Intelligent Analysis Model for Outsourced Software Project Risk Using Constraint-based Bayesian Network

Yong Hu*
Institute of Business Intelligence & Knowledge Discovery, Business School, Guangdong University of Foreign Studies, Sun Yat-Sen University, Guangzhou, 510006, PR China
Email: henryhu200211@163.com

Xizhu Mo*, Xiangzhou Zhang*, Yuran Zeng, Jianfeng Du and Kang Xie
Institute of Business Intelligence & Knowledge Discovery, Business School, Guangdong University of Foreign Studies, Sun Yat-Sen University, Guangzhou, 510006, PR China
Email: moxizhu@qq.com, zhxzhou86@foxmail.com, zengyuran@hotmail.com, jfdu@mail.gdufs.edu.cn, mnsxk@mail.sysu.edu.cn

***Abstract*—Software outsourcing is one of the leading methods in software development. However, it is also accompanied with higher risk than in-house software development. A risk intelligent analysis model based on Bayesian Network can effectively contribute to software project risk assessment. From the perspectives of both the customer and contractor, we propose a risk identification framework for outsourced software projects, and have collected real-life outsourced software project samples. Based on totally 154 valid samples, we established an intelligent analysis model for outsourced software project risk by incorporating expert knowledge as structural constraints into a Bayesian Network. Experimental results showed that the model has higher predictive accuracy than Decision Tree and Neural Network, and the derived management rules are consistent with the existing software engineering theory. The model would provide a great guideline for outsourced software project risk management in both theory and practice.**

***Index Terms*—outsourced software, software project risk management, Bayesian network, structural constraint, risk prediction**

I. INTRODUCTION

Software industry has become one of the mainstay industries of the world's economic development. Software outsourcing gradually developed from the 1990s and it's an important part of IT outsourcing [1]. The International Data Corporation (IDC) predicts that in the next few years, the global software outsourcing market will show expansive growth. By 2010, the global software outsourcing market has reached 83.69 billion dollars. Unfortunately, software development is of high failure rate. In 2010, the Standish Group of U.S. reported in *CHAOS Summary 2010* that the overall success rate of software project was only 32%, meanwhile, the complete failure rate accounted for 24% [2]. Different from in-house software project, outsourced software project involves two different stakeholders and decision-makers, namely customer and contractor. Software outsourcing has become one of the major ways for software development, though it may have a higher risk compared with in-house sourcing [3].

McManus's research on the reasons of project failure (based on 42 information systems projects which were completed during 1994 to 2001) showed that the management and technical issues account for 65% and 35% respectively [1]. Project management plays a crucial role in both software in-house sourcing and outsourcing [4]. It can help software project avoid the fate of failure, re-operation or cancel. Thus, how to identify, manage and remove these risks before they become a threat to the success of software project is the main objective of software risk management. Risk management has been regarded as a key activity (or process) to improve the management level of software project [5, 6]. Earl argued that there are 11 risks (possibility of weak management, inexperienced staff, business uncertainty, outdated technology skills, endemic uncertainty, hidden costs, lack of organizational learning, loss of innovative capacity, dangers of an eternal triangle, technological indivisibility, and fuzzy focus) of IT outsourcing and even if these risks are not universal, they made the IT outsourcing complex and uncertain [7].

Therefore, it's urgent and feasible to conduct further research on outsourced software project risk management. Chou pointed out that a successful IS outsourcing project needs suitable risk analysis and quality control process [8]. Risk intelligent analysis model can effectively carry out software project risk assessment. However, there is only a few works focusing on outsourced software project risk intelligent analysis model. In order to improve the risk decision-making performance in outsourced software

* contribute equally, regarded as first authors.

doi:10.4304/jsw.7.2.440-449

projects, our research constructs an outsourced software project risk intelligent analysis model based on real-life data, which applies quantitative method and Bayesian Network modeling to predict risks. Bayesian Network has the following advantages [9, 10]: 1) suitable for small and incomplete data sets and structural learning possibility; 2) visual modeling of cause-effect relationships which helps identify risk sources so as to provide explicit knowledge for risk analysis; 3) provide probabilistic estimates and explicit treatment of uncertainty. The key to establish a Bayesian Network analysis model is to not only find objective knowledge from data, but also make the model interpretable in a sound way. Therefore, our research combines expert knowledge and network structure learning method so as to construct a Bayesian-Network-based risk prediction model with good interpretability from data.

The remainder of this paper is structured as follows: Section II is a literature review of related works; Section III introduces the methodology of this research, including constraint-based Bayesian Network and network structure learning algorithm; Section IV presents how the model was built and validated; Section V is the conclusion and discussion of this research.

