Suitable Personality Traits for Learning Programming Subjects: A Rough-Fuzzy Model

Abdul Rehman Gilal[1], Jafreezal Jaafar[2], Mazni Omar[3], Shuib Basri[4], Izzatdin Abdul Aziz[5],Qamar Uddin Khand[6], Mohd Hilmi Hasan[7]

[1, 2, 4, 5, 7] Department of Computer and Information Sciences, Universiti Teknologi PETRONAS, Malaysia
[1, 6] Department of Computer Science, Sukkur IBA University, Pakistan
[3]School of Computing, Universiti Utara Malaysia

***Abstract*—Programming is a cognitive activity which requires logical reasoning to code for abstract presentation. This study aims to find out the personality traits of students who maintain the effective grades in learning programming courses such as structured programming (SP) and object oriented programming (OOP) by gender classification. Data were collected from three universities to develop, validate, and generalize the Rough-Fuzzy model. Genetic and Johnson algorithms were applied under Rough set theory's (RST) principles to extract the decision rules. In addition, Standard Voting, Naïve Bayesian, and Object Tracking procedures were applied on the generated decision rules to find the prediction accuracy of each algorithm. Mamdani's Fuzzy Inference System (FIS) was used for mapping the decision rules' condition (input) to decision (output) based on fuzzy set theory (FST) to develop the model. The results highlighted that certain personality compositions can be suitable for scoring good grades in programming subjects. For instance, a female student is capable enough to improve the programming skills if she is composed of introvert and sensing personality traits. Therefore, it is important to investigate an appropriate personality composition for programming learners.**

Keywords—Software development; personality; programming; rough sets; fuzzy sets

I. Introduction

Learning to program has always been a hard activity for students. Compare to other courses, programming courses are usually difficult and often have the high dropout rates [1]. Nevertheless, programming skills let students to find a bright future too. In recent years, students' interest for learning programming languages has been increasing rapidly. But, everyone cannot perform well in programming. Shneiderman [2] maintains it that even a similar background of programmers cannot assure the similar performance. In the same vein, Brooks [3] also faced a huge variability in the achievements of introductory programming classes students.

Learning programming acquires cognitive and mental skills to design, code, and debug. Robins et al., [1] mention that writing a program includes various mental models. On the other hand, it is also proposed that mental models are formed by personalities and life experiences [4]. Personality is a complex natural phenomenon and one of the important human factors [5]. For personality, one group of psychology experts declares that personality is an inherited property which does not change but gets improved by time within the same personality type [6]. Whereas, other group of experts say contrary statement that personality gets changed with time and age between 20 to 40 is stable [7]. Moreover, several questions can take place if personality gets changed or gets betterment into it. For example, which personality types are suitable for learning programming subjects or which personality gets changed into betterment for learning programming? Researchers also believe that many factors influence the type of personality: culture and gender [8]. According to our understanding, personality is also influenced by other personality types. It has certain natural equations which form interpersonal and mental skills. It is highlighted because it was found that certain personality types are flexible to work with each other and some are not [9], [10]. Certain studies have been conducted in the past which proposed several methods and models for finding effective personnel for programming by focusing personality. But, different results have been observed when those models were practically implemented [11]–[13]. Furthermore, several ambiguities have also been found in the literature of personality in software domain. For instance, Gorla and Lam [11] proposed extrovert trait of personality for programmer, whereas Capretz and Ahmed [14] proposed introvert personality trait for the same role.

It is believed that workforce for software industry is always prepared by the education institutes. Keeping all in view, this study was performed on the student population to measure the personality behavior while learning programming subjects. In other words, in order to know that learning programming language is not a random behavior but it has a natural relationship with personality. Therefore, the main objective of the paper was set: "to find out the personality traits which maintain the effective grades in learning programming courses: SP and OOP, by gender classification". Moreover, the future of this research may contribute in following ways:

1) It may help the students to select the programming courses (i.e., SP or OOP) as their semester or major course based on their personality type.

2) It may also help the subject teachers to design their course and focus particular personality types' students since the beginning of courses for the better outcomes.

3) It leaves a new idea for research community who wish to contribute in this area of software development.

The next section of this paper presents the related work for foundation of the study. The section after related work, methodology section, discusses the methods used for data collection and experimenting. The Section IV discusses the results emanated from this study in detail. Additionally, Section V discloses the threats to validity which can be considered for future work. In the end, the paper is concluded in the Section VI.

II. Related Work

Personality has been researched in several fields of science. The following section is organized to show the importance of personality in the software development. This study has also set the gender as a mediating variable between personality and performance. Therefore, Section *B (i.e., Gender and Software Development)* presents the gender in software development. In the last section of related work, programmer role is discussed under the shades of personality.

A. Personality and Software Development

Personality refers to the internal psychological patterns such as feelings and thoughts which curve the behavior of a person. In simple words, personality traits are formed from internal forces. Numerous studies have been carried out in software domain which applied psychological frameworks, widely used in the domain of psychology, to understand the developer personality [15]. These theoretical frameworks include: 1) dispositional, 2) biological, 3) psychoanalytic, 4) neoanalytic, 5) learning, 6) phenomenological, and 7) cognitive self-regulation. Cruz et al., [15] also mentioned that the past research studies have not only used dispositional perspective abundantly so as to determine the personality traits and types in organizational psychology. But, they have also been used commonly in the field of software engineering to determine the most suited personalities to form ideal team for software development. Similarly, this study explores the key importance of personality perspective handy for learning programming courses.

Dispositional perspective of psychology that sheds light on trait and type approach depicts the fact that the personality deals with internal stable qualities that vary from individual to individual and it also influences behavior. American Psychiatric Association defined “trait” as “enduring patterns of perceiving, relating to, and thinking about the environment and oneself that are exhibited in a wide range of social and personal contexts.” Thus, the personalities of the people are determined by their personality patterns classified by psychological differences. Moreover, personality and trait can be distinguished as the former demonstrates different levels and degrees. Whilst, types are discreet, because they cannot be distinguished by levels and degrees [15].

There are some key theories pertinent to personalities that have been profusely implemented in psychological and computing research studies [16]. The most prominent among them are: Keirsey Temperament Sorter [17], Five- Factor Model (FFM) [18], also known as Big Five, and Myers-Briggs Type Indicator (MBTI) [19]. The distinctive point amongst these three personality theories is the way of the describing personality types. Keirsey Temperament Sorter accentuates on the long term behavior of the individuals [20]. Whereas, Five-Factor Model (FFM) encompasses five distinctive personality traits such as: conscientiousness, agreeableness, openness to experience, extraversion, and agreeableness. On the contrary, MBTI mainly probes into what people think. According to Furham [21] both MBTI and Big Five personality tests are helpful when a researcher aims to examine behavioral and cognitive sides of individuals by correlating both the scales. However, there are many proponents of MBTI in the domain of software engineering as this theory has been widely used in the past research studies [10], [22]–[27]. Thus, keeping in view the wide acceptance of MBTI in terms of its effectiveness, the current study has used this theory.

