**CHAPTER 1**

**INTRODUCTION**

The amount of data stored into database are increased day by day in many organizations. The datasets are so complex and large in volume. And the attributes contained in datasets are so many and complex. So, it is hard to know which attributes are important in dataset to help decision making process and it becomes difficult to get useful and accurate knowledge for supporting decision to users. Data mining processes find out the hidden information which cannot be found by the analyst. And then it can find the relation among data in large dataset. There are many data mining tools to search for and extract useful and important information from large dataset.

The useful information which is hidden in a database cannot be seen easily especially in a large amount of data. This information can be shown as rule form to understand its patterns more easily and to help decision makers in problem solving and decision making. Rule forms are specified with two parts; one for condition part and one for decision part. Data mining application generates rules to understand data and its patterns more easily. Generated rules are also called as knowledge or useful information since they have meanings. Computer scientists and engineers have a new challenge to discover meaningful information automatically. It is hard for a human to generate meaningful rules manually since it takes a lot of time. All rules may not be meaningful and useful so they need to be eliminated. This is a process that a human can never do without the help of computer science and technology. In order to overcome these problems, Rough Set Theory (RST) was developed by Pawlak in 1980.

Rough set theory is a new mathematical tool in data mining to discover knowledge and support decision making. It has many stages to remove inconsistent data, search important and useful data, and directly generate rules from dataset. Heart disease dataset is used to generate rule set by rough set theory in this book. Inconsistent data may contain in dataset. It means that the objects which are the same value in all condition attributes are in different decision classes. These data may cause generated rules to have bad quality and lower accuracy. Upper and lower approximations are used to handle these inconsistent data in rough set theory.

Dataset is split into two parts; training dataset and testing dataset. Training part is used to generate rules and testing part is used to test these generated rules. The original dataset are randomly divided into 60%, 70%, 80%, and 90% of the data for training dataset and 40%, 30%, 20% and 10% of the data for testing dataset respectively with random number user chose. The splits are used to avoid results based on rules that were generated for a particular selection of objects. This makes the results more reliable and independent on one particular selection of objects. Finally, the system will choose one with the best accuracy among the rule sets getting from four split.

Rules can be generated directly from reducts in rough set theory. Reduct can be defined as the information obtained by omitting or neglecting the unwanted data from the dataset. The main objective of attribute reduction algorithm is to get reducts from the information system by eliminating redundant information and take minimum time and space complexity. The need of finding reducts is to remove redundancy and incompatible attributes to obtain key information and make decision rules. If the unimportant attributes are not removed, not only the time and space complexity of rule discovery increases but also the quality of the rules discovered degrades. The cost of finding the reducts depends on the size of the object and attribute sets.

The strength is calculated for each rule by using testing dataset to test the rule sets. It simply expresses the ratio of all facts which can be classified by the decision rule to all facts in the data table. The predicted decision is assigned for each object in testing dataset by using rule strength. If the rules that satisfy all the condition parts of the object in the testing part are more than one in rule sets, the decision part of rule which has maximum strength is used to assign as the predicted decision of these objects in testing part. And then the accuracy of rule set is calculated to know how many objects can be correctly assigned predicted decision as the actual decision. The calculation of accuracy of rule set is the ratio of the total number of supported objects to the number of all object in testing dataset. Rule set can be generated for each reduction of attributes by one split of dataset with one random number. Many rule sets may be induced with each of their accuracies. Finally, rule set with maximum accuracy is used to support decision making.

**1.1 Basic Concept of Theory**

In RST, datasets are formatted as a table. Each row in table represents an object and each column in table represents attributes of that object. This table is called an information system IS = (U, A), where the set U is set of objects and the set A is set of attributes.

Indiscernibility Relation is the main concept of rough set theory, and is considered as a same relation among objects in dataset, where all the values are identical in relation to a subset of considered attributes. Indiscernibility relation IND (A) is an equivalence relation.

Lower approximation consists of all objects surely belong to set and upper approximation consists of objects that probably belong to set while the boundary is a difference of upper and lower approximations. If the boundary exists, it may be considered that there is inconsistent data in the original dataset. These inconsistent data must be removed to get optimal rule set.

Approximation of sets are related to concept of dependency (total or partial) of attributes. A set of decision attributes depends totally on a set of condition attributes, if all values of decision attributes are determined by values of condition attributes. Otherwise, decision attributes depends partially on condition attributes.

Discernibility relation is a Boolean function constructed for each equivalence class. This function is true for all attributes’ combinations that discern this object from other objects with a different decision. It takes set of attributes from each object which has different values from other object in different decision.

Another important process in rough set theory is data reduction. Reduct is the minimal subset of condition attributes that is important and useful to classify decision class as the whole set of attributes. A set of condition attributes may have more than one reduct. Intersection of all reducts is called the core. The core is the set of attributes that cannot be removed from the dataset. JOHNSON’s algorithm is used for reduction of attributes. Johnson’s algorithm always select the attribute most frequently occurring in the clause. Rules are generated directly from reduct and these rules are defined as classifiers.

**1.2 Decision Rules**

Information table has two classes of attributes, condition and decision attributes, such table is called decision table. Each row of a decision table is considered as a decision rule, which specifies decision (actions) that should be taken when conditions pointed out by condition attributes are satisfied. Decision rules can be described as “if…, then…” that is human readable form.

**1.3 Objectives of the Thesis**

The objectives of the thesis are as follows:

* To get knowledge about data mining and its methods
* To understand directly rule generation from reduction of features of dataset
* To study background knowledge of Rough Set theory that handles inconsistent objects
* To know how to generate suggested rules by using JOHNSON’s Algorithm
* To suggest set of rules with maximum accuracy to decision maker about Heart Disease Dataset
* To be able to mine the important and minimal information that is meaningful and useful rules from large data
* To find hidden patterns and get knowledge from large databases and vast data

**1.4 Preview of the Thesis**

Introduction of data mining, rough set theory and objectives are presented in Chapter 1. Chapter 2 presents the background theory of Data Mining, Rough set and JOHNSON’s algorithm that makes the dataset to get suggested rules by reducing features and values of the dataset. In Chapter 3, system design and dataset are described. Chapter 4 includes the system implementation of Rule Generation of Heart Disease Dataset. Chapter 5 will present conclusion and limitations of the system.

**CHAPTER 2**

**THEORETICAL BACKGROUND OF THE SYSTEM**

The main purpose of this chapter is to describe the related work and to present the theory background of Data mining, Rough Set Theory and JOHNSON’s algorithm.

**2.1 Decision Support System**

Decision Support Systems (DSSs) are defined as interactive computer-based intended systems to help decision makers utilize data and models in order to identify and solve problems and make decisions. Some key words associated with DSSs are Decision Theory, Decision Analysis, Operation Research, Management Science and Artificial Intelligence (AI). However, the main problem of DSSs is to find ways of extracting useful information, called decision rules. These rules are something people using for decision making, DSSs [3] also, include a knowledge management component which stores and manages a new class of emerging AI tools such as machine learning and case-based reasoning. Machine learning refers to computational methods/tools of a computer system to learn from experience, data and observations, and consequently alter its behavior, triggered by a modification to the stored knowledge. Therefore, rough set is used as a mathematical tool for knowledge discovery on decision support system. J.Blaszczynski et al. [8], introduce a new version of the rule induction system. This system can be used to select decision rules that can be stored on knowledge base under the control of knowledge management system.

Decision support systems (DSSs) are prevalent information systems for decision making in many competitive business environments. In a DSS, decision making process is intimately related to some factors which determine the quality of information systems and their related products. Traditional approaches to data analysis cannot usually be implemented in sophisticated companies, where managers need some DSS tools for rapid decision making. In traditional approaches to decision making, scientific expertise together with statistical techniques are needed to support the managers. However, these approaches are not able to handle the huge amount of real data, and the processes are usually very slow. Recently, several innovative facilities have been presented for decision making process in enterprises. Presenting new techniques for development of huge databases, together with some heuristic models have enhanced the capabilities of DSSs to support managers in all levels of organizations.

Today, data mining and knowledge discovery is considered as the main module of development of advanced DSSs. In this paper, rough set theory is used for decision making process in a DSS.

In modern decision support systems, data mining is the most prevalent tool used to search for and extract useful information from volumes of data and for combining databases, artificial intelligence, and machine learning. In other words, data mining is a powerful tool for nontrivial extraction of implicit, previously unknown, and potentially useful information from large datasets. Obviously, a fairly large classes of data mining tasks can be described as search for interesting and frequently occurring patterns or rules from databases. This technology uses machine learning, statistical and visualization techniques to discover and present knowledge in a form which is easily comprehensible to humans [9]. Data mining is used in a process called Knowledge discovery in databases (KDD). The discovered knowledge can be rules describing properties of data or relationships among data, frequently occurring patterns, clustering of the objects in the dataset, etc. Most data mining systems in use have been designed using variants of traditional machine learning techniques [7].

Rough set theory is a powerful tool for data mining. This approach can be implemented for: a) Reduction of datasets; b) Finding hidden data patterns; and c) Generation of decision rules [1]. Rough set theory is a relatively new mathematical and artificial intelligence (AI) technique introduced in the early 1980’s by Pawlak. The technique is particularly suited to reasoning about imprecise or incomplete data, and discovering relationships among them. The main advantage of rough set theory is that it does not require any preliminary or additional information about data-like probability in statistics, or the value of possibility in fuzzy set theory. Recently, there has been a growing interest in rough set theory among researchers in modern intelligent information systems. The theory has found many real life applications in many areas. The primary applications of rough sets so far have been in data and decision analysis, databases, knowledge-based systems, and machine learning [4]. The concentration of this paper is on the application of rough set theory in data mining.

**2.2 Data Mining and Knowledge Discovery in Databases**

Data Mining is a discovery process of the hidden information from data which is yet unknown and potentially useful. On the other hand, according to Raghavan and Sever, data mining discovers the general patterns and relations hidden in the data (Sever and Oguz, 2003). Decision rules are one of the widely used techniques to present the obtained information. A decision rule summarizes the relation between the properties of the data. To transform the raw data residing in the database into valuable information, several stages of data processing is required. Data mining is an iterative process that acts as a bridge between logical decision making and the data, and is possible the classification for finding the useful samples or using and combining the classification rules from the samples. This process combines the approaches used in different disciplines like machine learning, statistics, database management systems, data warehousing, and constraint programming (Sever and Oguz, 2003).

