m dependent variables.

BP neural network has the ability of self-learning, it essentially achieves a mapping function from input to output, which can automatically select "reasonable" solution rules by learning the set of instances with correct answers. Its nonlinear processing power is a perfect way to deal with fuzzy information, incomplete and contradiction complex situation cognitive judgment problem, and problems with complicated internal mechanism. Therefore, it is suitable for software quality evaluation model based on the software management domain (sub domain) and SCT evaluation criterion proposed in this article.

III. RESEARCH GOAL AND APPROCAH

The goal of our research is to propose a new software quality evaluation model for analyzing and evaluating the quality of software projects which is based on the software project management data. For this, we plan 4 steps to achieve this goal.

1) STEP1: Redefining the basic concepts of management domain, sub-management domain, as well as relationship among management domain, sub-management domain and SCT with referencing the tradition software project management theory.

2) STEP2: Analyzing and extracting the logic relationships which are hidden in the management domain, sub-management domain and SCT, to establish the new model structure of software quality evaluation.

3) STEP3: Analyzing and defining a mathematical model for evaluating the impact contribution rate and cumulative impact contribution rate of the proposed model based on SCT evaluation criteria.

4) STEP4: Building and training the neural network training model based on the software project management data, to discover some key data, such as impact factor weights, the normal threshold range of evaluation standard and normal threshold range of SCT evaluation criteria.

IV. THE KEY IDEA OF PROPOSAL

A. THE DEFINATION OF SOFTWARE PROJECT MANAGEMENT DOMAIN

In this article, we select the proposal which defines the scope of software project management as 5 management domains and 17 sub-management domains. Based on the discussion of chapter II above, we know that the project management domain refers to a collection of some project sub-management domains with certain similar attribute, the project sub-management domains represent some specific and detailed contents during software project management process, and each sub-management domain has a certain mapping relationship with the process area in CMMI model.

Table 2 shows the details of these domains and the mapping relationship between them. The left side of the table 2 shows the specific contents of management domain and sub management domains, the right side shows the mapping relationship between sub-management domains and the process area in CMMI model. The number of project management domain in the table 2 represents each sub-management domain. This number will be used in the later discussion. The number of CMMI part in the table 2 corresponds to the number in Table 1 for easy reading.

In table 2, the management domain includes project management domain (*PM*), quality management domain (*QM*), process management domain (*PRM*), plan management domain (*PLM*) and configuration management domain (*CM*). Furthermore, the project management domain includes 4 sub-management domains, they are risk sub-management domain (*RM*), team sub-management domain (*TM*), HR sub-management domain (*HRM*) and team communication sub-management domain (*TCM*); The quality management domain includes software quality sub-management domain (*SQM*), software standards sub-management domain (*SSM*), software reviews sub-management domain (*SRM)* and software measurement sub-management domain (*SMM)*; The process management domain includes process measurement sub-management domain (*PMM*), process analysis sub-management domain (*PAM*) and process change sub-management domain (*PCM*); The plan management domain includes software pricing sub-management domain (*SPM*), project scheduling sub-management domain (*PSM*) and software design sub-management domain (*SDM*); The configuration management domain includes system version sub-management domain (*SVM*), system building sub-management domain (*SBM*) and system release sub-management domain (*SRM*).

B. THE METHOD FOR EXTRACTING INFLUENCE RELATIONSHIP

On the basis of further communication, discussion and analysis with actual project managers, a topological diagram that can reflect the direct influence relationship between different management domains, sub-management domains and project comprehensive evaluation standard SCT is established (see figure 4).

In the figure 4, the circle represents the 17 sub-management domains defined, the rectangle represents the three SCT evaluation criteria, and the vector arrow represents the relationship of direct influence, the affecting party and the affected party. Among them, since the influence

TABLE 2. Defination of project management domain.

Project Management Domain	Project Sub Management Domain		CMMI(Capability Maturity Model Integration)	
	No.	Sub Management Domain	Process Area	No.
PM Project Management	1	RM: Risk management	Risk Management	17
	2	TM: Team management	Organizational Process Performance	19
	3	HRM:HR management	Organizational Innovation and Deployment	21
	4	TCM: Team Communication management	Organizational Training	15
QM Quality Management	5	SQM: Software quality management	Process and Product Quality Assurance	6
	6	SSM: Software standards management	Quantitative Project Management	20
	7	SRM: Software reviews management	Validation	12
			Project Monitoring and Control	3
	8	SMM: Software measurement management	Verification	11
PRM Process Management	9	PMM: Process measurement management	Measurement and Analysis	5
	10	PAM: Process analysis management	Organizational Process Focus	13
	11	PCM: Process change management	Organizational Process Definition	14
PLM Plan Management	12	SPM: Software pricing management	Supplier Agreement Management	4
	13	PSM: Project scheduling management	Project Planning	2
	14	SDM: Software design management	Requirements Management	1
			Requirements Development	8
			Technical Solution	9
			Decision Analysis and Resolution	18
			Causal Analysis and Resolution	22
CM Configuration Management	15	SVM: System version management	Integrated Project Management	16
	16	SBM: System building management	Product Integration	10
	17	SRM: System release management	Configuration Management	7

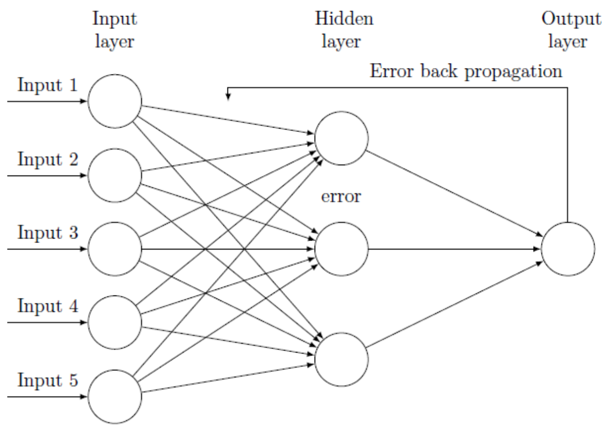


FIGURE 3. The structure of BP neural network model.

