A RULE-BASED APPROACH FOR DISCOVERING EFFECTIVE SOFTWARE TEAM COMPOSITION

[2]School of Computing, Universiti Utara Malaysia, Malaysia
[3]School of Technology Management & Logistic, Universiti Utara Malaysia, Malaysia

a-rehman@iba-suk.edu.my; mazni@uum.edu.my; kamalimran@uum.edu.my

ABSTRACT

Human aspects in software engineering play a key role in composing effective team members. However, to date there is no general consensus on the effective personality types and diversity based on software team roles. Thus, this paper aims to discover the effective personality types and diversity based on two software team roles – team leader and programmer by using a rule-based approach. The rule-based approach by employing the rough set technique was used to discover patterns of the data selected. In this study, four main steps were involved to discover the patterns – reduct generation rules, rules generation, rules filtering, and rules evaluation. The results show that the rules generated achieved acceptable prediction accuracy with more than 70 per cent accuracy. In addition, the ROC value achieved 0.65, which indicates the rule-based model is valid and useful. The results reveal that the extrovert personality type is dominant for both software team roles and a homogeneous or heterogeneous team plays an equal role to determine an effective team. This study provides useful rules for decision makers to understand and get insight into selecting effective team members that lead to producing high quality software.

Keywords: Software team composition, personality types, diversity, team roles, rule-based.

INTRODUCTION

Software engineering (SE) is perceived as a technical activity. However, there is growing evidence that the success of a software project depends on humanistic aspects (Dingsoyr & Dyba, 2012; Martínez, Licea, Rodríguez-Díaz & Castro, 2010). One of the humanistic aspects that may impact the quality of a software project is the composition of the personality types and behaviour among the team members (Acuña, Gómez, & Juristo, 2009; Cunha & Greathead, 2007; Koroutchev, Acuña, & Gómez, 2013; Mazni, Sharifah Lailee & Naimah, 2011; Ratnasingam, 2009). A number of studies were done in the past on team composition and personality types in software engineering, but the issue pertinent to a suitable personality type composition for effective teamwork is still being questioned (da Silva et al., 2013; Dingsoyr & Dyba, 2012).

Software team diversity is one of the significant elements in determining team effectiveness (Peslak, 2006; Woehr, Arciniega, & Poling, 2013). Diversity may refer to diversification of demographic factors, knowledge, skills, and personality types amongst team members. In this study, diversity is defined as the differences of the personality types among the members in a team. Currently, there is no general consensus on the advantages of having diversification amongst team members towards developing high quality software. This is because team dynamism plays a key role in software team composition.

To date, most techniques to analyse patterns of personality types in software engineering are based on statistical and qualitative analysis (Acuña et al., 2009; Bradley & Hebert, 1997; Koroutchev et al., 2013). There is a lack of research on applying the rule-based technique to understand the patterns that exist in the data selected. The rule-based approach offers significant advantages because the rules generated are easy to understand and can be easily interpreted. This is because the rules generated imply rules-decision in human understandable forms. Therefore, this study aims to discover the effective software team composition based on personality types, diversity, and software-team roles by using a rule-based approach.

LITERATURE REVIEW

Software development projects are usually highlighted by three key aspects; technical, managerial, and humanistic aspects (Hazzan & Hadar, 2008). These three aspects, all combined together, pave the way for high-performance software-development projects. Each aspect plays its role to contribute towards the positive outcome for the software-development projects. For instance, the technical aspect facilitates and provides smoothness in testing, implementation and design for software development, whereas the managerial aspects are also important to manage time and schedules for high quality software development. On the other hand, human aspects are also of equal importance in the sense that they enable teammates to communicate and maintain collaboration, thus accommodate the learning process among team members.

An individual performance in software development has direct interaction with the personality of an engineer (Gorla & Lam, 2004). According to Cruz (2011), personality is the major source to gain favourable outcomes in software development than process, tools, and technology. Moreover, job satisfaction, project success, and conflicts resolution can be conceived with the composition of the team with effective personality types. In the research of an effective team composition of personality types based on team roles, Martinez (2010) mentioned that team success can be achieved when the team members are assigned to appropriate or right roles.

In software engineering, personality types among team members plays a critical role to ensure that members in the team are comfortable and thus, working together effectively. One of the popular personality types in SE is the Myers-Briggs Type Indicator (MBTI). This personality type is widely used and accepted amongst researchers in the SE domains (Bradley & Hebert, 1997; Cunha & Greathead, 2007; Karn & Cowling, 2006; Karn, Syed-Abdullah, Cowling, & Holcombe, 2007; Mazni, 2012). The MBTI consists of 16 personality types that combine 4 pairs of personality type dimensions, which are:

1. Introvert (I) – Extrovert (E).
2. Sensing (S) – Intuitive (N).
3. Thinking (T) – Feeling (F).
4. Judging (J) – Perceiving (P).