In recent years, many successful machine learning applications have been developed, in particular in domain of data mining and knowledge discovery. One of the common tasks performed in knowledge discovery is classification. It consists of assigning a decision class label to a set of unclassified objects described by a fixed set of attributes (features). Learning algorithms induce various forms of classifiers. Decision rules are represented as logical expressions of the following form:

IF (conditions) THEN (decision class)

where conditions are formed as a conjunction of elementary tests on values of attributes. A number of various algorithms have already been developed to induce such rules. Decision rules are one of the most popular types of knowledge used in practice; one of the main reasons for their wide application is their expressive and easily human-readable representation. (Stefanowski, 2003).

There are many successful applications of data mining process in many different areas. Many methods to discover the useful patterns are available in data mining applications and each method has advantages and disadvantages over the others. However, when needed, the advantages of different methods can be combined and hybrid methods may be created. The process of creating hybrid methods is a work of combining computational intelligent tools. Many algorithms are used to implement a Decision Making(DM) process. The reason is that some technologies’ results are better than the others for different tasks, states and subjects do. There is a model creation process that represents a dataset in the core of the data mining. A model creation process that represents a dataset is generic for all DM products; on the other hand, the process itself is not generic.

Some methods used in DM processes are rough set theory, Bayesian networks, genetic algorithms, neural networks, fuzzy sets and inductive logic programming. DM functions are used to determine the pattern types that may exist in the DM tasks. Generally, DM tasks are classified into two categories as descriptive mining and estimator mining. Descriptive mining tasks characterize the general properties of data in the database. On the other hand, estimator mining makes inferences from the available data to make predictions (Han and Kamber, 2001). The samples of DM functions and resulting discovered pattern types are classification, clustering, summarization, estimation, time series analysis, association rules, sequence analysis and visualization.

Data mining and usage of the useful patterns that reside in the databases have become a very important research area because of the rapid developments in both computer hardware and software industries. In parallel with the rapid increase in the data stored in the databases, effective use of the data is becoming a problem. To discover the rules or interesting and useful patterns from these stored data, data mining techniques are used. If data is incomplete or inaccurate, the results extracted from the database during the data discovery phase would be inconsistent and meaningless. Rough set theory is used in the intelligent data analysis and data mining if data is uncertain or incomplete. This approach is of great importance in cognitive science and artificial intelligence, especially in machine learning, decision analysis, expert systems and inductive reasoning.

There are many advantages of rough set approach in intelligent data analysis. Some of these advantages are being suitable for parallel processing, finding minimal datasets, supplying effective algorithms to discover hidden patterns in data, valuation of the meaningfulness of the data, producing decision rule set from data, being easy to understand and the results obtained can be interpreted clearly.

In the last decades, the size of the data stored in the databases of the organizations has been growing each day and therefore users face difficulties about obtaining the valuable data. Databases are a collection of relational and non-recurring data to meet the demands of the organizations. Because the data stored in the databases is growing each day, it is getting harder for the users to reach the accurate and useful information. In the last few years, because of the rapid developments in both computer hardware and software industries, the increase in the storage capacities of huge databases, the data mining and the usage of the useful patterns that reside in the databases, became a very important research area.

To discover the rules or interesting and useful patterns among these stored data in the databases, data mining techniques are used. Storing huge amount of increasing data in the databases, which is called information explosion, it is necessary to transform these data into necessary and useful information. Using conventional statistics techniques fail to satisfy the requirements for analyzing the data, in the last years, the newly developed concepts Data Mining and Knowledge Discovery in Databases are getting more important.

One of the approaches used in data mining and knowledge discovery is rough set theory. According to this method, knowledge can be obtained from every object in the universe. In this study, the mathematical principles of rough set theory are explained and rule discovery from a decision table of heart disease dataset by using JOHNSON’s algorithm in rough set theory is presented.

**2.3 Theory Background**

Rough set theory, a new mathematical tool developed by Pawlak in 1980s [1], [2], is powerful tool among data mining methods and used to reduce unnecessary attributes and values from datasets, find important and useful data as reduct and generate decision rules directly from reduct. Some of the applications in which the rough set theory is efficiently employed are in the areas of medicine, social networking, aerospace engineering, market analysis etc. The main advantage of rough set theory is that it does not need any preliminary information about data.

Pawlak (2002) listed the advantages of RST as shown in below [11];

* RST provides an efficient algorithm to find patterns hidden in database.
* It works with minimum sets (Data Reduction).
* Evaluates importance of data.
* Generates decision information.
* Presents a simple evaluation of results.
* Easy to understand.

As it is shown in (Grzymala-Bussc 1988), rough set theory represents an objective approach to imperfections in data, all computations are performed directly on datasets, i.e., no feedback from additional experts is necessary. There is no need for any additional information about data, for example, a probability distribution function in statistics, a grade of membership from fuzzy set theory, etc. (Grzymala-Bussc, 1988). In general, rough set theory may be applied for consistent data (without conflicting cases) to study relations between attributes. For example, in this way, it may be eliminated redundant attributes. Another typical task is to find a minimal subset of the attribute set that may be used to identify all concepts. Yet, another task is to compute a family set of attribute-value pairs for the same reason: to identify all concepts.

Inconsistent datasets are handled by rough set theory using lower and upper approximations for every concept. These approximations are definable using existing attributes. In rough set theory, the approximations of sets are introduced to deal with inconsistency. Inconsistent objects are same in all value of any set of condition attributes, but they belong to different decision classes. The objects in lower approximation can be classified with certainty as on the basis of knowledge while the objects in upper approximation can be classified as the possible occurrence of decision class. The difference of lower and upper approximations is boundary region. If boundary region exists, there are inconsistent objects in dataset and then they have to be removed.

Rough set theory (RST) has been developed to manage uncertainties from information that presents some inexactitude, incompleteness and noises. When the available information is insufficient to determine the exact value of a given set, lower and upper approximations can be used by rough set for the representation of the concerned set. The approximation synthesis of concepts from the acquired data is the main objective of the rough set analysis. For example, if it is difficult to define a concept in a given knowledge base, rough set can ‘approximate’ with respect to that knowledge. In decision making, it has confirmed that rough set methods have a powerful essence in dealing with uncertainties.

The RST has been applied in several fields including image processing, data mining, pattern recognition, medical informatics, knowledge discovery and expert systems. In the current literature, several research works have been combined the rough set theory with other artificial intelligent methods such as neural networks, fuzzy logic, additionally to other methods resulting in some good results. The use of rough set theory to solve a specific complex problem has attracted world-wide attention of further research and development, extending the original theory and increasingly widening fields of application. Additionally, rough set is a computationally efficient technique. It presents a basic significance to many theoretical developments and practical applications of computing and automation, especially in the areas of machine learning and data mining, decision analysis and intelligent control. Among other computational problems, rough set addresses problems such as data significance evaluation, hidden pattern discovery from data, decision rule generation, data reduction and data-driven inference interpretation (Pawlak, 2004).

**2.3.1 Information Systems**

In rough set theory, a dataset is represented as a table and each row represents a state, an event or an object. Each column represents a measurable property for an object (a variable, an observation, etc.). This table is called an information system. More formally, the pair (*U,* ***A***) represents an information system. *U* is a finite nonempty set that is called universe and A is a finite nonempty set of properties. Here, for a ***A***, a: *U*. The set is called the value set of a. Another form of information systems is called decision systems. A decision system (i.e. decision table) expresses all the knowledge about the model. A decision system is (U, ***A*** {*d*}) form of any information system. Here, *d* ***A*** are decision attributes. Other attributes a ***A*** – {*d*} are called conditional attributes. Decision attributes can have many values, but usually they have a binary value like True or False (Komorowski, et.al., 1998, Hui, 2002).

**2.3.2 Indiscernibility Relation**

Indiscernibility Relation is a central concept in Rough Set Theory, and is considered as a relation between two objects or more, all the values are identical in all conditional attributes. Indiscernibility relation is an equivalence relation, where all identical objects of set are considered as elementary.

IND(C)={(x, y) ϵ U \* U : c(x) = c(y) , c ϵ C} (2.1)

where, IND(C) = Indiscernibility Relation

U = the set of objects

C = the set of condition attributes (patient’s symptoms)

c(x) = c(y) = the value of condition attributes in objects x and y.

**2.3.3 Approximation of Sets**

Suppose that an information system is S = (U, A), X U, and B A. The task is to describe the set X in terms of attribute values from B. To this end, two operations are defined by assigning to every X U two sets (X) and (X) called the B-lower and the B-upper approximation of X, respectively, and defined as follows:

(X) = (2.2)

(X) = (2.3)

Hence, the B-lower approximation of a set is the union of all B-granules that are included in the set, whereas the B-upper approximation of a set is the union of all B-granules that have a nonempty intersection with the set. The set

(X) = (X) - (X) (2.4)

will be referred to as the B-boundary region of X.

If the boundary region of X is the empty set, i.e., (X) =, then X is crisp (exact) with respect to B; in the opposite case, i.e., if (X) X is referred to as rough (inexact) with respect to B.

**2.3.4 Accuracy and Dependency of Attributes**

Accuracy is calculated for covering inconsistencies from dataset directly and dependency is for knowing the relation of condition attributes and decision attribute.

Accuracy= (X) / (X), {0 < accuracy < 1} (2.5)

If the boundary region is not empty and accuracy is less than 1.0, assume that there is inconsistent data in dataset. Otherwise, there is no inconsistent data. Approximations of sets are related to concept of dependency (total or partial) of attributes.

k = / |U| (2.6)

where ,|U| = total number of objects

n = number of decision class (1 or 0)

k = 1.0, decision attribute depends totally on condition attributes,

k< 1.0, decision attribute depends partially on condition attributes.