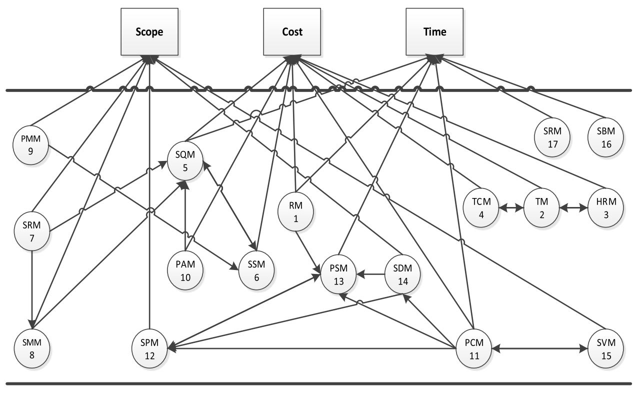


FIGURE 4. The topologies diagram of software management domain.

relation between different management domains can be replaced by the influence relation between the sub-management domains belonging to them, the topology diagram removes the description of the relationship between the management domains. In order to further facilitate the drawing and analysis, each sub-management domain is specified with a number and an abbreviated name.

In the framework of the above software project implementation process management system, the SCT evaluation criteria is also the comprehensive evaluation criteria for the management results of the entire management system in the above project implementation process. For example, when a project finished, there are fewer functional areas (S) actually completed than planned for the initial phase of the project, the reason is that the time (T) factors may be too tight or there is changes in the human resources sub-management domain of the cost management domain. If the time (T) spent exceeds the expected time (T) of the project, the possible reason is that the project budget (C) is reduced.

From the figure 4, we can see that there are complex influence relationships between specific management domain and SCT index, which means it has not only the direct effect relationship, but also the indirect effect relationships. Therefore, how to extract all the effect relationship is a important element in the quality evaluation of this study.

In this article, we proposed to use matrix calculation to solve the above issues. First, the topological diagram representing the direct impact relationship is described in the form of the adjacency matrix (A). The row position elements in the matrix is arranged from left to right, and the column position elements is arranged from top to bottom, which represent the sub-management domains from 1 to 17. If the median value of the position is 1 in the matrix (A), it indicates a direct relationship between row position elements and column position elements.

$$A = \begin{bmatrix}
0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0\\
0&0&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&1&0&0&1&0&0&0&0&0&0&0&0&0\\
0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&1&1&1&1&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&1&1&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0
\end{bmatrix}$$

After this, by calculating matrix (A), arrival matrix (RD) can be obtained. The arrival matrix (RD) contains all the influence relationships between sub-management domains (position of 1 in the matrix).

$$RD = \begin{bmatrix}
0&0&0&0&0&0&0&0&0&0&0&1&1&0&0&0&0\\
0&1&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&1&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&1&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&1&1&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&1&1&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&1&1&0&1&0&0&0&0&0&0&0&0&0\\
0&0&0&0&1&1&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&1&1&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&1&1&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&1&1&1&1&1&0&0\\
0&0&0&0&0&0&0&0&0&0&0&1&1&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&1&1&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&1&1&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&1&1&1&1&1&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0
\end{bmatrix}$$

Based on the proposed method above, it solves the extraction and analysis logical relationship problems among the management domain, sub-management domain and the SCT evaluation standard, so as to provide theoretical basis and solutions for software project management data logical relational model data mining.

C. THE ESTABLISHMENT OF EVALUATION INDEXES IN SOFTWARE QUALITY COMPREHENSIVE EVALUATION SYSTEM

Based on the proposal above, by analyzing the direct and indirect influence relationship among the project management domain, sub-management domain and the quality evaluation standard SCT for a specific project, qualitative analysis and evaluation can be realized for the completion quality of the project, but the disadvantages of this proposal are also obvious, such as the qualitative analysis is easy to be affected by subjective factors, lack of analysis accuracy, inaccurate comparison etc.

In order to further complete the evaluation index quantitative evaluation based on the above theory, this study defines the contribution rate of the output of a sub-management domain to the management domain as the impact contribution rate. Among them, its contribution rate to the superior management domain is called the direct impact contribution rate, while the contribution rate to other management domains is called the indirect impact contribution rate. The contribution rate of management domain to SCT was defined as cumulative impact contribution rate. The quantitative evaluation of the whole project can be achieved by calculating the impact contribution rate or corresponding cumulative impact contribution rate of a certain sub-management domain or management domain in the whole project, as well as by figuring out the comprehensive indexes SCT.

For example, for a specific project, we define the direct impact contribution rate as p and the indirect impact contribution rate as p'. A direct index matrix $M = [n,p]$ and a cross index matrix $M' = [n,p']$ can be obtained for the sub-management domain. n stands for the number of influence relations occurred. By calculating the characteristics of M and M' $(\lambda, \ldots \lambda p)$, $(\lambda', \ldots \lambda' p)$, the formula for calculating the direct contribution rate α and indirect contribution rate α' of number k's impact influence is as follows:

$$a = \lambda_k / \sum_{i=1}^{p} \lambda_i \quad \text{and} \quad a' = \lambda'_k / \sum_{i=1}^{p'} \lambda'$$

For a specific management domain, we assume that q and q' are the sum of the direct and indirect influence relations with the management domain, and the calculation formula of the cumulative direct impact contribution rate γ and the cumulative indirect impact contribution rate γ' of the management domain is as follows:

$$\Upsilon = \frac{\lambda_k / \sum_{i=1}^{q} \lambda_i}{\lambda_k / \sum_{i=1}^{p} \lambda i + \lambda'_k / \sum_{i=1}^{p'} \lambda'_i}$$

and

$$\Upsilon' = \frac{\lambda'_k / \sum_{i=1}^{q'} \lambda'_i}{\lambda_k / \sum_{i=1}^{p} \lambda i + \lambda'_k / \sum_{i=1}^{p'} \lambda'_i}$$

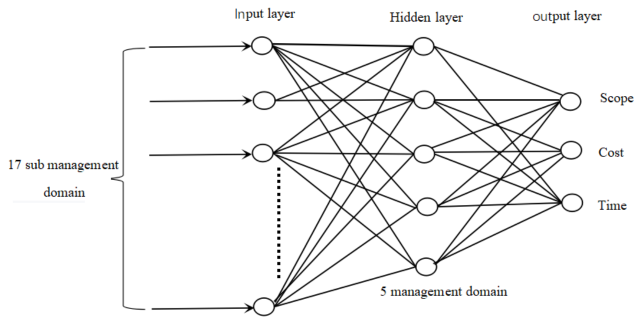


FIGURE 5. Neural network structure diagram.