II. Related Works

Boehm [11] and Charette [12] first introduced risk management into software project management and the risk management framework they proposed gave a great guideline for further research of this domain. Awareness and understanding of software project risk can help reduce the likelihood of project failure [3]. Successful software project always means that the project can be carried out on schedule and on budget; meanwhile it can meet customers' high demand of software product features, quality and performance. Software project risk management is a series of rules or practice. These rules or practices can identify, analyze and track risk factors and enhance the success rate of project [13]. Software project risk management has a positive effect on the project budget, schedule and scope etc. [14] Generally, risk management [15] includes two basic steps: risk assessment and risk control; risk assessment consists of risk identification, risk analysis, and risk prioritization. Risk identification requires systematically identifying and classifying project risks. Real project often manage risks according to the existing risk identification theoretical model. Thus, the model is the basis of risk management. Based on risk identification, risk analysis identifies the single or portfolio relationships between risk factors and between risk factors and project outcome and then prioritizes the risks in order to make key management. Risk control consists of risk-management planning, risk resolution, and risk monitoring. According to the sources and results of risk factors that risk analysis found, the further implementation of effective risk planning (necessary to or will generate revenue) and right actions can achieve the goal of minimum cost and maximum output. The first step of risk management planning is to establish a set of risk plan so that the risk factors are under control. The objective of risk management planning is to achieve the overall resource optimization, which means that it not only completes the plan of single risk factor, but also completes the plan of portfolio risks in order to realize the overall project risk planning [15]. During and after the implementation of risk planning, the risk state needs continuous monitoring and it is necessary to test the effectiveness of the risk planning and discover new risks in time.

There are two important steps to construct an effective model for outsourced software project risk management: first proposes the conceptual model for risk identification and then builds the risk intelligent analysis model.

A. Conceptual Model for Outsourced Software Project Risk Identification

From the customer perspective, Nakatsu and Iacovou [16] made a comparative study of key risk factors in the offshore and domestic software outsourcing. They totally identified three types of risks: risks that appeared in both offshore and domestic settings; risks that appeared in both but were aggravated in the offshore setting; risks that were unique to the offshore setting. Their findings implied that traditional project management risks were important in both domestically-and offshore- outsourced projects. Kliem et al. [17] regarded that the benefits related to the outsourcing would not be realized unless the risks are managed during the project lift cycle. They presented a framework of risks that related to outsourced projects, namely the financial, technical and management risks, and a process that can be applied to develop a matrix of risks.

From the contractor perspective, Rajkumar and Mani [18], Jennex and Adelakun [19] studied the risks of project management and skill of team members etc. Based on the data form 5 mid-tier offshore third party service providers in Bangalore, India, Aundhe and Mathew [20] found that there are three categories of risks that are faced by the contractors: 1) macroeconomic risks, namely government policy and regulatory, environment and exchange rate; 2) relationship specific risks, namely changes in client's corporate structure, client's experience, client culture etc.; 3) project specific risks, namely schedule and budget management, staffing, requirements capture, etc. They also identified that relationship maturity, nature of contract, nature of service or project and nature of client were important factors that affected the degree of risks using the principles of grounded theory.

From the perspectives of both the customer and contractor, Bahli and Rivard [21] conducted a survey of 132 IT executives to verified three main risk factors (namely, transaction process, customer and contractor) which have significant effects on information technology (IT) outsourcing.

However, the present research on conceptual model for outsourced software project risk identification has the following limitation:

Firstly, it lacks of authoritative model and there are significant differences among the existing researches. Lacity [22] made a literature review of IT outsourcing and identified 34 relevant articles. Lacity found that the proposed risk factors were confused, and lacked of consensus between different researches and the support of software engineering theories.

Secondly, majority of the existing models are just based on a single perspective (of customer or contractor) and the models that based on the perspectives of both the customer and contractor are lacking. For example, Boehm [15] listed "lack of top management support" as one of the top ten risks while he did not distinguish the source of this risk, i.e. whether it is from the contractor or from the customer. In risk management of outsourced software project, it is necessary to distinguish the risk source so that we can correctly analyze how the risk factors influence the project outcome and who is responsible for these risk factors. Therefore, we need a conceptual model for risk identification from the perspectives of both the customer and contractor.

B. Risk Intelligent Analysis Model

In order to guarantee the effectiveness of the model, we must select appropriate modeling method according to the requirements of risk intelligent analysis and data characteristics. The prevailing data mining algorithms used in software project risk intelligent analysis include hybrid Decision Tree [23, 24], Neural Network [25-27], Bayesian Network [28, 29] and so on. Their merits and limitations are listed as follows:

Decision Tree has a good interpretability and predictive accuracy when used in the domain of risk analysis modeling as it can provide objective facts of risk management based on statistics without having to add prior knowledge. The major practical challenge of Decision Tree is the over-fitting problem. Pruning is a pivot method to avoid this problem. However, the biggest limitation of pruning is that it will make the model unable to express all the information of risk factors. Meanwhile, the Decision Tree can sometimes be instable because of the "variable masking" problem (i.e. if one variable is highly relevant to another, then a small change may shift the split in the tree.) [30].This problem raises questions towards the stability and interpretability of the tree.

Neural Network has a good fitting ability for non-linear relationship between risk factors and project outcome, and the learning process of Neural Network is simple and easy for computer implementation. However, a Neural Network is unable to explain its reasoning. When the data are not sufficient, Neural Network will be unable to work. It is difficult for Neural Network to provide explicit decision-making knowledge to managers in the process of risk management because it only shows how the project outcome changes when a single or a portfolio risks change but cannot analysis the impact path of the risks on the project outcome. Zhang et al. [31] adopted Neural Network to establish a risk prediction model. The model has a sound description of performance for the nonlinear relationship between risk factors and outputs. It does not need the subjective definition between risk factors and between risk factors and outputs. However, the model is a "black box", which lacks of interpretability so that the project manager cannot understand the relationships between risk factors and outputs and consequently cannot provide decision knowledge.