**2.3.5 Discernibility Relation**

Discernibility relation is used to calculate reducts. Reduct is a minimal subset of attributes which is more important than other attributes in dataset. After reduct generation, rules are automatically generated.

m(, ) ={c ϵ C| c() ≠ c() }, for i, j=1, 2, 3, …n

= ᴧ {ᴠ m(, ) | 1<= i <= j <= n, m(, ) ≠ Ø} (2.7)

where,

m(, ) = the set of attributes which have different value between two objects , (different decision)

= the discernibility function of all objects

n = the number of objects

C = set of condition attributes in dataset

c(), c() = value of each attribute for two objects ,

**2.3.6 Johnson’s Algorithm**

Johnson’s algorithm is a heuristic algorithm using a greedy technique. The idea of Johnson’s algorithm is that it always selects the attribute most frequently occurring in the clause. The algorithm is described as follows:

JOHNSON (V, f)

{

V: set of attributes

f: a discernibility function

S ← ∅; //current reduct candidate

while (f ≠ ∅)

{

bCount = 0; //max count so far

for each(x ∈ V)

{

/\*

\* count (x, f) returns number of x in f

\*/

c = count (x, f);

if (c > bCount)

{

bCount = c;

bAttr ←x;

}

}

S ←S ∪ bAttr;

/\*

\* removeAttr (f, bAttr) removes all clauses in f containing bAttr

\*/

f ←removeAttr (f, bAttr);

}

return S;

}

Johnson’s algorithm starts by setting S, the current reduct candidate, to an empty set. Then, the algorithm counts the appearance of each attribute within the clauses. The attribute with the highest count is added into S, and all clauses in f containing this attribute are removed from the discernibility function. The algorithm continues until all clauses are removed from the discernibility function, the algorithm then returns S as a reduct. [4]

For example, the steps of getting the reduct of f = (Led2 ∨ Led4 ∨ Led6) ∧ (Led1 ∨ Led2 ∨ Led4 ∨ Led6) ∧ (Led2 ∨ Led5 ∨ Led7) ∧ (Led5) are:

1. Count the appearance of the attributes, Led1 = 1, Led2 = 3, Led4 = 2, Led5 = 2, Led6 = 2, Led7 = 1.

2. Led2 is the most frequently occurring attribute, so it adds Led2 into S. The classifier removes all clauses containing Led2 from f. After this step, f = Led5 and S = {Led2}.

3. Count the appearance of attributes in f and it finds that Led5 is the most frequently occurring attribute. Then, remove all clauses containing Led5. After this step, f becomes ∅ and S = {Led2, Led5}.

4. The algorithm finishes because f = ∅, and we get reduct Led2 ∧ Led5.

In Johnson’s algorithm, the attribute that appears more frequently is considered to be the most significant. Even though this is not true in all cases, Johnson’s algorithm generally finds out a solution close to the optimal [5].

**2.3.7 Rule Accuracy and Assessment**

A decision rule can be denoted α β, read as “if α then β “. The pattern α is called the rule’s antecedence while the pattern β is called the rule’s consequence. Three units of measure shown below can be used to evaluate the quality of a given decision rule [2]:

1. Support: the number of events that possesses both property α then β.
2. Strength: A decision rule α β may only reveal partially the overall picture of the derived decision system. Given pattern α, the probability of the conclusion β can be assessed by measuring how trustworthy the rule is in drawing the conclusion β on the basis of evidence α.

Strength (α β) = (2.8)

1. Coverage: The strength of the rule relies upon the large support basis that describes the number of events, which support each rule. The quantity coverage (α β) is required in order to measure how well the evidence α describes the decision class. It can be defined via β:

Coverage (α β) = (2.9)

**2.3.8 Classifier Performance**

The set of rules derived from the reducts must be assessed on its classification performance, readability and usefulness before it can be used effectively in decision support. For assessing the classifier performance, the dataset is divided into a training set and a test set. The training set is a set of examples used for learning that is to fit the parameters, whereas the test set is a set of examples used only to assess the performance of a classifier. Rules are mined from a selection of events in each training set using rough sets. They are then used to classify the events in the test set. This method can be carried out for two purposes. First, the rule set can be viewed as a classifier, used for the purpose of classifying only. Second, the computed reducts and the generated rules can be used by domain experts to learn more about the data.

The original dataset is randomly divided into four different training sets and test sets respectively with a partition of 90%, 80%, 70% and 60% of the data for training and 10%, 20%, 30% and 40% for testing. The splits are used to avoid results based on rules that were generated for a particular selection of events [6]. This makes the results more reliable and independent on one particular selection of events.

**CHAPTER 3**

**ROUGH SET THEORY**

**3.1 Overview of the System**

This chapter provides the system design and explains the proposed system flow that describes how it works to generate decision rules from heart disease dataset. Brief and general rules are generated from user’s input, dataset by using Rough Set Theory. Rough Set theory is a new method that deals with vagueness and uncertainty emphasized in decision making. It is used to generate rules for decision making by using data reduction. In decision making, rough set methods have a powerful essence in dealing with uncertainties.

There are many datasets already available in websites on the Internet which are used for scientific research especially for data mining field. The system used heart disease dataset for knowledge discovery. The dataset is saved in the database file and used by rough set theory for extracting rules and saved them in the knowledge base to suggest and support decision making process. Rules are displayed as a model of antecedence – consequence pairs which are mined from dataset by Rough Set Theory. The antecedence or condition part is represented the attributes (Symptoms) and consequence or action part represents class attributes (disease).

Data mining aims to extract exact information from real world databases. It also aims to generate some rules which are meaningful and helpful for human to support decision making. All the patterns which are hidden are not able to see in a database, especially in a large amount of data. Data mining applications generate rules to make understand data and its patterns more easily and generated rules may also be called as knowledge or useful information since they have meanings. Computer scientists and engineers have a new challenge to discover meaningful information automatically. It is hard for a human to generate meaningful rules manually since it takes a lot of time. And then the datasets are so complex and large in volume. It is difficult to handle with the existing Database Management tools.

To overcome above problems, Rough Set Theory is used to make decisions under uncertainty and vagueness. RST approach can reduce datasets, find hidden patterns and generate meaningful rules for users. Inconsistent datasets are handled by RST using lower and upper approximations for every concept. In RST, the approximations of sets are introduced to deal with inconsistency. Inconsistent objects are same in all value of any set of condition attributes, but they belong to different decision classes. The objects in lower approximation can be classified with certainty as on the basis of knowledge while the objects in upper approximation can only be classified as the possible occurrence of decision class. The difference of lower and upper approximations is boundary region. If boundary region exists, there are inconsistent objects in dataset and then they are removed.

The input dataset is transformed into a decision system which is subsequently split into two parts: 60%, 70%, 80%, and 90% of the training dataset and 40%, 30%, 20% and 10% for the testing dataset respectively. Decision rules will be induced from the training dataset and applied to the testing dataset to obtain a performance estimate. The decision rules are generated by using the reducts computed by the attribute reduction with discernibility function, etc. Then, the strength are computed for each of the generated rules. After calculating strength for all rules, the predicted decision is assigned for each object in testing dataset. After assigning predicted class for all testing objects, the system calculates accuracy for generated rules. The rule set with maximum accuracy is produced as suggested rules.

**3.2 Design of the System**

At the first state, Original dataset is stored into MySQL database. They can contain redundant data and inconsistent data. The system calculates upper and lower approximations of the dataset. From this calculation, accuracy of dataset can be available. If accuracy is less than one, the system assumes there is inconsistent data and remove them. After removing any inconsistencies, the dataset is split into two parts: training dataset and testing dataset.

Dataset

Database

Store Dataset

Upper and Lower Approximation

Calculate Accuracy

Calculate Accuracy

Is Accuracy 1.0?

<1.0

=>1.0

Remove Inconsistency

Split Training and Testing Dataset

**Figure 3.1 System Design for Removing Inconsistency and Splitting Dataset**

Training dataset is used to reduce unnecessary attributes and generate decision rules. The unimportant attributes and values in making decision can be reduced by using JOHNSON’s algorithm of Rough Set Theory. The decision rules can be directly generated from these reducts computed by the attribute reduction with JOHNSON’s algorithm. One or more reducts can be generated and the set of decision rules can be generated for each of these reducts. Generated rules are stored into database.

Training Dataset

Reduce Attributes By using JOHNSON’s algorithm

Reduce Values By using JOHNSON’s algorithm

Generate Rules

Rule base

Store Rules

**Figure 3.2 System Design for Generating Rules**

Testing dataset is used to get performance estimation and accuracy rate of generated rule sets. The strength is calculated for each generated rule to know the percentage of correctness in making decision. After calculating the strength for all generated rules, the predicted decision is assigned for each object in testing dataset. In predicting decision for testing objects, the decision of rule with maximum strength is assigned into testing object when there are more than one rules with same condition value and different value for this testing object. After assigning predicted decision value for all testing objects, accuracy for the whole generated rule set can be calculated and the rule set with maximum accuracy is produced as the suggested rules.

Testing Dataset

Rule base

Calculate Strength of Rules

Assign Predicted Decision Class in each Testing Dataset

Calculate Accuracy

Store Accuracy of Rule set

**Figure 3.3 System Design for Calculating Rule Accuracy**

**3.3 Dataset**

In Rough Set Theory, the dataset is defined as decision table denoted by DS= (U, A),

where, U = the set of objects

A = the set of features (attributes),

there are two subsets of features in A = C U d

C = the set of condition features of objects

d = decision feature of objects

U = {p1, p2, p3, p4, p5, p6, p7, p8, p9, p10, p11, p12, p13, p14, p15, p16, p17, p18, p19, p20, p21, p22}

C = {a1, a2, a3, a4, a5, a6, a7, a8, a9, a10, a11, a12, a13}

d = a14

**Table 3.1 Attribute Lists of Heart Disease Dataset**

|  |  |
| --- | --- |
|  | **Attributes** |
| a1. | Age |
| a2. | Sex |
| a3. | cp (chest pain type) |
| a4. | trestbps (resting blood pressure in mmHg) |
| a5. | chol (serum cholestoral in mg/dl) |
| a6. | fbs (fasting blood sugar ) >120mg/dl |
| a7. | restecg (resting electrocardiographic results) |
| a8. | thalach (maximum heart rate achieved) |
| a9. | exang (exercise induced angina) |
| a10. | oldpeak (ST depression induced by exercise relative to rest) |
| a11. | slope (the slope of the peak exercise ST segment) |
| a12. | ca (number of major vessels (o-3) colored by flouroscopy) |
| a13. | thal (thallium scan) |
| a14. | Num (Diagnosis of heart disease) |