MBTI primarily focuses on four pairs of the personality which can be further classified into 16 types. The four pairs are: Extroversion-Introversion (I-E), Sensing-Intuitive (S-N), Thinking-Feeling (T-F), and Judging-Perceiving (J-P). These four dimensions also beget sixteen possible combinations of personality types as shown in the following Table 1:

TABLE I. The 16 MBTI Personality Types

ISTJ (1)	ISFJ (2)	INFJ (3)	INTJ (4)
ISTP (5)	ISFP (6)	INFP (7)	INTP (8)
ESTP (9)	ESFP (10)	ENFP (11)	ENTP (12)
ESTJ (13)	ESFJ (14)	ENFJ (15)	ENTJ (16)

Based on the performance and the score obtained, a person can be attributed with one of the 16 personality types cited in the above Table 1. For instance, a person scoring higher on Introversion (I) than Extroversion, Sensing (S) than Intuition (N); Thinking (T) than Feeling (F) and Judging (J) than Perceiving (P) would be categorized as an ISTJ.

B. Gender and Software Development

In social sciences, many research studies have explored personality and gender, either collectively or separately, to address the grave problems of teamwork in organizations and have achieved the acute success as well. However, this problem is still persistent in the field of software development since few researchers have ever tried to test personality and gender collectively to test the suitability of the team handy for software development. In this regard, Richards and Busch [28], Gilal et al., [10], and Rehman et al., [29] also assert that maturity level is yet to find in software development research. In the same vein, Trauth [30] also recommends that the need of improvement is required in the theoretical work on software development.

Study conducted by Gilal et al., [10] comes among the few studies which focused personality with gender. This study investigated the performance variation among software development team members caused by genders’ personality types. For instance, the male-dominated teams create reasons for females for being ineffective in teams if the personality type of female is with “E” trait. Furthermore, the study also revealed that the female-leader are more convenient with only female or majority-female (i.e., having female in majority) groups. Whereas, male-leaders are acceptable with all kind of team compositions. Critically, this study was just based on tabulated calculation and could not give any statistical or predictive evidences. However, the study also recommended some future research on gender with personality types to obtain appropriate conclusions. Moreover, Richards and Busch [28] study explored the gender and culture parameters to find their effects on the performance in IT workplaces. This study focused the knowing and doing gaps in software development workplaces. The researchers tried to find the effectiveness of diversity on the overall performance of the team. Moreover, authors acknowledged that these results are too weak to generalize that was one of the limitations of the study. But, these limitations can be overcome by inclusion of personality in the study. Because, it is also believed that inclusion of personality can help to achieve the efficiency, productivity, and quality [31].

C. *Software Programmer and Personality*

In the software development process, programmer has the key position for implementing the designs of system. The sensitivity of programmer's role lie in a fact that the programmer must be adept in syntax of the programming and good at analytical and logical sharpness for finding the code of the program with an ease. The lack of these qualities could make programmer to face the terrible failure. Because, coding phase has the crucial importance which is used to apply and identify data structures, control structure of the program and determines relevant variables [32]. Moreover, the past research studies have tried to empirically prove the relationship between personality and computer programming activities. Capretz [33] conducted experiments on Brazilian software engineering students to propose a personality profile for software developers. In his study, total 68 students participated and majority of them were male. Moreover, author concluded that ISFP, INTP, and ESTP personality types were significantly overrepresented among Brazilian software engineering students and, whereas, ENTP, ESTJ, and ENTJ personality types were significantly underrepresented among them. In the same vein, Martínez et al., [34] proposed a methodology for assigning roles to software developers. They divided the research experiments into two cases: training and testing with 12 and 16 participants respectively. The findings of the study revealed that ISTP personality type is best fitted for programmer role. Additionally, study conducted by Capretz and Ahmed [35] also highlighted the same objective in which software development tasks were contrasted with personality types. In their study, personality types were mapped with job requirements collected from newspaper, magazines, and online forums. At the end, authors recommended ISTJ and ISTP personality types for programmer role.

III. Methodology

This study presents the methodology section into two major subsections: Data collection and preprocessing and model development. Data collection and preprocessing section is all about the process of data collection: variables, algorithm, population and criteria of data collection for better understanding of the results of the study. Whereas, the second section highlights the whole process of the model from development to generalization.

A. *Data Collection and Preprocessing*

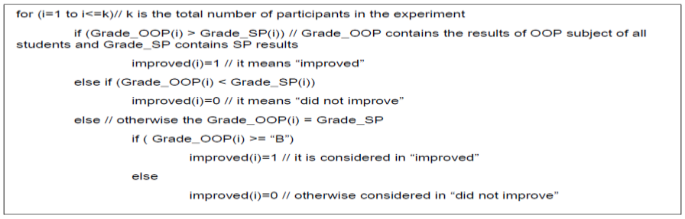
In order to achieve the objective of the study, data was collected from three universities: University Teknologi Petronas (UTP), Universiti Utara Malaysia (UUM), and Sukkur Institute of Business Administration (SIBA). Total size of the main dataset was 270, in which 110, 120, and 40 students participated from UTP, UUM and SIBA respectively. In the year of 2015, students who were learning software engineering subject during their bachelor from the universities participated voluntarily in the process of data collection. Software engineering class was chosen to collect data with the reason that in the all three universities, software engineering course is only offered after SP and OOP courses are already learnt by students. It is, because, to maintain the main objective of the study to see that which personality types maintain the effectiveness in learning programming languages in these two subjects. Moreover, in the all three universities, SP and OOP courses were of 4 credit hours per week. In these universities SP is the prerequisite course for OOP. Importantly, content and time duration (i.e., 16 weeks) of the courses were almost same in the all universities because these all three universities offer culture exchange program for international students. Therefore, they have to make a standard course contents and time durations.

MBTI instrument was used to measure the personality types of the participants. It stores the responses of personality in four pairs, as mentioned above, IE pair, SN pair, TF pair, and JP pair. Therefore, this study has 5 independent predictor variables (i.e., gender, IE, SN, TF, and JP) and 1 dependent outcome variable (i.e., improved; where this variable holds the final results whether or not the results of students in SP and OOP are improved). The following Table 2 shows the possible inputs which can be passed to study variables.