Bayesian Network combines the accurate probability distribution and prior knowledge of experts, which are adequate for uncertainty modeling. It is relatively easy to build and the visual modeling can help identify the source of risks. Lauría and Duchessi [28] introduced a methodology for building an information technology (IT) implementation BN. This research incorporates structure constraints between risk factors so as to search for the best network. In our research, we not only incorporate structure constraints between risk factors but also between risk dimensions. Constraints among risk dimensions are perspicuous and can easily obtain literature support and industry approval. However, some researchers have subjective biases on the constraint among specific risk factors. Setting constraints for each factor will affect the knowledge finding ability of the network.

III. METHODOLOGY

A. Constraint-Based Bayesian Network

Bayesian Network, also known as Belief Network, or Bayesian Belief Network [32], is one of the most effective theoretical models in the field of uncertain knowledge expressing and reasoning. Based on graph theory and probability theory, Bayesian Network can describe the relationship between variables reasoning from incomplete, imprecise or uncertain information. Nowadays, Bayesian Network has been successfully applied to a wide variety of fields including medical diagnosis, statistical decision making, expert systems and prediction and so on. Bayesian Network consists of two parts: 1) a Directed Acyclic Graph (DAG), also known as Bayesian Network structure, which consists of nodes and directed links between these nodes. Each node is corresponding to a variable, while the directed links represent the correlation or causality between nodes; 2) and a set of Conditional Probability Tables (CPTs), which are a set of local probability distribution (also known as probability parameters) that reflect the relationship between the variables .

Campos and Castellano's research [32] showed that adding prior knowledge to the learning process of Bayesian Network could achieve better results.

There are three types of structural constraints:

1) Existence constraint

We introduce two types of existence constraints: existence of arcs and existence of edges. Let $\xi_a, \xi_e \subseteq \gamma \times \gamma$ be two subsets of pairs of variables, with $\xi_a \cap \xi_e = \varnothing$. We define:

- $(x, y) \in \xi_a$: The arc $x \rightarrow y$ must be part of any graph in the search space.
- $(x, y) \in \xi_e$: In any graph in the search space, there must be a directed link between node *x* and *y*.

An application of existence constraint is the BAN algorithm (Bayesian Network Augmented Naive Bayes), which predefines the Naïve Bayes structure (i.e. add arcs from the class variables to all the attribute variables) and searches for the appropriate additional arcs in pairs of attribute variables.

2) Absence constraint

We also introduce two kinds of constraints: the absence of arcs and absence of edges. Let $\eta_a,\eta_e \subseteq \gamma\times\gamma$, namely η_a,η_e are two subsets of pairs of variable $\gamma\times\gamma$, with $\eta_a \bigcap$. We define:

- $(x,y)\in\eta_a$: The arc $x \rightarrow y$ must not be part of any graph in the search space.
- $(x,y)\in\eta_e$: In any graph in the search space, there must not be a directed link between node *x* and *y*.

An application of absence constraint is the selective Naïve Bayesian classifier, which forbids directed links between attribute variables as well as directed links from attribute variables to the class variable.

3) Partial ordering constraint

Let $\Re_0 \subseteq \gamma\times\gamma$, namely $\Re_0$ is a subset of pairs of variable $\gamma\times\gamma$. We define:

- $(x,y)\in\Re_0$: In any graph in the search space, *x* must precede *y*.

An application of partial ordering constraint is the K2 algorithm (a well-known Bayesian Network learning algorithm), which requires a predefined ordering of variables.

B. Network Structure Learning Algorithm

Bayesian Network structure learning algorithm is generally classified into two types: the search+score algorithm and the dependence-analysis-based algorithm. The search+score algorithm aims at finding an optimal network structure. In order to measure the goodness of each explored structure in the space of available solutions, the algorithm uses a scoring function (often defined as a measure of fit between the network structure and the data) and a search method. The optimal network structure is the one that has the highest degree of fit with the training data [32]. Dependence-analysis-based algorithms are Bayesian Network learning algorithms based on quantitative validation of mutual information, and Cheng's three-stage algorithm is one of the most representative algorithms based on dependence analysis. Hence, in this research, we apply Cheng's algorithm [33]. The algorithm is divided into three phases: drafting, thickening and thinning. In the drafting phase, a network sketch will be established based on the mutual information of each pair of nodes. In the thickening phase, if any pair of nodes is not independent given some specific condition set, it adds an arc between these two nodes. In the thinning phase, it carries out conditional independence test for each arc that obtained in the thickening stage, and removes an arc if the linked pair of nodes is conditional independent given some specific condition set.

IV. Outsourced Software Project Risk Intelligent Analysis Model

A. Framework of Outsourced Software Project Risks

Because of outsourcing, software project will have two different stakeholders and decision-makers, namely customer and contractor. Therefore, risk management requires the coordination of both sides. Apart from the risks derived from the customer and contractor, outsourced software projects are also affected by the requirement complexity risks of projects such as development costs, development period, technology complexity and schedule and budget management.

