ISSN: 2442-6571

Generated rules for AIDS and e-learning classifier using rough set approach

Sarina Sulaiman [a,1,*], Nor Amalina Abdul Rahim [b,c,2], Andri Pranolo [b,3]

[a] *UTM Big Data Centre, Universiti Teknologi Malaysia, Skudai Johor, Malaysia*
[b] *Faculty of Computing, Universiti Teknologi Malaysia, Skudai Johor, Malaysia*
c Informormatics Department, Universitas Ahmad Dahlan, Yogyakarta, Indonesia
[1] *sarina@utm.my *;* [2] *haura.malina@gmail.com;* [3] *andri.pranolo@tif.uad.ac.id*
** corresponding author*

ARTICLE INFO

Article history:
Received June 6, 2016
Revised July 17, 2016
Accepted July 30, 2016

Keywords:
Rough Set
AIDS blog data
E-Learning log data
Rules derivation
Cross validation

ABSTRACT

The emergence and growth of internet usage has accumulated an extensive amount of data. These data contain a wealth of undiscovered valuable information and problems of incomplete data set may lead to observation error. This research explored a technique to analyze data that transforms meaningless data to meaningful information. The work focused on Rough Set (RS) to deal with incomplete data and rules derivation. Rules with high and low left-hand-side (LHS) support value generated by RS were used as query statements to form a cluster of data. The model was tested on AIDS blog data set consisting of 146 bloggers and E-Learning@UTM (EL) log data set comprising 23105 URLs. 5-fold and 10-fold cross validation were used to split the data. Naïve algorithm and Boolean algorithm as discretization techniques and Johnson's algorithm (Johnson) and Genetic algorithm (GA) as reduction techniques were employed to compare the results. 5-fold cross validation tended to suit AIDS data well while 10-fold cross validation was the best for EL data set. Johnson and GA yielded the same number of rules for both data sets. These findings are significant as evidence in terms of accuracy that was achieved using the proposed model.

I. Introduction

Web mining extracts the information from the World Wide Web (WWW) by using data mining techniques. The extraction of hidden pattern, or predictive information from huge database, and useful knowledge and unknown information can be discovered by using data mining. Data mining is one of the parts in Knowledge Discovery in Database (KDD). KDD is one of the processes used to transform data into knowledge. Data mining as an analysis of enormous datasets to discover hidden information or unsuspected relationships inside a network and to concise the data in novel ways and produce useful and meaningful information to the owner of the data [1]. In addition, data mining can be defined as a method of automatically extracting implicit and useful patterns from databases [2]. It encompasses many different techniques and algorithm, including classification, clustering, association rules and others. Over the years, Rough Set Theory (RST) has become an interest for researches and has been applied to many domains, such as data classification, data clustering, and association rules mining.

Rough Set (RS) analyzes uncertainty of a dataset that is used to determine the crucial attributes of objects and build the upper and lower approximate sets of objects sets [3], [4]. The main advantage of using RST instead of fuzzy set in data analysis is that it does not need any preliminary or additional information about data − like probability in statistics, grade of membership or the value of possibility in fuzzy set theory [5], [6]. In the real world data varies in size and complexity, difficult to analyze and also hard to manage from computational view point. The major objectives of RS analysis are to reduce data size and to handle inconsistency in data [4]. Moreover, it is being used for the extraction of rules from database. Decision rules extracted by RS algorithms are

valuable and concise, which can be beneficial by enlightening some hidden knowledge in the data [7]. Another research is

Came out with the question of problem regarding large log dataset on how to remove the messy data timely with low cost and find out useful information in huge dataset [8]. Therefore, to solve the problem of incomplete dataset, RST will be used since RS can deal with uncertainty data. RST is a new mathematical tool that can handle uncertainty and incomplete information. A principal goal of RST analysis is to synthesize or construct approximations (upper and lower) offsets concepts from the acquired data [9]. RS for rules generation and rules extraction for better classification in Web usage mining using the Web log dataset since RS can deal with uncertain data has applied by [10], [11], and [12]. The generated rules will be used as a guideline to query a large dataset and get the accurate relationship among the parameters from the database.

The rest of this paper is organized as follows. Section II reviews the related works; Section III presents the experimental design; Section IV provides the experimental results and analysis. Finally, the last section in this paper, Section V describes a conclusion as summary of the research.

II. Related Works

With the enormous growth of data especially large size data sets, mixed types of data, data change, incomplete and uncertain data, the information system may contain a number of redundancies that will not assist in any knowledge discovery and may in fact deceive the process. One of the methods which can be used to deal with these issues is the RST, proposed by [13], a mathematical tool used to deal with imperfect knowledge and discover pattern hidden in data. RST deals with uncertainty and vagueness, allowing generation of the sets of decision rules from data. Reduct set can be generated or the core of the contribute set can be constructed by eliminating the redundant attributes [14].

This simple idea leads to many competent applications of RS such as data mining, machine learning, and also in granular computing. RS have also been applied in many real life applications such as web transaction [15], [16], web search clustering [17], medical [7], [18], [19], [20], e-learning [2], [3], and marketing [21]–[23]. In real world data varies in size and complexity, which is difficult to analyze and also hard to manage from computational view point. The major objectives of RS are to reduce data size and to handle inconsistency or redundancy in data [4]. Hidden patterns or hidden information or relationship can be identified from large data sets. Therefore, RS is used in this research to generate rules.

A. Reduct and Rules Generation

Computation of reduct is conducted to determine minimal attributes that represent the patterns of knowledge in the data. Attributes that are irrelevant will be eliminated through reduction process and rules will be produced from the reduced number of attributes. Thus, unimportant and redundant knowledge need to be eliminated in order to generate an effective reduct set and a more reliable model. Johnson's algorithm (Johnson) and Genetic algorithm (GA) are two reduction methods that can be used to generate rules. These two reduction methods are provided in ROSETTA software.

ROSETTA is a toolkit designed to support the overall data mining and knowledge discovery process, and for analyzing tabular data within the framework of RST that could be applied in the original dataset to compute the reduced set without the loss of the knowledge of the original set [24]. The whole RS processes can be applied in ROSETTA; from the initial browsing and pre-processing of the data via computation of minimal attribute sets and generation of if-then rules or descriptive patterns, up to the validation and analysis of the induced rules or patterns [25].

Liu [26] stated that system performance is more effective if the rules are less. Performing reduction on a set of data is one mechanism to decrease the number of rules. Reduct provided by RS generates comprehensible rules compared to other methods [27]. Liang et.al. [27] used RS and Rough Set-based Inductive Learning to help instructors and students with WebCT learning. Rough Set-based Inductive Learning was used to obtain the decision rules to provide the reasons for the lack of success of students. As a result, the Web learning system improved and increased the effectiveness of WebCT.

Back in 2001, RST used for the analysis of diabetic databases [12]. They applied RS to Pima Indian Diabetic Database (PIDD) by using ROSETTA software. 392 complete cases in the PIDD was randomly divided into training set (n=300) and testing set (n=92). The training set was then discretized. They used the Equal Frequency Binning (EFB) with k=5 bins. Then, they applied Johnson reducer algorithm to create the reduct. Next, classification method was applied to the testing set by using batch classifier with the standard/tuned voting method (RSES). The generated rules were applied to the testing set. The result showed that the prediction accuracy was increased. The workflow of the main steps in conducting rough set analysis has proposed by [28]. The workflow in Fig. 1 is the same as the process conducted by [12].

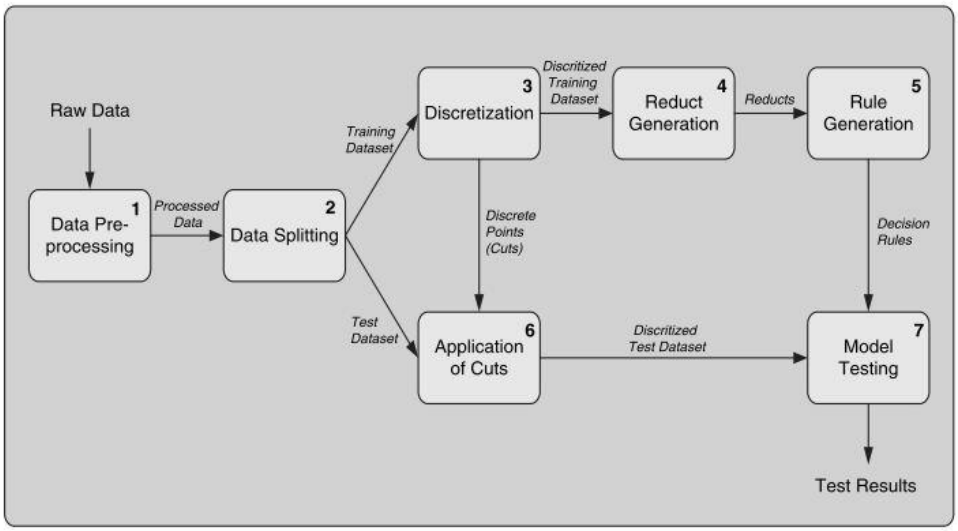

Fig. 1. Main steps of rough sets analysis by [28]

On the other hand, RST applied to feature granularity of Cardiac datasets with 70% training and 30% testing set [7]. Standard voting classifier (SVC) was used to implement classification. Fig. 2 shows the RS classification modeling as shown by [29]. EFB discretization technique with k=3 was used to get the same number of data for each interval. New decision table was constructed based on core attributes and minimal cardinality in the generated reduct. Highest support values, less length and highest percentage of Rule Importance Measure (RIM) were the parameters used to analyze the generated rules.