Mazni and Sharifah-Lailee (2010) indicated that certain personality types, namely extrovert (E), sensing (S), feeling (F) and judging (J), affect the software project's success the most, where the last type inevitably affects project success, as most software team members are the judging types. Their findings were also emphasized by Mazni, Sharifah-Laileee and Naimah (2011). However, their study did not consider software team role as one of the key variables to investigate the effective personality types among the team members.

Another study by Martinez et al. (2010) proposed a RAMSET (Role Assignment Methodology for Software Engineering Teams) in which the personality of a team member was defined with team roles. In the study, the researcher focused on students' expertise in the particular area of programming languages and databases. The MBTI personality test was used to gain the personality of an individual student. The research revealed the fact that Extrovert (E)'s and Sensing (S)'s can be suitable for analysts and designers, and Introvert (I)'s for programmers/developers. Moreover, the researcher recommended that the ISTP personality type is suitable for programmers, ENTJ for designers, and ESTJ for analysts and testers.

In order to mitigate the risks of assigning ineffective personality types of team members, Capretz and Ahmed (2010) proposed a conceptual model of the effective personality types team members. The authors suggested a suitable personality type according to software team roles. For example, programmers must have introvert (I) personality, whereas system analysts must have extrovert (E) personality in order to compose an effective team. They also suggested that software designers should be with intuitive (N) and thinking (T) personalities. Moreover, programmers must have IST personality to be effective members. Lastly, sensing (S) and judging (J) personality types can benefit the testers. Nevertheless, this model is non-empirical which is difficult to test its effectiveness. The following Table 1 demonstrates the suitable MBTI personality types of software team members based on their roles.

Table 1

MBTI and Roles in Software Development

No.	Role in software development	Author	Personality type	Role Explanation
1	Team leader	Gorla and Lam (2004)	• Extrovert, Intuition and Feeling (ENF) 1. ENFP 2. ENFJ	• assigning tasks • reporting • dealing external stakeholders • responsible of progress of project
2	System analyst	Ahmed and Capretz (2010); Gorla and Lam (2004)	• Extrovert and Feeling (EF) 1. ENFJ 2. ESFJ 3. ESFP 4. ENFP	• requirement gathering • logical modelling

(continued)

No.	Role in software development	Author	Personality type	Role Explanation
3	Designer	Ahmed and Capretz (2010); Martinez et al. (2010)	• Intuition and Thinking (NT) 1. ENTJ 2. ENTP 3. INTP 4. INTJ	• provide solution to problem • design the flow of project
4	Programmer	Ahmed and Capretz (2010); Martinez et al. (2010)	• Introversion, Sensing, and Thinking (IST) 1. ISTJ 2. ISTP	• identifies control structures • code the problem
5	Tester	Ahmed and Capretz (2010); Martinez et al. (2010)	• Sensing and Judging (SJ) 1. ESTJ 2. ESFJ 3. ISFJ	• bug finding • bug fixing

Table 1 shows the research conducted in the past on personality types based on their roles. According to past researches, the programmer role is discussed with two major personality types: ISTJ and ISTP. Moreover, based on the job requirements of the team leader, the personality types disclosed are ENFP and ENFJ.

Diversity has remained the prominent part of research in software development which includes team diversity, process diversity and culture diversity. The need of diversity such as member's variation of skills and attitudes is important to ensure that team members can solve problem in different ways (Beck & Andres, 2005). According to Gorla and Lam (2004), the composition of a team based on heterogeneity could be considered to be better in results for some phases of the software-development life cycle. For example, heterogeneous teams could be optimal for requirement gathering and team leadership tasks, whereas homogenous teams perform better in programming kind of jobs.

A myriad of data-mining techniques that are being followed by and most of them are followed using classical statistics, whereas few data-mining techniques are performed following artificial intelligence and machine learning. The data-mining techniques that are profusely put into use are decision tree, logistic regression, artificial neutral network (ANN), rough set and support vector machine (SVM). Each of these techniques has its own merits and demerits that can be taken into account before implementing them into data analysis.

The key advantage of the statistical logistic regression lies in the fact that it is quite effective while handling multiple predictors of mixed-data types that yield reliable binary outcomes. Despite this advantage, this technique is not devoid of disadvantages. Firstly, it requires extensive data so as to perform modelling and secondly, researchers should be well qualified in statistical and domain knowledge to operate this technique adroitly. ANN is another popular technique in the field of data-mining that not only enables researchers to predict tasks but it has also the tendency to help secure higher and accurate performance. But its darker sides cannot be neglected while performing this technique for it lacks the tendency in terms of over-fitting. Moreover, it is considered as a black-box technique that is less effective to assess categorical data.