From this perspective, the direct or indirect contribution rate of any sub-management domain and the direct or indirect cumulative contribution rate of the management domain can be calculated through the above formula, providing data basis for quantitative analysis and evaluation of the whole project.

D. THE ESTABLISHMENT OF TRAINING MODEL BASED ON BP NEURAL NETWORK

The above research solves the quantitative evaluation issue of software project quality, and improves the scientific and objectivity of our proposal. However, for software projects with different characteristics, such as code quantity, technical framework, and team culture etc., the impact contribution rate, cumulative impact contribution rate of management domain and sub- management domain have different weight of effect on SCT. In traditional, the weight of effect on SCT is mainly evaluated by the expert group, and it has strong subjectivity.

In order to improve the scientific nature and accuracy of our proposal, this article proposed to construct a three-layer neural network training model based on BP artificial neural network technology, which includes the logical relationship of the evaluation model as well as the quantitative evaluation index of SCT. In generally, the input and output of BP neural network are highly nonlinear, which is difficult to be expressed by formula. Only through continuous training and learning of certain data sets can the weight of evaluation be adjusted step by step to achieve the most suitable state. By doing this, we can overcome the shortcomings of the subjective factors of the traditional evaluation methods, and reflect the objectivity of the input samples to the greatest extent, so we select three-layer neural network model as the data training model for avoiding this defect in our proposal.

Combined the characteristics of 17 sub-management domains, 5 management domains and SCT, we decided to adopt 17 sub-management domains as the input layer nodes of neural network, 5 management domains as the hidden layer nodes of neural network, and SCT as the output layer nodes of neural network. The structure is shown in the figure 5.

Among them, the series weights between the input layer and the hidden layer represent the impact contribution rate of the sub-management domain to the management domain (including direct and indirect contribution rate), while the series weights between the hidden layer and the output layer represent the cumulative impact contribution rate of the management layer to SCT. By doing this, we establish the BP neural network training model based on SCT evaluation criteria and software management domain (sub-management domain).''

V. CASE STUDY

A. THE GOAL OF THE EXPERIMENT

The proposed model can be used to analyze the direct, indirect or cross-influence relationship between each management domain (sub-management domain) and SCT index which hidden in the project management data, as well as the direct impact contribution rate, indirect impact contribution rate and cumulative impact contribution rate based on these relationships. However, the analysis is not a thorough quantitative analysis, which is susceptible to a variety of subjective factors.

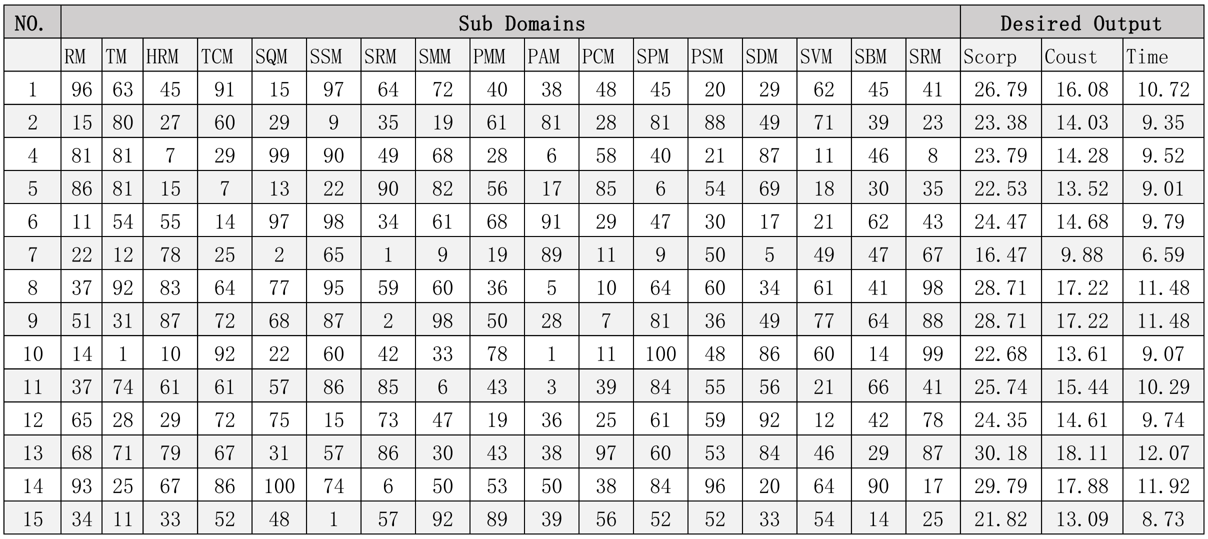
Therefore, in order to improve the scientific and accuracy of evaluation system, we will build neural network training model including logical relationship model as well as SCT quantitative evaluation index based on artificial neural network technology. Besides, we will make use of software project management data for training and testing, so as to prove that our proposal is valid and discover some key data, such as impact factor weights, the normal threshold range of evaluation standard and normal threshold range of SCT evaluation criteria.

B. THE PREPARATION FOR TRAINING DATA

The data at the input end of the training model are integers within 100, which are randomly generated in advance. The total number of the training data includes 300 groups, each group contains 20 data. Among them, 90% of them are used as the training data, and 10% of them are used as the test data. The data is divided into 5 grades: 81∼100 (excellent), 61∼80 (good), 41∼60 (medium), 11∼40 (poor), 0∼10 (poorest). These data are evenly and reasonably distributed, whose value represents the completion degree of the corresponding sub-management domains in the actual project execution process. Because of the length of this article, figure 6 only shows a part of the training data.