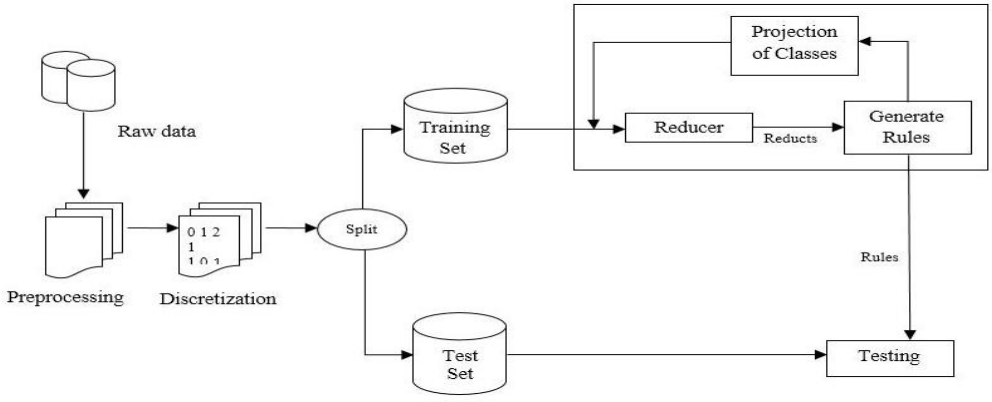

Fig. 2. Rough Set classification modeling by [29]

In case of reduct, the experiment performed by [30] using k-fold method resulted with more rules produced by GA compared to Johnson which led to less accuracy. Their results also proved that k=10 was convenient for model validation. Fig. 3 shows the general steps to develop performance prediction model proposed by [30].

In 2012, the concept of cross validation with k=10 also applied [3]. The generated rules enhanced the prediction performance of Web pre-caching and the rules were then used to construct

queries for the datasets using Social Network Analysis (SNA). Fig. 4 shows the illustration of RS classification procedure by using ROSETTA System.

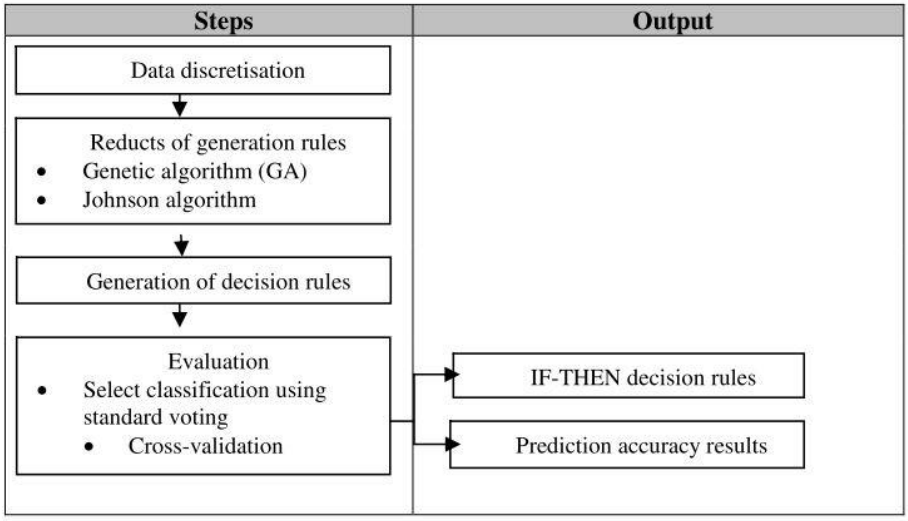

Fig. 3. General steps for development of prediction model [30]

In recent years, a new technique to solve the issue of unorganized large multimedia data proposed by [31]. They used RST and web services technology in the proposed model to classify and analyze data. The proposed technique that involved 50% of testing data and 50% training data proved the effectiveness of RST in classifying data into respective clusters. RST for customer classification also applied [23]. The generated rules presented the factors that influenced the client's purchase. They claimed that RST had no information loss, extendable and flexible compared to other data mining technologies. The generated rules helped to make their products better and organized the customer accurately. Both of these studies used ROSETTA software for validation and data processing. A year after, decision rules were used to classify real world Web services, done by [32] to improve the classification accuracy. Fig. 5 shows the RS steps they proposed.

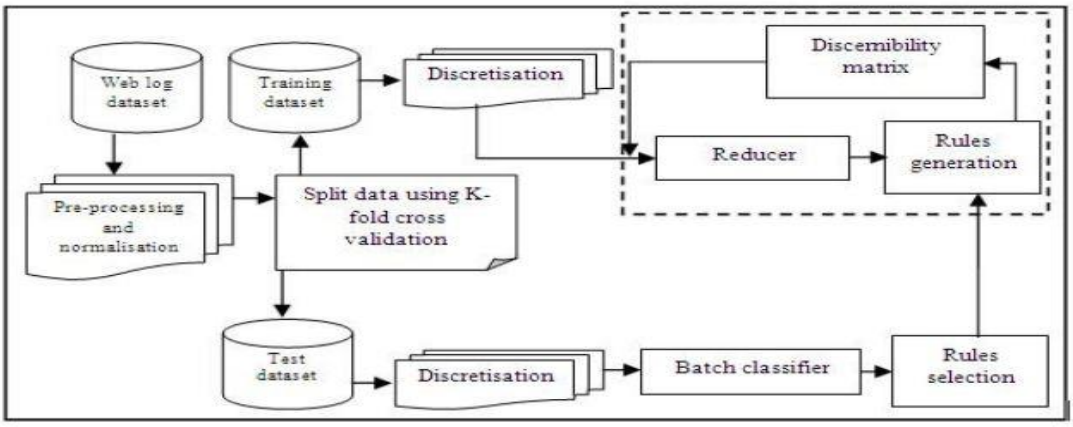

Fig. 4. RS classification procedure [3]

A new approach of Karnaugh map for the reduction of attributes and RST to generate rules proposed by [14]. They claimed that the major objectives of RS analysis were to reduce data size and to handle inconsistency in data. They dealt with uncertainty and extracted useful information from the database. The proposed work used Flu Data Set where the data was discretized by using RST and K-map. The data about six patients was used as training data. Using k-map and RS approach, data was analyzed; redundant data was eliminated; attributes were reduced and set of rules were developed.

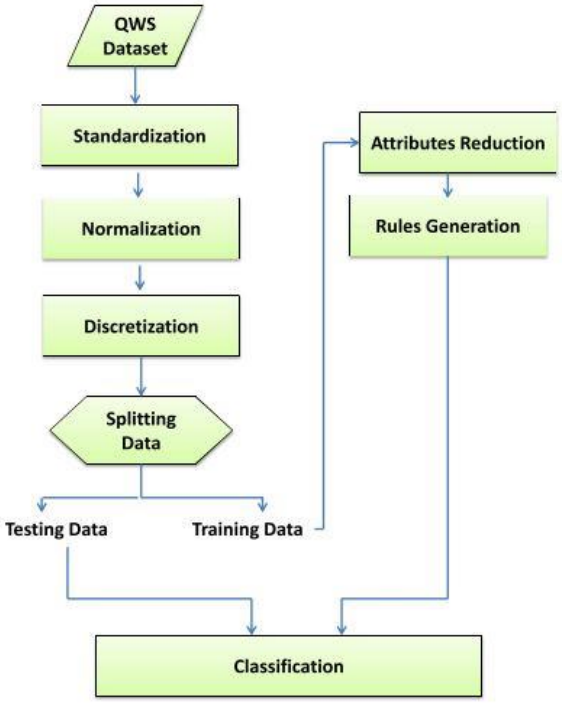

Fig. 5. Rough set-based Web services classification technique [32]

The cross validation technique is used to define a validation dataset to test the model after training phase. In k-fold cross validation, the dataset is divided into k subsets of equal size. For each k experiment, a single subset is used as the testing set, and the remaining k-1 subsets will be used as the training set, as shown in Fig. 6. The advantage of using k-fold cross validation is all the data are eventually used for both training and testing. This technique ensures that each data subset is tested once, and thus has the same proportion of data, reducing bias in the model evaluation. k-Fold technique allows the accuracy for each fold to be calculated. The fold with the highest prediction accuracy can be identified.

