2012 10th International Conference on Frontiers of Information Technology

# Identifying Similar Software Datasets Through Fuzzy Inference System

Saba Anwar, Zeeshan A. Rana, Mian M. Awais
*Department of Computer Science,*
*School of Science and Engineering (SSE),*
*LUMS, DHA, Lahore 54792, Pakistan.*
{*10030040, zeeshanr, awais*}*@lums.edu.com*

***Abstract*—Similar software have similar software measurements. Defect data from one software can be used to anticipate defects in a similar software. Although, not many defect datasets are made public in software engineering domain, PROMISE repository is a reasonable collection of software data. This paper presents a two step approach to identify similar software and applies the proposed technique to find similar datasets in PROMISE repository. As step 1, the approach generates associations rules for each dataset to determine dataset's behavior in terms of frequent patterns. As step 2, overlap between the association rules is calculated using Fuzzy Inference Systems (FIS). The FIS generated for the study have been expert-based as well as auto-generated. Similarity between 28 dataset pairs has been found KC2 and PC1 turned out to be most similar datasets with 86% similarity using Mamdani, 92% with Sugeno models. Results from expert-based and autogenerated FIS have been comparable.**

## I. Introduction

Current software projects can benefit from already developed similar projects in number of aspects. This benefit can be in terms of co-occurrences of certain values of software measures and software defects. This defect information can be helpful in avoiding similar defects through better resource planning and thorough testing.

A company may store and interpret its past experiences to avoid future defects but similar projects across different companies cannot benefit unless there is a mechanism to represent the software in a common manner. A software can be encoded as a dataset by recording module wise software measurements and defect data. These datasets can then be compared to identify similar software. The software measurements can have similar values and may exhibit similar defect patterns. Similarity between current and past datasets can be exploited to reduce likelihood of defects in current projects.

Various approaches have been employed to find similar datasets including clustering [1], substructure in condensed models of datasets [2], and analogy based methods [3]. Further, association rules based techniques have also been used for the purpose [4], [5], [6], [7]. Parthasarathy et al. [4] have found similarity between homogeneous datasets through support count and a parameter $\alpha$ to reflect variations in support count. Li et al. [5] have proposed an associations based measure to find similarity between basket datasets to mine distributed datasets. Dudek et al. [6] have used association rules to measure uniformity of source datasets applied for qualitative evaluation of incremental association rule mining methods, testing uniformity of a dataset, and tracking stability of a sequential data source. Rana et al. [7] have found similarity between software product datasets by comparing association rules obtained from each dataset. Software process measures and fuzzy logic have also been employed to find software project similarity where numerical data was not available. Other studies [8], [9] and [10] have proposed measures to evaluate similarity between two software projects by using linguistic values. Barreto [11] have found software project similarity by identifying characteristics from software process to improve software project monitoring process.

The above mentioned approaches have ascertained similar software with numeric and non-numeric data. There are more public datasets with numeric data as compared to non-numeric datasets [12]. Learning defect patterns from the available public datasets is a useful approach. To this end this paper proposes a association mining and fuzzy rules based approach to identify similar datasets.

This paper suggests the generation of association rules for two datasets and then develop Fuzzy Inference System (FIS) to find similarity between the rulesets. The basic principal of the proposed method is that if two datasets have similar association rules they have similar behavior hence the analogous defect occurrence patterns. Association rules capture the frequent patterns in the data. If associations rules of two data sets are similar it means that they both have similar patterns of attributes and hence they are same. Association rules are generated using Apriori algorithm [13] followed by two types of fuzzy rule generation. Fuzzy rules have been generated by experts as well as through automatic rule generation methods. Mamdani [14] and Sugeno [15] based inference systems have been developed for expert based rules. The auto generated FIS has used Sugeno inference only. The fuzzy models allow the comparison between the datasets with different number of attributes. Unlike other association mining approaches this approach can find similarity between the datasets that do not use same software attributes.