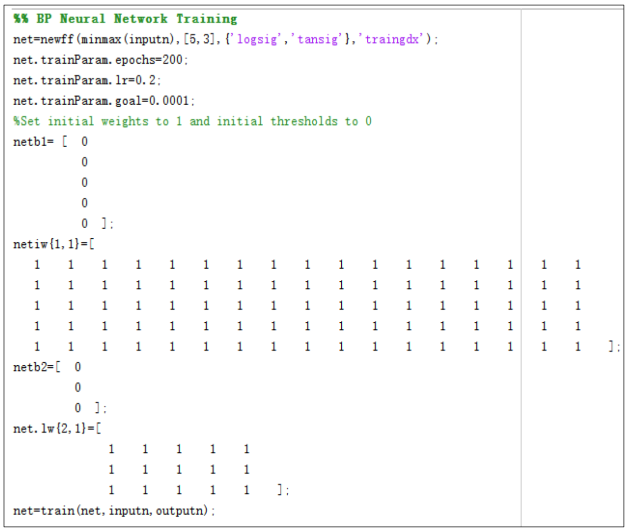
On the other hand, the characteristic of BP neural network is to calculate a network calculation result through the input value passes through various weights, thresholds and excitation functions, and then make comparison with the expected output, we adjust the weight threshold of the network by the error between the calculated output and the expected output. Therefore, the expected output of the training model should have an internal logic connection with the input.

Because according to the actual project experience, SCT have different impact on the quality of project strength for different size, different architecture and technology, we initialize the experimental data output expectations by the way



NO.	Sub Domains																	Desired Output		
	RM	TM	HRM	TCM	SQM	SSM	SRM	SMM	PMM	PAM	PCM	SPM	PSM	SDM	SVM	SBM	SRM	Scorp	Coust	Time
1	96	63	45	91	15	97	64	72	40	38	48	45	20	29	62	45	41	26. 79	16. 08	10. 72
2	15	80	27	60	29	9	35	19	61	81	28	81	88	49	71	39	23	23. 38	14. 03	9. 35
4	81	81	7	29	99	90	49	68	28	6	58	40	21	87	11	46	8	23. 79	14. 28	9. 52
5	86	81	15	7	13	22	90	82	56	17	85	6	54	69	18	30	35	22. 53	13. 52	9. 01
6	11	54	55	14	97	98	34	61	68	91	29	47	30	17	21	62	43	24. 47	14. 68	9. 79
7	22	12	78	25	2	65	1	9	19	89	11	9	50	5	49	47	67	16. 47	9. 88	6. 59
8	37	92	83	64	77	95	59	60	36	5	10	64	60	34	61	41	98	28. 71	17. 22	11. 48
9	51	31	87	72	68	87	2	98	50	28	7	81	36	49	77	64	88	28. 71	17. 22	11. 48
10	14	1	10	92	22	60	42	33	78	1	11	100	48	86	60	14	99	22. 68	13. 61	9. 07
11	37	74	61	61	57	86	85	6	43	3	39	84	55	56	21	66	41	25. 74	15. 44	10. 29
12	65	28	29	72	75	15	73	47	19	36	25	61	59	92	12	42	78	24. 35	14. 61	9. 74
13	68	71	79	67	31	57	86	30	43	38	97	60	53	84	46	29	87	30. 18	18. 11	12. 07
14	93	25	67	86	100	74	6	50	53	50	38	84	96	20	64	90	17	29. 79	17. 88	11. 92
15	34	11	33	52	48	1	57	92	89	39	56	52	52	33	54	14	25	21. 82	13. 09	8. 73

FIGURE 6. **Experimental data schematic diagram.**



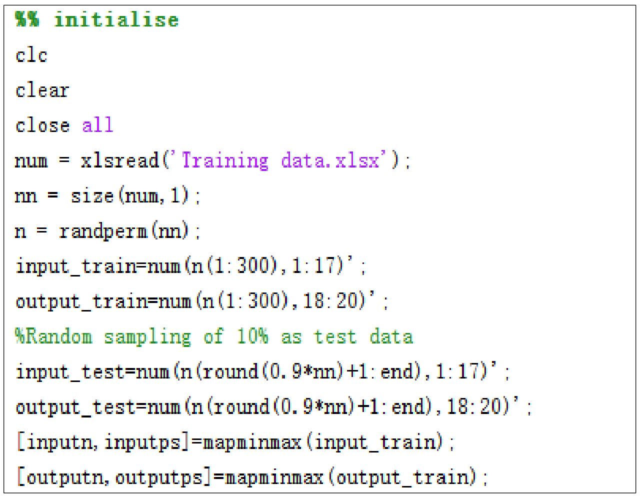
```
%% BP Neural Network Training
net=newff(minmax(inputn),[5,3],{'logsig','tansig'},'traingdx');
net.trainParam.epochs=200;
net.trainParam.lr=0.2;
net.trainParam.goal=0.0001;
%Set initial weights to 1 and initial thresholds to 0
netb1= [  0
          0
          0
          0
          0  ];
netiw{1,1}=[
   1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
   1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
   1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
   1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
   1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    ];
netb2=[  0
         0
         0  ];
net.lw{2,1}=[
              1    1    1    1    1
              1    1    1    1    1
              1    1    1    1    1    ];
net=train(net,inputn,outputn);
```

FIGURE 7. **The setting of parameter for the training model.**

of mean value of the 17 sub-management domains multiplied the intensity coefficient. (coefficient of S is 0.5, coefficient of C is 03, and coefficient of T is 0.2). Among them, the coefficient of strength stands for the strength of influence of SCT on project quality in the project. We can adjust it during evaluation model training for different specific projects, so as to achieve the best state of evaluation.

In addition, because the initial impact contribution rate of experiment assumes are the same, we set the initial value as 1, the initial threshold as 0, and training learning rate as 0.2. Besides, we set the largest number of training as 5000, learning objective as 0.00001.