Based on previous studies such as by Omar *et al*. (2011), Sulaiman *et al*. (2012) and Phillips *et al*. (2015), 10-fold cross validation is convenient for model validation. However, according to Omar *et al*. (2011), it is possible to divide k = 5 and k = 10 depending on the data size. Moreover, as stated by [34], the value of k is often 5 or 10, but there is no specific requirement.

In ROSETTA, rules are constructed based on IF-THEN rules. Then terms used are LHS which stands for Left Hand Side that refers to the IF-part of the rule; and RHS which is Right Hand Side that refers to the THEN part of the rule. Rules are evaluated according to how general they are such as for coverage, the fraction of objects from the decision class in the THEN-part matches the IF-part; and how specific they are such as for accuracy, the fraction of objects matches the IF-part that are from the decision class of the THEN-part [35]. Rosetta lists the rules and provides some statistics for the rules which are support, accuracy, coverage, stability and length. Below is the definition of the rule statistics: "i) the rule of LHS support is defined as the number of records in the training data that matches the IF condition, ii) the rule of RHS support is defined as the number of records in the training data that matches the THEN condition, iii) the rule of RHS accuracy is defined as the number of RHS support divided by the number of LHS support, iv) the rule of LHS coverage is the fraction of the records that satisfies the IF conditions of the rule. It is obtained by dividing the support of the rule by the total number of records in the training sample, v) The rule of RHS coverage is the fraction of the training records that satisfies the THEN conditions. It is obtained by dividing the support of the rule by the number of records in the training that satisfies the THEN condition, vi) The rule of LHS length is defined as the number of conditional elements in the IF part, vii) The rule of RHS length is defined as the number of conditional elements in the THEN part" [7], [35].

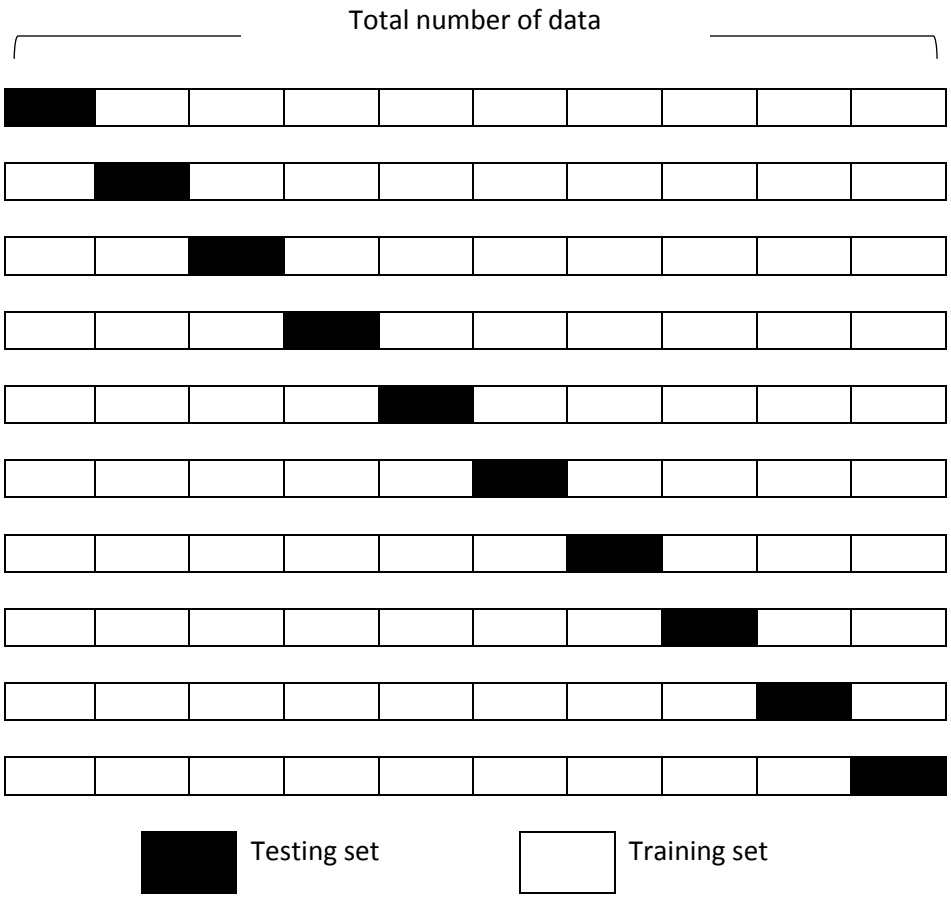

Fig. 6. Example of 10-fold cross validation

Most of these previous researchers used ROSETTA for the entire RS processes starting from data pre-processing until data classification stage. Some of the studies used the traditional technique to split data into training and testing set, and some used the k-fold cross validation technique. Discretization is applied to training and testing set. Discretization technique is one of the pre-processing techniques. The use of continuous attributes involves huge storage, misinterpretation and long rules. Hence, discretization is needed to change from continuous attributes to discrete attribute in order to increase the accuracy in prediction [36]. Then, reduction is performed by using GA or Johnson to generate rules. Johnson used by [12] while [3] applied GA, and [30] applied these two reduction methods to make comparison. The generated rules were then used to classify the testing set. Hence, the classification accuracy was obtained.

Therefore, in this research, Rough Set is so far considered as popular approaches to generate rules. The generated rules will then be selected based on LHS support in order to query the dataset in Social Network Analysis part. The reason of using LHS support to select the significant rules is discussed in the next section.

B. Significant Rules

Reduct is possible to generate large number of rules that can be important or unimportant. Therefore, many analysis have been using approaches to identify the significant rules. Reference [37] suggested sorting the rules based on the support value in order to find the most important rules for each set. Value of length is not much different between each rules, thus support is used as the criteria to rank the rules. Furthermore, reference [7] claimed that rules with less length were not effective to measure the significance of rules.

On the other hand, the rules that had the highest support of objects in LHS support was the most significant rules mentioned by [30]. Previously, reference [38] proposed a new measure called Rule Importance Measure (RIM) to evaluate association rules based on rough sets theory. It is possible for rules from different reduct sets to contain dissimilar representative information. Thus, important information might be excluded if only one set of reduct is examined for rules generation. Multiple

reducts will generate the rules many times. Rules that occur more frequently are considered to be more important. If a rule is generated more frequently across different rule sets, we say that this rule is more important than other rules [32].

III. Experimental Design

The proposed model of Rough Set Rules Generation (RSRG) by generating RS and set of rules is illustrated in Fig. 7. There are two components involved in the model. The two components are data pre-processing in which data was converted into a format that is acquainted for experiment; and rough set rules generation to generate rules and select rules based on high and low LHS support.

A. Data Pre-processing

This phase is the corresponding activity of component one as depicted in Fig. 7. This step includes two subsections, data cleaning and data transformation. Data collection and data analysis were involved at the beginning of this phase. In this research, two different datasets, including AIDS and EL log datasets were used as the datasets. The raw dataset would undergo the pre-processing process. Data pre-processing involved manipulating input data into a suitable form.

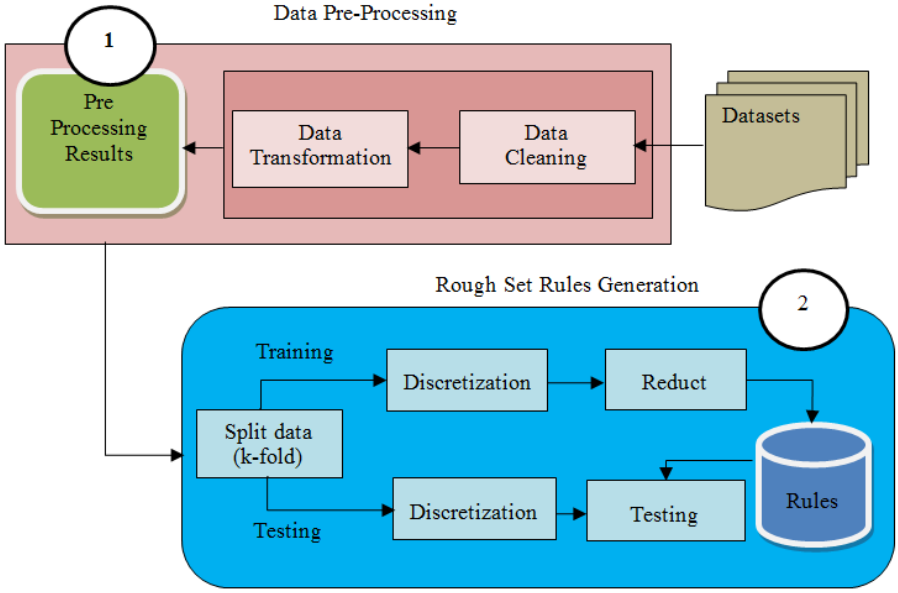

Fig. 7. The proposed model of RSRG

The data pre-processing involved the following two steps:

- Filtering the data to remove unnecessary fields.
- Finalizing the data into a format that is acquainted for experiment.