**Table 3.2 Attribute and Value Lists of Heart Disease Dataset**

|  |  |  |
| --- | --- | --- |
|  | **Attributes** | **Values** |
| a1. | Age | Young(<35), Mild(36-60), Old(60-75), Very Old(>75) |
| a2. | Sex | Male, Female |
| a3. | cp | 1(Typical angina), 2(Atypical angina), 3(Non-angina pain), 4(Asymptomatic) |
| a4. | trestbps | Low(<120),Normal(120-139), Medium(140-159), High(160-179), Very high(>=180) |
| a5. | chol | Low(<160), Medium(160-190), High(190-250), Very high(>250) |
| a6. | fbs | false, true |
| a7. | restecg | 1(Normal), 2(ST-T Abnormal), 3(Hypertrophy) |
| a8. | thalach | Medium, Normal, High |
| a9. | exang | yes, no |
| a10. | oldpeak | Low(<2), Risk(2-3), Terrible(>3) |
| a11. | slope | Up sloping, flat, Down sloping |
| a12. | ca | 0, 1, 2, 3 |
| a13. | thal | Normal, Fixed defect, Reversible defect |
| a14. | Num | Value 0(there is no heart disease),  Value 1 (there is heart disease), |

**Table 3.3 Example Heart Disease Dataset**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | a1 | a2 | a3 | a4 | a5 | a6 | a7 | a8 | a9 | a10 | a11 | a12 | a13 | a14 |
| p1 | old | male | 1 | medium | high | t | hyper | high | no | risk | down | 0 | fixed | 0 |
| p2 | old | male | 4 | high | very | f | hyper | high | yes | low | flat | 3 | normal | 1 |
| p3 | old | male | 4 | normal | high | f | hyper | high | yes | risk | flat | 2 | reverse | 1 |
| p4 | old | male | 4 | normal | high | f | hyper | high | yes | risk | flat | 2 | reverse | 0 |
| p5 | mild | female | 2 | normal | high | f | hyper | high | no | low | up | 0 | normal | 0 |
| p6 | mild | male | 2 | normal | high | f | normal | high | no | low | up | 0 | normal | 0 |
| p7 | old | female | 4 | medium | very | f | hyper | high | no | terrible | down | 2 | normal | 1 |
| p8 | mild | female | 4 | normal | very | f | normal | high | yes | low | up | 0 | normal | 0 |
| p9 | old | male | 4 | normal | very | f | hyper | high | no | low | flat | 1 | reverse | 1 |
| p10 | mild | male | 4 | medium | high | t | hyper | high | yes | terrible | down | 0 | reverse | 1 |
| p11 | mild | male | 4 | medium | high | f | normal | high | no | low | flat | 0 | fixed | 0 |
| p12 | mild | female | 2 | medium | very | f | hyper | high | no | low | flat | 0 | normal | 0 |
| p13 | mild | male | 3 | normal | very | t | hyper | high | yes | low | flat | 1 | fixed | 1 |
| p14 | mild | male | 2 | normal | very | f | normal | high | no | low | up | 0 | reverse | 0 |
| p15 | mild | male | 3 | high | high | t | normal | high | no | low | up | 0 | reverse | 0 |
| p16 | mild | male | 3 | medium | medium | f | normal | high | no | low | up | 0 | normal | 0 |
| p17 | mild | male | 2 | very | high | f | normal | high | no | low | down | 0 | reverse | 1 |
| p18 | mild | male | 4 | medium | high | f | normal | high | no | low | up | 0 | normal | 0 |
| p19 | old | female | 3 | normal | medium | t | normal | normal | no | low | up | 0 | normal | 0 |
| p20 | mild | male | 3 | normal | high | f | normal | high | no | terrible | down | 0 | normal | 0 |

**3.4 Rule Generation**

Table 3.3 shows an example of heart disease dataset of 20 patients as training dataset. This training dataset is used to remove inconsistencies and generate decision rules.

**3.4.1 Removing Inconsistencies**

Indiscernibility Relation is a central concept in Rough Set Theory, and is considered as a relation between two objects or more, all the values are identical in all conditional attributes. Indiscernibility relation is an equivalence relation, where all identical objects of set are considered as elementary.

IND(C)={(x, y) ϵ U \* U : c(x) = c(y) , c ϵ C}

IND(C)={{p1}, {p2}, {p3}, {p4}, {p5}, {p6}, {p7}, {p8}, {p9}, {p10}, {p11}, {p12}, {p13}, {p14}, {p15}, {p16}, {p17}, {p18}, {p19}, {p20}, {p21, p22}}

Upper Approximation is the description of the objects that possibly belong to the subset of concept.

Up(X) ={ Y ϵ U / IND(C) | Y ∩ X ≠ Ø}

Lower Approximation is a description of the domain objects that certainly belong to the subset of concept.

Low(X) ={ Y ϵ U / IND(C) | Y Ϲ̲ X}

For d=1, X={p2, p3, p7, p9, p10, p13, p17, p22}

Up(X)={p2, p3, p7, p9, p10, p13, p17, p21, p22}

Low(X)={p2, p3, p7, p9, p10, p13, p17}

For d=0, X={p1, p4, p5, p6, p8, p11, p12, p14, p15, p16, p18, p19, p20, p21}

Up(X)={p1, p4, p5, p6, p8, p11, p12, p14, p15, p16, p18, p19, p20, p21, p22}

Low(X)={p1, p4, p5, p6, p8, p11, p12, p14, p15, p16, p18, p19, p20}

Boundary Region of the set is the difference of upper and lower approximations.

BNX = Up(X)-Low(X)

BNX = Up(X)-Low(X) ={p21, p22}

If the boundary region is not empty and accuracy is less than 1.0, assume that there is inconsistent data in dataset. Otherwise, there is no inconsistent data. In boundary region, there are p21 and p22 as inconsistencies. They are removed to make generated rules strong and powerful in decision making.

accuracy= |Low(X)|/|Up(X)|, {0 < accuracy < 1}

Accuracy= |Low(X)|/|Up(X)|

=7/9 = 0.78 (for decision=1)

=13/15=0.87 (for decision=0)

**3.4.2 Calculating Dependency**

Approximations of sets are related to concept of dependency (total or partial) of attributes.

k= / |U|

k= (7+13)/20

=1.0

where , |U|=total number of objects

n =number of decision class (1 or 0)

k = 1.0, so, decision attribute depends totally on condition attributes,

If k< 1.0, decision attribute depends partially on condition attributes.

**3.4.3 Attribute Reduction**

Discernibility function is used to calculate reducts. Reduct is a minimal subset of attributes which is more important than other attributes in dataset. After reduct generation, rules are automatically generated.

m(, ) ={c ϵ C| c() ≠ c() }, for i, j=1, 2, 3, …n

= ᴧ {ᴠ m(, ) | 1<= i <= j <= n, m(, ) ≠ Ø}

where,

m(, ) = the set of attributes which have different value between two objects , (different decision)

= the discernibility function of all objects

n = number of objects

C = set of condition attributes in dataset

c(), c() = value of each attribute for two objects ,

Boolean Function; a ᴠ (a ᴧ b) =a

a ᴧ(a ᴠ b) =a

a ᴠ (b ᴧ c)=(a ᴠ b) ᴧ (a ᴠ c)

a ᴧ (b ᴠ c)=(a ᴧ b) ᴠ (a ᴧ c)

F(U)=F(p1) ᴧ F(p2) ᴧ F(p3) ᴧ F(p4) ᴧ F(p5) ᴧ F(p6) ᴧ F(p7) ᴧ F(p8) ᴧ F(p9) ᴧ F(p10) ᴧ F(p11) ᴧ F(p12) ᴧ F(p13) ᴧ F(p14) ᴧ F(p15) ᴧ F(p16) ᴧ F(p17) ᴧ F(p18) ᴧ F(p19) ᴧ F(p20)

Eg. F(p1)= (a3 ᴠ a4 ᴠ a5 ᴠ a6 ᴠ a9 ᴠ a10 ᴠ a11 ᴠ a12 ᴠ a13) ᴧ

(a3 ᴠ a4 ᴠ a6 ᴠ a9 ᴠ a11 ᴠ a12 ᴠ a13) ᴧ

(a2 ᴠ a3 ᴠ a5 ᴠ a6 ᴠ a9 ᴠ a10 ᴠ a12 ᴠ a13) ᴧ

(a3 ᴠ a4 ᴠ a5 ᴠ a6 ᴠ a10 ᴠ a11 ᴠ a12 ᴠ a13) ᴧ

(a1 ᴠ a3 ᴠ a9 ᴠ a10 ᴠ a13) ᴧ

(a1 ᴠ a3 ᴠ a4 ᴠ a5 ᴠ a9 ᴠ a10 ᴠ a11 ᴠ a12) ᴧ

(a1 ᴠ a3 ᴠ a4 ᴠ a6 ᴠ a7 ᴠ a10 ᴠ a13) ᴧ

(a3 ᴠ a4 ᴠ a6 ᴠ a7 ᴠ a8 ᴠ a10 ᴠ a11 ᴠ a13)

F(U)= a3 a4 a11 ᴠ a3 a4 a7 ᴠ a3 a11 a13 ᴠ a4 a11 a13

After calculating reducts, four sets of reduct are generated as ( a3 a4 a11), (a3 a4 a7 ), ( a3 a11 a13) and (a4 a11 a13). The set ( a3 a4 a11) is used for generating rules.

**Table 3.4 Attribute Reduction of Training Dataset**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | a3 | a4 | a11 | a14 |
| p1 | 1 | medium | down | 0 |
| p2 | 4 | high | flat | 1 |
| p3 | 4 | normal | flat | 1 |
| p4 | 4 | normal | flat | 0 |
| p5 | 2 | normal | up | 0 |
| p6 | 2 | normal | up | 0 |
| p7 | 4 | medium | down | 1 |
| p8 | 4 | normal | up | 0 |
| p9 | 4 | normal | flat | 1 |
| p10 | 4 | medium | down | 1 |
| p11 | 4 | medium | flat | 0 |
| p12 | 2 | medium | flat | 0 |
| p13 | 3 | normal | flat | 1 |
| p14 | 2 | normal | up | 0 |
| p15 | 3 | high | up | 0 |
| p16 | 3 | medium | up | 0 |
| p17 | 2 | very | down | 1 |
| p18 | 4 | medium | up | 0 |
| p19 | 3 | normal | up | 0 |
| p20 | 3 | normal | down | 0 |

**3.4.4 Attribute Value Reduction**

After reducing unnecessary attributes, the redundant values can be reduced to directly generate decision rules. The symbol “\*” is defined as the unnecessary value of attributes in generating final minimal rule set.