Rest of the paper is organized as follows. Section II

978-0-7695-4927-9/12 $26.00 © 2012 IEEE
DOI 10.1109/FIT.2012.40

Table I
LIST OF DATASETS AND THEIR CHARACTERISTICS

| | CM1 | KC1 | KC2 | PC1 | MW1 | PC4 | MC2 | KC3 |
|---|---|---|---|---|---|---|---|---|
| No. of Modules | 498 | 2109 | 522 | 1109 | 403 | 1458 | 161 | 458 |
| No. of Attributes | 22 | 22 | 22 | 22 | 38 | 38 | 40 | 40 |
| Percentage of Defective Modules | 9.84% | 15.46% | 20.50% | 6.95% | 7.70% | 12.21% | 32.30% | 9.39% |
| Language | C | C++ | C++ | C | C++ | C | C++ | Java |

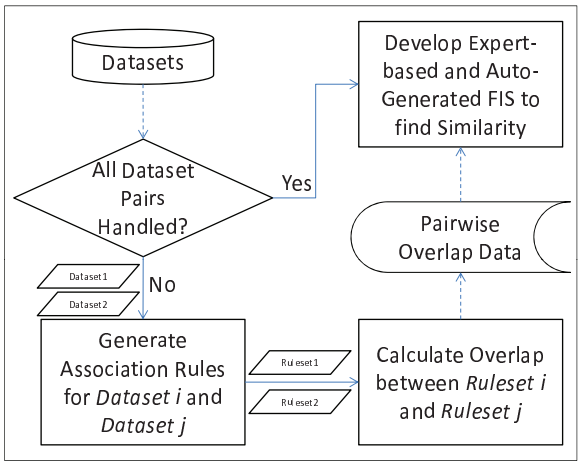


Figure 1. Our Methodology

presents the proposed methodology, section III discusses results, section IV gives an analysis of the results and section V concludes the paper and highlights future directions..

## II. METHODOLOGY

This section describes the steps performed to identify similar datasets. Attributes of the 8 datasets used in this study are measurements collected from software code through static analysis. Summarized information about the datasets is given in Table I and details about the dataset attributes can be found elsewhere [7], [12]. Using the 8 datasets, this study has prepared 28 dataset pairs based on the attributes used, language of the software, size of the dataset and the number of attributes. After *Dataset Selection* the next step is to *Find Similar Association Rules*. After normalizing the attributes to same ranges, Apriori algorithm [13] has been used to generate association rules for each dataset. *Overlap* between each pair of rulesets is calculated and the overlap value is fuzzified to develop fuzzy rules and inference systems. The fuzzy rules have been generated in two ways: through expert knowledge and through auto-rule generation method. As a next step, different *Fuzzy Inference System (FIS)* have been developed to infer about similarity of corresponding dataset pairs. Figure 1 depicts flow of proposed approach. Dotted lines in the figure represent the flow of data.

### A. Finding Similar Association Rules

The attributes of the datasets used are not categorical in nature whereas Apriori algorithm requires categorical attributes. Therefore all attributes have been discretized. Further all attributes havebeen normalized between 0 and 1 to make the association rules comparable. Ranges of the attributes have been replaced with linguistic labels using the approach taken by Anwar et al. [7]. After replacing the ranges with labels, next step is to find similar rules and extent of their similarity.

### B. Calculating Rules Overlap

Two rules can be either completely similar, partially similar or completely dissimilar. The completely similar rules have same antecedents and same consequent and are assigned $Overlap = 1$. Overcoming the limitation of existing approaches [7], this method also calculates partial similarity between rules. As a first precondition to call two rules partially similar, their consequents must be same. Partial similarity between arbitrary rules $rule_i$ and $rule_j$, which satisfy the precondition, is calculated as follows:

$$Overlap(rule_i, rule_j) = \frac{CommonAttributes(rule_i, rule_j)}{TotalAtributes(rule_i, rule_j)} \quad (1)$$

where $CommonAttributes = CountofCommonAntecedents+1$ and $TotalAtributes = NumberofUniqueAntecedents + 1$. The value 1 is added in both the expressions to include the effect of consequent. Completely dissimilar rules have no common antecedents and overlapping extent of the completely dissimilar rules is 0. Each rule from one ruleset is compared with all rules from the second ruleset. Rule with maximum similarity are selected for further processing.

### C. Developing Fuzzy Inference Systems

The overlap values for the similar association rules are used as input to the FIS. Since it is possible to have different number of overlapping rules for different datasets, it can be difficult to develop FIS with variable number of inputs. In order to keep the number of FIS inputs constant, the proposed approach aggregates all the rule overlaps into 10 overlaps. Each of the 10 overlaps is an average overlap value of the aggregated overlaps. Expert-based and auto-generated FIS have been generated using these aggregated overlap values.