TABLE II. Controlling the Inputs to Variable

Variable	**Input**
Predictor	
1. Gender	1=Male 2=Female
2. IE	1=introvert 2=extrovert
3. SN	1=sensing 2=intuiting
4. TF	1=thinking 2=feeling
5. JP	1=judging 2=perceiving
Outcome	
1. Improved	0= did not improve 1= improved

The calculation of the outcome variable was made from the obtained marks of students in the SP and OOP subjects. For example, if a student obtained grade "B" in SP and grade "A" in OOP then it means the student improved the grades. Similarly, if the student obtained grade "B" in SP and grade "C" in OOP then it represents that student could not manage to improve the results. Another possibility could also occur that student neither "improved" nor "did not improve". In that situation, the input was adjusted in "improved" if the grades are still greater than "B" grade otherwise considered as "did not improve". It was applied, because, all these universities consider that grade "B" or Grade Point Average (GPA) 3.0 and above are good to excellent academic levels. Appendix 1 is highlighting the chart of grading scales in the universities. The following algorithm script defines the process of assigning values to outcome variable.



```
for (i=1 to i<=k)// k is the total number of participants in the experiment
          if (Grade_OOP(i) > Grade_SP(i)) // Grade_OOP contains the results of OOP subject of all
students and Grade_SP contains SP results
                    improved(i)=1 // it means "improved"
          else if (Grade_OOP(i) < Grade_SP(i))
                    improved(i)=0 // it means "did not improve"
          else // otherwise the Grade_OOP(i) = Grade_SP
                    if ( Grade_OOP(i) >= "B")
                              improved(i)=1 // it is considered in "improved"
                    else
                              improved(i)=0 // otherwise considered in "did not improve"
```

Fig. 1. Determining the values for outcome variable.

B. *Model Development*

To develop the model, the combination of RST and FST was used together to build an efficient model [36]. In this study, RST is used to extract the rules (IF-THEN) and, which were, then used as rules' database to fuzzy controller with Mamdani inference. The details from rules extraction to implementation to Mamdani inference are presented in the next sections. Basically, the model development phase was based on several steps:

a) Rules Generation

Model development was started with extracting the useful rules for decisions. In this study, the rules generation and validation were performed by using ROSETTA toolkit: analyzing toolkit for tabular data within RST framework [37]. This step of modeling also helped to remove the redundancy and unimportant data through reduction process. Moreover, Genetic Algorithm (GA) and Johnson's Algorithm (JA) were applied on data. Because, Hvidsten [38] says that GA is one of the effective solutions to searching problems. On another hand, Johnson [39] stated that JA invokes a variation of a simple greedy algorithm to compute a single reduct only. Therefore, both algorithms were applied to use the most effective one's rules for decision. The following experiment objectives were set for this step:

1) Find the personality traits of students who obtained effective (good) or low grades in SP and OOP courses (where term effective (good) refers to grade B and above or GPA 3.0 and above).

2) Find the personality traits of students who managed to improve grades in learning OOP (or managed to be consistent in the effective grades while learning SP and OOP).

It is important to note that these experiments were performed on each dataset to extract the effective rules separately. But, the rules extracted from UTP dataset were compared with UUM and SIBA datasets for results generalization purpose. It means that UTP dataset was used for model development and data from UUM and SIBA were used for validation and generalization.

b) Rules Evaluations and Generalization

The measurement of effectiveness of rules was computed through hold-out methods. For that purpose, this study used 70% of data for training and remaining 30% for testing the prediction accuracy only on UTP dataset (as mentioned above). It means that selection of algorithm results was based on the prediction accuracy. Hence, higher the prediction accuracy will increase the efficiency of model. Standard Voting, Naïve Bayesian, and Object Tracking procedures were applied on generated rules to find the prediction accuracy of each algorithm.

Moreover, for further validation and generalization, datasets from UUM and SIBA were equally distributed as the size of 30% of testing dataset of UTP. Basically, UUM data was used to see whether or not the personality preferences remain the same within Malaysian university students (i.e., UTP and UUM, during learning SP and OOP). Similarly, it was also validated to see the personality preferences behavior with Pakistani students. Basically, the performances of the model were measured in two ways: prediction accuracy and based on ROC, Area under Curve (AUC) results. Hence, 70% was considered as a benchmark for the effective prediction accuracy. Because, according to Bakar [40], the predication results can be known effective if the prediction accuracy is at least 70%. Similarly, Hvidsten [38] also mentioned that the 70% prediction accuracy is acceptable for prediction modeling. Moreover, Fawcett [41] asserted that the model is perfect if the obtained AUC is 1. On another hand, the model can be accepted if the computed AUC is 0.5 or above otherwise the generalization is rejected (if AUC<0.5). Therefore, the model results can be generalized if the prediction accuracy is at least 70% and AUC curve was computed 0.5 or above.

c) Fuzzy Inference System (FIS) Development

Once the rules were extracted and validated, the FIS; a system that maps the input to output based on fuzzy set theory, was used with Mamdani [42] inference system. The selection of Mamdani, instead of Sugeno inference system, was basically because of defuzzificaiton process. In the same vein, Govinderajan [43] also states that Mamdani is mostly used in pure fuzzy systems. Moreover, the Mamdani FIS system was developed by following its four basic parts [44] by using Matlab:

1) Fuzzifier: With the help of membership function, it helped to convert the crisp inputs into fuzzy inputs. Linear triangular, one of the mostly used membership functions [44], was used to define membership functions.

2) Rules: IF-THEN statements that were already defined from RST experiments.

3) Interface Engine: A part which converted the fuzzy input sets to fuzzy output from defined rules database.

4) Defuzzifier: With the help of membership function, the fuzzy outputs were converted to crisp output. Centroid or Center of area method (COA), a popular approach, was used to perform defuzzification.

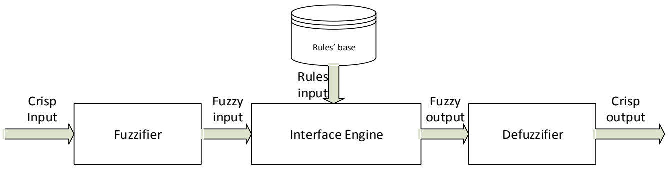


Fig. 2. Mamdani inference system.

Fig. 2 represents the general form of Mamdani inference system which independently acts for fuzzy controller development. On another hand, Fig. 3 is showing the overall process of rough and fuzzy approaches integration within the model development.

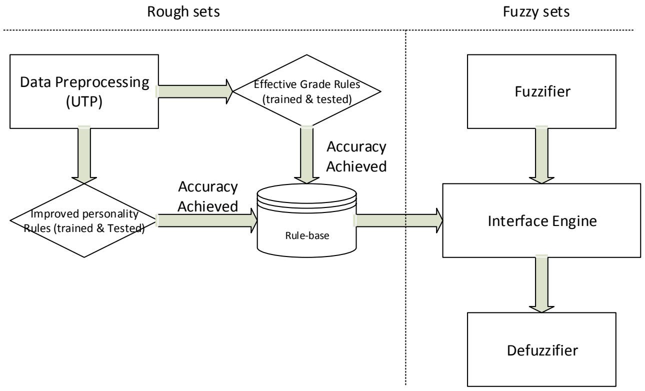


Fig. 3. Integration of rough and fuzzy approaches.

IV. Results and Discussion

The first step was to decide which algorithm (i.e., GA or JA) results were effective for model development. The obtained accuracy of rules was set as a benchmark (i.e., 70%) for algorithm selection. Table 3 summarizes the overall results of experiments on the both algorithms.