**Table 3.5 Attribute Value Reduction of Training Dataset**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | a3 | a4 | a11 | a14 |
| p1 | 1 | \* | \* | 0 |
| p2 | 4 | high | flat | 1 |
| p3 | \* | normal | flat | 1 |
| p4 | 4 | normal | flat | 0 |
| p5 | 2 | normal | up | 0 |
| p6 | 2 | normal | up | 0 |
| p7 | 4 | \* | down | 1 |
| p8 | \* | \* | up | 0 |
| p9 | \* | normal | flat | 1 |
| p10 | 4 | \* | down | 1 |
| p11 | \* | medium | flat | 0 |
| p12 | 2 | medium | flat | 0 |
| p13 | 3 | normal | flat | 1 |
| p14 | 2 | normal | up | 0 |
| p15 | 3 | high | up | 0 |
| p16 | 3 | medium | up | 0 |
| p17 | 2 | very | down | 1 |
| p18 | \* | \* | up | 0 |
| p19 | \* | \* | up | 0 |
| p20 | \* | normal | down | 0 |

The necessary and important attributes for each row of the table are as follows. They are changed from the form of Table 3.5. The final minimal rule may be directly generated from the following lines.

F(p1)=a3

F(p2)=a3a4 ᴠ a4a11

F(p3)=a4a11

F(p4)=a3a11 ᴠ a4a11

F(p5)=a3a4 ᴠ a11

F(p6)=a3a4 ᴠ a11

F(p7)=a3a11

F(p8)=a11

F(p9)=a4a11

F(p10)=a3a11

F(p11)=a4a11

F(p12)=a3a4 ᴠ a3a11 ᴠ a4a11

F(p13)=a3a11 ᴠ a4a11

F(p14)=a3a4 ᴠ a3a11 ᴠ a4a11

F(p15)=a3a4 ᴠ a3a11 ᴠ a4a11

F(p16)=a3a4 ᴠ a3a11 ᴠ a4a11

F(p17)=a4 ᴠ a3a11

F(p18)=a11

F(p19)=a11

F(p20)=a4a11

**3.4.5 Generating Minimum Rules**

*F(p1)*

If cp=Typical angina then TD=0

*F(p2)*

If cp=Asymptomatic, and trestbps=High, then TD=1

If trestbps=High, and slope=Flat, then TD=1

*F(p3)*

If trestbps=Normal and slope=Flat then TD=1

*F(p4)*

If cp=Non-anginal pain, and slope=Down then TD=0

If trestbps=Normal, and slope=Down then TD=0

*F(p5)*

If cp=Atypical angina, and trestbps=Normal then TD=0

If slope=Up then TD=0

*F(p6)*

If cp=Atypical angina, and trestbps=Normal then TD=0

If slope=Up then TD=0

*F(p7)*

If cp=Asymptomatic, and slope=Down then TD=1

*F(p8)*

If slope=Up then TD=0

*F(p9)*

If trestbps=Normal and slope=Flat then TD=1

*F(p10)*

If cp=Asymptomatic, and slope=Down then TD=1

*F(p11)*

If trestbps=Medium, and slope=Flat then TD=0

*F(p12)*

If cp=Atypical angina, and trestbps=Medium then TD=0

If cp=Atypical angina, and slope=Flat then TD=0

If trestbps=Medium, and slope=Flat then TD=0

*F(p13)*

If cp=Non-anginal pain, and slope=Flat then TD=1

If trestbps=Normal and slope=Flat then TD=1

*F(p14)*

If cp=Atypical angina, and trestbps=Normal then TD=0

If cp=Atypical angina, and slope=Up then TD=0

If trestbps=Normal, and slope=Up then TD=0

*F(p15)*

If cp=Non-anginal pain, and trestbps=High then TD=0

If cp=Non-anginal pain, and slope=Up then TD=0

If trestbps=High, and slope=Up then TD=0

*F(p16)*

If cp=Non-anginal pain, and trestbps=Medium, then TD=0

If cp=Non-anginal pain, and slope=Up then TD=0

If trestbps=Medium, and slope=Up then TD=0

*F(p17)*

If trestbps=Very High, then TD=1

If cp=Atypical angina, and slope=Down then TD=1

*F(p18)*

If slope=Up then TD=0

*F(p19)*

If slope=Up then TD=0

*F(p20)*

If trestbps=Normal and slope=Flat then TD=1

**3.4.6 Final Minimum Rules**

* If cp=Typical angina then TD=0
* If slope=Up then TD=0
* If cp=Asymptomatic, and trestbps=High, then TD=1
* If trestbps=High, and slope=Flat, then TD=1
* If trestbps=Normal and slope=Flat then TD=1
* If cp=Non-anginal pain, and slope=Down then TD=0
* If trestbps=Normal, and slope=Down then TD=0
* If cp=Atypical angina, and trestbps=Normal then TD=0
* If cp=Asymptomatic, and slope=Down then TD=1
* If trestbps=Medium, and slope=Flat then TD=0
* If cp=Atypical angina, and trestbps=Medium then TD=0
* If cp=Atypical angina, and slope=Flat then TD=0
* If cp=Non-anginal pain, and slope=Flat then TD=1
* If cp=Non-anginal pain, and trestbps=High then TD=0
* If cp=Non-anginal pain, and trestbps=Medium, then TD=0
* If trestbps=Very High, then TD=1
* If cp=Atypical angina, and slope=Down then TD=1

**3.5 Apply Generated Rules with Testing Dataset**

Table 3.6 shows an example of heart disease dataset of 10 patients as testing dataset. This testing dataset is used to estimate performance and accuracy of decision rules.

**Table 3.6 Example Heart Disease Testing Dataset**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | a1 | a2 | a3 | a4 | a5 | a6 | a7 | a8 | a9 | a10 | a11 | a12 | a13 | a14 |
| p1 | mild | female | 2 | normal | high | f | hyper | high | no | low | flat | 1 | normal | 1 |
| p2 | mild | male | 4 | normal | low | f | normal | high | yes | low | flat | 1 | reverse | 1 |
| p3 | old | male | 1 | high | high | f | hyper | high | no | low | flat | 0 | reverse | 0 |
| p4 | mild | male | 3 | normal | med | f | normal | high | no | low | up | 0 | normal | 0 |
| p5 | mild | female | 1 | medium | high | f | normal | high | no | low | up | 0 | normal | 0 |
| p6 | old | male | 4 | high | high | f | hyper | high | no | risk | up | 0 | fix | 0 |
| p7 | mild | male | 2 | normal | very | f | hyper | high | no | low | down | 0 | normal | 0 |
| p8 | mild | female | 4 | very | very | t | normal | high | yes | terrible | down | 2 | reverse | 1 |
| p9 | mild | male | 1 | high | very | f | hyper | high | no | low | flat | 0 | reverse | 1 |
| p10 | old | male | 1 | high | high | t | hyper | high | no | low | flat | 1 | normal | 0 |

**3.5.1 Calculation of Rule Strength**

strengthᵣ(C, D)= support ᵣ(C, D)/support ᵣ (C)

For rule r,

If cp=Typical angina then TD=0,

There are three objects in which cp(a3) is typical angina(1) and decision is 0. So, supportᵣ(C, D) is 3. There are four objects in which cp(a3) is typical angina(1).

So, support ᵣ(C) is 4.

strengthᵣ(C, D) = (3/4)\*100%

= 75%

**3.5.2 Rules with Strength**

* If cp=Typical angina then TD=0 75%
* If slope=Up then TD=0 100%
* If cp=Asymptomatic, and trestbps=High, then TD=1 0%
* If trestbps=High, and slope=Flat, then TD=1 33%
* If trestbps=Normal and slope=Flat then TD=1 100%
* If cp=Non-anginal pain, and slope=Down then TD=0 -
* If trestbps=Normal, and slope=Down then TD=0 100%
* If cp=Atypical angina, and trestbps=Normal then TD=0 50%
* If cp=Asymptomatic, and slope=Down then TD=1 0%
* If trestbps=Medium, and slope=Flat then TD=0 -
* If cp=Atypical angina, and trestbps=Medium then TD=0 -
* If cp=Atypical angina, and slope=Flat then TD=0 0%
* If cp=Non-anginal pain, and slope=Flat then TD=1 -
* If cp=Non-anginal pain, and trestbps=High then TD=0 -
* If cp=Non-anginal pain, and trestbps=Medium, then TD=0 -
* If trestbps=Very High, then TD=1 100%
* If cp=Atypical angina, and slope=Down then TD=1 0%

**3.5.3 Making Predicted Decision for Testing Data**

For patient 1,

If trestbps=Normal and slope=Flat then TD=1 100%

If cp=Atypical angina, and trestbps=Normal then TD=0 50%

If cp=Atypical angina, and slope=Flat then TD=0 0%

Among these rules, the decision of the rule with maximum strength is assigned for patient 1.

**Table 3.7 Testing Dataset with Predicted Decision**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | a1 | a2 | a3 | a4 | a5 | a6 | a7 | a8 | a9 | a10 | a11 | a12 | a13 | a14 | a15 |
| p1 | mild | female | 2 | normal | high | f | hyper | high | no | low | flat | 1 | normal | 1 | 1 |
| p2 | mild | male | 4 | normal | low | f | normal | high | yes | low | flat | 1 | reverse | 1 | 1 |
| p3 | old | male | 1 | high | high | f | hyper | high | no | low | flat | 0 | reverse | 0 | 0 |
| p4 | mild | male | 3 | normal | med | f | normal | high | no | low | up | 0 | normal | 0 | 0 |
| p5 | mild | female | 1 | medium | high | f | normal | high | no | low | up | 0 | normal | 0 | 0 |
| p6 | old | male | 4 | high | high | f | hyper | high | no | risk | up | 0 | fix | 0 | 0 |
| p7 | mild | male | 2 | normal | very | f | hyper | high | no | low | down | 0 | normal | 0 | 1 |
| p8 | mild | female | 4 | very | very | t | normal | high | yes | terrible | down | 2 | reverse | 1 | 0 |
| p9 | mild | male | 1 | high | very | f | hyper | high | no | low | flat | 0 | reverse | 1 | 0 |
| p10 | old | male | 1 | high | high | t | hyper | high | no | low | flat | 1 | normal | 0 |  |

**3.5.4 Calculation of Accuracy**

After assigning predicted values to all objects in testing dataset, the system calculates the whole rate of accuracy for each rule sets and generates the rule set with maximum accuracy as the suggested rules to support decision making process.

accuracy = total number of supported objects/ number of all objects

There are 9 objects in which the actual decision and the predicted decision are the same. And then there are 10 total number of objects in testing data.