Table II
SAMPLE FUZZY RULES

| |
|---|
| If (Overlap1 is HIGH) and (Overlap2 is HIGH) and (Overlap3 is HIGH) and (Overlap4 is LOW) and (Overlap5 is LOW) and (Overlap6 is LOW) and (Overlap7 is LOW) and (Overlap8 is LOW) and (Overlap9 is LOW) and (Overlap10 is LOW) then (Similarity is LOW) (1) |
| If (Overlap1 is HIGH) and (Overlap2 is HIGH) and (Overlap3 is MEDIUM) and (Overlap4 is MEDIUM) and (Overlap5 is MEDIUM) and (Overlap6 is MEDIUM) and (Overlap7 is LOW) and (Overlap8 is LOW) and (Overlap9 is LOW) and (Overlap10 is LOW) then (Similarity is LOW) (1) |
| If (Overlap1 is HIGH) and (Overlap2 is HIGH) and (Overlap3 is HIGH) and (Overlap4 is HIGH) and (Overlap5 is MEDIUM) and (Overlap6 is MEDIUM) and (Overlap7 is LOW) and (Overlap8 is LOW) and (Overlap9 is LOW) and (Overlap10 is LOW) then (Similarity is MEDIUM) (1) |
| If (Overlap1 is HIGH) and (Overlap2 is HIGH) and (Overlap3 is HIGH) and (Overlap4 is HIGH) and (Overlap5 is HIGH) and (Overlap6 is MEDIUM) and (Overlap7 is MEDIUM) and (Overlap8 is MEDIUM) and (Overlap9 is MEDIUM) and (Overlap10 is LOW) then (Similarity is MEDIUM) (0.85) |
| If (Overlap1 is HIGH) and (Overlap2 is HIGH) and (Overlap3 is HIGH) and (Overlap4 is HIGH) and (Overlap5 is HIGH) and (Overlap6 is HIGH) and (Overlap7 is LOW) and (Overlap8 is LOW) and (Overlap9 is LOW) and (Overlap10 is LOW) then (Similarity is MEDIUM) (0.75) |
| If (Overlap1 is HIGH) and (Overlap2 is HIGH) and (Overlap3 is HIGH) and (Overlap4 is HIGH) and (Overlap5 is HIGH) and (Overlap6 is HIGH) and (Overlap7 is HIGH) and (Overlap8 is MEDIUM) and (Overlap9 is MEDIUM) and (Overlap10 is MEDIUM) then (Similarity is HIGH) (0.85) |
| If (Overlap1 is HIGH) and (Overlap2 is HIGH) and (Overlap3 is HIGH) and (Overlap4 is HIGH) and (Overlap5 is HIGH) and (Overlap6 is HIGH) and (Overlap7 is HIGH) and (Overlap8 is HIGH) and (Overlap9 is MEDIUM) and (Overlap10 is LOW) then (Similarity is HIGH) (1) |

*1) Expert-based :* The 10 overlaps have been assigned three linguistic labels each: LOW, MEDIUM and HIGH and rules have been developed. To assign the linguistic labels to the inputs, three membership functions have been designed for each input. After the assignment, fuzzy rules have been generated as shown in Table II. These rules have been used to develop Mamdani and Sugeno-based inference systems. Difference between these two systems is the type of output function. The output function for a Mamdani FIS is also fuzzy whereas the output function $y$ for Sugeno FIS can be linear as well as constant. Both the models have been developed for three different types of membership functions namely triangular, trapezoidal and gaussian.

*2) Auto-Generated:* It is not always possible to have an expert available for rules development. Therefore, this paper demonstrate the application of auto-generated rules to find dataset similarity. To generate the automatic rules subtractive clustering [16] has been used to determine number of input membership functions and the number of rules where the resultant number of clusters corresponds to these two parameters. Subtractive clustering uses the data points as a first estimate to cluster centers. The data points at a distance more than $\alpha$ are considered part of other clusters. Keeping all the memberships functions as Gaussian, Sugeno FIS has been generated.

## III. RESULTS

Association rulesets have been generated for all the datasets and then overlap between the rulesets has been calculated. In order to perform pairwise comparison of the eight datasets, 28 pairs have been generated. From these 28 pairs, approximately 19 (70%) pairs have been used as training data to build FIS and the rest of the pairs have been used to validate the developed FIS. Rest of the section presents each step in detail.