The SVM data-mining technique is also eminent among researchers for its prime advantageous quality that enables researchers to obtain satisfactory performance and accuracy based on normal distributed data. However, unlike ANN, this technique is less lucrative and ineffective when employed to analyse categorical data. To cope with this problem of handling categorical data, researchers mostly rely on the decision-tree technique that is lucrative for analysing mixed data in general, and analysing categorical data in particular. But, this technique is often taken into due consideration prior to employing it because of its drawbacks for generating complex tree structures that cannot be fitted easily in the data. Rough set is quite a new technique, compared to all other techniques, that facilitates researchers to generate IF-THEN rule that can be easily interpreted and can easily be suited to both small sample size and categorical data types. But, this technique is often criticized for generating rules in excess that cannot be followed easily while making pattern interpretation. Additionally, this technique works on continuous data discretization that mostly lessens data knowledge representation.

Keeping in view the merits and the demerits of the data-mining techniques, Kotsiantis (2007) maintains that since data-mining is an exploratory process in nature, no data-mining technique or learning algorithm can be called as the best suited technique to analyse different data sets and domains. In the same vein, Dreiseitl and Ohno-Machado (2002) state that researchers need to know the nature of data and employ the techniques accordingly. However, past literature reveals the fact that three kinds of data-mining techniques-cum-rough set, decision tree and logistic regression can be employed to devise prediction models. These techniques were declared safe techniques for devising prediction models based on the data normality assumption, sample size and type of data.

In this study, rough set, which is a supervised machine learning technique, was chosen because of the following reasons (Düntsch & Gediga, 2000; Hui, 2011; Mazni et al., 2010; Pawlak, 1997):

1. It is free from making assumptions of data and its size. As most of the nominal data type, i.e. personality type and team role, was used in this study, its use was quite useful and helpful. It is the reason why normal data distribution was not achieved. In addition, rough set is also useful in handling the small-size sample which was used in this study.
2. Rough set generates IF-THEN decision rules which pave the way for researchers to identify and understand different patterns in the collected data. Thus, rough set can also be interpreted as a white-box model for it reveals hidden patterns of data with rule-based patterns. This quality of rough set not only makes the model to be easily conceived but also eases the interpreting data relationship.
3. Rough set functions are available within a Rough Set Toolkit for Analysis of Data (ROSETTA) tool, and can be used to analyse data using the rough set theory. The tool was designed within the rough set discernibility framework and it was integrated with a collection of rough set algorithms, thus it is an important tool for analysing research data.

The ROSETTA tool provides extensive support to researchers of mining whose research is based on rough set. Penalty of work on rough set has already been carried out by using this software. For instance, Strömbergsson et al. (2006) used the ROSETTA tool in their research of "Rough set-based proteochemometrics modelling of G-protein-coupled receptor-ligand interactions". In the same vein, Wang et al. (2002) also used the services of ROSETTA in their work - "RIDAS-a rough set-based intelligent data analysis system". Moreover, Swiniarski and Skowron (2003) used this software in discussing the experiments in their research on rough set methods "pattern recognition letters". Additionally, in the latest research of Shen and Chen (2013), in the management of customer relationships research, rough set approach was used with the ROSETTA tool.

In sum, it can be concluded that most researches conducted in the past were lacking in discovering the effective personality types and diversity based on software team roles. Furthermore, the researches were non-empirical and thus difficult to claim the generalization of results. Therefore, this research intends to fill this gap by exploring the effective pattern of personality types, diversity and software team roles by using the rule-based approach.

METHODOLOGY

In order to investigate the relationships amongst the variables investigated; personality types, diversity and team roles, data from Mazni (2012) was chosen. This data was chosen because the nature of the present study requires a similar kind and the same number of research participants that were used in the previous research. In addition, the research also focuses on a software development team that has personality type differences which suit this study. The dataset selected consists of 220 participants from two different settings; 184 participants were involved in an academic setting and 41 participants were involved in an industrial setting. The main purpose of the researcher for collecting data from two different settings was to produce effective and unbiased results.

The data selected was examined by using a rule-based approach. The rule-based approach is based on the rough set technique. The rough set technique was chosen because it could handle imprecise and uncertain datasets. Moreover, data-size assumptions and normality of data are not required in rough sets (Düntsch & Gediga, 2000; Hui, 2011; Pawlak, 1997). Rough set generates IF-THEN decision rules which pave a way for researchers to identify and understand the different patterns in the collected data.

The variable denotes the selection of a predictor and outcome variables which were used to discover the effective personality. The selections of the variables in the data sets were based on the research problem, which involved three predictor variables and one outcome variable. The predictor variables were team role, personality type and team diversity whereas the team performance was an outcome variable. Table 2 shows the details of these predictor and outcome variables and its values.