So, accuracy = (9/10)\*100%

= 90%

Accuracy of all final rules is 90%.

**CHAPTER 4**

**SYSTEM IMPLEMENTATION**

This chapter describes implementation of rule generation from heart disease dataset by using JOHNSON’s algorithm of Rough Set Theory. The system was implemented by C#.Net programming language, and MySQL database. Heart disease dataset is obtained from the website “https://github.com”.

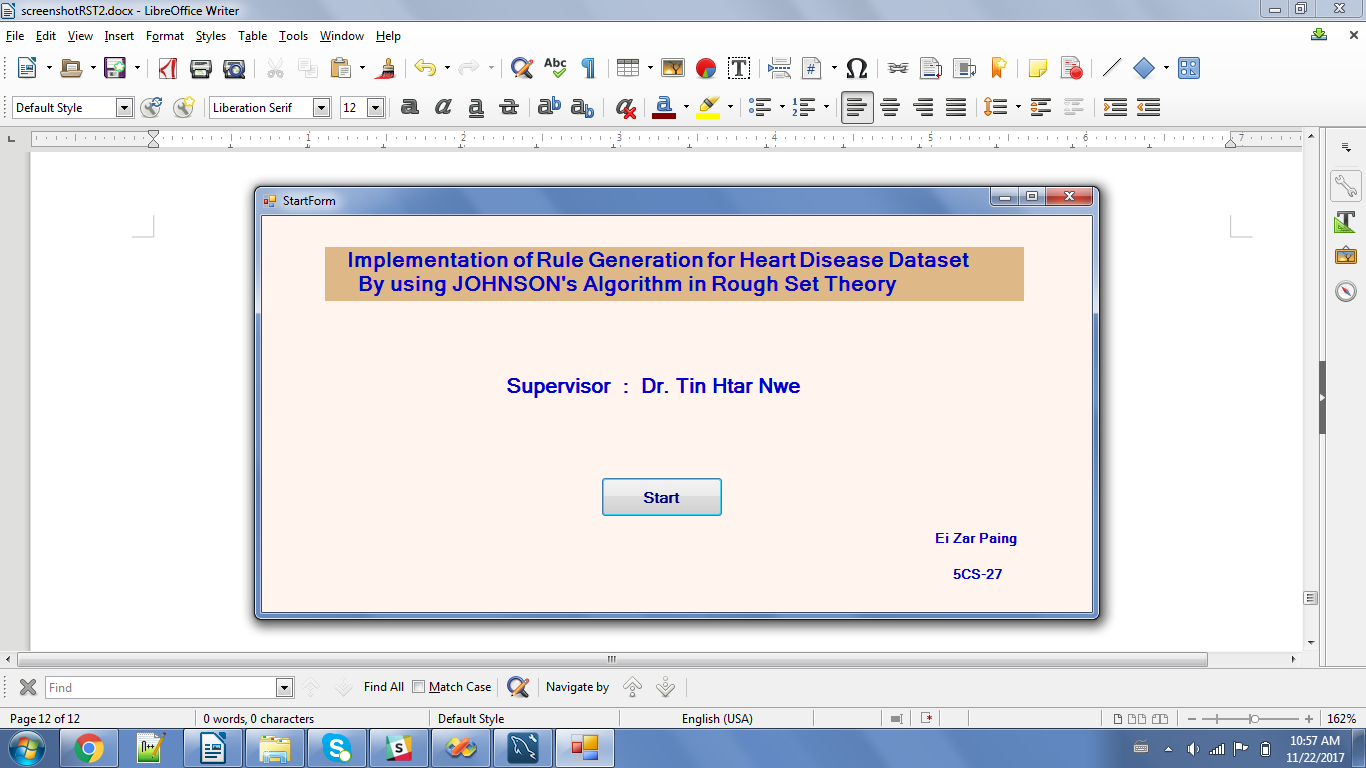
**4.1 Overview of the System**

At the first state, the original dataset of patients who suffer heart disease or not are inserted into MySQL database. By using them, the system “Implementation of rule generation for heart disease dataset by using Rough Set Theory” will check whether there is inconsistent data or not in this heart dataset. Inconsistent data are same in symptoms of any set of patients, but they are in different states of disease (one suffer heart disease and other can’t suffer heart disease, but they have same symptoms).

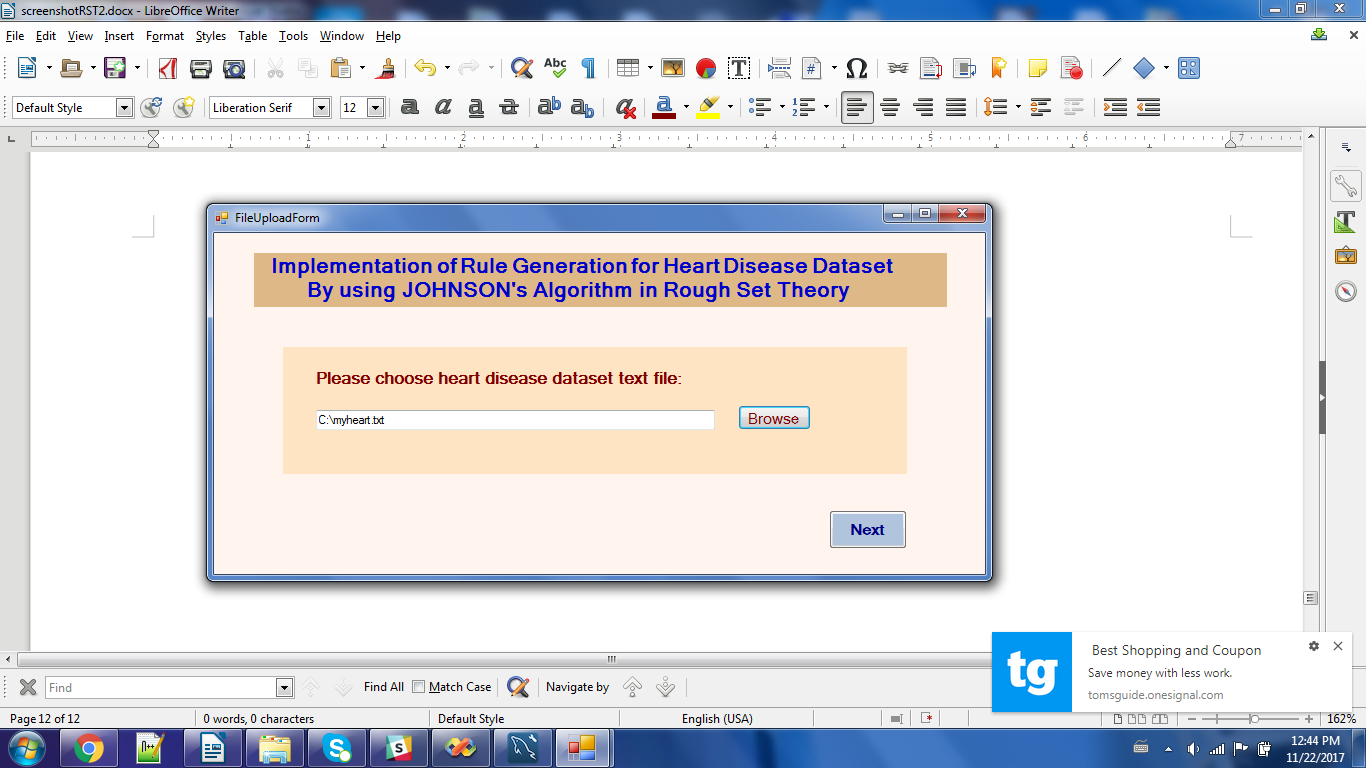
After removing inconsistent data, the user needs to chose random number and percentages of training data and testing data. The system splits original dataset into two parts training dataset and testing dataset according to percentage user chose. Random number is for relocating randomly all objects in the dataset. The system uses training dataset and reduces unimportant attributes by calculating set of reducts. Reducts can include only useful attributes to generate strong rules. One set of rules can be generated from each of reducts with JOHNSON’s algorithm. The system calculates strength for each generated rules and assigns predicted decision for each objects of testing dataset by using these generated rules. After assigning predicted decision for all testing objects, the system calculates accuracy rate for generated rules. Finally, the system shows the suggested rule sets that have maximum accuracy.