The first step in pre-processing involved the process of identifying incorrect and unused records, and removing unnecessary attributes. Second step involved the formatting of data to be acquainted and amenable for experiment. The pre-processing output were then passed and processed as inputs for the next process.

B. Rough Set Rules Generation

Next, the filtered data would undergo the process in second component as illustrated in Fig. 7. The generated rules would be selected based on highest and lowest support value that would be used as the queries to cluster the data. Fig. 8 illustrates the procedure of RS using ROSETTA system. The procedure involved data splitting, data discretization, data reduction, classification and selection of significant rules based on LHS support.

1) Data Splitting

In this research, k-fold cross validation was used to split the data into testing and training set. The aim of using this technique was to validate the dataset and to ensure the consistency of results.

In fact, according to [28], the main advantage of k-fold cross validation is to reduce the bias by repeating the experiment ten times. Even though this methodology is rather time consuming, it is a viable option for small datasets. This research clearly expressed that 10-fold cross validation does not require more data compared to the conventional single split. Furthermore, 10-fold cross validation is the best and has been the common practice. In fact, in data mining community, for methods-comparison studies with relatively smaller datasets, k-fold cross validation is recommended [39].

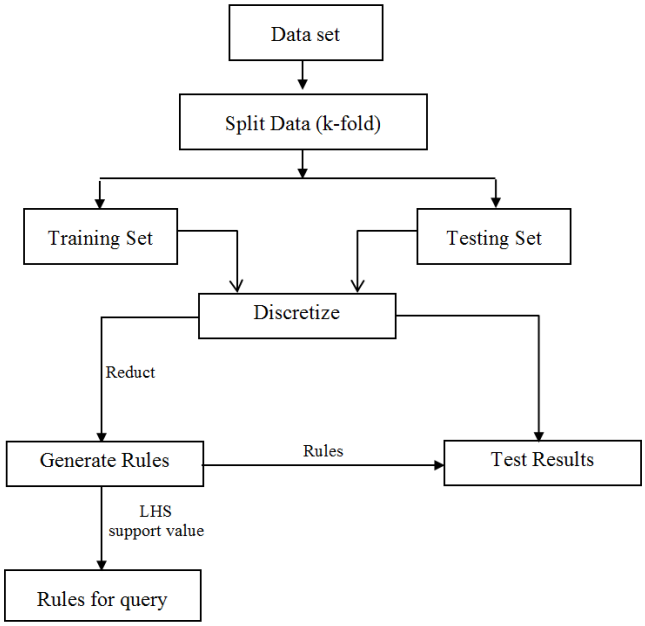

Fig. 8. Procedures of RS

Nevertheless, as discussed in Section *II.A*, it is possible to divide k=5 and k=10 depending on the size of data. Hence, in this research, 5-fold and 10-fold cross validations were applied on both datasets to test different types of k-fold on different sizes of datasets. Data was divided into 5-fold (80% training, 20% testing) and 10-fold (90% training, 10% testing) as shown in Tables 1 and 2, respectively.

Table 1. 5-fold cross validation of AIDS and EL datasets

Fold	AIDS	EL
1	1 – 38	1 – 4621
2	39 – 75	4622 – 9242
3	76 – 112	9243 – 13863
4	113 – 149	13864 – 18484
5	150 – 187	18485 – 23105

Table 2. 10-fold cross validation of AIDS and EL datasets

Fold	AIDS	EL
1	1 – 19	1– 2,311
2	20 – 38	2,312 – 4,622
3	39 – 57	4,623 – 6,933
4	58 – 76	6,934 – 9,244
5	77 – 95	9,245 – 11,555
6	96 – 114	11,556 – 13, 865
7	115 – 133	13, 866 – 16, 175
8	134 – 151	16,176 – 18,485
9	152 – 169	18,486 – 20,795
10	170 – 187	20,796 – 23,105

2) Discretization

Next, each training and testing set executed the discretization process. Discretization process involves converting continuous values into categories or classes. Reference [40] claimed that Naïve and Boolean Reasoning were ranked first as these two algorithms were the most suitable discretization methods in medical area that provided better accuracy. Similarly for engineering data with a specific class distribution, Naïve, semi-naïve and entropy gave better results compared to other methods.

Therefore, in this research, two techniques of discretization provided by ROSETTA Toolkit which is Naïve Algorithm [10], [38], [41] and Boolean Reasoning Algorithm [42]–[44] were tested to establish a technique that present high accuracy in classification. This research also compared the accuracy of non-discretization technique with discretization technique. The end result of this process is data was transformed into several categories.

3) Reduct

Subsequently, the training sets went through the reduction process and rules were generated from this data. Different reduct techniques were compared between Genetic algorithm (GA) and Johnson's algorithm (Johnson). Reduct generation had two options; full object reduction and object related reduction. Full object reduction produced a set of minimal attributes subset that defines functional dependencies, while reduct with object related produced a set of decision rules or general pattern through minimal attributes subset that discern on a per object basis. The classification accuracy for reduct with object related is higher than using full reduct [45]. Hence, reduct with object related was preferred in this research due to its ability in generating reduct based on discernibility function of each object.

4) Classification

Lastly, the testing sets were used to verify the rules generated from training sets. The classification was implemented using Standard Voting Classifier (SVC). The performance of SVC was more optimal and more accurate compared to Batch classifier performance [46]. They concluded that, SVC was a better classifier in ROSETTA. Reference [47] claimed that SVC was an efficient algorithm under RS. Therefore, in this research, SVC was used to enhance accuracy of classification. The rules generated were used to classify the testing dataset.

IV. Experimental Results and Analysis

In this research, the classification accuracy of non-discretized datasets were compared with discretized datasets. The following section will discuss the results of non-discretized technique, followed by results of discretization technique, and results of reduct and rules generation which presents the number of reduct and rules.

A. Non-Discretization

The aim of this process is to compare the accuracy of non-discretize and discretize datasets with different reduct methods. The two reduct methods are Johnson's algorithm (Johnson) and Genetic algorithm (GA). 10-fold and 5-fold cross validations were used in this research since according to Omar *et al*., (2011), depending on the data size, it is possible to divide k = 5 and k = 10. K-folds are labeled as 1, 2, 3,.. to 10.

Table 3 shows prediction accuracy of each fold using k=10 for AIDS dataset. Both reduct methods obtained the same accuracy. The highest prediction accuracy was 84.21% and the lowest was 66.67%. The average accuracy for 10-fold cross validation was 75.32%.

Table 3. Classification accuracy for non-discretization technique for AIDS dataset 10-fold

Reduct Technique	**Johnson**		**GA**	
K-Fold	***Prediction Accuracy (%)***	***Average Accuracy (%)***	***Prediction Accuracy (%)***	***Average Accuracy (%)***
1	78.95	75.32	78.95	75.32
2	78.95		78.95	
3	73.68		73.68	
4	78.95		78.95	

Reduct Technique	**Johnson**		**GA**	
K-Fold	***Prediction Accuracy (%)***	***Average Accuracy (%)***	***Prediction Accuracy (%)***	***Average Accuracy (%)***
5	84.21		84.21	
6	78.95		78.95	
7	68.42		68.42	
8	66.67		66.67	
9	66.67		66.67	
10	77.78		77.78	

GA and Johnson obtained the same accuracy as 5-fold cross validation as depicted in Table 4. The highest prediction accuracy was 81.08%, the lowest was 65.79% and the average was 72.75%. This indicates that cross validation k=10 produced higher accuracy than k=5 for non-discretize AIDS data.