Table 2

Variables Used in the Research

No.	Variables	Values
1	Predictor variables:	
	1. Team role	1=Team leader 2=Programmer
	2. Personality types	Four pairs in MBTI: I-E(Introvert-Extrovert) • 1=Introvert • 2=Extrovert

(continued)

No.	Variables	Values
		S-N(Sensing-Intuitive) • 1=Sensing • 2=Intuitive T-F(Thinking-Feeling) • 1=Thinking • 2=Feeling J-P(Judging-Perceiving) • 1= Judging • 2= Perceiving
	3. Team diversity	Divided into two groups: • 0-3=Homogeneous • 4-8=Heterogeneous
2	Outcome variable	
	1. Team performance	0=Ineffective 1=effective

The data selected was examined using the ROSETTA tool, a tool for analysing data using the rough-set technique. In order to investigate the patterns of data, several tasks were performed. The tasks were:

Reduct Generation Rules

In this step, the computation for the reducts was carried out using the rough sets approach by the ROSETTA tool. This process was carried out to determine the minimal attributes that show knowledge patterns in the data. To design a reliable model, redundant and unimportant data was screened out through reduction.

Both Genetic Algorithm (GA) and Johnson's were used to reduct the generation rules. Holland (1992) and Hvidsten (2010) state that GA is an effective method to search optimal solutions and to solve searching problems. In addition, Johnson (1974) stated that the Johnson algorithm appeals to a speedy algorithm for computing single reduct only. Both algorithms for generating reducts i.e GA and Johnson algorithm were employed for identifying the algorithm which enables to produce better and accurate classifications.

Rules Generation

Under these rule patterns, the predictor variable and the outcome variables were extracted from the data which was generated in the form of IF-ELSE rules based on results obtained from the reducts generation. Moreover, the decision-rules were comprised of any or all of three predictor variables: personality types, team role, and team diversity. These three variables predict the outcome variable, i.e. team performance, and also indicate the different patterns showing relationships amongst these variables. Detailed results for generating rules from this step by using the ROSETTA tool are discussed in the results and discussion section.

Rules Filtering

The rules-filtering task was carried out to determine the most frequently appearing variables in the rules generated. The higher the appearance of the variables the more significant the variable is to determine team effectiveness (Clark, 2009; Wong & Chung, 2007). Only the rules that have effective decisions were considered to be analysed. Once rules were constructed and filtered, the evaluation of the rules was performed so as to ensure that the generated patterns could be generalized to design an accurate prediction model. For this purpose, rough set rules were used along with the ROSETTA tool for it provides a myriad of classifications procedures that consist of standard voting, object tracking and naïve Bayesian.

Rules Evaluation

For evaluating the rules, the hold-out validation method was used to ensure prediction accuracy and validate the rules for the prediction model. Moreover, in the hold-out method, the training sets (academic data sets) were examined with the test set of data (industrial sets). It was decided to use standard voting for it was handy for creating accuracy in classification (Olson & Delen, 2008; Witlox & Tindemans, 2004). Standard voting is considered to be an ad-hoc classification technique which is used to assign a numerical certainty factor for each object in each decision class. Moreover, this technique is also handy for computing condition probability for each correct-decision class (Øhrn, 1999).

In this study, prediction accuracy was benchmarked at 70 per cent because it has the feasibility to deal with new data sets. According to Bakar (2011) and Hvidsten (1999), 70 per cent prediction accuracy is acceptable for modelling. The results of standard voting and classification experiments performed on both team roles were discussed in the previous section.

In addition, the proposed rule-based model based on team members' roles was evaluated by using the receiver-operating characteristic (ROC) method. The ROC curve method is considered an effective method for testing and evaluating the performance of diagnostic tests. This technique has been widely used in medical, psychological, social science, machine learning, and engineering researches (Kumar & Vijayalakshmi, 2011; Park, 2004).

RESULTS OF RULES GENERATED

In this section, rules generated using the ROSETTA tool will be presented based on the software team's roles.

Team Leader Role

Initially, 22 rules were generated for the team leader role. In order to understand the relationships of the patterns that existed in the data, every pair of MBTI personality type indicator was observed based on decision (effective) rules. In this study, Q(1) refers to the effective decision rules. After filtering the rules, only 9 rules were considered as effective decision rules.