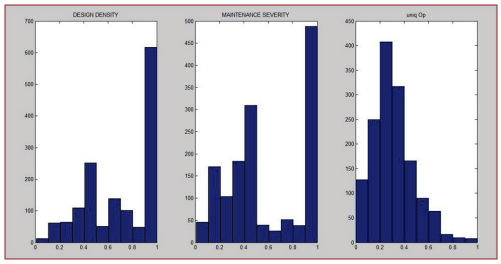


Figure 2. Frequency Distribution of 3 Attributes from KC1

### A. Finding Similar Association Rules

As a preprocessing step to finding association rules attributes of the datasets have been discretized and normalized to range [0-1]. The the association rules for each dataset have been generated using WEKA [**?**]. For each dataset 100 strongest rules have been selected and kept as a ruleset. These rules have been assigned linguistic labels LOW, MEDIUM and HIGH. These labels have been assigned by observing distribution of the attribute. Sample distribution of three attributes from KC1 dataset are given in Figure 2. For the attributes mentioned in the figure labels can be LOW = 0 - 0.3, MEDIUM = 0.25 - 0.65, HIGH = 0.6 - 1. These are essentially the parameters for fuzzy membership functions for the three attributes. After this labeling step similar rules look like the sample rules shown in Table III

### B. Calculating Rules Overlap

Rule overlap is calculated for each pair of rulesets using the expression 1. Examples of best matching rule and similarity calculation is given in TableIV. Rule 1

Table V
AVERAGE OVERLAP VALUES FOR DATASETS

| DataSetPair | Overlap1 | Overlap2 | Overlap3 | Overlap4 | Overlap5 | Overlap6 | Overlap7 | Overlap8 | Overlap9 | Overlap10 |
|---|---|---|---|---|---|---|---|---|---|---|
| CM1-KC1 | 1.00 | 1.00 | 1.00 | 0.80 | 0.75 | 0.75 | 0.67 | 0.65 | 0.45 | 0.14 |
| CM1-KC2 | 1.00 | 1.00 | 0.93 | 0.75 | 0.72 | 0.65 | 0.49 | 0.28 | 0.00 | 0.00 |
| CM1-KC3 | 1.00 | 1.00 | 0.88 | 0.75 | 0.75 | 0.69 | 0.67 | 0.46 | 0.08 | 0.00 |
| CM1-PC1 | 1.00 | 1.00 | 1.00 | 0.85 | 0.75 | 0.75 | 0.67 | 0.48 | 0.00 | 0.00 |
| CM1-PC4 | 1.00 | 0.80 | 0.68 | 0.67 | 0.54 | 0.50 | 0.27 | 0.00 | 0.00 | 0.00 |
| CM1-MC2 | 1.00 | 1.00 | 0.88 | 0.75 | 0.73 | 0.67 | 0.55 | 0.28 | 0.00 | 0.00 |
| CM1-MW1 | 1.00 | 1.00 | 0.83 | 0.75 | 0.73 | 0.67 | 0.60 | 0.43 | 0.08 | 0.00 |
| KC1-KC2 | 1.00 | 1.00 | 0.93 | 0.67 | 0.45 | 0.18 | 0.00 | 0.00 | 0.00 | 0.00 |
| KC1-KC3 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 0.90 | 0.67 | 0.30 | 0.00 | 0.00 |
| KC1-PC1 | 1.00 | 1.00 | 1.00 | 0.78 | 0.63 | 0.26 | 0.00 | 0.00 | 0.00 | 0.00 |
| KC1-PC4 | 1.00 | 0.95 | 0.72 | 0.67 | 0.43 | 0.12 | 0.00 | 0.00 | 0.00 | 0.00 |
| KC1-MC2 | 1.00 | 1.00 | 1.00 | 0.83 | 0.67 | 0.44 | 0.08 | 0.00 | 0.00 | 0.00 |
| KC1-MW1 | 1.00 | 1.00 | 1.00 | 0.93 | 0.70 | 0.63 | 0.41 | 0.00 | 0.00 | 0.00 |
| KC2-KC3 | 1.00 | 1.00 | 0.79 | 0.75 | 0.73 | 0.67 | 0.58 | 0.36 | 0.15 | 0.00 |
| KC2-PC1 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 0.81 | 0.38 |
| KC2-PC4 | 1.00 | 0.81 | 0.75 | 0.67 | 0.56 | 0.48 | 0.33 | 0.08 | 0.00 | 0.00 |
| KC2-MC2 | 1.00 | 1.00 | 0.90 | 0.78 | 0.75 | 0.73 | 0.67 | 0.55 | 0.17 | 0.00 |
| KC2-MW1 | 1.00 | 1.00 | 0.82 | 0.75 | 0.75 | 0.67 | 0.61 | 0.41 | 0.17 | 0.00 |
| KC3-PC1 | 1.00 | 1.00 | 0.79 | 0.56 | 0.13 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| KC3-PC4 | 1.00 | 1.00 | 0.75 | 0.67 | 0.53 | 0.15 | 0.00 | 0.00 | 0.00 | 0.00 |
| KC3-MC2 | 1.00 | 1.00 | 1.00 | 0.88 | 0.67 | 0.42 | 0.12 | 0.00 | 0.00 | 0.00 |
| KC3-MW1 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 0.88 | 0.71 | 0.63 | 0.19 | 0.00 |
| PC1-PC4 | 1.00 | 0.83 | 0.75 | 0.68 | 0.65 | 0.54 | 0.40 | 0.00 | 0.00 | 0.00 |
| PC1-MC2 | 1.00 | 1.00 | 1.00 | 0.82 | 0.76 | 0.75 | 0.67 | 0.61 | 0.00 | 0.00 |
| PC1-MW1 | 1.00 | 1.00 | 0.90 | 0.77 | 0.75 | 0.71 | 0.67 | 0.51 | 0.00 | 0.00 |
| PC4-MC2 | 1.00 | 1.00 | 0.79 | 0.75 | 0.63 | 0.10 | 0.00 | 0.00 | 0.00 | 0.00 |
| PC4-MW1 | 1.00 | 1.00 | 0.79 | 0.75 | 0.63 | 0.10 | 0.00 | 0.00 | 0.00 | 0.00 |
| MC2-MW1 | 1.00 | 1.00 | 1.00 | 0.92 | 0.67 | 0.65 | 0.45 | 0.21 | 0.00 | 0.00 |