**4.2 Implementation of the System**

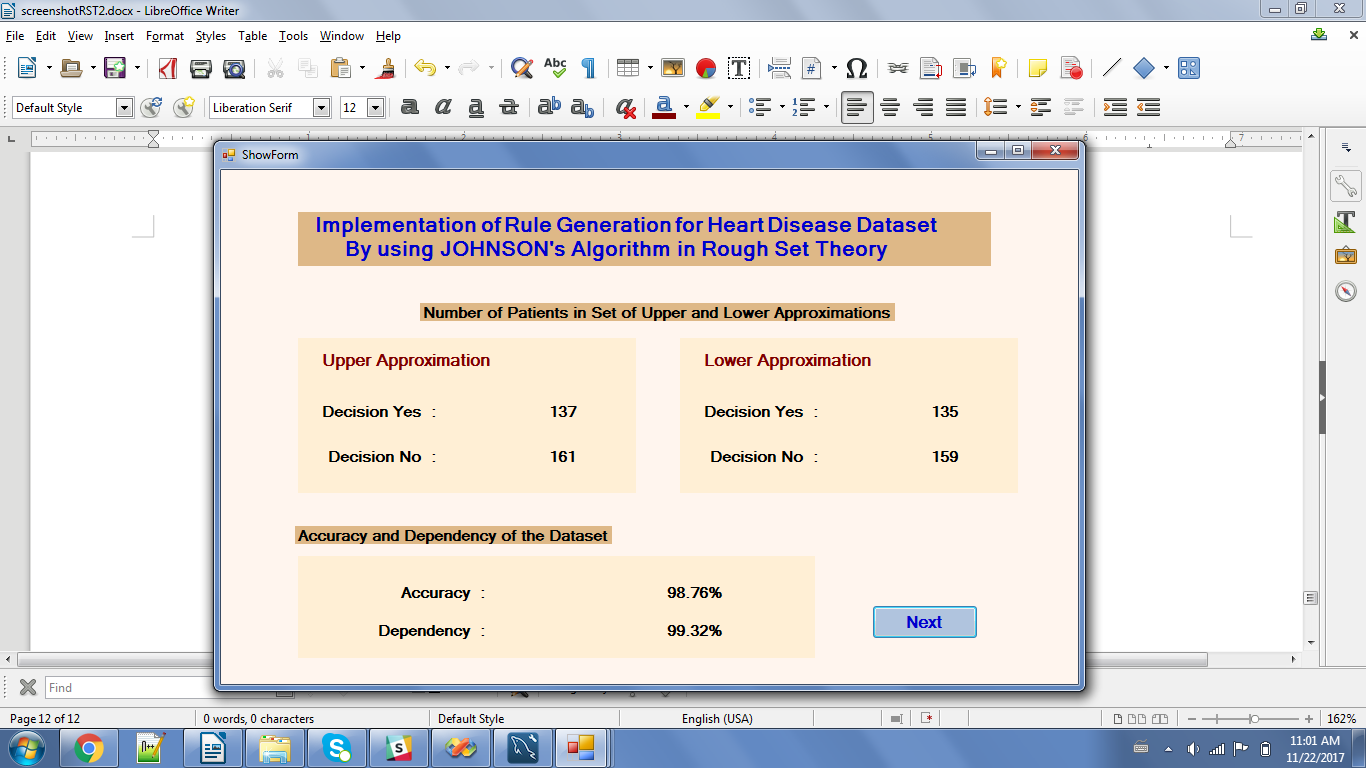
The home page of the system for “Implementation of rule generation for heart disease dataset by using Rough Set Theory” is illustrated in Figure 4.1. From this interface, user can start the system by clicking “Start” button.Firstly, the heart disease dataset will be stored into database. To store the heart disease dataset, text file of dataset is chosen from the browser in Figure 4.2.



**Figure 4.1 Home Page**



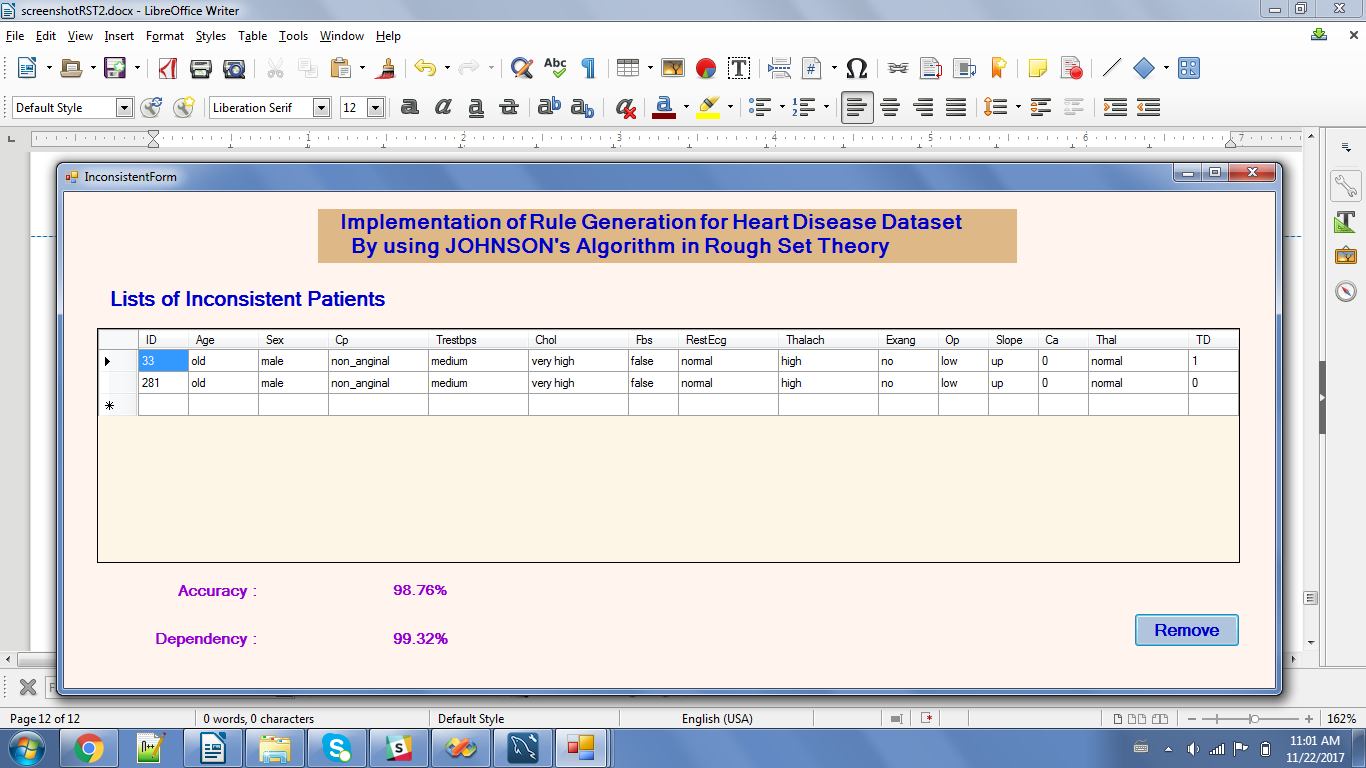
**Figure 4.2 Upload File**

****

**Figure 4.3 Show Upper and Lower Approximations**

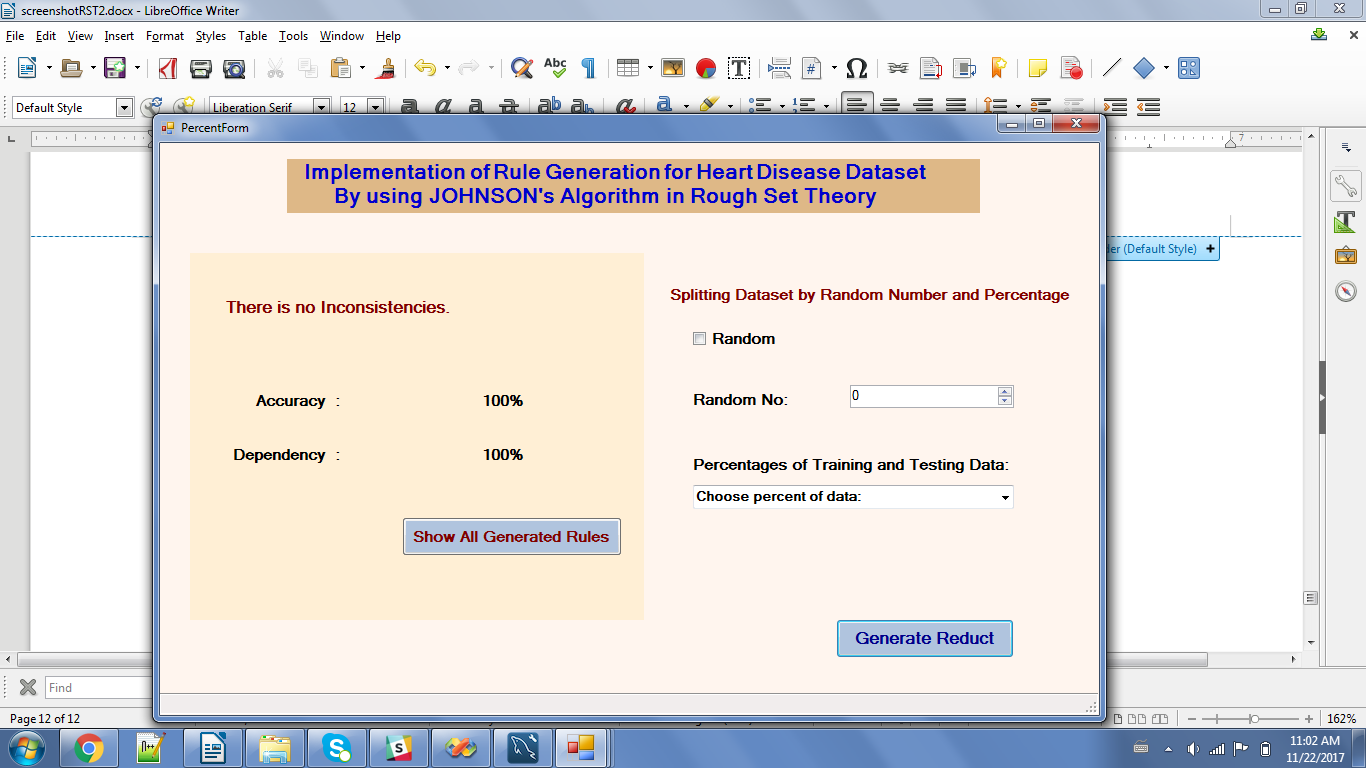
After storing the dataset into database, Upper and Lower Approximation sets are calculated to handle inconsistent objects. The numbers of patients in Upper and Lower Approximation sets and the rate of Accuracy and Dependency of the dataset are shown in Figure 4.3. If there are inconsistent objects in the dataset, it shows that accuracy and dependency is under 100%.

After the “Next” button is clicked, the inconsistent patients are shown in Figure 4.4 when the accuracy is under 100%. The user can choose random number and type of percentages splitting training dataset and testing dataset in Figure 4.5 when the accuracy is 100% and there is no inconsistent objects. The user can see inconsistent patients data and accuracy in Figure 4.4. And then the user can remove inconsistent data from original dataset by using “Remove” button.