Table 3

Decision Rules for Team Leader Role

Rule number	Rules
1	ie(2) AND sn(2) AND Diversity(2) => Q(1)
2	sn(2) AND tf(2) AND Diversity(3) => Q(0) OR Q(1)
3	sn(2) AND Diversity(6) => Q(1)
4	ie(2) AND Diversity(1) => Q(1)
5	ie(2) AND sn(2) AND tf(2) AND Diversity(5) => Q(1)
6	ie(1) AND sn(2) AND Diversity(4) => Q(1)
7	Diversity(7) => Q(1)
8	ie(1) AND tf(2) AND Diversity(4) => Q(1)
9	ie(2) AND sn(1) AND tf(1) AND Diversity(4) => Q(1)

Based on Table 3, the first pair of MBTI, I-E (Introvert-Extrovert) occurred six times (see rule numbers 1, 4, 5, 6, 7 and 8) in which the E [i.e. (2)] personality types appeared four times and I [i.e. (1)] types of personality appeared two times. Therefore, the E personality type was observed to be more effective with appearances four rules (66.7%) than the I type of personality which appeared two rules (33.3%). The second pair of MBTI personality indicators, S-N (Sensing- Intuitive), was also observed to be effective (decision) in the team leader's rules. This pair also appeared six times in the effective rules (see rule numbers 1, 2,3, 5, 6 and 9), in which N [sn(2)] got higher appearances in the rules by appearing five times (83.3 per cent), whereas S [sn(1)] personality types got lower appearances than N [sn(2)] by obtaining 16.7 per cent. Here the N types of personality were shown as dominant personality types in the second pair.

In the T-F (Thinking-Feeling) pair of MBTI personality indicators, the personality types were observed based on 9 effective rules as perceived in Table 2. It shows that the effective rules in the T-F pair appeared four times. The personality type F dominated T by appearing three times (75 per cent). The last pair of MBTI personality indicators J-P were not found in the effective rules of the team leader role in the data set. This is because the majority of the team leaders have J personality type.

The team personality diversity was also analysed in the team leader role based on the effective rules. It was seen that heterogeneous team leaders appeared more effective than the homogeneous team leaders. In this study, heterogeneous refers to the team that has more than four diversities. In the results, the percentage for the heterogeneous team leaders was 66.7 per cent by appearing 6 times in the effective rules whereas the homogeneous teams appeared three times with 33.3 per cent in the effective rules in the dataset.

Programmer Role

For the programmer role, 17 rules were generated. After filtering the rules, only seven rules were considered as effective decision rules.

Table 4

Decision Rules for Programmer Role

Rule number	Rules
1	sn(1) AND tf(1) AND Diversity(2) => Q(0) OR Q(1)
2	ie(1) AND sn(1) AND Diversity(6) => Q(0) OR Q(1)
3	ie(2) AND tf(1) AND Diversity(1) => Q(0) OR Q(1)
4	tf(2) AND Diversity(1) => Q(1)
5	sn(2) AND tf(1) AND Diversity(4) => Q(1)
6	Diversity(7) => Q(1)
7	sn(2) AND tf(1) AND Diversity(2) => Q(1)

Based on the MBTI personality indicator, the I-E pair was observed from the effective decision rules generated by ROSETTA. The first pair of MBTI was shown only two times (rule numbers 2 and 3) in the decision rules. Both personality types I [i.e. (1)] and E [i.e. (2)] appeared once in the rules. E obtained 57.1 per cent and I got 42.9 per cent in the first pair. The second pair of MBTI personality indicators, S-N (Sensing-Intuitive), was repeated four times in the decision rules (rule numbers 1, 2, 5 and 7) in which both personality types S and N appeared 2 times. The T-F pair of MBTI appeared 5 times (rule numbers 1, 3, 4, 5, and 7) in the decision rules.

In the T-F pair, the T personality types appeared more frequently in the decision rules than F. In the results, it was seen that T appeared four times with 80 per cent appearance. On the other hand, F only got 1 appearance with 20 per cent T personality types could be categorized as effective as they appeared 80 per cent in the rules. It was observed that no rule was extracted from the experiment for the J-P pair of MBTI personality indicators. In descriptive analysis, it was observed that J remained more dominant P.

Team personality diversity was similarly observed as in the team leader section. Based on the rules generated by using the rough sets technique, it was also obtained that homogeneous programmers were appeared more effective than the heterogeneous programmers. In the results, the percentage of the homogeneous teams was 57.1 per cent by appearing 4 times in the effective rules whereas the heterogeneous teams appeared 3 times with 42.9 per cent in the effective rules.

Prediction Accuracy Results

In this study, prediction accuracy is used in order to assess the quality in the learned model and to check the accuracy of the prediction given by the model. Prediction accuracy was determined by applying the hold-out method using the ROSETTA tool (as mentioned in the previous section). Once the results were obtained, the evaluation was assessed based on the produced results.

The academic data set was used to generate rules and the industrial set was kept for validation purpose. Therefore, the accuracy of the rules was observed by applying the industrial data set on the obtained results from the academic data set. Likewise, the academic data set was divided into two subsets: team leader and programmer, in order to construct the model based on roles. Similarly, the industrial data set was also distributed into two subsets, in order to evaluate the accuracy rules and validate the model.