Table III
AN EXAMPLE OF LABELED ASSOCIATION RULES

| |
|---|
| ev(g)=L, iv(g)=L==> e=L, t=L |
| ev(g)=L, iv(g)=L ==> e=L, t=L |

Table IV
SAMPLE OVERLAPPING RULES

| |
|---|
| 1. e=L ==> t=L |
| 2. e=L ==> t=L |
| 3. b=L ==> e=L |
| 4. locCodeAndComment=L, t=L ==> e=L |
| 5. t=L, locCodeAndComment=L ==> e=L |
| 6. b = L , v = L ==> e = L |
| 7. ev(g)=L, iv(g)=L ==> e=L, t=L |

and 2 are exactly similar and their $Overlap = 1$. Rule 3 has the same consequent as Rules 4, 5 and 6 have i.e. Rule 3 has partial similarity with these rules. However, Rule 3 and Rule 6 have the highest overlap where the total number of attributes are 3 and 2 of them are same. Hence, $Overlap(Rule_3, Rule_6) = (2/3) = 0.67$. Whereas $Overlap(Rule_3, Rule_4) = Overlap(Rule_3, Rule_5) = (1/4) = 0.25$. Rule 7 is not similar to any rule because there is no rule with the same consequent so its overlap with all the rules is 0. It is important to note that $Overlap(Rule_4, Rule_5) = 1$ despite of the different arrangement of attributes in antecedent part. By the end of this step 100 overlap values have been calculated for each dataset pair.

### C. Fuzzy Inference System

Instead of using the 100 overlap values as input to generate FIS, the overlap values have been aggregated. This preprocessing step reduces the FIS input space by averaging 10 overlap values for a pair and representing them as one value. The 10 overlap values for each pair shown in Table V are averages of best matching 100 rules. Table V is input to our FIS and the output is the extent of similarity.

*1) Expert-based :* Based on the frequency distribution of average overlap values, membership have been defined for each pair and the rules (shown in Table II) have been generated through Matlab Fuzzy Toolbox. As an example, frequency distribution for $Overlap1$, $Overlap3$, and $Overlap10$ and their respective membership functions have been given in Figure 3. Same number of functions have been used to develop rules for trapezoidal and Gaussian membership functions.

After development of fuzzy rules the output is inferred using Mamdani and Sugeno methods. All the rules were conjunctive rules therefore t-norm operator [17] has been applied to calculate the final output for both types of FIS. In Mamdani FIS output was defuzzified using centroid method and output was aggregated. Table VI reports the results for