Table 4. Classification accuracy for non-discretization technique for AIDS dataset 5-fold

Reduct Technique	**Johnson**		**GA**	
K-Fold	***Prediction Accuracy (%)***	***Average Accuracy (%)***	***Prediction Accuracy (%)***	***Average Accuracy (%)***
1	65.79		65.79	
2	76.32		76.32	
3	81.08	72.75	81.08	72.75
4	70.27		70.27	
5	70.27		70.27	

However, for EL dataset, GA outperformed Johnson with average accuracy of 97.86%, while Johnson yielded 97.74% when using 10-fold cross validation as shown in Table 5. The highest prediction accuracy was 99.31% (fold 6) obtained by both reduct techniques and the lowest was 95.80% (fold 1) obtained by Johnson.

Table 5. Classification accuracy for non-discretization technique for EL dataset - 10- fold

Reduct Technique	**Johnson**		**GA**	
K-Fold	***Prediction Accuracy (%)***	***Average Accuracy (%)***	***Prediction Accuracy (%)***	***Average Accuracy (%)***
1	95.80		95.89	
2	99.00		99.00	
3	98.78	97.74	98.83	97.86
4	97.71		97.75	
5	97.53		97.53	
6	99.31		99.31	
7	95.93		95.93	
8	98.05		98.05	
9	98.40		98.48	
10	96.84		96.84	

Table 6 shows 5-fold cross validation where the average accuracy of GA was 0.02% over Johnson. The highest prediction accuracy was 98.25% (fold 2) obtained by GA and the lowest was 96.88% (fold 4) obtained by both reduct techniques.

Table 6. Classification accuracy of non-discretization technique for EL dataset - 5-fold

Reduct Technique	**Johnson**		**GA**	
K-Fold	***Prediction Accuracy (%)***	***Average Accuracy (%)***	***Prediction Accuracy (%)***	***Average Accuracy (%)***
1	97.36		97.36	
2	98.20		98.25	
3	98.07	97.60	98.07	97.62
4	96.88		96.88	
5	97.49		97.53	

B. Discretization

Both datasets were then discretized using two discretization techniques provided by ROSETTA; Naïve algorithm and Boolean reasoning (BR) algorithm. Both algorithms were compared to determine the highest accuracy. Tables 7, 8, 9 and 10 show the results of discretization technique for AIDS and EL datasets, respectively.

Table 7 illustrates the prediction accuracy for each fold when using cross validation k=10 for AIDS dataset. The highest prediction accuracy for Naïve algorithm was 68.42% and the lowest was 61.11%. The highest prediction accuracy obtained by BR was 84.21% and the lowest was 52.63%. It was discovered that Boolean yielded higher average accuracy than Naïve with difference of 8.98%

Table 7. Classification accuracy for discretization technique for AIDS dataset - 10-fold

Discretization	**Naive algorithm**		**Boolean algorithm**	
K-Fold	***Prediction Accuracy (%)***	***Average Accuracy (%)***	***Prediction Accuracy (%)***	***Average Accuracy (%)***
1	63.16	63.13	78.95	72.11
2	63.16		78.95	
3	63.16		52.63	
4	63.16		78.95	
5	63.16		84.21	
6	52.63		78.95	
7	68.42		68.42	
8	66.67		66.67	
9	61.11		55.56	
10	66.67		77.78	

Table 8 depicts the prediction accuracy for each fold when using cross validation k=5 for AIDS dataset. The highest accuracy of 81.08% was obtained by Boolean reasoning algorithm and the lowest, 54.05% was obtained by Naïve algorithm. It was discovered that Boolean yielded higher average accuracy than Naïve with difference of 11.18%. Nevertheless, the comparison between the average accuracy of k=5 and k=10 shows that average accuracy of Naïve and BR when k=5 was higher than k=10.

Table 8. Classification accuracy of discretization technique for AIDS dataset - 5-fold

Discretization	**Naive algorithm**		**Boolean algorithm**	
K-Fold	***Prediction Accuracy (%)***	***Average Accuracy (%)***	***Prediction Accuracy (%)***	***Average Accuracy (%)***
1	73.68	64.12	76.31	75.93
2	63.16		76.31	
3	54.05		81.08	
4	70.27		72.97	
5	59.46		72.97	

Meanwhile, surprisingly for 10-fold cross validation of EL dataset, Boolean yielded lower average of accuracy than Naïve algorithm as shown in Table 9. The higher accuracy was 100% obtained by Naïve algorithm and the lowest was 51.17% obtained by BR algorithm. There was a clear difference between average accuracy of Naïve and Boolean as much as 23.56%.

While as depicted in Table 10, the highest accuracy for 5-fold cross validation was also 100% obtained by Naïve algorithm and the lowest was 59.08 obtained by BR algorithm. Thus, this indicates that Naïve outperformed BR when cross validation k=5 and k=10. However, the difference of average accuracy between these two discretization techniques when k=5 was only 8.19%.

Table 9. Classification accuracy for discretization technique for EL dataset - 10-fold

Discretization	**Naive algorithm**		**Boolean algorithm**	
K-Fold	***Prediction Accuracy (%)***	***Average Accuracy (%)***	***Prediction Accuracy (%)***	***Average Accuracy (%)***
1	100.00	99.98	70.40	76.42
2	99.96		99.96	
3	100.00		64.26	
4	99.96		99.96	
5	99.91		99.91	
6	100.00		60.39	
7	99.96		51.17	
8	99.96		99.96	
9	100.00		63.12	
10	100.00		55.02	

Table 10. Classification accuracy for discretization technique for EL dataset - 5-fold

Discretization	**Naive algorithm**		**Boolean algorithm**	
K-Fold	***Prediction Accuracy (%)***	***Average Accuracy (%)***	***Prediction Accuracy (%)***	***Average Accuracy (%)***
1	99.98	99.98	99.98	91.79
2	99.98		99.98	
3	99.96		99.96	
4	99.96		99.96	
5	100.00		59.08	

Table 11 summarizes the results for non-discretize and discretize datasets. Based on the results for AIDS dataset, it was discovered that BR outperformed Naïve and non-discretized AIDS data when using k=5, while k-10 non-discretize outperformed Naïve and BR. As for k-fold of discretize data, the average accuracy of Naïve and BR increased when using k=5 compared to k=10. Thus, k=5 was well suited for discretize AIDS data. While for EL dataset, Naïve algorithm obtained the same average accuracy for k=10 and k=5 and outperformed BR by 99.98%. Moreover, discretize EL data by Naïve algorithm also yielded higher accuracy than non-discretize EL data. Thus, it was revealed that the best classification accuracy for EL dataset was generated by Naïve algorithm, and k=10 was well suited for cross validation of EL dataset since the highest prediction accuracy when k=10 was 100%.

Table 11. Summarize results of non-discretization and discretization

Discretization	**AIDS**		**EL**	
Technique	***K=5***	***K=10***	***K=5***	***K=10***
Naive algorithm	64.12	63.13	99.98	99.98
Boolean algorithm	75.93	72.11	91.79	76.42
Non – Discretize (Johnson)	72.75	75.32	97.60	97.74
Non - Discretize (GA)	72.75	75.32	97.62	97.86

C. Reduct and Rules Generation

The purpose of this process is to generate reduct and rules from each training set. Reduct was generated from discretized data using k=5 for AIDS dataset, since in previous process the average accuracy when k=5 is increased compared to k=10 for discretize AIDS data. Two reduct algorithms for rough set rules generation were used, Johnson and GA to compare the prediction accuracy for each fold.

The results of reduct for discretize AIDS data are shown in Table 12. Johnson and GA yielded the same prediction accuracy for each fold. Hence, the same average of accuracy for these two types of reduction technique. Non-discretize AIDS dataset (refer Table 3) also obtained the same prediction accuracy for both reduct algorithms.

Table 12. Number of reduct and rules for AIDS dataset - 5-fold

Discretization	Reduct Technique	K-Fold	Num of reduct	Num of rules	Prediction Accuracy (%)
Naive algorithm	*Johnson*	1	1	8	73.68
		2	1	8	63.16
		3	1	8	54.05
		4	1	8	70.27
		5	1	8	59.46
	GA	1	1	8	73.68
		2	1	8	63.16
		3	1	8	54.05
		4	1	8	70.27
		5	1	8	59.46
Boolean algorithm	*Johnson*	1	1	5	76.31
		2	1	5	76.31
		3	1	5	81.08
		4	1	5	72.97
		5	1	5	72.97
	GA	1	1	5	76.31
		2	1	5	76.31
		3	1	5	81.08
		4	1	5	72.97
		5	1	5	72.97

In terms of generated rules, AIDS data had 1 reduct and 8 rules for Naïve algorithm and 5 rules for BR. BR obtained higher prediction accuracy with lesser number of rules. Naïve produced lower accuracy with more number of rules. Both findings have their own advantages and drawbacks. Although BR gives the best accuracy, shorter rules generated may contribute to the loss of knowledge [36]. On the other hand, Naïve showed comparative performance towards BR with more number of rules. Therefore, rules generated from AIDS data that had been discretized by Naïve algorithm using k=5 were selected to be used in this research.