Prediction Accuracy of Team Leader Role Subset

As discussed in Section 3.0, the evaluation of the generated rules was obtained by applying standard voting. Standard voting is a classifier to recognize the patterns of the data that map input data to a category. 70 per cent was decided as a benchmark for checking prediction accuracy. The prediction accuracy of the team leader subset is presented in Table 5.

Table 5

Prediction Accuracy of Team Leader Subset

		Predicted		
		0	1	(*100)
Actual	0	6	0	100
	1	1	1	50
	(*100)	85.71	100	87.5

Based on Table 5, the prediction accuracy was 87.5 per cent which was greater than the benchmark (i.e. 70 per cent), and thus could be considered as satisfactory prediction accuracy. Therefore, the model was considered as effective for predicting the team leader personality type.

Prediction Accuracy of Programmer Role Subset

Standard voting classification was applied on the industrial programmer set based on the rules generated by the academic programmer set. The prediction accuracy of the programmer subset is presented in Table 6.

Table 6

Programmer Subset's Prediction Accuracy

		Predicted		
		0	1	(*100)
Actual	0	24	1	96
	1	7	1	12.5
	(*100)	77.41	0.5	75.7

Based on Table 6, the results of prediction accuracy was quite satisfactory by obtaining 75.7 per cent which is greater than the benchmark of 70 per cent. The results of this model indicated that a programmer personality can be predicted by these results.

Average Prediction Accuracy

In this study, the academic data set was firstly divided into two subsets in order to gain the rules dependent on team role. After getting the individual prediction accuracy, the accuracy of both these subsets were joined together in mandate to find the average accuracy that could be generalized for the entire model. Table 7 shows the average results of prediction accuracy of the model.

Table 7

Average Prediction Accuracy

Team leader accuracy	Programmer accuracy	Average accuracy
87.5	75.7	81.6

Based on Table 7, the overall accuracy of the model (81.6 per cent) was obtained from team leader and programmer accuracy. Hence, the results of this model could be generalized for determining the effective rules for personality type and diversity based on team leader and programmer roles.

Receiver Operating Curve (ROC) Result

In this study, the rule-based model was evaluated by using the Receiver Operating Characteristic (ROC) curve. According to Fawcett (2006), if the model area under the ROC curve is equal to 1 then the model is perfect. But, it is considered as ineffective if it is equal to 0.5 or less. Moreover, the author maintained that the model is considered acceptable if the ROC value is higher than 0.5. In this study, the ROSETTA tool was used to obtain the area under the ROC curve based on both data sets (i.e. the academic dataset was used to construct the model and the industrial data set was kept for validation). Table 8 shows the details of the area under ROC curve in both sub sets of the academic data set, which were tested by the industrial data set.

Table 8

Area under ROC Curve Results

Area of team leader sub set	Area of programmer sub set	ROC results
0.67	0.62	0.65

In this study, the average ROC value is 0.65 which demonstrates that the proposed rule-based model is acceptable. Therefore, based on these results, the model can be generalized to discriminate effective software team composition of personality types and diversity based on software team roles.

DISCUSSION

Overall, nine rules were obtained as effective rules for team leader role. Decision rules were then observed based on the MBTI personality indicators, in which each pair was discussed separately. In the first pair, E remained dominant over on I by obtaining 66.7 per cent. N outperformed the S in second pair of MBTI by obtaining 88.3 per cent. In the third T-F pair, decision rules T outperformed F with 75 per cent. The last pair of MBTI, J-P, remained invisible in the rules. Moreover, heterogeneous teams outperformed the homogenous by obtaining 66.7 per cent.

For the programmer role, seven rules were considered as effective decision rules. Based on the MBTI personality indicators decision rules were then observed. Each pair was discussed separately. In the first pair, I and E were found equal in the rules but E got (57.1 per cent) a higher percentage based on the conditions supporting compared to I. N and S in the second pair of MBTI also appeared equal in the rules. In the third pair T-F, T outperformed F with 75 per cent. The last pair of MBTI, J-P, was not visible in any decision rule. Finally, homogenous teams outperformed the heterogeneous with 57.1 per cent.

The results show that the E personality type is significant in determining team effectiveness for both team roles. A good leader and programmer needs to actively communicate with the users in order to get clear requirements. This finding is in line with a previous study which demonstrated that extrovert members give an impact on the quality of software produced by a team (Acuña et al., 2009). In addition, T, thinking personality types are dominant for the programmers since it is natural for a programmer to have the ability to make logical and objective decisions. These results are supported by other studies (Capretz & Ahmed, 2010; Peslak, 2006).

Data sets selected from Mazni (2012) were analysed and mined to extract significant rules that can determine effective personality types and diversity based on software team roles. Rules generating algorithms (i.e. GA and Johnson Algorithm) were applied on the data sets to generate rules which included the three predictor variables investigated (team role, personality types and diversity) to determine team effectiveness. Once the rules were obtained, classification techniques and standard voting were applied on those rules in order to check the validity and reliability of the rules generated. Several experiments were carried out to demonstrate the applicability of the rules and the results demonstrated that the rules achieved acceptable prediction accuracy, which more than 70 per cent.