This research interviews the industry experts and specialists of software engineering theory on the basis of analyzing a large number of literatures on software project risk identification to build the prototype model. After multiple rounds of research, discussion and revision, the final model contains three risk dimensions (totally 23 risk factors) and eight performance attributes of outcome, as listed in Table I and Table II.

B. Expert Knowledge of Outsourced Software Project Risk

Outsourced software project risks are divided into three types: requirement complexity risks of project, customer risks and contractor risks. CMMI (Capability Maturity Model Integration) classifies its 25 defined key process areas into four types, namely process management, project management, engineering and support. According to CMMI, the customer risks and project management risks will affect the software engineering risks [6]. Wallace's research [44] shows that the requirement complexity risks will have significant effects on project management risks and contractor risks. Meanwhile, our interviews with experts indicate that the requirement complexity risks of project will affect customer risks and software engineering risks will affect the project outcome. According to the above expert knowledge, our research defines partial orderings among risk dimensions as follows:

1) Requirement complexity risks of project precede customer risks and contractor risks, i.e.

Req_Comp → Cust_R, Req_Comp → Contr_R

2) Customer risks precede software engineering risks, which is a sub-dimension of contractor risks, i.e.

Cust_R → Soft_E

3) In the dimension of contractor risks, project management risks precede software engineering risks and software engineering risks precede project outcome, i.e.

Pro_M → Soft_E,Soft_E → Outcome

TABLE I.
RISK DIMENSIONS AND RISK FACTORS OF OUTSOURCED SOFTWARE

Requirement Complexity Risks (Req_Comp)	**Reference**	**Contractor Risks (Contr_R)**		**Reference**
1. Development Cost (DC)	[34]	Project Management (Pro_M)	1. Development Team (CDT)	[15, 35, 36]
2. Development Period (DP)	[37]		2. Project Manager (PM)	[35]
3. Function Point (FP)	[37]		3. Number of Team Members (TS)	[38]
4. Requirement of Real-time and Security (SC)	[37]		4. Number of Collaborators (EC)	[35, 36, 38]
5. Technology Complexity (TC)	[34]		5. Industry Experience (ESP)	[35, 36]
6. Requirements Stability (RS)	[35, 39, 40]	Software Engineering (Soft_E)	6. Requirement Development (RD)	[38]
7. Schedule and Budget Management (TBR)	[34, 35]		7. Requirement Management (RM)	[38, 40]
Customer Risks (Cust_R)	**References**		8. Development and Testing (IT)	[41]
1. Level of IT Application (DI)	[35]		9. Engineering Support (ES)	[35, 40]
2. Business Process (BP)	[36]		10. Plan and Control (PC)	[15, 35, 38]
3. Top Management Support (TMS)	[35, 36, 38]			
4. Client Department Support (CDS)	[35, 36, 38, 42]			
5. Client Experience (ECM)	[36, 38]			
6. Collaboration of Client Team (CTC)	[35, 36, 38]			

TABLE II.
PROJECT PERFORMANCE ATTRIBUTES

	Performances	**Attributes**	**Reference**
Project Outcome	Product Performance	1 The users perceive that the system meets intended functional requirements.	[3]
		2 The system meets user expectations with respect to ease of use, response time and reliability.	[3]
		3 The application developed is easy to maintain.	[3]
		4 The information quality which the system provide to users and organizations.	[43]
		5 The users are satisfied with the developed application.	[3]
		6 The overall quality of the developed application is high.	[37]
	Process Performance	7 The system was completed within schedule.	[3]
		8 The system was completed within budget.	[3]

C. Data Preparation and Preprocessing

Our research totally collected 154 valid samples of outsourced software project. A wide variety of industries such as government (18.83%), information industry (24.68%), manufacturing (12.99%), commerce (17.53%) etc. were represented in our samples. From the view of project scale, nearly 40% of the project had team members over 10 and development time exceeding 6 months. The development scale ranged from under 500 to over 10000 function points, and the project which had the function points under 200 was the largest contributor (51.3%). Over 87% of the respondents are project manager (34.42%), project technical director (18.18%) or members of development team (21.43%) with a related project working experience over 3 years. Therefore, the respondents are eligible to provide the credible information to the study. A summary of the demographics variables are showed in Table III.

Because of the limited numbers of collected project samples, we must discretize the sample data into binary data to make the information fully reflect in the conditional probability tables of the learned Bayesian Network. In this research, we define the state s1 is a success of project (Outcome=s1), while the state s0 (Outcome=s0) is regarded as a failure of project. As for attribute "*Number of team members (TS)*", the team which has team members under 10 is regarded as a small team (express by s0), while over 10 it is regarded as a big team (express by s1). Similarly, attribute *"Development cost" (DC), "Development period" (DP)* and *"Function point" (FP)* use 100 million, 1 year and 200 as the demarcation point respectively. The final data discretization processing results are showed in Table IV.