Whereas for EL dataset, rules were generated from discretize EL data using Naïve algorithm using k=10 because it produced higher accuracy compared to BR and non-discretize EL data. Both reduct algorithms also produced the same prediction accuracy for each fold as depicted in Table 13. However, for non-discretize EL dataset (refer to Table 4), GA obtained higher accuracy than Johnson with the small difference of only 0.12%. EL dataset had 3 reduct and generated 18 rules for each fold.

Table 13. Number of reduct and rules for EL dataset - 10-fold

Discretization	Reduct Technique	K-Fold	Num of reduct	Num of rules	Prediction Accuracy (%)
Naive algorithm	*Johnson*	1	3	18	100.00
		2	3	18	99.96
		3	3	18	100.00
		4	3	18	99.96
		5	3	18	99.91
		6	3	18	100.00
		7	3	18	99.96
		8	3	18	99.96
		9	3	18	100.00
		10	3	18	100.00
	GA	1	3	18	100.00
		2	3	18	99.96
		3	3	18	100.00
		4	3	18	99.96
		5	3	18	99.91
		6	3	18	100.00
		7	3	18	99.96
		8	3	18	99.96
		9	3	18	100.00
		10	3	18	100.00

Based on the results for reduct process, Johnson and GA produced the same prediction accuracy for each fold for both datasets. This pattern of result where Johnson produced the same accuracy as GA is the same as obtained by [25] and [48] in their research in which GA and Johnson produced the same accuracy for the same dataset. Moreover, both reduct algorithms also generated the same number of reduct and number of rules.

Then, the most significant and less significant were selected from the 8 rules for AIDS dataset and 18 rules for EL dataset that had been generated. According to [30], the most significant rules have the highest support value of Left-Hand-Side (LHS) support. Thus, in order to find the most and least significant rules that would be used to visualize SNA, comparison of high and low LHS support value for each fold was made as shown in Tables 14 and 15. For AIDS data, the highest LHS was 36 obtained by fold 4 and the lowest LHS was 11 obtained by folds 4 and 1. For EL data, the highest LHS was 10597 acquired by fold 1 and the lowest LHS was 13 acquired by folds 2 and 5.

Table 14. High and low LHS support value for AIDS dataset

Discretization	Reduct Technique	K-Fold	Num. of Reduct	Num. of Rules	High LHS support value	Low LHS support value
Naive algorithm	***Johnson***	1	1	8	33	11
		2	1	8	35	12
		3	1	8	35	12
		4	1	8	36	11
		5	1	8	29	13
	GA	1	1	8	33	11
		2	1	8	35	12
		3	1	8	35	12
		4	1	8	36	11
		5	1	8	29	13

Table 15. High and low LHS support value for EL dataset

Discretization	Reduct Technique	K-Fold	Num. of Reduct	Num. of Rules	High LHS support value	Low LHS support value
Naive algorithm	***Johnson***	1	3	18	10597	17
		2	3	18	10396	13
		3	3	18	10400	15
		4	3	18	10466	17
		5	3	18	10474	13
		6	3	18	10580	14
		7	3	18	10497	16
		8	3	18	10564	15
		9	3	18	10540	17
		10	3	18	10480	16
	GA	1	3	18	10597	17
		2	3	18	10396	13
		3	3	18	10400	15
		4	3	18	10466	17
		5	3	18	10474	13
		6	3	18	10580	14
		7	3	18	10497	16
		8	3	18	10564	15
		9	3	18	10540	17
		10	3	18	10480	16

D. Rules Derivation

Tables 16 and 17 demonstrate the sample of rules derivation from AIDS and EL datasets, respectively. AIDS dataset consists of eight rules while EL dataset consists of eighteen rules. For AIDS dataset, support value of LHS showed the total number of support including VALUE(1) and VALUE(0), while RHS showed the number of support for each VALUE(1) or VALUE(0) separately. The generated rule of RESPONSE ([*, 10)) => VALUE(1) OR VALUE(0) was considered as the most significant rule. The rule was supported by 36 support values of LHS and 33 support values of RHS for VALUE(1) and 3 support values of RHS for VALUE(0). The RHS support values had two different values, depending on the numbers of records in the training dataset described by the THEN condition; VALUE (0) or VALUE (1). The RHS stability and LHS length was equal to one for all rules. There were two groups of rules for RHS length which are rules of length less than or equal to 1 and greater than 1. According to Sulaiman (2011), rules with length of greater than 1 contribute to better classification compared to rules of length less than or equal to 1.

The most significant rules based on high support value are often considered as the rule to query the dataset. Nonetheless, in this scenario, generated rule of RESPONSE ([*, 10)) => VALUE(1) OR VALUE(0) could not be considered as the most significant rule for query statement. This is because the rule had an infinite (*) value for the 'from, including' value. Rule of 'from * (including *)' is not valid to be used as a query statement. Therefore, other rules with high support value were chosen to be used as query statement to cluster the dataset.

Table 16. Sample rules of AIDS dataset

Rules	LHS Support	RHS Support	RHS Accuracy	LHS Coverage	RHS Coverage	RHS Stability	LHS Length	RHS Length
RESPONSE([143, *)) => VALUE(1) OR VALUE(0)	27	10, 17	0.37037, 0.62963	0.181208	0.185185, 0.178947	1.0,1.0	1	2
RESPONSE([*, 10)) => VALUE(1) OR VALUE(0)	36	27, 9		0.221477	0.185185, 0.242105	1.0, 1,0	1	2
RESPONSE([96, 126)) => VALUE(1) OR VALUE(0)	18	9,9	0.5,0.5	0.120805	0.166667, 0.094737	1.0, 1.0	1	2
RESPONSE([126, 138)) => VALUE(0)	13	13	1.0	0.087248	0.240741	1.0	1	1
RESPONSE([25, 55)) => VALUE(0)	13	13	1.0	0.087248	0.136842	1.0, 1.0	1	1
RESPONSE([10, 25)) => VALUE(1) OR VALUE (0)	21	20,1	0.952381, 0.047619	0.14094	0.210526, 0.018519	1.0, 1.0	1	1
RESPONSE([55, 96)) => VALUE(1) OR VALUE(0)	11	8,3	0.727273, 0.272727	0.073826	0.084211, 0.055556	1.0, 1.0	1	2
RESPONSE([138, 143)) => VALUE(1) OR VALUE(0)	13	8,5	0.615385, 0.384615	0.087248	0.148148, 0.052632	1.0, 1.0	1	2

Tables 18 and 19 sorted the rules according to their support value. The higher the support value the more significant the rules.

From Table 18, RESPONSE([143, *)) => VALUE(1) OR VALUE(0) was considered as the top highest support value of LHS support. The first rule was supported by 36 support values of LHS and there were 27 support values of LHS for the second rule. Although the second rule contained infinite (*) value, the rule did not include * for instance, * was not included in 143 to *. Therefore,

the rule statement that would be used for query was considered as RESPONSE>=143 and RESPONSE<=143.

Table 17. Sample rules of EL dataset

Rule	LHS Support	RHS Support	RHS Accuracy	LHS Coverage	RHS Coverage	RHS Stability	LHS Length	RHS Length
NUM_OF_HITS([0.00132,*)) => CACHE(1)	2531	2531	1	0.121712	0.207204	1	1	1
SIZE([0.00008,*)) =>CACHE(1)	10480	10480	1	0.503967	0.857962	1	1	1
SIZE([0.00006, 0.00007)) AND NUM_OF_HITS([0.00044, 0.00132)) => CACHE(1)	143	143	1	0.006877	0.011707	1	2	1
SIZE([0.00001, 0.00002)) AND NUM_OF_HITS([0.00044, 0.00132)) => CACHE(0)	415	415	1	0.019957	0.048368	1	2	1
SIZE([*, 0.00001)) AND NUM_OF_HITS([*, 0.00044)) => CACHE(0)	2790	2790	1	0.134167	0.325175	1	2	1
SIZE([0.00001, 0.00002)) AND NUM_OF_HITS([*, 0.00044)) => CACHE(0)	2416	2416	1	0.116182	0.281585	1	2	1
SIZE([0.00002, 0.00003)) AND NUM_OF_HITS([*, 0.00044)) => CACHE(0)	785	785	1	0.037749	0.091492	1	2	1
SIZE([0.00007, 0.00008)) AND NUM_OF_HITS([*, 0.00044)) => CACHE(0) OR CACHE(1)	131	131	0.954198, 0.045802	0.0063	0.014569, 0.000491	1, 1	2	2
SIZE([0.00002, 0.00003)) AND NUM_OF_HITS([0.00044, 0.00132)) => CACHE(0)	185	185	1	0.008896	0.021562	1	2	1
SIZE([0.00006, 0.00007)) AND NUM_OF_HITS([*, 0.00044)) => CACHE(0)	658	658	1	0.031642	0.07669	1	2	1
SIZE([0.00007, 0.00008)) AND NUM_OF_HITS([0.00044, 0.00132)) => CACHE(1)	27	27	1	0.001298	0.00221	1	2	1
SIZE([0.00005, 0.00006)) AND NUM_OF_HITS([0.00044, 0.00132)) => CACHE(1)	68	68	1	0.00327	0.005567	1	2	1