The area under the ROC curve values determines the validity of a model developed. The model is considered valid if the ROC value is greater than 0.5. The method was applied separately on both data sets and then the results were combined to obtain the average value for model validation. The team leader subset of the area under the ROC value obtained is 0.67, whereas for the programmer subset the value is 0.62. Therefore, the average ROC value for the constructed model is 0.65. This suggests that the proposed constructed rule-based model is valid and useful.

CONCLUSION

The major contribution of this study is in the field of human aspects in software engineering (SE), which offers a rule-based model of the software team roles, personality types and diversity. The rules generated by using the rough sets technique can serve as a foundation for decision-makers to compose effective team compositions based on humanistic aspects. This study revealed that there are significant relationships between personality types and diversity based on software team roles. The results demonstrated that the extrovert personality types play a dominant role to induce a team to be effective team. In addition, this research also revealed that heterogeneous team leaders with diverse personality types in a team can be effective leaders; whereas programmers can work better in homogeneous software development teams.

This study only focuses on two main team roles in constructing the rule-based model, which are team leaders and programmers. Therefore, future research should find out the effective personality types for system analysts, software designers, and software testers, in order to improve the constructed rule-based model. This research can be extended by collecting data from a multi-cultural context as the empirical data used in this study is based on only the Malaysian context. Having data from a multi-cultural context may reduce the cultural boundaries and values differences in software team composition. Finally, the rule-based model from this study is based on the rough sets technique. Therefore, future work can explore and integrate other techniques such as decision tree by working on the same problem and using the same data to improve the reliability and accuracy of the rule-based model.

REFERENCES

Acuña, S. T., Gómez, M., & Juristo, N. (2009). How do personality, team processes and task characteristics relate to job satisfaction and software quality? *Journal of Information and Software Technology*, *51*(3), 627−639.

Bakar, A. A., Kefli, Z., Abdullah, S., & Sahani, M. (2011). *Predictive models for dengue outbreak using multiple rulebase classifiers*. Paper presented at the International Conference on Electrical Engineering and Informatics (ICEEI 2011), Bandung, Indonesia.

Beck, K., & Andres, C. (2005). *Extreme programming explained: Embrace change* (2nd ed.) USA: Addison Wesley.

Bradley, J. H., & Hebert, F. J. (1997). The effect of personality type on team performance. *Journal of Management Development*, *16*(5), 337−353.

Capretz, L. F., & Ahmed, F. (2010). Making sense of software development and personality types. *It Professional*, *12*(1), 6−13.

Clark, S. D. (2009). Characterising and predicting car ownership using rough sets. *Transportation research part C: Emerging technologies, 17*(4), 381−393.

Cruz, S. S., da Silva, F. Q., Monteiro, C. V., Santos, P., & Rossilei, I. (2011). Personality in software engineering: Preliminary findings from a systematic literature review. *In Evaluation & Assessment in Software Engineering* (EASE 2011), *15th Annual Conference on* IET, pp. 1−10.

Cunha, A. D. D., & Greathead, D. (2007). Does personality matter? An analysis of code-review ability. *Commun. ACM, 50*(5), 109−112.

Da Silva, F. Q. B., França, A. C. C., Suassuna, M., De Sousa Mariz, L. M. R., Rossiley, I., De Miranda, R. C. G. et al. (2013). Team building criteria in software projects: A mix-method replicated study. *Information and Software Technology*.

Dingsoyr, T., & Dyba, T. (2012). *Team effectiveness in software development: Human and cooperative aspects in team effectiveness models and priorities for future studies.* Paper presented at the 5th International Workshop on Cooperative and Human Aspects of Software Engineering (CHASE).

Dreiseitl, S., & Ohno-Machado, L. (2002). Logistic regression and artificial neural network classification models: A methodology review. *Journal of Biomedical Informatics*, *35*(5−6), 352−359.

Düntsch, I., & Gediga, G. (2000). *Rough set data analysis: A road to non-invasive knowledge discovery* (Vol. 2). Bangor, Bissendorf: Metoδos Publisher.

Fawcett, T. (2006). An introduction to ROC analysis. *Pattern Recogn. Lett., 27*(8), 861−874.

Gorla, N., & Lam, Y. W. (2004). Who should work with whom? *Communications of the ACM* June, (No. 6).

Hazzan, O., & Hadar, I. (2008). Why and how can human-related measures support software development processes? *Journal of Systems and Software*, *81*(7), 1248−1252.

Hui, Y. (2011). Ad hoc networks based on rough set distance learning method. *Information Technology Journal*, *10*(5), 1038−1043.