TABLE III. Extracted Reducts with Obtained Accuracy

	Objective:1 (good grades)		**Objective: 2 (improved grades?)**	
	GA	**JA**	**GA**	**JA**
Redcuts	16	14	18	15
Standard Voting	71.25%	74.24%	69.50%	72.72%
Object Tracking	70.59%	73.43%	67.76%	71.80%
Naïve Bayesian	65.34%	69.10%	68.54%	70.23%

JA algorithm was found suitable in both objectives. For example, in both cases, JA algorithm produced less reducts, than GA, which create a lesser complexity in the model. Moreover, JA algorithm maintained the effective accuracy in the all mentioned classifying techniques. Only in Naïve Bayesian, JA algorithm reducts could not obtained the said benchmark accuracy. But, overall, JA got the acceptable accuracy in both objectives. Therefore, this model used reducts of JA algorithm for finalizing the rule-base for fuzzy controller.

A. *Reducts for Finding Personality Traits who Obtained Good and Low Grades in Learning SP and OOP*

In order to maintain the main theme of this paper, it was first objective to see that what personality traits could earn good and low grades while learning SP and OOP. Each participant's results of SP and OOP programming were collected during experiment. In the first objective, gender, personality traits, and the results (of SP and OOP) were used to form the reducts without caring subject specification. Because, if a student managed to earn good grade in any subject it means that particular personality has capability to earn good grades. Therefore, it was even more precise discovery within dataset about effective personality traits. Table 4 shows the extracted reducts from JA algorithm on the experiments for finding good and low grades' personality traits.

TABLE IV. Reducts Obtained from JA Algorithm for the First Objective

No	**Reducts**	**LHS Sup port**	**RH S Sup port**	**RHS Accur acy**	**LHS Cove rage**	**RHS Cover age**
1	Female AND Extrovert AND Judging => Good-grade OR low-grade	45	34, 11	0.7555 56, 0.2444 44	0.29 2208	0.2905 98, 0.2972 97
2	Female AND Introvert AND Sensing => Good-grade OR low-grade	30	22, 8	0.7333 33, 0.2666 67	0.19 4805	0.1880 34, 0.2162 16
3	Female AND Thinking => Good-grade OR low-grade	34	26, 8	0.7647 06, 0.2352 94	0.22 0779	0.2222 22, 0.2162 16
4	Extrovert AND iNtuiting AND Judging => Good-grade OR low-grade	56	43, 13	0.7678 57, 0.2321 43	0.36 3636	0.3675 21, 0.3513 51
5	Extrovert AND Feeling AND Judging => Good-grade OR low-grade	52	38, 14	0.7307 69, 0.2692 31	0.33 7662	0.3247 86, 0.3783 78
6	Male AND Extrovert AND Sensing AND Thinking AND Judging => Good-grade	5	5	1	0.03 2468	0.0427 35
7	iNtuiting AND Thinking => Good-grade OR low-grade	32	23, 9	0.7187 5, 0.2812 5	0.20 7792	0.1965 81, 0.2432 43
8	Male AND Introvert AND Feeling => Good-grade	15	15	1	0.09 7403	0.1282 05
9	Thinking AND Perceiving => Good-grade OR low-grade	4	3, 1	0.75, 0.25	0.02 5974	0.0256 41, 0.0270 27
10	Introvert AND Thinking => Good-grade OR low-grade	27	22, 5	0.8148 15, 0.1851 85	0.17 5325	0.1880 34, 0.1351 35
11	Female AND Extrovert AND Sensing AND Perceiving => low-grade	2	2	1	0.01 2987	0.0540 54
12	Male AND Feeling AND Perceiving =>	10	10	1	0.06 4935	0.0854 7

	Good-grade					
13	iNtuiting AND Perceiving => Good-grade	8	8	1	0.051948	0.068376
14	Introvert AND iNtuiting AND Feeling => Good-grade	12	12	1	0.077922	0.102564

In Table 4, the term reducts (also called rules) refers to the extracted information of condition where "IF" a situation occurs "THEN" what decision should be taken. "Left Hand Side (LHS) support" informs that how many objects belong to "IF condition" and, whereas, "Right Hand Side (RHS) support" shows the number of decision objects on the "IF condition" within whole training dataset. Moreover, in this objective, the decision variable contains only two values: "Good-grade" and "low-grade", hence the RHS support has sometimes returned two numbers if the decision is divided. The rules which have two decision are called bi-dimension (i.e., rule no. 1). Otherwise, rules are called uni-dimension if it has only one decision (i.e., rule no. 6). Similarly, RHS accuracy is obtained to highlight the weight of decision within rule. It is simply obtained by dividing LHS support with the one value from RHS support. The accuracy of decision is always 1 if the rule has only one decision. In the same vein, LHS coverage shows the overall appearance of "IF condition" within the training dataset. Whereas, RHS coverage is almost similar to LHS coverage but it is only applied on decision part of reducts upon the class listed in the then part. For instance, the appearance of "low-grade" in the training set was 37 and, hence, the RHS coverage of rule no 1 for "low-grade" class is 0.297297.

First of all, total eight rules (i.e., Table 4 rule no. 1-5, 7, 9, and 10) out of fourteen were bi-dimension and, whereas, only remaining six rules were uni-dimension. Overall, most of bi-dimension rules seemed more towards "good-grade" if simply rely on the RHS support computation. Nevertheless, in this case, the high coverage of RHS support was not sufficient to decide the impact of these bi-dimension rules on one side of class when the dataset is not normally distributed. Therefore, the RHS coverage computation was necessary to consider defining the impacting class. More precisely, rule no. 1, 2, 5, and 7 were classified into low-grade class because their coverage was higher than good-grade class. On another hand, rule no. 3, 4, 9 and 10 were found more to good-grades class with the same reason. But, it should also be noted that rule no. 1, 3, and 4 did not show the higher difference between good and low grades. The following figures show the coverage of IF and THEN parts within graphs:

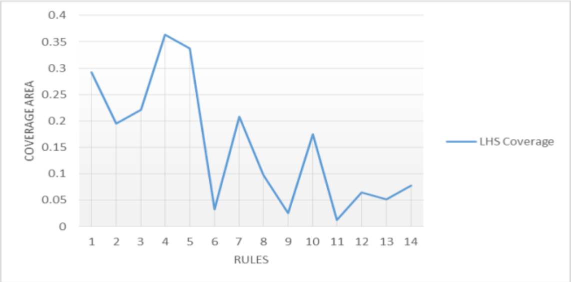


Fig. 4. "if-statement" coverage within training set for the first objective

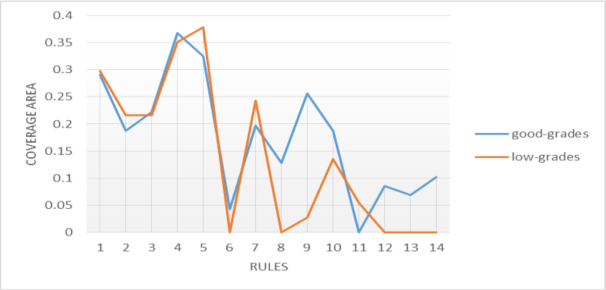


Fig. 5. "Then" part coverage grades within training set for the first objective

Moreover, total seven rules (i.e., Table 4 rule no. 1-3, 6, 8, 11, and 12) were found on gender classification. It shows that 50% of rules were gender-based decisions. More specifically, four rules: 1, 2, 3, and 11, were listed for female decision and in which rule 1, 2, and 3 were bi-dimension and rule number 11 was uni-dimension. Based on RHS coverage computations, rule 1 and 2 were listed for low grade decision. Therefore, it was extracted from rules that if a female learner is E with J personality traits or I with S or E with S and P then she may get low-grades in learning SP and OOP. On another, thinking (T) females were found effective in earning good-grades. This was also found in our previous study [25], [45] that T-trait females are effective for programming jobs. Furthermore, combinations of ESTJ personality traits were found progressive for male leaners. The ESTJ personality type is also appeared in the past studies [46] but the gender is missing. Hence, results from this study highlight that ESTJ can be progressive for male learners. In the same vein, I with F and F with P personality traits combinations were also found effective for male learners in SP and OOP classes. Lastly, N with P and I with N and F (rule no. 13 and 14) personality combinations were suitable for both genders for learning SP and OOP.

B. *Reducts for Finding Consistent Effective Personality Traits in Learning SP and OOP*

The section above underlined the personality traits of students that achieved good and low grades in learning SP and OOP subjects. On the other hand, this section was prepared to highlight those personality traits which maintained the good or low grades in the both subjects. For example, it does not happen always that the personality combination which managed the good grades in SP will manage the good grades in OOP too or other way around. Therefore, in order to see that whether the behavior of personality traits appeared identical in these both subjects or it has variations upon the subjects need. Table 5 summarizes the reducts extracted from the experiments for finding the answer on personality consistency.

The structure of the table is totally same as Table 4 but the description of decision class is different with "improve" and "didn't improve" outputs. Where "improve" denotes that personality combination appeared in IF statement (or LHS) of the rule was found effective or improved in learning SP and OOP. Whereas, "didn't improve" classify that the personality combination in the IF statement did not manage to improve the results in OOP.

TABLE V. REDUCTS OR FINDING CONSISTENCY OF PERSONALITY TRAITS IN LEARNING SP AND OOP

No	Reducts	LHS Support	RHS Support	RHS Accuracy	LHS Coverage	RHS Coverage
1	iNtuiting AND Judging => Improved OR didn't-improve	37	17, 20	0.459459, 0.540541	0.480519	0.395349, 0.588235
2	iNtuiting AND perceiving => Improved	3	3	1	0.038961	0.069767
3	Female AND Feeling AND Judging => Improved OR didn't-improve	22	12, 10	0.545455, 0.454545	0.285714	0.27907, 0.294118
4	Extrovert AND iNtuiting => didn't-improve OR Improved	28	16, 12	0.571429, 0.428571	0.363636	0.470588, 0.27907
5	iNtuiting AND Thinking => didn't-improve OR Improved	18	11, 7	0.611111, 0.388889	0.233766	0.323529, 0.162791
6	Female AND Introvert AND Sensing => Improved OR didn't-improve	13	8, 5	0.615385, 0.384615	0.168831	0.186047, 0.147059
7	Male AND Extrovert AND Thinking => didn't-improve OR Improved	10	5, 5	0.5, 0.5	0.12987	0.147059, 0.116279
8	Female AND Extrovert AND Sensing AND Thinking => Improved	2	2	1	0.025974	0.046512
9	Male AND Introvert AND Sensing AND Thinking => didn't-improve	1	1	1	0.012987	0.029412
10	Male AND Introvert AND Sensing AND Feeling => Improved	4	4	1	0.051948	0.093023
11	Thinking AND perceiving => Improved OR didn't-improve	2	1, 1	0.5, 0.5	0.025974	0.023256, 0.029412
12	Female AND Extrovert AND perceiving => Improved	1	1	1	0.012987	0.023256
13	Male AND Extrovert AND Judging => didn't-improve OR Improved	17	8, 9	0.470588, 0.529412	0.220779	0.235294, 0.209302
14	Male AND Introvert AND perceiving => Improved	4	4	1	0.051948	0.093023
15	Male AND Extrovert AND Feeling AND perceiving => didn't-improve	1	1	1	0.012987	0.029412

Table 5 comprised total eight bi-dimension rules (i.e., 1, 3-7, 11, and 13) and seven rules (i.e., 2, 8, 9, 10, 12, 14, and 15) as uni-dimension. From those bi-dimension rules, number 1, 4, 5 and 7 were classified to "didn't improve" class based on RHS support and coverage. It was found in these classified rules that combination of N trait with E or F or J (i.e., Table 5 rule no. 1, 4, and 5) trait does not guarantee the improvement in learning OOP. Similarly, male with E and T personality traits appeared inconsistent in the training set. On the other hand, rule no. 3, 6, and 13 were computed for "improved" class. Moreover, rule no 11 remained uncertain at this stage because it was computed almost same for both decision classes. But, it was considered in "didn't improve" class as it had very little higher coverage than "improved". The following Fig. 6 displays the RHS coverage against the rules.

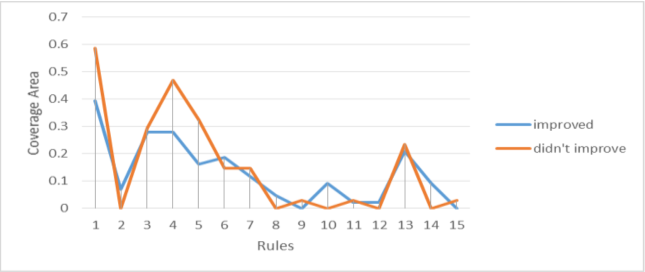


Fig. 6. RHS coverage for second objective from training set.

In this objective, gender was appeared highly impacting variable on the extracted rules. Total 10 rules (i.e., 3, 6-10, 12-15) were listed in gender classification; in which 4 rules (i.e., 3, 6, 8, and 12) belonged to female and remaining 6 (i.e., 7, 9, 10, 11, 13, 14, and 15) for male learners. Moreover, in the female rules, number 3 and 6 were bi-dimension and classified to "improved class". Whereas, two more rules: 8 and 12 were uni-dimension with "improved" class. Based on these rules, one can say that female gender can produce consistently effective results in learning SP and OOP if she is composed of feeling (F) and judging (J) or introvert (I) and sensing (S) or extrovert (E) and sensing (S) or extrovert (E) and perceiving (P). On another hand, number 7, 9 and 15 rules were straightly classified to "didn't improve" class in the male rules. In which, male with I, S, and T traits or E, F, and P traits or E with T traits' composition were found inconsistent in learning SP and OOP subjects. Moreover, a male was found consistent or improved when I with S and F traits or I with P traits or E with J traits (i.e., Table 5 rule no 10, 13 and 14). Furthermore, in the past studies [46], [47], E or S or J traits are found effective for programming job. But, to what extent they are consistent and for which gender they are more suitable or what other traits should be aligned for better results. Therefore, these results can bring some interesting and new information for users.