Rule	LHS Support	RHS Support	RHS Accuracy	LHS Coverage	RHS Coverage	RHS Stability	LHS Length	RHS Length
SIZE([0.00005, 0.00006)) AND NUM_OF_HITS([*, 0.00044)) => CACHE(0)	278	278	1	0.013369	0.032401	1	2	1
SIZE([0.00003, 0.00004)) AND NUM_OF_HITS([*, 0.00044)) => CACHE(0)	359	359	1	0.017264	0.041841	1	2	1
SIZE([*, 0.00001)) AND NUM_OF_HITS([0.00044, 0.00132)) => CACHE(0)	389	389	1	0.018706	0.045338	1	2	1
SIZE([0.00003, 0.00004)) AND NUM_OF_HITS([0.00044, 0.00132)) => CACHE(0)	116	116	1	0.005578	0.01352	1	2	1
SIZE([0.00004, 0.00005)) AND NUM_OF_HITS([*, 0.00044)) => CACHE(0)	64	64	1	0.003078	0.007459	1	2	1
SIZE([0.00004, 0.00005)) AND NUM_OF_HITS([0.00044, 0.00132)) => CACHE(1)	16	16	1	0.000769	0.00131	1	2	1

The rule with the lowest support value was also selected to be used to cluster the dataset. This was to determine the relationship between rules and LHS support value in visualization. Table 18 presents the generated rule of RESPONSE([55, 96)) => VALUE(0), which was considered as rule with less support value as it was only supported by 11 support values of LHS.

Table 18. Sorted highest rules support values for AIDS dataset

Rule	LHS Support	RHS Support
RESPONSE([*, 10)) => VALUE(1) OR VALUE(0)	36	33, 3
RESPONSE([143, *)) => VALUE(1) OR VALUE(0)	27	10, 17
RESPONSE([10, 25)) => VALUE(1) OR VALUE(0)	21	20, 1
RESPONSE([96, 126)) => VALUE(1) OR VALUE(0)	18	9, 9
RESPONSE([22, 55)) => VALUE(1)	13	13
RESPONSE([138, 143)) => VALUE(1) OR VALUE(0)	13	8, 5
RESPONSE([126, 138)) => VALUE(0)	13	13
RESPONSE([55, 96)) => VALUE(1) OR VALUE(0)	11	8, 3

For EL dataset, the generated rule of SIZE ([0.00008,*)) =>CACHE(1) was considered as the most significant rule. The rule was supported by 10480 support values for RHS and LHS. LHS and RHS support affected the total of LHS and RHS coverage. The RHS accuracy and stability were equal to one for all rules. On the other hand, the rule with the highest value of LHS and RHS support also obtained the highest value of coverage. Despite that, the same value of LHS and RHS support did not produce the same value of RHS and LHS coverage. The highest coverage for LHS was 0.503967 and 0.857962 for RHS.

The most significant rules based on high support value will be considered as the rule to query the dataset. Nonetheless, in this scenario, generated rule of SIZE ([0.00008,*)) =>CACHE(1) could not be used as the query statement. Although the rule statement did not include *, yet the dataset consisted of three reduct. Thus, attributes of NUM_OF_HITS were also needed to be used in the

query statement. Another rule with high support value was chosen to be used as query statement to cluster the dataset.

Based on Table 19,

SIZE([0.00001, 0.00002)) AND NUM_OF_HITS([0.00044, 0.00132)) => CACHE(0)

was considered as the rule with the highest support value of LHS support with the outcome of no cache (output=0). This rule was supported by 415 of LHS support value. Generated rule of,

SIZE([0.00007, 0.00008)) AND NUM_OF_HITS([0.00044, 0.00132)) => CACHE(1)

was selected as the rule with less support value of LHS. This rule was only supported by 27 support values of LHS support.

Table 19. Sorted highest rules support values for EL dataset

Rule	LHS Support	RHS Support
SIZE([0.00008,*)) =>CACHE(1)	10480	10480
SIZE([*, 0.00001)) AND NUM_OF_HITS([*, 0.00044)) => CACHE(0)	2790	2790
NUM_OF_HITS([0.00132,*)) => CACHE(1)	2531	2531
SIZE([0.00001, 0.00002)) AND NUM_OF_HITS([*, 0.00044)) => CACHE(0)	2416	2416
SIZE([0.00002, 0.00003)) AND NUM_OF_HITS([*, 0.00044)) => CACHE(0)	785	785
SIZE([0.00006, 0.00007)) AND NUM_OF_HITS([*, 0.00044)) => CACHE(0)	658	658
SIZE([0.00001, 0.00002)) AND NUM_OF_HITS([0.00044, 0.00132)) => CACHE(0)	415	415
SIZE([*, 0.00001)) AND NUM_OF_HITS([0.00044, 0.00132)) => CACHE(0)	389	389
SIZE([0.00003, 0.00004)) AND NUM_OF_HITS([*, 0.00044)) => CACHE(0)	359	359
SIZE([0.00005, 0.00006)) AND NUM_OF_HITS([*, 0.00044)) => CACHE(0)	278	278
SIZE([0.00002, 0.00003)) AND NUM_OF_HITS([0.00044, 0.00132)) => CACHE(0)	185	185
SIZE([0.00006, 0.00007)) AND NUM_OF_HITS([0.00044, 0.00132)) => CACHE(1)	143	143
SIZE([0.00007, 0.00008)) AND NUM_OF_HITS([*, 0.00044)) => CACHE(0) OR CACHE(1)	131	131
SIZE([0.00003, 0.00004)) AND NUM_OF_HITS([0.00044, 0.00132)) => CACHE(0)	116	116
SIZE([0.00005, 0.00006)) AND NUM_OF_HITS([0.00044, 0.00132)) => CACHE(1)	68	68
SIZE([0.00004, 0.00005)) AND NUM_OF_HITS([*, 0.00044)) => CACHE(0)	64	64
SIZE([0.00007, 0.00008)) AND NUM_OF_HITS([0.00044, 0.00132)) => CACHE(1)	27	27
SIZE([0.00004, 0.00005)) AND NUM_OF_HITS([0.00044, 0.00132)) => CACHE(1)	16	16

V. Conclusion

This paper discusses the analysis and experimental results of Rough set. The generated rules of Rough set were analyzed based on high LHS support value to identify the significant rules. Selected significant rules were used to query the dataset. Furthermore, rules with low LHS support value also will be selected to be used in SNA part in order to compare the visualization of data based on rules with high and low LHS support values.

Some limitations of this research that may serve as a guide for future work. Instead of only using k-fold cross validation and various discretization algorithms, analysis of various percent of training and testing should be done to compare the better result of classification. Different percent of training and testing and different discretization techniques contribute to different results of accuracy classification. Moreover, the available literature in RS opens a promising domain towards future research and more intensive experiments in other complex area such as big data analysis.

Acknowledgement

This work is supported by Ministry of Higher Education Malaysia (MOHE), Ministry of Science, Technology and Innovation Malaysia (MOSTI) and Universiti Teknologi Malaysia (UTM). This paper is financially supported by UTM Flagship Grant Q.J130000.2428.02G70, FRGS Grant, R.J130000.7828.4F634, E-Science Fund, R.J130000.7928.4S117, UTM IDG, R.J130000.7728.4J170, UTM GUP Tier 1, Q.J130000.2528.13H48 and PRGS, R.J130000.7828.4L680. The authors would like to express their deepest gratitude to the Research Management Centre (RMC), UTM for the support in research and development, and Soft

Computing Research Group (SCRG) for the inspiration in making this research a success.

References

[1] D. Hand, *Principles of Data Mining*, vol. 2001. 2001.

[2] A. Anitha and N. Krishnan, "A Dynamic Web Mining Framework for E-Learning Recommendations using Rough Sets and Association Rule Mining," vol. 12, no. 11, pp. 36–41, 2011.

[3] S. Sulaiman, S. M. Shamsuddin, and A. Abraham, "Meaningless to Meaningful Web Log Data for Generation of Web Pre-caching Decision Rules Using Rough Set," vol. 1, no. September, pp. 2–4, 2012.