Hvidsten, T. R. (1999). *Fault diagnosis in rotating machinery using rough set theory and ROSETTA*. Trondheim, Norway: Department of Computer Science and Information Science, Norwegian University of Science and Technology.

Karn, J., & Cowling, T. (2006). *A follow up study of the effect of personality on the performance of software engineering teams*. Paper presented at the International Symposium on Empirical Software Engineering, Rio De Janeiro, Brazil.

Karn, J., Syed-Abdullah S., Cowling, A. J., & Holcombe, M. (2007). A study into the effects of personality types and methodology on cohesion in software engineering teams. *Behaviour & Information Technology, 26*(2), 99−11.

Koroutchev, K., Acuña, S. T., & Gómez, M. N. (2013). The social environment as a determinant for the impact of the big five personality factors and the group's performance. *International Journal of Human Capital and Information Technology Professionals, IJHCITP, 4*(1), 1−8.

Kotsiantis, S. B. (2007). Supervised machine learning: A review of classification techniques. *Informatica, 31*, 249−268.

Kumar, S. A., & Vijayalakshmi, M. N. (2011). Efficiency of decision trees in predicting student's academic performance. In D.C. Wyld (Ed.). *CCSEA, CS & IT 02, First International Conference on Computer Science, Engineering and Applications* (pp. 335−343). AIRCC.

Martínez, L. G., Licea, G., Rodríguez-Díaz, A., & Castro, J. R. (2010). *Experiences in software engineering courses using psychometrics with ramset*. Paper presented at the Proceedings of the Fifteenth Annual Conference on Innovation and Technology in Computer Science Education.

Mazni, O. (2012). *The effectiveness of an agile software methodology: Empirical evidence on humanistic aspects*. Universiti Teknologi Mara, Shah Alam.

Mazni, O., Sharifah Lailee, S.-A., & Naimah, M. H. (2010). *Analyzing personality types to predict team performance*. Paper presented at the Cssr'10, Kuala Lumpur, Malaysia.

Mazni, O., Sharifah Lailee, S.-A., & Naimah, M. H. (2011). Developing a team performance prediction model: A rough sets approach. In A. A. Manaf (Ed.), *ICIEIS 2011 Commmunication in Computer and Information Sciences, CCIS Part II,* (*252*, 691−705). Springer-Heidelberg.

Øhrn, A. (1999). *ROSETTA technical reference manual*. Trondheim: Norwegian University of Science and Technology.

Olson, D. L., & Delen, D. (2008). *Advanced data mining techniques*. Berlin Heidelberg: Springer-Verlag.

Park, S. H., Goo, J. M., & Jo, C. H. (2004). Receiver operating characteristic (ROC) curve: Practical review for radiologists. *Korean Journal of Radiology*, *5*(1), 11−18.

Pawlak, Z. (1997). Rough set approach to knowledge-based decision support. *European Journal of Operational Research, 99*(1), 48−57.

Peslak, A. R. (2006). *The impact of personality on information technology team projects.* Paper Presented at the ACM Sigmis Cpr Conference on Computer Personnel Research, Claremont, California, USA.

Ratnasingam, M. (2009). The contribution of teamwork, thinking styles, and innovation towards knowledge management. *Journal of Information and Communication Technology*, *9*, 29−39.

Shen, L., & Chen, S. (2013, January). Research of customer classification based on rough set using rosetta software. In *Proceedings of the 2012 International Conference on Communication, Electronics and Automation Engineering* (pp. 837−843). Springer Berlin Heidelberg.

Strömbergsson, H., Prusis, P., Midelfart, H., Lapinsh, M., Wikberg, J. E., & Komorowski, J. (2006). Rough set-based proteochemometrics modeling of G-protein-coupled receptorligand interactions. *PROTEINS: Structure, Function, and Bioinformatics*, *63*(1), 24−34.

Swiniarski, R. W., & Skowron, A. (2003). Rough set methods in feature selection and recognition. *Pattern Recognition Letters*, *24*(6), 833−849.

Wang, G. Y., Zheng, Z., & Zhang, Y. (2002). RIDAS-a rough set based intelligent data analysis system. In Proceedings of the 2002 International Conferences on IEEE. *Machine Learning and Cybernetics, 2*, 646−649.

Witlox, F., & Tindemans, H. (2004). The application of rough sets analysis in activity-based modelling. Opportunities and constraints. *Expert Systems with Application, 27*(4), 585−592.

Woehr, D., Arciniega, L., & Poling, T. (2013). Exploring the effects of value diversity on team effectiveness. *Journal of Business and Psychology, 28*(1), 107−121.

Wong, J.-T., & Chung, Y. S. (2007). Rough set approach for accident chains exploration. *Accident Analysis & Prevention, 39*(3), 629−637.