C. Comparison between both Objectives

In the previous sections, personality traits were extracted either based on low and good grades or consistency in improving grades in learning SP or OOP subjects. The first objective helped to extract the personality traits that can produce effective or ineffective results in learning SP and OOP. Whereas, second objective extracted the personality traits which were consistent to achieve improvement in learning SP and OOP or other way around. However, this section is designed to compare the both objectives and to see that whether the obtaining effective results and consistency are with reasons or it is random. Table 6 contains the rules of both objectives after deciding bi-dimensional rules in its suitable classes.

TABLE VI. COMPARISON BETWEEN 1^{ST} AND 2^{ND} OBJECTIVES

	1st Objective (personality traits based on low and good grades)	**Classified in**	**2nd objective (personality traits based on consistency in results)**	**Classified in**
1	Female AND Extrovert AND Judging	low-grade	iNtuiting AND Judging	didn't improve
2	Female AND Introvert AND Sensing	low-grade	iNtuiting AND perceiving	Improved
3	Female AND Thinking	good-grade	Female AND Feeling AND Judging	Improved
4	Extrovert AND iNtuiting AND Judging	good-grade	Extrovert AND iNtuiting	didn't improve
5	Extrovert AND Feeling AND Judging	low-grade	iNtuiting AND Thinking	didn't improve
6	Male AND Extrovert AND Sensing AND Thinking AND Judging	good-grade	Female AND Introvert AND Sensing	Improved
7	iNtuiting AND Thinking	low-grade	Male AND Extrovert AND Thinking	didn't improve
8	Male AND Introvert AND Feeling	good-grade	Female AND Extrovert AND Sensing AND Thinking	Improved
9	Thinking AND Perceiving	good-grade	Male AND Introvert AND Sensing AND Thinking	didn't improve
10	Introvert AND Thinking	good-grade	Male AND Introvert AND Sensing AND Feeling	Improved
11	Female AND Extrovert AND Sensing AND Perceiving	low-grade	Thinking AND perceiving	didn't improve
12	Male AND Feeling AND Perceiving	good-grade	Female AND Extrovert AND perceiving	Improved
13	iNtuiting AND Perceiving	good-grade	Male AND Extrovert AND Judging	Improved
14	Introvert AND iNtuiting AND Feeling	good-grade	Male AND Introvert AND perceiving	Improved
15			Male AND Extrovert AND Feeling AND perceiving	didn't improve

It was already mentioned above that total fourteen (14) rules were obtained from first objective and fifteen from second objective. In the first objective, five (5) rules were classified into “low grade” and nine (9) into “good grade” class. Whereas, on another side, seven (7) rules were classified into “didn’t improve” and eight (8) into “improved” class in the second objective. Moreover, based on the results, it was found that in the first objective rule no. 2 and 11 (i.e., Table 6, “Female AND Introvert AND Sensing” and “Female AND Extrovert AND Sensing AND Perceiving”) were listed in “low grade” class. But, both rules were appeared in the “improved” class in the second objective (see Table 6 rule no. 6 and 12 in the column of second objective). Similarly, rule no. 7 (iNtuitive AND Thinking) in the first objective was classified into “low grade” class and found into “didn’t improve” class in the second objective (rule no. 5). Whereas, rule no. 9 was in “good grade” class in the first objective but it was found in “didn’t improve” class in the second objective (i.e., rule no. 11). In the same way, rule no. 3 and 13 in the first objective (“Female AND Thinking” and “iNtuiting AND Perceiving”) were considered in the “good grade” class and found in the “improved” class of second objective (i.e., rule no. 8 and 2).

Finally, from three “good grade” personalities (i.e., rule no. 3, 9 and 13 in 1^{st} objective), two had capability to improve (i.e., rule no. 2 and 8 in 2^{nd} objective) and one did not improve (i.e., rule no. 11 in 2^{nd} objective). On another hand, two “low grade” personality (i.e., rule no. 2 and 11 in 1^{st} objective) combination were found improved (i.e., rule no. 6 and 12 in 2^{nd} objective) and one could not improve (i.e., rule no. 7 in 1^{st} objective and 5 in 2^{nd} objective) in learning. Therefore, based on the personality pairs found common in the both objectives, it could be summarized that there are some personality traits which can grade good in programming subjects and remain consistent or other way around.

D. Rules Generalization

UTP dataset was used to extract rules for making rule-base for FIS system. Rules extracted from it fulfilled the demand of efficiency benchmark [38], [40]. But, to what extent these results can be utilized for finding effective personality traits for programming learners. Data from UUM and Sukkur IBA were used to find the generalization within Malaysia and out of it. Table 7 summarizes the results extracted from these datasets.

TABLE VII. GENERALIZATION OF RESULTS BASED ON UUM AND SUKKUR IBA DATASETS

Objective1					
	sub-sets	Standard Voting	Object Tracking	Naïve Bayesian	ROC
UUM	uum1_ob1	69%	64%	65%	0.48
	uum2_ob1	73%	71%	69%	0.54
	uum3_ob1	70%	74%	71%	0.51
	uum4_ob1	76%	72%	71%	0.59
Sukkur IBA	iba1_ob1	54%	55%	51%	0.35
Objective 2					
	sub-sets	Standard Voting	Object Tracking	Naïve Bayesian	ROC
UUM	uum1_ob2	81%	76%	72%	0.65
	uum2_ob2	77%	79%	73%	0.63
Sukkur IBA	iba1_ob2	51%	55%	47%	0.41

It was clearly found that the results extracted from Malaysia cannot be generalized with Pakistan. In the first objective, one subset (i.e., uum_ob1) from UUM dataset was appeared slightly lower than the benchmark of accuracy but it showed the accepted benchmark on ROC curve. But, generally the rules accuracy was above 70%. Hence, it can be inferred that these results can be used within Malaysian universities. Whereas, for further expansion in the model, rules can be extracted from other countries data.

E. Fuzzy Inference System (FIS) Development

It is already mentioned in the methodology section that Mamdani inference system was used for fuzzification and defuzzification. For fuzzification process, the input variables were simply used with defined ranges in their membership functions. But, for output variable, the “low-grade” and “didn’t improve” classes were merged into “ineffective” membership function and, similarly, “good-grade” and “improved” classes were merged into “effective” membership function. Additionally, the output variable was ranged from 0 to 1 in which it was considered “ineffective” if the computed range is less than or equal to 0.5 and it was set “effective” if the range is greater than 0.5. Fig. 7 displays the control on output variable. During rule defining process, two rules (“iNtuitive AND Thinking” and “iNtuitive AND Perceiving”) were appeared twice with the same computation results because they were listed in the both objectives. Hence, they were kept once in the rule-base with double weight.