D. Bayesian Network Construction and Analysis

Using Cheng's algorithm [33] and the previously defined partial ordering constraints, we got the final Bayesian Network, as showed in Fig. 1. And the conditional probability tables (CPTs) of the direct parent nodes of the node *Outcome* are listed in the Appendix. Although the network is a bit complicated, it has good interpretability for each directly linked pair of nodes and contains well explainable connections within the same risk dimension or across different dimensions. The network has one project outcome, namely the target variable (labeled as *Outcome*). We listed some obtained findings from Fig. 1 as follows:

1) The project outcome, *or Outcome*, is directly affected by three risk factors, including "*Requirement development" (RD),* "*Requirement management" (RM) and* "*Development team" (CDT),* indicating that the requirement risk plays the most important role in the project outcome. Requirement development is the process that can turn users' requirements into the project requirement. Not fully understanding the users' needs will result in unclear requirement and then lead to frequent changes of the project. Frequent changing requirement and lack of efficient change management of requirement always lead to project failure [4]; CMMI defined requirement management as one of the 25 key process areas [6]. As the requirement frequently changes in software development, lack of systemic and normative requirement management is an important reason for project failure; At present, software development still cannot realize (automatic or semi-automatic) industrialized production, which determines that the human factor is one of the most important risk factors of software project [12]. Only with experienced and well trained team members effectively planning, coordinating and managing their work together can the project lead to success [35].

TABLE III.
DETAILS OF SAMPLES

Demographics Variables	Freq.	Percent
Level of the Respondents		
Company manager	8	5.19%
Project manager	53	34.42%
Project technical director	28	18.18%
Member of development team	33	21.43%
Customer representative of project	26	16.88%
Other	6	3.90%
Work Experience		
Under 3 years	42	27.27%
3-6 years	65	42.21%
7 or above	46	29.87%
Industry		
Government	29	18.83%
Education	12	7.79%
Finance	12	7.79%
Information	38	24.68%
Health	7	4.55%
Manufacturing	20	12.99%
Commerce	27	17.53%
Transportation	7	4.55%
Other	2	1.30%
Function Point		
≤ 200	79	51.30%
>200	75	48.70%

TABLE IV.
DATA DISCRETIZATION

Attributes	After discretization
Number of Team member (TS)	s0 (<10)
	s1 (≥10)
Development Cost (DC)	s0 (≤1 million)
	s1 (>1 million)
Development Period (DP)	s0 (≤1 year)
	s1 (>1 year)
Function Point (FP)	s0 (≤200)
	s1 (>200)

2) In the aspect of project requirement complexity risk dimension, *"Requirements stability" (RS)* and *"Schedule and budget management" (TBR)* directly affect "Requirement *development" (RD),* which belongs to the software engineering risk dimension (sub-dimension of the contractor risk dimension). With stable requirement, adequate time and funding, we can identify the users' needs more comprehensive and clear during the requirement development phase; *"Requirement of real-time and security" (SC)*, directly affects *"Number of collaborators" (EC)*, which belongs to the project management risk dimension (sub-dimension of the contractor risk dimension). Project with high real-time and security requirement often requires more complicated technology and needs the cooperation of multiple development teams which have different core technologies; *"Requirements stability" (RS)* directly affects *"Schedule and budget management" (TBR)*. It will be difficult to calculate the time and costs of a project if the requirement changes frequently. Consequently, the pressure to complete the project increase and will finally heightens the failure rate of project [35]; *"Function point" (FP)* directly affects the

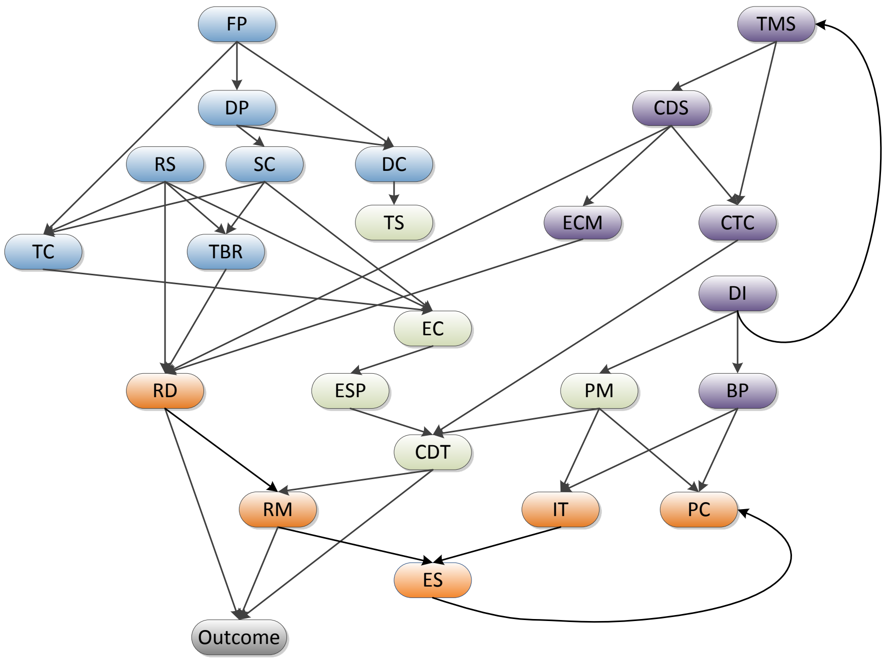

Figure 1. Bayesian Network of Software Project Risk

"*Development period" (DP), "Development cost" (DC) and "Technology complexity" (TC).* The more functions the project has, the more complicated the code implementation and consequently the more costs and time are needed.