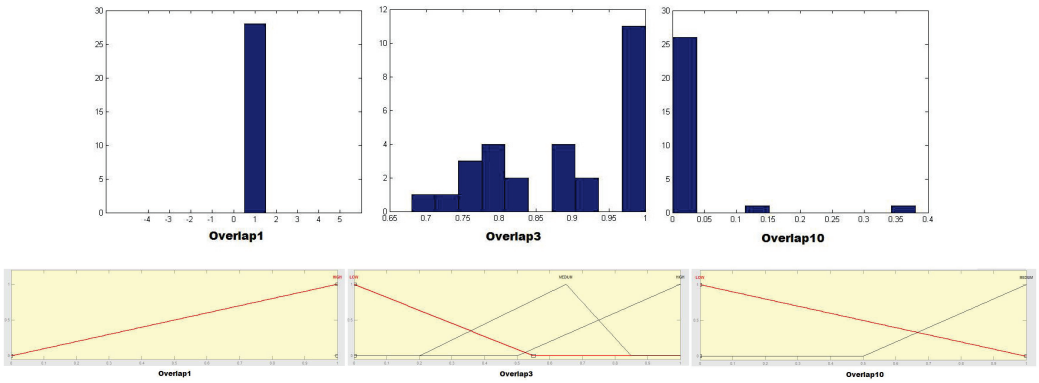


Figure 3. Frequency distribution and corresponding membership functions for three input variables

Table VI
SIMILARITY BETWEEN DATASETS CALCULATED BY EXPERT-BASED FIS

| DataSetPairs | Mam-Triangular | Mam-Trapezoidal | Mam-Gaussian | Sug-Triangular | Sug-Trapezoidal | Sug-Gaussian |
|---|---|---|---|---|---|---|
| CM1-KC1 | 0.656 | 0.661 | 0.620 | 0.702 | 0.691 | 0.680 |
| CM1-KC2 | 0.603 | 0.562 | 0.579 | 0.602 | 0.560 | 0.587 |
| CM1-KC3 | 0.579 | 0.558 | 0.555 | 0.581 | 0.558 | 0.557 |
| CM1-PC1 | 0.655 | 0.660 | 0.621 | 0.686 | 0.690 | 0.649 |
| CM1-PC4 | 0.352 | 0.343 | 0.376 | 0.322 | 0.312 | 0.381 |
| CM1-MC2 | 0.621 | 0.601 | 0.578 | 0.621 | 0.611 | 0.598 |
| CM1-MW1 | 0.532 | 0.560 | 0.518 | 0.541 | 0.561 | 0.530 |
| KC1-KC2 | 0.500 | 0.500 | 0.501 | 0.500 | 0.500 | 0.474 |
| KC1-KC3 | 0.828 | 0.841 | 0.756 | 0.870 | 0.882 | 0.813 |
| KC1-PC1 | 0.544 | 0.541 | 0.505 | 0.533 | 0.530 | 0.501 |
| KC1-PC4 | 0.230 | 0.218 | 0.427 | 0.163 | 0.149 | 0.349 |
| KC1-MC2 | 0.540 | 0.537 | 0.505 | 0.533 | 0.530 | 0.501 |
| KC1-MW1 | 0.558 | 0.536 | 0.559 | 0.555 | 0.530 | 0.561 |
| KC2-KC3 | 0.452 | 0.441 | 0.470 | 0.503 | 0.449 | 0.514 |
| KC2-PC1 | 0.849 | 0.861 | 0.832 | 0.870 | 0.882 | 0.851 |
| KC2-PC4 | 0.360 | 0.351 | 0.411 | 0.322 | 0.312 | 0.417 |
| KC2-MC2 | 0.609 | 0.598 | 0.590 | 0.629 | 0.594 | 0.591 |
| KC2-MW1 | 0.510 | 0.526 | 0.509 | 0.523 | 0.530 | 0.526 |
| KC3-PC1 | 0.212 | 0.199 | 0.290 | 0.163 | 0.149 | 0.160 |
| KC3-PC4 | 0.500 | 0.500 | 0.490 | 0.500 | 0.500 | 0.447 |
| KC3-MC2 | 0.539 | 0.536 | 0.505 | 0.533 | 0.530 | 0.499 |
| KC3-MW1 | 0.843 | 0.856 | 0.784 | 0.870 | 0.882 | 0.830 |
| PC1-PC4 | 0.500 | 0.500 | 0.499 | 0.500 | 0.500 | 0.497 |
| PC1-MC2 | 0.651 | 0.655 | 0.618 | 0.675 | 0.666 | 0.641 |
| PC1-MW1 | 0.596 | 0.581 | 0.572 | 0.599 | 0.579 | 0.573 |
| PC4-MC2 | 0.546 | 0.544 | 0.504 | 0.533 | 0.530 | 0.496 |
| PC4-MW1 | 0.546 | 0.544 | 0.504 | 0.533 | 0.530 | 0.496 |
| MC2-MW1 | 0.575 | 0.546 | 0.553 | 0.583 | 0.542 | 0.578 |

both the FIS with all three types of membership functions.