The training algorithm adopts the steepest descent method, and optimize weights and thresholds with repeated training data on network. After completion of the training, we verify the result with test data. Figure 7 shows the setting of parameter for the training model.



```
%% initialise
clc
clear
close all
num = xlsread('Training data.xlsx');
nn = size(num,1);
n = randperm(nn);
input_train=num(n(1:300),1:17)';
output_train=num(n(1:300),18:20)';
%Random sampling of 10% as test data
input_test=num(n(round(0.9*nn)+1:end),1:17)';
output_test=num(n(round(0.9*nn)+1:end),18:20)';
[inputn,inputps]=mapminmax(input_train);
[outputn,outputps]=mapminmax(output_train);
```

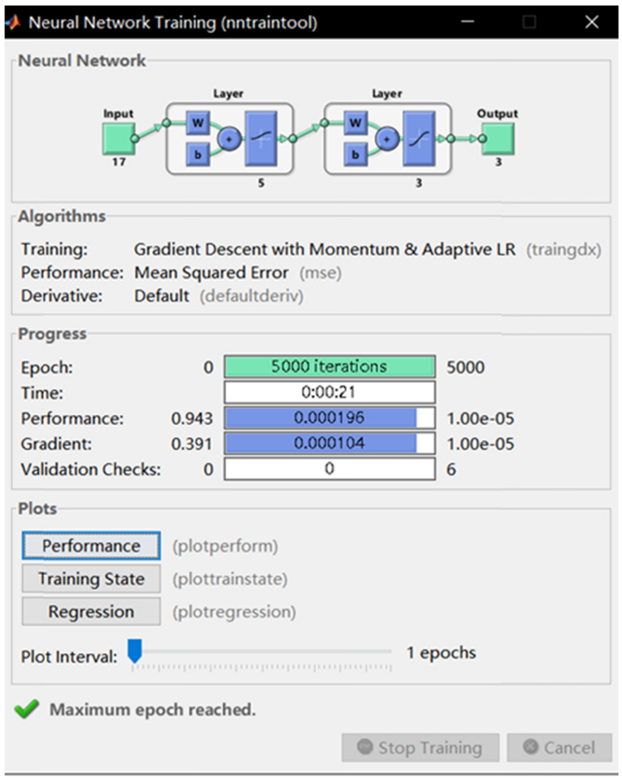
FIGURE 8. **The execution code of the experiment.**

C. EXPERIMENT AND RESULT

In this experiment, MATLAB is used as the training platform of BP neural network. Under the environment of i5CPU, 8g memory and Windows10, MATLAB R2012a execution environment and the MATLAB BP software toolbox were used for the experiment.

Figure 8 shows the execution code of this experiment. In this case study, experimental process are divided into two steps, the first is to use the *xlsread()* function to import training data which was stored in the excel file, the second is to start building and training with the import data. Among them, *logsig* logarithmic *S* transfer function is used in middle layer, *tansig* tangent *S* transfer function is used in output layer, and training function *traingdx* adopts the steepest descent method.

Figure 9 shows the feedback information of the program execution results, it can be seen that the number of times of this training is 5000, which takes 21 seconds. Among them, algorithms part in figure 9 stands for related parameter



Neural Network Training (nntraintool)

Neural Network

Input 17 — Layer 5 — Layer 3 — Output 3

Algorithms

Training: Gradient Descent with Momentum & Adaptive LR (traingdx)
Performance: Mean Squared Error (mse)
Derivative: Default (defaultderiv)

Progress

Epoch:	0	5000 iterations	5000
Time:		0:00:21	
Performance:	0.943	0.000196	1.00e-05
Gradient:	0.391	0.000104	1.00e-05
Validation Checks:	0	0	6

Plots

Performance (plotperform)
Training State (plottrainstate)
Regression (plotregression)

Plot Interval: 1 epochs

✓ Maximum epoch reached.

FIGURE 9. Feedback information of experiment result.

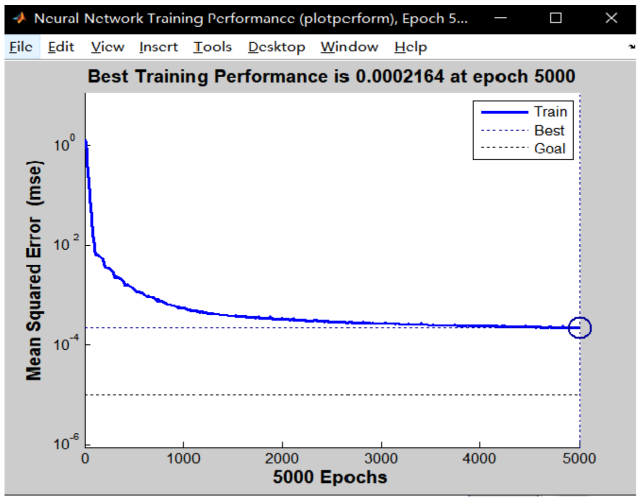


FIGURE 10. Training performance diagram.

information. Progress part in figure 9 stands for information of the termination condition (If one of them is satisfied, it will be stopped), and Plots part in figure 9 are various graph curves.

Figure 10 is the experimental training performance result diagram. In the diagram, the abscissa represents the training times, the ordinate represents the mean square error (*mse*), the blue curve represents the training error change curve, and the dotted black line represents the target line.

From figure 10, it can be seen that the error of the training sample data decreases gradually with the number of training times. The mean square error (*mse*) decreases rapidly in the

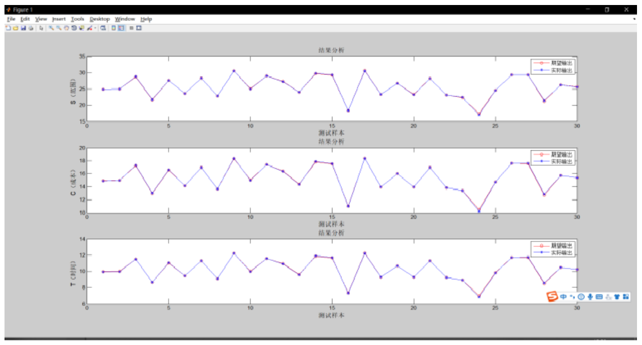


FIGURE 11. Experimental data schematic diagram.

first 500 times, and the rate of decline from 500 to 2000 times is basically linear with the number of iterations, it is close to the expected result for about 2000 times. The mean square error has reached 0.0002164 in the 5000 times of training, which is close to the expected output value.

Figure 11 is the output layer's (SCT) comparison diagram of the experimental results of the training model. In the diagram, the red curve represents the expected output value while the blue curve represents the actual training value. It can be seen from the experimental result diagram that the measured value and the training value are close to coincide, which indicates that the BP neural network model designed in this article can better reflect the characteristics of the simulation data, and can be well applied in our proposal in this article.