3) In the aspect of customer risk dimension, it directly affects *"Development team" (CDT)*, which belongs to the project management risk dimension (sub-dimension of the contractor risk dimension), mainly through *"Collaboration of client team" (CTC)*. A generalized development team also contains a client team. Therefore, in a relatively big project, there is always a client team to track the project and cooperate with the development team of contractor. Hence, if the client team cannot effectively cooperate, it may lead to poor performance of the contractor development team and the failure rate will soar consequently [45]; *"Business process" (BP)*, directly affects *"Plan and control" (PC)*, which belongs to the software engineering risk dimension (sub-dimension of the contractor risk dimension). Disordered business process is an important reason that leads to project failure [46]. If the business process of client is standardized, the requirements and processing logic will be clearer. Hence, it will be easier to make the project under control and then improve the success rate of project; *"Top management support" (TMS)* directly affects *"Collaboration of client team" (CTC)* and *"Client department support" (CDS)*. "Lack of management support" is regard as one of the top ten risks [47], and collaborative culture requires effective support from all levels of management. Strong support of leadership can help build a good team as well as coordinate interest between departments, making the client team collaborate effectively so as to enhance the success rate of project.

4) In the aspect of project management risk dimension (sub-dimension of the contractor risk dimension), *"Project manager" (PM)*, directly affects "*Plan and control" (PC)*, which belongs to the software engineering risk dimension. A project manager is the core of a project team whose main task is to plan, organize and control the overall project. As the project manager must conduct a comprehensive plan and control management, poor performance of the manager will easily lead to a project failure [35, 48, 49].

5) In the aspect of software engineering risk dimension (sub-dimension of the contractor risk dimension), *"Requirement development" (RD)*, directly affects *"Requirement management" (RM)*. If the users' requirements are not correctly understood in the process of requirement development, it will easily lead to frequent changes of requirement and then affect the requirement change management [4].

The above conclusions are consistent with the software engineering theory, which is a great guideline for software project risk management.

Moreover, the model also has higher predictive accuracy. We applied 10-fold cross-validation on the target node *Outcome* to obtain its average predictive accuracy. The final average predictive accuracy of the model is 80%. Meanwhile, we adopt other measurements, namely *Precision, Recall* and *F-Measure*, to make a comparison with Decision Trees and Neural Network. The verification results show that the Bayesian Network is better than Decision Trees and Neural Network on the above measurements. A predication performance comparison of the three algorithms is showed in Table V.

TABLE V.
PERFORMANCE COMPARISON OF DIFFERENT MODELS

Measurement	Constraint-based Bayesian Network	C4.5	Neural Network
Accuracy	**80%**	77.27%	76.62%
Precision	**0.851**	0.757	0.761
Recall	**0.917**	0.773	0.766
F-Measure	**0.877**	0.763	0.764

V. CONCLUSION AND DISCUSSION

In this study, we build an intelligent analysis model for outsourced software project risk based on empirical data. In order to find objective knowledge from data, and make the model have a better interpretability, our research codified expert knowledge of outsourced software project risk management into structure constraint and combined real-life data to learn the Bayesian network structure and its parameters. Experimental results showed that the model not only has a better interpretability, but also reach a predictive accuracy rate of 80%. Meanwhile, the *Precision, Recall* and *F-Measure* are 0.851, 0.917, and 0.877 respectively. All the indicators are better than Decision Tree and Neural Network.

We acknowledge that there are some limitations in this model. Firstly, there are only 154 samples of outsourced software projects, which may not be sufficient for a Bayesian Network with 23 nodes. Secondly, data discretization may affect the information accuracy. As for future prospects, we will collect more project samples to revise the model. Currently, the risk identification framework of our project is relatively simple. Hence, we plan to explore a more systematic and comprehensive outsourced software project risk framework to better describe the software project risks.

APPENDIX CONDITIONAL PROBABILITY TABLES

Node RD:

s0	s1	RS	TBR	CDS	ECM
0.715191	0.284809	s0	s0	s0	s0
6.25E-06	0.999994	s0	s0	s0	s1
3.34E-06	0.999997	s0	s0	s1	s0
0.493728	0.506272	s0	s0	s1	s1
0.676958	0.323042	s0	s1	s0	s0
1.45E-05	0.999985	s0	s1	s0	s1
0.288756	0.711244	s0	s1	s1	s0
0.722326	0.277674	s0	s1	s1	s1
0.414231	0.585769	s1	s0	s0	s0
0.999996	4.41E-06	s1	s0	s0	s1
0.999996	3.6E-06	s1	s0	s1	s0
0.998756	0.001244	s1	s0	s1	s1
0.567152	0.432848	s1	s1	s0	s0
0.999971	2.91E-05	s1	s1	s0	s1
2.71E-06	0.999997	s1	s1	s1	s0
0.762075	0.237925	s1	s1	s1	s1