*2) Auto-Generated :* In order to perform the identification of similar datasets in absence of expert, we have generated the automatic rules and FIS. In this paper subtractive clustering (SC) has been used to determine number of rules and membership functions for each pair. Auto rule generation generates sugeno type FIS and uses Gaussian membership function for all the inputs. After generating rules and developing the FIS on training data, evaluation has been performed using test data and the results have been reported in Table VII.

## IV. DISCUSSION

Datasets used for experimentation can be categorized in three groups depending upon language, number of attributes and percentage of defective modules. Data sets belong to same group should be more similar to each other than data sets from different groups. There are four data sets that have 22 attributes name CM1, KC1, KC2 and PC1, similarly MW1 and PC4 fall in same group and MC2 and KC3 in same. CM1 has more similarity with its group members than data sets from other two groups. Only its similarity with one MC2 from third group 62.1% more than

its group member KC2. PC4 and MW1 from second group are 54.6% similar, MW1 is 4 times less similar to data sets from other group except its similarity with KC3 and PC1 but they are more close to each other due to having approximately close percentage of defective modules. PC4's highest similarity is with its own group member and have very little similarity with other groups. Third group of MC2 and KC3 have similarity 53.9% that does not differentiate them well being a same group members. Possible reason might be programming languages and significant difference between defective modules that makes them more close to other data sets.

On the basis of programming languages CM1, PC1 and PC4 belongs to one group and similarity of PC4 with CM1 and PC1 is more than its similarity with KC1, KC2 whereas its more close to MC2 and MW1 but MW1 more close because they have same number of attributes. Data sets KC1 and KC2, MW1 and MC2 developed in same programming language. MC2 is more close to its group member on programming language basis except the case where its similarity with CM1 is 62.9% that is more than similarity with KC2. Third group of just one data set KC3 have high similarity with KC1 and MW1. Percentage of defective modules seems not to be good criteria to group similar data sets. It fails in most of the cases.

Nine least similar dataset pairs with similarity less than equal to 50% belong to different groups on the basis of number of attributes. That are KC2-MW1, KC1-KC2, KC3-PC4, PC1-PC4, KC2-KC3, KC2-PC4, CM1-PC4, KC1-PC4 and KC3-PC1. Ten most similar data set pairs also belong to same groups, they are in same group either on number of attributes, programming language or defective modules. KC2-PC1, CM1-KC1, CM1-PC1, CM1-KC2 are in high similarity with respect to number of attributes, KC3 and MW1 second highest similar pair belong to same category on basis of defective modules. KC2-MC2 are similar on basis of programming language. KC1-KC3 the third highest pair does not belong to same group on any basis. PC1-MC2, CM1-MC2 are more similar even though not from same group.

Results are consistent most of the time on number of attributes and programming language basis to categorize similar data sets. Despite of few wrong classifications overall performance good. But some more valid criteria might be needed to compare results that can group data sets on basis of their patterns. Results with genfis2 were quite close to actual output. Except one case where genfis2 predicted similarity for KC3-PC1 is quite higher than similarity value produced by sugeno gaussian.

## V. Conclusion and Future Work

This paper has presented an approach to compare software datasets with the help of association rules extracted from the datasets. The similarity has been calculated through Fuzzy Inference System (FIS). In the approach presented here, partial similarity of association rules is also handled. This similarity between software datasets can be used to monitor and improve the running or future projects. This can further make it possible to benefit from public datasets.

Table VII
Dataset Similarity Calculated by Auto-Generated FIS

| DataSet Pairs | Sug-Gaussian) | SC-Gaussian |
|---|---|---|
| KC3-PC1 | 0.16 | 0.35 |
| KC3-PC4 | 0.45 | 0.45 |
| KC3-MC2 | 0.50 | 0.50 |
| KC3-MW1 | 0.83 | 0.81 |
| PC1-PC4 | 0.50 | 0.42 |
| PC1-MC2 | 0.64 | 0.65 |
| PC1-MW1 | 0.57 | 0.56 |
| PC4-MC2 | 0.50 | 0.50 |
| MC2-MW1 | 0.58 | 0.56 |

The association rule mining has been applied to find frequent patterns of attributes and the datasets with similar attributes have been assigned overlap value. This overlap has been used to find pairwise similarity between the datasets through Fuzzy inference. For the 28 dataset pairs, expert based FIS having 27 fuzzy rules has been developed. In order to demonstrate development of FIS in absence of an expert, subtractive clustering has been used to generate automatic rules and FIS. The datasets used in study have been grouped on the basis of number of attributes, programming language and defectiveness. Datasets belonging to the same groups have been more close to each other than their the datasets from another group. Number of attributes has emerged as an appropriate grouping strategy. Comparable results from expert-based and auto-generated FIS show that it is not required to have an expert available for fuzzy rules development. The automatic rules can achieve the same performance with input from a domain expert. The auto-generated and the expert based Sugeno FIS can further be optimized to improve identification of similar datasets.