VI. DISCUSSION

A. SUMMARY OF CONCLUSION

In this article, we propose a software project quality evaluation system based on software project management data and quality evaluation standard SCT. The proposal includes the definition of software project management domain (sub-management domain) in software project development management process, the analysis and mining method of the direct (indirect) influence relationship between project management domain (sub-management domain), and the quantitative evaluation index (direct impact contribution rate, indirect impact contribution rate and cumulative impact contribution rate) and its calculation method. Then the proposal above basically solves the problem of quantitative evaluation method based on software project management domain as well as project quality evaluation standard SCT.

During the continuous research, we find that according the different characteristics of software projects, such as code volume, technical framework, and even team culture, the quantitative evaluation indexes also have different weights on the effect of SCT in the whole evaluation system of the proposal. For solving this issue, we proposed to build the neural network training model which includes logical relationship model as well as SCT quantitative evaluation index based on artificial neural network technology.

In this article, we also finished an experiment with software project management data for training and testing, so as to prove that our proposal is valid and discover some key data, such as impact factor weights, the normal threshold range of evaluation standard and normal threshold range of SCT evaluation criteria.

For this experiment we prepared the total number of the training data includes 300 groups 6000 data for implementing, and hope that we can figure out some key data including the normal threshold range of influence factor weights, evaluation criteria and SCT evaluation criteria by learning, training and testing of simulated software project management data.

According the experiment result, the measured value and the training value are close to coincide, which indicates that the BP neural network model designed in this article can better reflect the characteristics of the simulation data, and can be well applied in our proposal in this article. By doing this, the scientific nature and accuracy of the evaluation system is improved and a complete evaluation system is established.

B. RELATED WORK

At present, researchers mainly focus on the application of neural network combined with other technologies in software quality evaluation. For example, Li *et al.* [39] proposed a software quality evaluation method based on fuzzy neural network, and established a network evaluation model by using the improved asymmetric fuzzy trigonometric regression method, so as to evaluate software performance from the perspective of users. Zhang and Song [40] applied neural network combined with genetic algorithm to the software quality evaluation system of aerospace system, which reduced the subjective arbitrariness and uncertainty in thinking, thus the evaluation results can be more scientific and accurate. Yang [41] added BP neural network to the software requirements analysis risk assessment model, combined with fuzzy theory, which used the nonlinear mapping attribute of BP neural network and the super expression ability and comprehension of fuzzy theory to improve the effectiveness and predictability of risk assessment.

Other researchers mainly focus on the research of software quality evaluation system. For example, researchers of Sener [42] vividly illustrated user's ideas and software attributes through expressions, and introduced the theory of fuzzy trigonometric number and regression idea into software product quality evaluation. Osmundson *et al.* [43] and other scholars believed that the management of software quality decides the success or failure of software project. Therefore, they proposed a measurement method based on software quality management, which includes the following four aspects: Demand management, evaluation/plan management, role management and risk management. They obtained a comprehensive score for software management measures by the method of asking the software researchers questions. Based on the hierarchical McCall software quality assessment framework, Pedrycz *et al.* [44] and other scholars compared the application of fuzzy calculation method based on fuzzy set with rough calculation method based on rough set in software quality assessment.

C. FUTURE WORK

The software quality evaluation model based on BP neural network integrates the quantitative process into the learning process of BP neural network, which reflects the objective weight of the input layer samples and overcomes the influence of subjective factors in the traditional evaluation method. At the same time, because of the strong expansibility of the software quality evaluation of BP neural network, the input and output layer can be adjusted according to different types of indicators and different evaluation levels, which is of great significance to the comprehensive evaluation of software quality.

The BP neural network simulation algorithm adopted the steepest descent method in the experiment. The algorithm convergence process is relatively slow, and the algorithm training gradually decreases to a minimum value from a starting point along the error function can negative gradient direction. Therefore, in the process of training, it may fall into a local minimum value of dell, which will affect the accuracy of the model [40], [38].

In the future work, in order to solve the problem of slow convergence in the learning process, the algorithm in this paper can be optimized by using the additional momentum method. To solve the problem of convergence to local minimum, BP neural network algorithm can be optimized by genetic algorithm, quasi-newton method, conjugate gradient method and simulated annealing algorithm. At the same time, due to the limitation of BP neural network prediction, more advanced methods can be used to replace BP neural network to achieve better fitting and prediction results.

REFERENCES

[1] (2015). *Standish Group 2015 Chaos Report—Q&A with Jennifer Lynch [EB/OL]*. [Online]. Available: http://www.infoq.com/articles/standish-chaos-2015?utm_source=tuicool&utm_medium=referral,2015-10-04/2016-01-21

[2] N. Gorla and S.-C. Lin, "Determinants of software quality: A survey of information systems project managers," *Inf. Softw. Technol.*, vol. 52, no. 6, pp. 602–610, Jun. 2010.

[3] J. Favaro, "Guest editor's introduction: Renewing the software project management life cycle," *IEEE Softw.*, vol. 27, no. 1, pp. 17–19, Jan. 2010.

[4] S. Mohapatra, "Improvised process for quality through quantitative project management: An experience from software development projects," *Int. J. Inf. Commun. Technol.*, vol. 2, no. 4, pp. 355–373, Aug. 2010.

[5] Z. Khadija, "Software project risk management by using six sigma approach," *Int. J. Eng. Res. Gen. Sci.*, vol. 3, no. 4, pp. 17–21, Aug. 2015.

[6] C. C. Chen, J. Y.-C. Liu, and H.-G. Chen, "Discriminative effect of user influence and user responsibility on information system development processes and project management," *Inf. Softw. Technol.*, vol. 53, no. 2, pp. 149–158, Feb. 2011.

[7] W. H. Hou, "The research on the method of software project evaluation," M.S. thesis, School Civil Eng. & Archit., Wuhan Univ. Technol., Wuhan, China, 2011.

[8] J. J. Ding, K. G. HAO Hao, and H. Hao, "Research of software project risk management based on rough set theory," *Comput. Sci.*, vol. 37, no. 4, pp. 117–119, Apr. 2010.

[9] P. Gong, "Research of software project risk management based on data mining theory," M.S. thesis, School Civil Eng., Dalian Univ. Technol., Dalian, China, 2014.

[10] L. Cai, "Application of KANO model in software project risk management," M.S. thesis, Inf. School, Capital Univ. Econ. Bus., Beijing, China, 2014.

[11] Z. Y. Wei, "The design and realization of recognition system of handwritten numeric characters based on BP neural network," M.S. thesis, School Inf. Sci. Eng., Hebei Univ. Sci. Technol., Shijiazhuang, China, 2013.