Node CDT:

s0	s1	ESP	CTC	PM
0.923077	0.076923	s0	s0	s0
0.625	0.375	s0	s0	s1
0.76923	0.23077	s0	s1	s0
0.172414	0.827586	s0	s1	s1
0.857142	0.142858	s1	s0	s0
0.09091	0.90909	s1	s0	s1
0.583333	0.416667	s1	s1	s0
0.125	0.875	s1	s1	s1

Node RM:

s0	s1	ESP	CTC	PM
0.923077	0.076923	s0	s0	s0
0.625	0.375	s0	s0	s1
0.76923	0.23077	s0	s1	s0
0.172414	0.827586	s0	s1	s1
0.857142	0.142858	s1	s0	s0
0.09091	0.90909	s1	s0	s1
0.583333	0.416667	s1	s1	s0
0.125	0.875	s1	s1	s1

Node Outcome:

s0	s1	RD	RM	CDT
1	3.33E-07	s0	s0	s0
0.751073	0.248927	s0	s0	s1
0.856381	0.143619	s0	s1	s0
6.74E-06	0.999993	s0	s1	s1
0.904553	0.095447	s1	s0	s0
0.999998	1.96E-06	s1	s0	s1
0.999997	2.9E-06	s1	s1	s0
0.999999	1.12E-06	s1	s1	s1

ACKNOWLEDGMENT

This research was partly supported by the National Natural Science Foundation of China (70801020, 61005043), the Science and Technology Planning Project of Guangdong Province, China (2010B010600034), and National Social Science Foundation of China (08AJY038).

REFERENCES

[1] J. McManus and T. Wood-Harper, "Understanding the Sources of Information Systems Project Failure," *Management services,* vol. 51, pp. 38-43, 2007.

[2] The Standish Group. (2010, 2010-10-6). *New Standish Group Report Shows More Project Failing and Less Successful Projects*. Available: http://www.standishgroup.com/newsroom/chaos_2009.php

[3] L. Wallace, *et al.*, "Understanding Software Project Risk: A Cluster Analysis," *Information & Management,* vol. 42, pp. 115-125, 2004.
[4] J. Verner and W. Evanco, "In-house software development: what project management practices lead to success?," *IEEE software,* vol. 22, pp. 86-93, 2005.
[5] Project Management Institute (PMI), *A Guide to the Project Management Body of Knowledge (PMBOK Guide)*, 4 ed. Newtown Square PA, USA: Project Management Institute, 2008.
[6] CMMI Product Team, "Capability Maturity Model Integration (CMMI SM) Version 1.1," CMU/SEI-2002-TR-011, ESC-TR-2002-011, 2002.
[7] M. J. Earl, "The risks of outsourcing IT," *Sloan Management Review,* vol. 37, pp. 26-32, 1996.
[8] D. C. Chou and A. Y. Chou, "Information systems outsourcing life cycle and risks analysis," *Computer Standards & Interfaces,* vol. 31, pp. 1036-1043, 2009.
[9] C. Fan and Y. Yu, "BBN-based Software Project Risk Management," *Journal of Systems and Software,* vol. 73, pp. 193-203, 2004.
[10] J. Pearl, *Causality: Models, Reasoning, and Inference*. Los Angeles, USA: Cambridge University Press, 2000.
[11] B. Boehm, *Software Risk Management*: IEEE Computer Society Press, 1989.
[12] R. Charette, *Software Engineering: Risk Analysis and Management*. New York, NY: McGraw-Hill, Inc., 1989.
[13] P. L. Bannerman, "Risk and risk management in software projects: A reassessment," *Journal of Systems and Software,* vol. 81, pp. 2118-2133, 2008.
[14] J. Jiang, *et al.*, "A measure of software development risk," *Project Management Journal,* vol. 33, pp. 30-41, 2002.
[15] B. Boehm, "Software Risk Management: Principles and Practices," *IEEE Software,* pp. 32-41, 1991.
[16] R. T. Nakatsu and C. L. Iacovou, "A comparative study of important risk factors involved in offshore and domestic outsourcing of software development projects: A two-panel Delphi study," *Information & Management,* vol. 46, pp. 57-68, 2009.
[17] R. Kliem, "Managing the risks of offshore IT development projects," *EDPACS,* vol. 32, pp. 12-20, 2004.
[18] T. Rajkumar and R. Mani, "Offshore software development: The view from Indian suppliers," *Information Systems Management,* vol. 18, pp. 1-11, 2001.
[19] M. E. Jennex and O. Adelakun, "Success factors for offshore information system development," *Journal of Information Technology Cases and Applications,* vol. 5, pp. 12-31, 2003.
[20] M. D. Aundhe and S. K. Mathew, "Risks in offshore IT outsourcing: A service provider perspective," *European Management Journal,* vol. 27, pp. 418-428, 2009.
[21] B. Bahli and S. Rivard, "Validating measures of information technology outsourcing risk factors," *Omega,* vol. 33, pp. 175-187, 2005.
[22] M. C. Lacity, *et al.*, "A review of the IT outsourcing literature: Insights for practice," *The Journal of Strategic Information Systems,* vol. 18, pp. 130-146, 2009.
[23] S. J. Huang, *et al.*, "Fuzzy Decision Tree Approach for Embedding Risk Assessment Information into Software Cost Estimation Model," *Journal of Information Science and Engineering,* vol. 22, pp. 297-313, 2006.
[24] Z. Xu, *et al.*, "Software Risk Prediction Based on the Hybrid Algorithm of Genetic Algorithm and Decision Tree," *Communications in Computer and Information Science,* vol. 2, pp. 266-274, 2007.
[25] T. M. Khoshgoftaar and D. L. Lanning, "A Neural Network Approach for Early Detection of Program Modules Having High Risk in The Maintenance Phase," *Journal of Systems and Software,* vol. 29, pp. 85-91, 1995.
[26] D. Neumann, "An Enhanced Neural Network Technique for Software Risk Analysis," *IEEE Transactions on Software Engineering,* vol. 28, 2002.
[27] Y. Hu, *et al.*, "Software Project Risk Management Modeling with Neural Network and Support Vector Machine Approaches," 2007, pp. 358-362.
[28] E. Lauría and P. Duchessi, "A Methodology for Developing Bayesian Networks: An Application to Information Iechnology (IT) Implementation," *European Journal of operational research,* vol. 179, pp. 234-252, 2007.
[29] I. Gashi, *et al.*, "Uncertainty Explicit Assessment of Off-the-shelf Software: A Bayesian Approach," *Information and Software Technology,* vol. 51, pp. 497-511, 2009.
[30] D. Janssens, *et al.*, "Integrating Bayesian Networks and Decision Trees in a Sequential Rule-based Transportation Model," *European Journal of operational research,* vol. 175, pp. 16-34, 2006.
[31] G. Zhang, *et al.*, "Predicting information technology project escalation: A neural network approach," *European Journal of operational research,* vol. 146, pp. 115-129, 2003.
[32] L. de Campos and J. Castellano, "Bayesian Network Learning Algorithms using Structural Restrictions," *International Journal of Approximate Reasoning,* vol. 45, pp. 233-254, 2007.
[33] J. Cheng, *et al.*, "Learning Bayesian Networks from Data: An Information-theory Based Approach," *Artificial Intelligence,* vol. 137, pp. 43-90, 2002.
[34] Z. Xu, *et al.*, "Application of Fuzzy Expert Systems in Assessing Operational Risk of Software," *Information and Software Technology,* vol. 45, pp. 373-388, 2003.
[35] L. Wallace, *et al.*, "Software Project Risks and Their Effect on Outcomes," *Communications of the ACM,* vol. 47, pp. 68-73, 2004.