## References


[1] H. Wang, W. Wang, J. Yang, and P. S. Yu, "Clustering by pattern similarity in large data sets," in *Proceedings of the 2002 ACM SIGMOD international conference on Management of data*, ser. SIGMOD '02. New York, NY, USA: ACM, 2002, pp. 394–405. [Online]. Available: http://doi.acm.org/10.1145/564691.564737

[2] K. Sequeira, "Exploring similarities in high-dimensional datasets," Ph.D. dissertation, Troy, NY, USA, 2005, aAI3183647.

[3] M. Shepperd and C. Schofield, "Estimating software project effort using analogies," *Software Engineering, IEEE Transactions on*, vol. 23, no. 11, pp. 736 –743, nov 1997.

[4] S. Parthasarathy and M. Ogihara, "Exploiting dataset similarity for distributed mining," in *Proceedings of the 15 IPDPS 2000 Workshops on Parallel and*

*Distributed Processing*, ser. IPDPS '00. London, UK, UK: Springer-Verlag, 2000, pp. 399–406. [Online]. Available: http://dl.acm.org/citation.cfm?id=645612.662849

[5] T. Li, M. Ogihara, and S. Zhu, "Association-based similarity testing and its applications," *Intell. Data Anal.*, vol. 7, pp. 209–232, August 2003. [Online]. Available: http://dl.acm.org/citation.cfm?id=1293875.1293878

[6] D. Dudek, "Measures for comparing association rule sets," in *Proceedings of the 10th international conference on Artificial intelligence and soft computing: Part I*, ser. ICAISC'10. Berlin, Heidelberg: Springer-Verlag, 2010, pp. 315–322. [Online]. Available: http://dl.acm.org/citation.cfm?id=1894214.1894256

[7] S. Anwar, Z. Rana, S. Shamail, and M. M. Awais, "Using association rules to identify similarities between software datasets," in *Proceedings of the 8th International Conference on Quality of Information and Communications Technology (QUATIC'12)*. IEEE Computer Society, 2012.

[8] M. Azzeh, D. Neagu, and P. Cowling, "Software project similarity measurement based on fuzzy c-means," in *Proceedings of the Software process, 2008 international conference on Making globally distributed software development a success story*, ser. ICSP'08. Berlin, Heidelberg: Springer-Verlag, 2008, pp. 123–134. [Online]. Available: http://dl.acm.org/citation.cfm?id=1789757.1789773

[9] A. Idri and A. Abran, "A fuzzy logic based set of measures for software project similarity: validation and possible improvements," in *Software Metrics Symposium, 2001. METRICS 2001. Proceedings. Seventh International*, 2001, pp. 85 –96.

[10] A. Idri. and A. Abran., "Evaluating software project similarity by using linguistic quantifier guided aggregations," in *IFSA World Congress and 20th NAFIPS International Conference, 2001. Joint 9th*, vol. 1, july 2001, pp. 470 –475 vol.1.

[11] A. Barreto and A. Rocha, "Analyzing the similarity among software projects to improve software project monitoring processes," in *Quality of Information and Communications Technology (QUATIC), 2010 Seventh International Conference on the*, 29 2010-oct. 2 2010, pp. 441 –446.

[12] G. Boetticher, T. Menzies, and T. Ostrand, "Promise repository of empirical software engineering data." West Virginia University, Department of Computer Science, 2007. [Online]. Available: http://promisedata.org/ repository

[13] T. M. Mitchell, *Machine Learning*. New York: McGraw-Hill, 1997.

[14] E. H. Mamdani and S. Assilian, "An experiment in linguistic synthesis with a fuzzy logic controller," *International Journal of Man-Machine Studies*, vol. 7, no. 1, pp. 1–13, 1975.

[15] M. Sugeno, *Industrial Applications of Fuzzy Control*. New York, NY, USA: Elsevier Science Inc., 1985.

[16] J. Jang, C. Sun, and E. Mizutani, "Neuro-fuzzy and soft computing-a computational approach to learning and machine intelligence [book review]," *Automatic Control, IEEE Transactions on*, vol. 42, no. 10, pp. 1482 –1484, oct 1997.

[17] T. J. Ross, *Fuzzy Logic with Engineering Applications*, second edition ed. John Wiley and Sons, 2004.