[12] R. L. Chen, Z. H. Guo, and Z. W. Zhu, "The realization of speaker recognition technology based on BP neural network," *Intell. Comput. Appl.*, vol. 2, no. 2, pp. 51–53, Apr. 2012.

[13] S. Y. Liu, H. Zhai, and D. S. Liu, "Cross-project software defect prediction based on domain adaptive neural network," *Comput. Digit. Eng.*, vol. 4, no. 47, pp. 869–891, Apr. 2019.

[14] K. A. Alam, R. Ahmad, A. Akhunzada, M. H. N. M. Nasir, and S. U. Khan, "Impact analysis and change propagation in service-oriented enterprises: A systematic review," *Inf. Syst.*, vol. 54, pp. 43–73, Dec. 2015.

[15] A. Fatwanto, "Software requirements specification analysis using natural language processing technique," in *Proc. Int. Conf. QiR*, Jun. 2013, pp. 105–110.

[16] D. Maevskiy and Y. Kozina, "Where and when is formed of software quality," *Electr. Comput. Syst.*, vol. 18, pp. 55–59, Mar. 2016.

[17] H. Nakai, N. Tsuda, K. Honda, H. Washizaki, and Y. Fukazawa, "A SQuaRE-based software quality evaluation framework and its case study," in *Proc. IEEE Region 10 Conf. (TENCON)*, Singapore, Nov. 2016, pp. 3704–3707.

[18] A. Yamashita, "Experiences from performing software quality evaluations via combining benchmark-based metrics analysis, software visualization, and expert assessment," in *Proc. IEEE Int. Conf. Softw. Maintenance Evol. (ICSME)*, Bremen, Germany, Sep. 2015, pp. 421–428.

[19] N. Tsuda, H. Washizaki, K. Honda, H. Nakai, Y. Fukazawa, M. Azuma, T. Komiyama, T. Nakano, H. Suzuki, S. Morita, K. Kojima, and A. Hando, "WSQF: Comprehensive software quality evaluation framework and benchmark based on SQuaRE," in *Proc. IEEE/ACM 41st Int. Conf. Softw. Eng., Softw. Eng. Pract. (ICSE-SEIP)*, Montreal, QC, Canada, May 2019, pp. 312–321.

[20] *Systems and Software Engineering—Systems and Software Quality Requirements and Evaluation (SQuaRE)—Measurement of Quality in Use*, Standard ISO/IEC 25022:2016, 2016.

[21] A. Magazinius and R. B. Svensson, "Effects of feature complexity on software effort estimates - an exploratory study," in *Proc. 40th EUROMICRO Conf. Softw. Eng. Adv. Appl.*, Verona, Italy, Aug. 2014, pp. 301–304.

[22] B. H. Sun, N. Hu, and D. Y. Li, "Analysis research of software requirement safety based on neural network and NLP," *Comput. Sci.*, vol. 46, no. 6A, pp. 348–352, Jun. 2019.

[23] A. Sekhon and P. Agarwal, "Face recognition using back propagation neural network technique," in *Proc. Int. Conf. Adv. Comput. Eng. Appl.*, Ghaziabad, India, Mar. 2015, pp. 226–230.

[24] E. Tanuar, B. S. Abbas, A. Trisetyarso, C.-H. Kang, F. L. Gaol, and W. Suparta, "Back propagation neural network experiment on team matchmaking MOBA game," in *Proc. Int. Conf. Inf. Commun. Technol. (ICOIACT)*, Yogyakarta, Indonesia, Mar. 2018, pp. 240–243.

[25] Z. Li, D.-Y. Sun, J. Li, and Z.-F. Li, "Social network change detection using a genetic algorithm based back propagation neural network model," in *Proc. IEEE/ACM Int. Conf. Adv. Social Netw. Anal. Mining (ASONAM)*, San Francisco, CA, USA, Aug. 2016, pp. 1386–1387.

[26] F. Hu, L. Wang, S. Wang, X. Liu, and G. He, "A human body posture recognition algorithm based on BP neural network for wireless body area networks," *China Commun.*, vol. 13, no. 8, pp. 198–208, Aug. 2016.

[27] A. A. Al-Hameed, S. H. Younus, A. T. Hussein, M. T. Alresheedi, and J. M. H. Elmirghani, "Artificial neural network for LiDAL systems," *IEEE Access*, vol. 7, pp. 109427–109438, 2019.

[28] M. Chanda and M. Biswas, "Plant disease identification and classification using back-propagation neural network with particle swarm optimization," in *Proc. 3rd Int. Conf. Trends Electron. Informat. (ICOEI)*, Tirunelveli, India, Apr. 2019, pp. 1029–1036.

[29] Y. Liu, W. Jing, and L. Xu, "Cascading model based back propagation neural network in enabling precise classification," in *Proc. 12th Int. Conf. Natural Comput., Fuzzy Syst. Knowl. Discovery (ICNC-FSKD)*, Changsha, China, Aug. 2016, pp. 7–11.

[30] M. A. Abuzneid and A. Mahmood, "Enhanced human face recognition using LBPH descriptor, multi-KNN, and back-propagation neural network," *IEEE Access*, vol. 6, pp. 20641–20651, 2018.

[31] C. Chao and X. M. Cao, "Improved differential evolution algorithm to optimize BP neural network for intrusion detection," *Comput. Appl. Softw.*, vol. 4, no. 35, pp. 310–316 and 324, Apr. 2018.

[32] S. Sinha and N. Mandal, "Design and analysis of an intelligent flow transmitter using artificial neural network," *IEEE Sensors Lett.*, vol. 1, no. 3, pp. 1–4, Jun. 2017.

[33] K. H. Wagner and S. McComb, "Optical rectifying linear units for back-propagation learning in a deep holographic convolutional neural network," *IEEE J. Sel. Topics Quantum Electron.*, vol. 26, no. 1, pp. 1–18, Jan./Feb. 2020.

[34] K. Moshkbar-Bakhshayesh and M. B. Ghofrani, "Development of an efficient identifier for nuclear power plant transients based on latest advances of error back-propagation learning algorithm," *IEEE Trans. Nucl. Sci.*, vol. 61, no. 1, pp. 602–610, Feb. 2014.