[4] P. Mahajan, "Rough Set Approach in Machine Learning : A Review," vol. 56, no. 10, pp. 1–13, 2012.

[5] Z. Pawlak, "Rough Sets," *Int. J. Comput. Inf. Sci.*, pp. 1–51, 1982.

[6] M. Kumar and N. Yadav, "Fuzzy Rough Sets and Its Application in Data Mining Field," vol. 2, no. 3, pp. 237–240, 2015.

[7] N. S. Sulaiman and S. M. Shamsuddin, "Feature granularity for cardiac datasets using Rough Set," *2011 IEEE Int. Conf. Comput. Sci. Autom. Eng.*, pp. 346–352, Jun. 2011.

[8] L. K. Dqj *et al.*, "An evolutionary approach for solving the job shop scheduling problem in a service industry," *Int. J. Adv. Intell. Informatics*, vol. 1, no. 1, pp. 1–6, Apr. 2014.

[9] A. Mitra, "Clustering Analysis in Social Network using Rough set and Soft set," *Univers. J. Appl. Comput. Sci. Technol.*, vol. 2, no. 2, pp. 282–285, 2012.

[10] S. Sulaiman, S. M. Shamsuddin, and A. Abraham, "An Implementation of Rough Set in Optimizing Mobile Web Caching Performance (Invited Paper)," *Tenth Int. Conf. Comput. Model. Simul. (uksim 2008)*, pp. 655–660, 2008.

[11] S. Sulaiman, S. M. Shamsuddin, and A. Abraham, "Rough Neuro-PSO Web caching and XML prefetching for accessing Facebook from mobile environment," *2009 World Congr. Nat. Biol. Inspired Comput.*, pp. 884–889, 2009.

[12] J. L. Breault, "Data Mining Diabetic Databases : Are Rough Sets a Useful Addition ?," 2001.

[13] Z. Pawlak, "Rough sets," *Int. J. Comput. Inf. Sci.*, vol. 11, no. 5, pp. 341–356, Oct. 1982.

[14] D. Srivastava, S. Batra, and S. Bhalothia, "Efficient Rule Set Generation using K-Map & Rough Set Theory (RST)," vol. 2, no. 3, pp. 6–10, 2015.

[15] S. K. De and P. R. Krishna, "Clustering web transactions using rough approximation," *Fuzzy Sets Syst.*, vol. 148, no. 1, pp. 131–138, Nov. 2004.

[16] I. Tri, R. Yanto, T. Herawan, and M. M. Deris, "A Framework of Rough Clustering for Web," pp. 265–277, 2010.

[17] C. L. Ngo and H. S. Nguyen, "A Method of Web Search Result Clustering Based on Rough Sets," *2005 IEEE/WIC/ACM Int. Conf. Web Intell.*, pp. 673–679, 2005.

[18] S. Karthik, A. Priyadarishini, J. Anuradha, and B. K. Tripathy, "Classification and Rule Extraction using Rough Set for Diagnosis of Liver Disease and its Types," vol. 2, no. 3, pp. 334–345, 2011.

[19] A. Szyma and A. W. Przybyszewski, "Rough Set Rules Help to Optimize Parameters of Deep Brain Stimulation in Parkinson â€™s Patients," pp. 345–356, 2014.

[20] X. Yu, L. Su, and P. Gou, "Study on Knowledge Discovery for Lifestyle Diseases Using Rough Set," *2013 6th Int. Conf. Intell. Networks Intell. Syst.*, pp. 13–16, Nov. 2013.

[21] Z. Akbar, "Marketing data classification using Johnson's algorithm.pdf." 2003.

[22] M. Reichle, P. Perner, and K. Althoff, "Data Preparation of Web Log Files for Marketing," pp. 131–145, 2006.

[23] L. Shen and S. Chen, "Research of Customer Classification Based on Rough Set Using Rosetta Software," pp. 837–843, 2013.

[24] a Ohrn and T. Rowland, "Rough sets: a knowledge discovery technique for multifactorial medical outcomes.," *Am. J. Phys. Med. Rehabil.*, vol. 79, no. 1, pp. 100–108.

[25] A. K. Muda, "AUTHORSHIP INVARIANCENESS FOR WRITER," 2009.

[26] Z. Liu, "A New Heuristic Algorithm of Rules Generation Based on Rough Sets," *2008 Int. Semin. Bus. Inf. Manag.*, no. 3, pp. 291–294, Dec. 2008.

[27] A. H. Liang, B. Maguire, and J. Johnson, "Rough Set Based WebCT Learning," pp. 425–436, 2000.

[28] D. L. D. Olson and D. Delen, *Advanced data mining techniques*. 2008.

[29] N. S. Sulaiman, "Generation of Rough Set (RS) Significant Reducts and Rules for Cardiac Dataset Classification," Universiti Teknologi Malaysia, 2007.

[30] M. Omar, S.-L. Syed-Abdullah, and N. M. Hussin, "Developing a Team Performance Prediction Model: A Rough Sets Approach," *Informatics Eng. Inf. Sci.*, pp. 691–705, 2011.

[31] M. N. A. Rahman, Y. M. Lazim, F. Mohamed, S. I. A. Saany, and M. K. M. Yusof, "Rules Generation for Multimedia Data Classifying using Rough Sets Theory," vol. 6, no. 5, pp. 209–218, 2013.

[32] H. S. Own and H. Yahyaoui, "Rough set based classification of real world Web services," *Inf. Syst. Front.*, no. May 2014, pp. 1301–1311, 2014.

[33] Phillips, Elizabeth, N. J. R.C., M. Goldsmith, and S. Creese, "Applying Social Network Analysis to Security," *Int. Conf. Cyber Secur. Sustain. Soc.*, no. June, pp. 11–27, 2015.

[34] C. Xiao, "Using Machine Learning for Exploratory Data Analysis and Predictive Models on Large Datasets," UNIVERSITY OF STAVANGER, 2015.

[35] T. R. Hvidsten, "A tutorial-based guide to the ROSETTA system : A Rough Set Toolkit for Analysis of Data," no. October, 2013.

[36] N. Shuib, A. Bakar, and Z. Othman, "Performance Study on Data Discretization Techniques Using Nutrition Dataset," *Int. Symp. Comput. Commun. Control*, vol. 1, no. Isccc 2009, pp. 304–308, 2011.

[37] I. Bose, "Deciding the financial health of dot-coms using rough sets," *Inf. Manag.*, vol. 43, no. 7, pp. 835–846, 2006.

[38] J. Li and N. Cercone, "Empirical Analysis on the Geriatric Care Data Set Using Rough Sets Theory," 2005.

[39] R. Kohavi, "A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection," *Int. Jt. Conf. Artif. Intell.*, vol. 14, no. 12, pp. 1137–1143, 1995.

[40] Z. Marzuki and F. Ahmad, "Data Mining Discretization Methods and Performances," *Mach. Learn.*, no. 1, pp. 978–980, 2007.

[41] J. Jiang, "System model of college students' network behavior research based on rough sets," *J. Chem. Pharm. Res.*, vol. 6, no. 7, pp. 2264–2270, 2014.

[42] V. Brtka, I. Berkovic, E. Brtka, and V. Jevtic, "A comparison of rule sets induced by techniques based on rough set theory," *2008 6th Int. Symp. Intell. Syst. Informatics*, no. 3, pp. 1–4, Sep. 2008.

[43] Y. Chen, D. Miao, and R. Wang, "A rough set approach to feature selection based on ant colony optimization," *Pattern Recognit. Lett.*, vol. 31, no. 3, pp. 226–233, Feb. 2010.

[44] H. I. Elshazly, N. I. Ghali, A. M. El Korany, and A. E. Hassanien, "Rough Sets and Genetic Algorithms: A hybrid approach to breast cancer classification," pp. 260–265, 2012.

[45] Q. A. Al-radaideh, M. N. Sulaiman, M. H. Selamat, and H. Ibrahim, "an Empirical Comparison of Reduct Generation Approaches in the Context of Rough Set Based Classification," 2003.

[46] M. Durairaj and T. Sathyavathi, "Applying Rough Set Theory for Medical Informatics Data Analysis," no. 5, pp. 1–8, 2013.

[47] A. Lebbe, S. Saabith, E. Sundararajan, and A. A. Bakar, "Comparative Study on Different Classification Techniques for Breast Cancer Dataset," vol. 3, no. 10, pp. 185–191, 2014.

[48] N. S. Jaddi and S. Abdullah, "Hybrid of genetic algorithm and great deluge algorithm for rough set attribute," pp. 1737–1750, 2013.