Moreover, in order to ensure the performance, the controller was used within Simulink. An array of values was passed to it by using “from workspace” block and the response of all returns was also captured in the workspace for further verifications. Therefore, the datasets used in training and testing were supplied from workspace to the controller without passing decision or outcome variable. It is because, at this stage, the accuracy of the controller was being measured rather than the performance of the model. However, it was expected to achieve the performance, at least, the same like obtained accuracy (i.e., at least 70%) as the datasets were same. Table 8 presents the confusion matrix based on the real dataset outcomes and obtained from controller.

TABLE VIII. CONFUSION MATRIX FOR CONTROLLER PERFORMANCE

	Predicted ineffective	Predicted effective	
Actual ineffective	29	12	41
Actual effective	13	56	69
	42	68	**0.7273**

Finally, as mentioned above, the obtained prediction accuracy was much similar as it was obtained in the classification. Moreover, the sensitivity of the predicted results was 0.81 and, whereas, computed specificity was 0.71. Therefore, based on the results obtained from these experiments, the developed model can be considered satisfied for future use.

V. THREANTS TO VALIDITY

Personality is a complex part of human factors which can be vague in shapes. It can be impacted from several internal and external factors: culture and language. Thus, the results from this study cannot be generalized other than Malaysian universities. In order to generalize it, the model can be expanded with multicultural data for more rules. Moreover, only two subjects (i.e., SP and OOP) learning was measured to develop the model. It restricts the results of the model for other programming and development subjects. The model can be enriched if it is trend with several other subjects: Database languages or web development languages. Furthermore, only MBTI based personality compositions are offered in the model. It can be one of the limitations of the model. Therefore, this model can include the new rules based on personality assessment other than MBTI: Big Five or Keirsey Temperament Sorter.

VI. CONCLUSION

Some personality combinations have capability to improve their programming learning skills. Other way around, some personality compositions are weak in learning and improving the programming skills. It was also found that in some cases each gender (i.e., male or female) demands different composition. For example, compositions “Female AND Introvert AND Sensing” or “Female AND Extrovert AND Sensing AND Perceiving” have capabilities to improve the programming skills. This study concludes that learning programming subjects has direct relation with certain personality compositions. Hence, it is very much important to investigate an appropriate personality composition for programming learners. Moreover, personality is complex in nature, extensively hidden results can be produced if complex networks approaches are applied. We aim to extend this study with complex networks approaches. Currently, weighted degree centrality, betweenness centrality, and closeness centrality techniques are proposed for future work.

REFERENCES

[1] Robins, J. Rountree, and N. Rountree, “Learning and teaching programming: A review and discussion,” Comput. Sci. Educ., vol. 13, no. 2, pp. 137–172, 2003.

[2] B. Shneiderman, “Software psychology: Human factors in computer and information systems (Winthrop computer systems series),” 1980.

[3] R. E. Brooks, “Studying programmer behavior experimentally: the problems of proper methodology,” Commun. ACM, vol. 23, no. 4, pp. 207–213, 1980.

[4] N. Halevy, T. R. Cohen, E. Y. Chou, J. J. Katz, and A. T. Panter, “Mental Models at Work Cognitive Causes and Consequences of Conflict in Organizations,” Personal. Soc. Psychol. Bull., p. 146167213506468, 2013.

[5] F. P. Brooks Jr, The Mythical Man-Month, Anniversary Edition: Essays on Software Engineering. Pearson Education, 1995.

[6] C. J. Boyce, A. M. Wood, and N. Powdthavee, “Is personality fixed? Personality changes as much as ‘variable’ economic factors and more strongly predicts changes to life satisfaction,” Soc. Indic. Res., vol. 111, no. 1, pp. 287–305, 2013.

[7] P. T. Costa Jr and R. R. McCrae, “Personality stability and its implications for clinical psychology,” Clin. Psychol. Rev., vol. 6, no. 5, pp. 407–423, 1986.

[8] J. S. Karn and a J. Cowling, “Using ethnographic methods to carry out human factors research in software engineering.,” Behav. Res. Methods, vol. 38, no. 3, pp. 495–503, Aug. 2006.

[9] R. Feldt, L. Angelis, R. Torkar, and M. Samuelsson, “Links between the personalities, views and attitudes of software engineers,” Inf. Softw. Technol., vol. 52, no. 6, pp. 611–624, 2010.

[10] A. R. Gilal, J. Jaafar, M. Omar, and M. Z. Tunio, “Impact of Personality and Gender Diversity on Software Development Teams’ Performance,”

in International Conference on Computer, Communication, and Control Technology (I4CT 2014), 2014, no. 2014 IEEE 2014, pp. 261–265.

[11] N. Gorla and Y. W. Lam, "Who should work with whom?: building effective software project teams," Commun. ACM, vol. 47, no. 6, pp. 79–82, 2004.

[12] A. R. Gilal, J. Jaafar, S. Basri, M. Omar, and A. Abro, "Impact of software team composition methodology on the personality preferences of Malaysian students," in 2016 3rd International Conference on Computer and Information Sciences (ICCOINS), 2016, pp. 454–458.

[13] A. R. Gilal, M. Omar, J. Jaafar, K. I. Sharif, A. W. Mahesar, and S. Basri, "Software Development Team Composition: Personality Types of Programmer and Complex Networks," in 6th International Conference on Computing and Informatics (ICOCI-2017), 2017, pp. 153–159.

[14] L. F. Capretz and F. Ahmed, "Making sense of software development and personality types," IT Prof., vol. 12, no. 1, pp. 6–13, 2010.

[15] S. Cruz, F. Q. B. da Silva, and L. F. Capretz, "Forty years of research on personality in software engineering: A mapping study," Comput. Human Behav., vol. 46, pp. 94–113, 2015.

[16] M. Omar, N. Katuk, S. L. S. Abdullah, N. L. Hashim, and R. Romli, "ASSESSING PERSONALITY TYPES PREFERENCES AMONGST SOFTWARE DEVELOPERS: A CASE OF MALAYSIA," ARPN J. Eng. Appl. Sci., vol. VOL. 10, N, no. FEBRUARY 2015, pp. 1499–1504, 2015.

[17] M. Bates and D. Keirsey, "Please Understand Me: Character and Temperament Types," Del Mar Prometh. Nemesis B. Co, 1984.

[18] R. R. McCrae and O. P. John, "An introduction to the five-factor model and its applications," J. Pers., vol. 60, no. 2, pp. 175–215, 1992.

[19] I. B. Myers, M. H. McCaulley, N. L. Quenk, and A. L. Hammer, MBTI manual: A guide to the development and use of the Myers-Briggs Type Indicator, vol. 3. Consulting Psychologists Press Palo Alto, CA, 1998.

[20] L. J. Francis, C. L. Craig, and M. Robbins, "The relationship between the Keirsey Temperament Sorter and the short-form revised Eysenck Personality Questionnaire," J. Individ. Differ., vol. 29, no. 2, pp. 116–120, 2008.