[35] T. Nitta and Y. Kuroe, "Hyperbolic gradient operator and hyperbolic back-propagation learning algorithms," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 5, pp. 1689–1702, May 2018.

[36] F. E. Asimakopoulou, V. T. Kontargyri, G. J. Tsekouras, I. F. Gonos, and I. A. Stathopulos, "Estimation of the Earth resistance by artificial neural network model," *IEEE Trans. Ind. Appl.*, vol. 51, no. 6, pp. 5149–5158, Nov./Dec. 2015.

[37] R. Mukhaiyar and R. Safitri, "Implementation of artificial neural network: Back propagation method on face recognition system," in *Proc. 16th Int. Conf. Qual. Res. (QIR): Int. Symp. Electr. Comput. Eng.*, Padang, Indonesia, Jul. 2019, pp. 1–5.

[38] H. Singh, G. Kaur, and N. Gupta, "Robust edge detector using back propagation neural network with multi-thresholding," in *Proc. IEEE Int. Conf. Comput. Intell. Comput. Res.*, Coimbatore, India, Dec. 2014, pp. 1–6.

[39] K. W. Li, Y. Zhang, and J. F. Ma, "Software quality evaluation method based on fuzzy neural network with fuzzy triangle numbers," *Comput. Eng. Sci.*, vol. 7, no. 36, pp. 1301–1306, Jul. 2017.

[40] D. Zhang and X. Q. Song, "Research on software quality evaluation model based on genetic algorithm and BP artificial neural network," *China New Telecommun.*, vol. 4, no. 18, pp. 88–91, Apr. 2016.

[41] M. L. Yang, "Research on software requirement analysis risk assessment model based on BP neural network," *Shandong Ind. Technol.*, vol. 4, p. 139 Apr. 2016.

[42] Z. Sener and E. E. Karsak, "A fuzzy regression and optimization approach for setting target levels in software quality function deployment," *Softw. Qual. J.*, vol. 18, no. 3, pp. 323–339, May 2010.

[43] J. S. Osmundson, J. B. Michael, and M. J. Machniak, "Quality management metrics for software development," *Inf. Manage.*, vol. 8, no. 40, pp. 799–812, Aug. 2003.

[44] W. Pedrycz, L. Han, J. F. Peters, S. Ramanna, and R. Zhai, "Calibration of software quality: Fuzzy neural and rough neural computing approaches," *Neurocomputing*, vol. 36, nos. 1–4, pp. 149–170, Feb. 2001.

[45] J. P. Wang, X. D. Kong, and J. M. Yang, "Research on CMMI," *Appl. Res. Comput.*, vol. 10, no. 18, pp. 10–13, Mar. 2001.

[46] P. Orgun, D. Gungor, Y. Y. Kuru, O. O. Metin, and M. Yilmaz, "Software development overall efficiency improvement in a CMMI level 5 organization within the scope of a case study," in *Proc. 3rd Int. Conf. Comput. Sci. Eng. (UBMK)*, Sarajevo, Bosnia and Herzegovina, Sep. 2018, pp. 258–263.

[47] W. Wang, X. Lu, R. Jia, and F. Li, "Development of mass spectrometer software project based on CMMI," in *Proc. 3rd IEEE Int. Conf. Comput. Commun. (ICCC)*, Chengdu, China, Dec. 2017, pp. 2508–2511.

[48] C. Rojrattanakorn and W. Vatanawood, "Automated risk identification of CMMI project planning using ontology," in *Proc. 5th Int. Conf. Appl. Comput. Inf. Technol./4th Int. Conf. Comput. Sci./Intell. Appl. Inform./2nd Int. Conf. Big Data, Cloud Comput., Data Sci. (ACIT-CSII-BCD)*, Hamamatsu, Japan, Jul. 2017, pp. 19–24.

[49] S. Sanchez-Gordon and D. Viera-Bautista, "Mapping between CMMI-DEV v1.3 and ISO/IEC 90003:2014," in *Proc. Int. Conf. Inf. Syst. Softw. Technol. (ICI2ST)*, Quito, Ecuador, Nov. 2019, pp. 134–140.

BEN YAN received the B.E. degree from the Henan University of Science and Technology, China, in 1999, the M.E. degree from the Department of Information Science, Okayama University of Science, Japan, in 2006, and the Ph.D. degree from the Graduate School of Information Science, Nara Institute of Science and Technology, Japan, in 2008. From 2009 to 2014, he worked with the Panasonic Group, SANYO Information Technology Solutions Company Ltd., Osaka, Japan. He is currently a Professor with the Department of Computer and Information Engineering, Luoyang Institute of Science and Technology (LIT). His main research interests include software engineering, service-oriented architecture, the V and V of home network systems, and requirements engineering for safety critical systems.

HUA-PING YAO received the B.E. degree from the Henan University of Science and Technology, China, in 1999, and the M.E. degree from the Department of Information Science, Okayama University of Science, Japan, in 2006. From 2006 to 2014, she worked with CSI and Trend Creates Company Ltd., Osaka, Japan. She is currently a Lecturer with the Department of Computer and Information Engineering, Luoyang Institute of Science and Technology (LIT). Her main research interests include e-learning, software engineering, and home network systems.

MASAHIDE NAKAMURA (Member, IEEE) received the B.E., M.E., and Ph.D. degrees in information and computer sciences from Osaka University, Japan, in 1994, 1996, and 1999, respectively. From 1999 to 2000, he has been a Postdoctoral Fellow with SITE, University of Ottawa, Canada. He was with the Cyber media Center, Osaka University, from 2000 to 2002. From 2002 to 2007, he worked with the Graduate School of Information Science, Nara Institute of Science and Technology, Japan. He is currently an Associate Professor with the Graduate School of System Informatics, Kobe University. His research interests include service /cloud computing, smart home, smart city, and life log. He is a member of the IEICE and IPSJ.

ZHI-FENG LI (Student Member, IEEE) received the B.E. degree from the Luoyang Institute of Science and Technology, China, in 2019. His main research interests include software engineering, neural networks, and machine learning.

DONG WANG received the M.E. degree from the Nara Institute of Science and Technology, in Japan, where he is currently pursuing the Ph.D. degree. His research interests include code review and mining software repositories.