[36] W. Xia and G. Lee, "Complexity of Information Systems Development Projects: Conceptualization and Measurement Development," *Journal of Management Information Systems,* vol. 22, pp. 45-83, 2003.
[37] B. W. Boehm, "Software Engineering Economics," in *Prentice-hall, Englewood Cliffs, New Jersey*, ed, 1981.
[38] R. Schmidt*, et al.*, "Identifying Software Project Risks: An International Delphi Study," *Journal of Management Information Systems,* vol. 17, pp. 5-36, 2001.
[39] C. Jones, *Assessment and Control of Software Risks*: Yourdon Press, 1994.
[40] K. D. Walter, "Software Engineering Risk Management," ed. Los Alamitos: IEEE Computer Society Press, 1996.
[41] J. J. Jiang*, et al.*, "An Exploration of the Relationship Between Software Development Process Maturity and Project Performance," *Information & Management,* vol. 41, pp. 279-288, 2004.
[42] H. Barki*, et al.*, "Toward an Assessment of Software Development Risk," *Journal of Management Information Systems,* vol. 10, pp. 203-225, 1993.
[43] S. Nidumolu, "The Effect of Coordination and Uncertainty on Software Project Performance: Residual Performance Risk as an Intervening Variable," *Information Systems Research,* vol. 6, p. 191, 1995.
[44] L. Wallace*, et al.*, "How Software Project Risk Affects Project Performance: An Investigation of the Dimensions of Risk and an Exploratory Model," *Decision Sciences,* vol. 35, pp. 289-321, 2004.
[45] M. Carr*, et al.*, "Taxonomy-based risk identification," Software Engineering Institute, Carnegie Mellon University (CMU/SEI) CMU/SEI-93-TR-6, ESC-TR-93-183, 1993.
[46] F. Reyes*, et al.*, "The optimization of success probability for software projects using genetic algorithms," *Journal of Systems and Software,* 2011.
[47] The Standish Group. (2001). *EXTREME CHAOS*. Available: www.standishgroup.com/sample_research/PDFpages/extreme_chaos.pdf
[48] M. Keil*, et al.*, "A Framework for Identifying Software Project Risks," *Communications of the ACM,* vol. 41, p. 83, 1998.
[49] S. Huang and W. Han, "Exploring the Relationship between Software Project Duration and Risk Exposure: A Cluster Analysis," *Information & Management,* vol. 45, pp. 175-182, 2008.