[21] A. Furnham, "The big five versus the big four: the relationship between the Myers-Briggs Type Indicator (MBTI) and NEO-PI five factor model of personality," Pers. Individ. Dif., vol. 21, no. 2, pp. 303–307, 1996.

[22] A. D. Da Cunha and D. Greathead, "Does personality matter?: an analysis of code-review ability," Commun. ACM, vol. 50, no. 5, pp. 109–112, 2007.

[23] A. R. Gilal, J. Jaafar, M. Omar, S. Basri, and A. Waqas, "A Rule-Based Model for Software Development Team Composition: Team Leader Role with Personality Types and Gender Classification," Inf. Softw. Technol., vol. 74, pp. 105–113, 2016.

[24] J. S. Karn, S. Syed-Abdullah, A. J. Cowling, and M. Holcombe, "A study into the effects of personality type and methodology on cohesion in software engineering teams," Behav. Inf. Technol., vol. 26, no. 2, pp. 99–111, 2007.

[25] A. R. Gilal, J. Jaafar, M. Omar, S. Basri, A. Aziz, and I. Din, "A Set of Rules for Constructing Gender-based Personality types ' Composition for Software Programmer," Lect. Notes Electr. Eng. by Springer, 2015.

[26] M. Omar and S.-L. Syed-Abdullah, "Identifying Effective S oftware Engineering (SE) Team Personality Types Composition using Rough Set Approach," in IEEE, 2010, pp. 1499–1503.

[27] A. R. Gilal, M. Omar, and K. I. Sharif, "A Rule-Based Approach For Discovering Effective Software Team Composition," J. Inf. Commun. Technol., vol. 13, pp. 1–20, 2014.

[28] D. Richards and P. Busch, "Knowing-doing gaps in ICT: gender and culture," J. Inf. Knowl. Manag. Syst., vol. 43, no. 3, pp. 264–295, 2013.

[29] M. Rehman, A. K. Mahmood, R. Salleh, and A. Amin, "Mapping job requirements of software engineers to Big Five Personality Traits," in Computer & Information Science (ICCIS), 2012 International Conference on, 2012, vol. 2, pp. 1115–1122.

[30] E. M. Trauth, "Theorizing gender and information technology research," Encycl. Gend. Inf. Technol., vol. 2, pp. 1154–1159, 2006.

[31] D. Varona and L. F. Capretz, "Evolution of Software Engineers ' Personality Profile," vol. 37, no. 1, pp. 2–6, 2012.

[32] S. Licorish, A. Philpott, and S. G. MacDonell, "Supporting agile team composition: A prototype tool for identifying personality (in) compatibilities," in Proceedings of the 2009 ICSE Workshop on Cooperative and Human Aspects on Software Engineering, 2009, pp. 66–73.

[33] L. F. Capretz, "Psychological Types of Brazilian Software Engineering Students," J. Psychol. Type, no. 5, pp. 37–42, 2008.

[34] L. G. Martínez, G. Licea, A. Rodríguez-Díaz, and J. R. Castro, "Experiences in Software Engineering Courses Using Psychometrics with RAMSET," in Proceedings of the Fifteenth Annual Conference on Innovation and Technology in Computer Science Education, 2010, pp. 244–248.

[35] L. F. Capretz and F. Ahmed, "Making Sense of Software Development and Personality Types," IT Prof., vol. 12, no. February, pp. 6–13, 2010.

[36] J. Krupka and P. Jirava, "Rough-fuzzy Classifier Modeling Using Data Repository Sets," Procedia Comput. Sci., vol. 35, pp. 701–709, 2014.

[37] A. Øhrn, J. Komorowski, and others, "Rosetta--a rough set toolkit for analysis of data," in Proc. Third International Joint Conference on Information Sciences, 1997.

[38] T. R. en Hvidsten, "Fault diagnosis in rotating machinery using rough set theory and ROSETTA," 1999.

[39] D. S. Johnson, "Approximation algorithms for combinatorial problems," J. Comput. Syst. Sci., vol. 9, no. 3, pp. 256–278, 1974.

[40] A. A. Bakar, Z. Kefli, S. Abdullah, and M. Sahani, "Predictive models for dengue outbreak using multiple rulebase classifiers," in Electrical Engineering and Informatics (ICEEI), 2011 International Conference on, 2011, pp. 1–6.

[41] T. Fawcett, "An introduction to ROC analysis," Pattern Recognit. Lett., vol. 27, no. 8, pp. 861–874, 2006.

[42] E. H. Mamdani, "Application of fuzzy algorithms for control of simple dynamic plant," Proc. Inst. Electr. Eng., vol. 121, no. 12, pp. 1585–1588, 1974.

[43] A. Govindarajan, "A soft computing framework to evaluate the efficacy of software project management," in Intelligent Systems and Control (ISCO), 2015 IEEE 9th International Conference on, 2015, pp. 1–6.

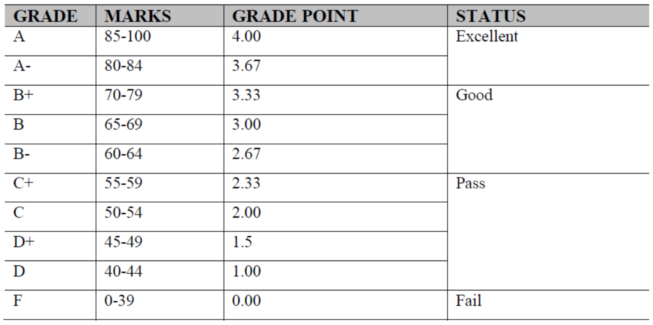
[44] A. Amindoust, S. Ahmed, A. Saghafinia, and A. Bahreininejad, "Sustainable supplier selection: A ranking model based on fuzzy inference system," Appl. Soft Comput., vol. 12, no. 6, pp. 1668–1677, 2012.

[45] A. R. Gilal, J. Jaafar, M. Omar, S. Basri, and I. Din, "Balancing the Personality of Programmer: Software Development Team Composition.," Malaysian J. Comput. Sci., vol. 29, no. 2, pp. 145–155, 2016.

[46] D. Varona, L. F. Capretz, and Y. Piñero, "Personality types of Cuban software developers," Glob. J. Eng. Educ., vol. 13, no. 2, pp. 77–81, 2011.

[47] G. Mourmant and M. Gallivan, "How Personality Type Influences Decision Paths in the Unfolding Model of Voluntary Job Turnover: An Application to IS Professionals," Business, pp. 134–143, 2007.

APPENDIX 1



GRADE	MARKS	GRADE POINT	STATUS
A	85-100	4.00	Excellent
A-	80-84	3.67	
B+	70-79	3.33	Good
B	65-69	3.00	
B-	60-64	2.67	
C+	55-59	2.33	Pass
C	50-54	2.00	
D+	45-49	1.5	
D	40-44	1.00	
F	0-39	0.00	Fail