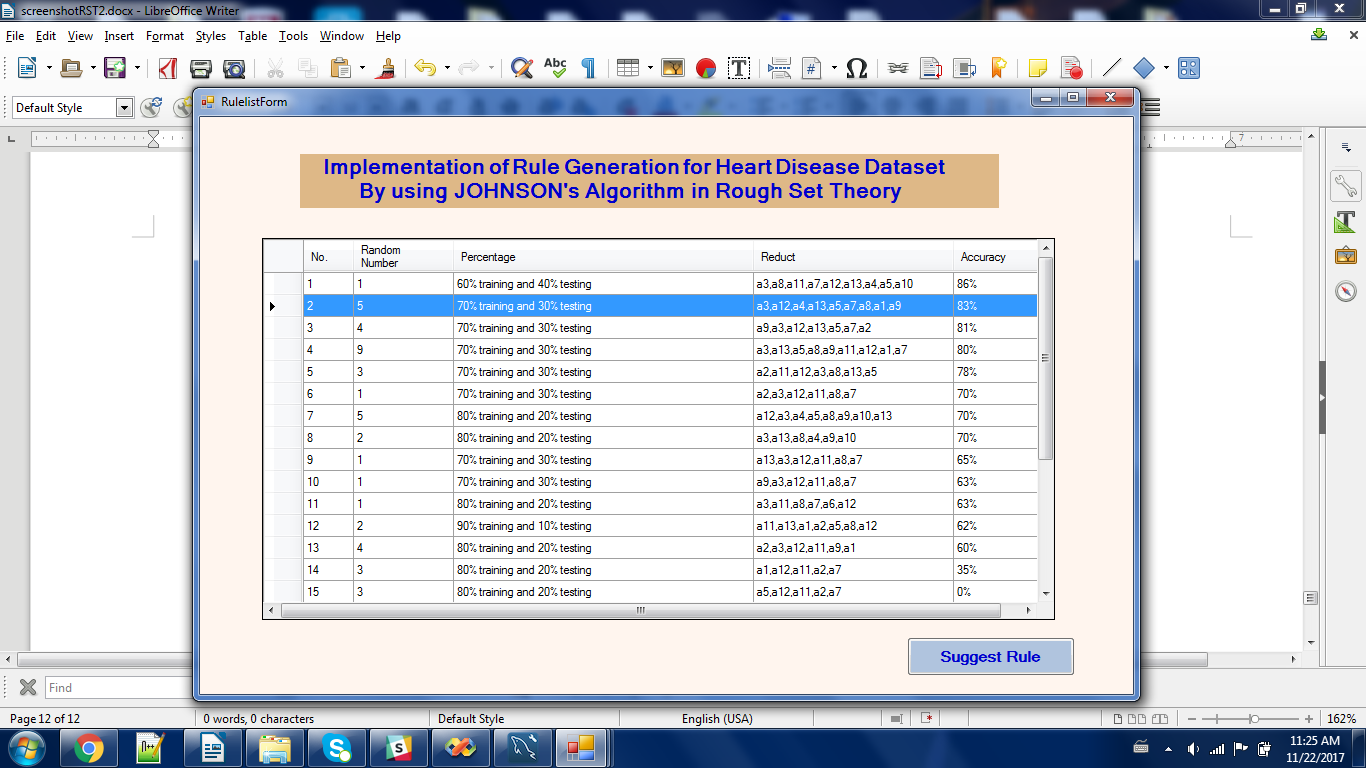
**Figure 4.4 Inconsistent Data Page**

In Figure 4.3, the accuracy is 100% and there is no inconsistent objects in the dataset. The user can choose random numbers and percentages to split training and testing dataset as they want. The system gives four types of percentages to user.



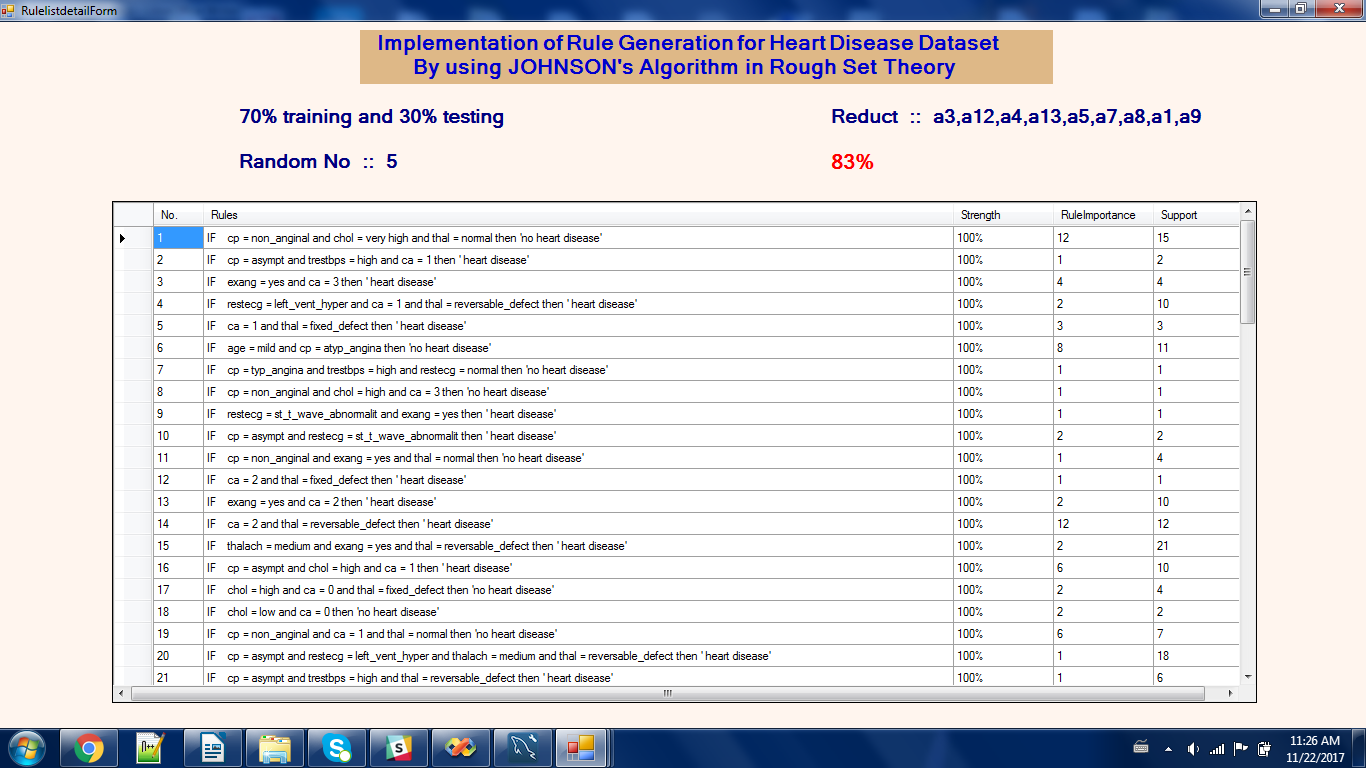
**Figure 4.5 Show Accuracy**

In figure 4.5, the user can see the lists of already generated rules with accuracy by clicking the button “Show All Generated Rules”. All pairs of reduct, percentage and random number list that the user already chose to generate rules are shown in Figure 4.6. The user can do “double click” and see the rule lists of each row in the table of Figure 4.6. And then the rule set with maximum accuracy can be seen by clicking the button “Suggest Rule”.

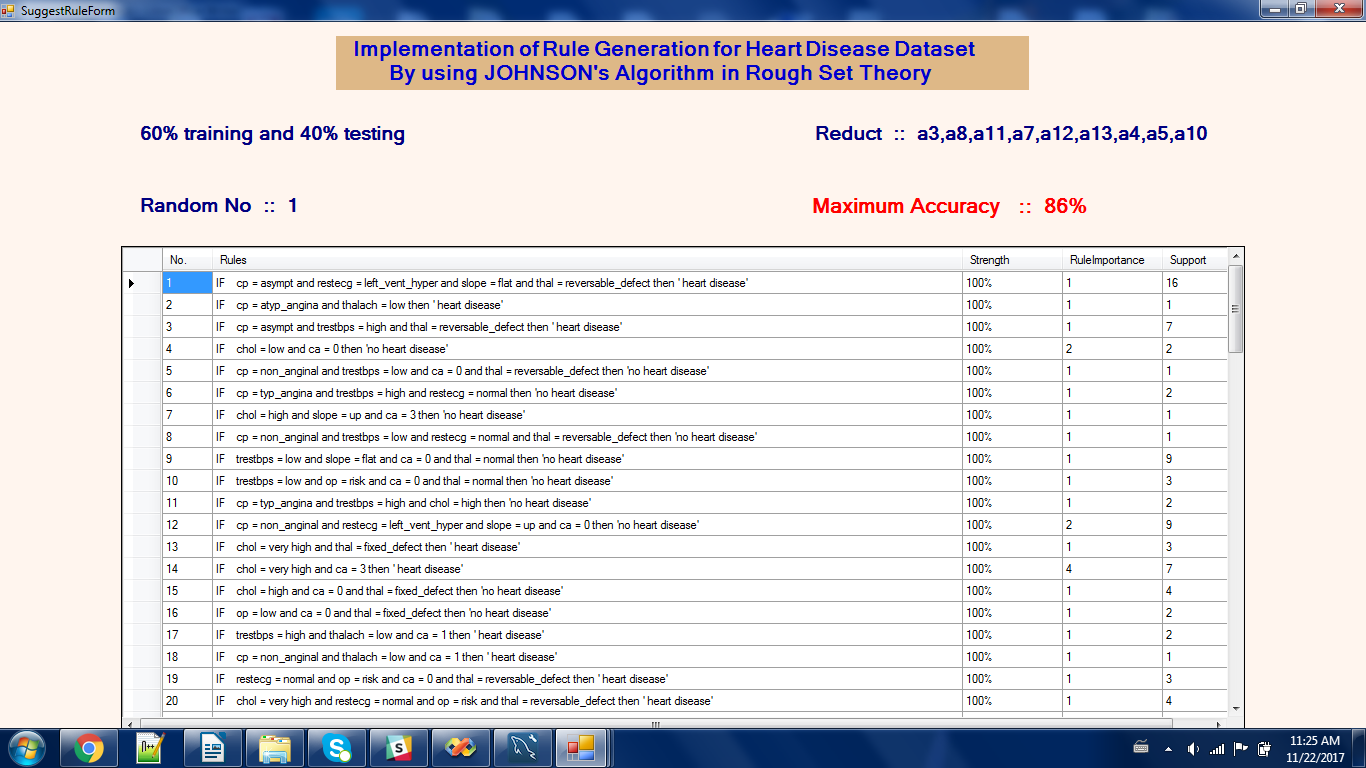


**Figure 4.6 Show All Rule Lists with Accuracy**

In figure 4.7, the list of rules with the accuracy are shown according to the row that user selected in table of Figure 4.6. The rule set with the maximum accuracy are shown in Figure 4.8 as the suggested rule set.

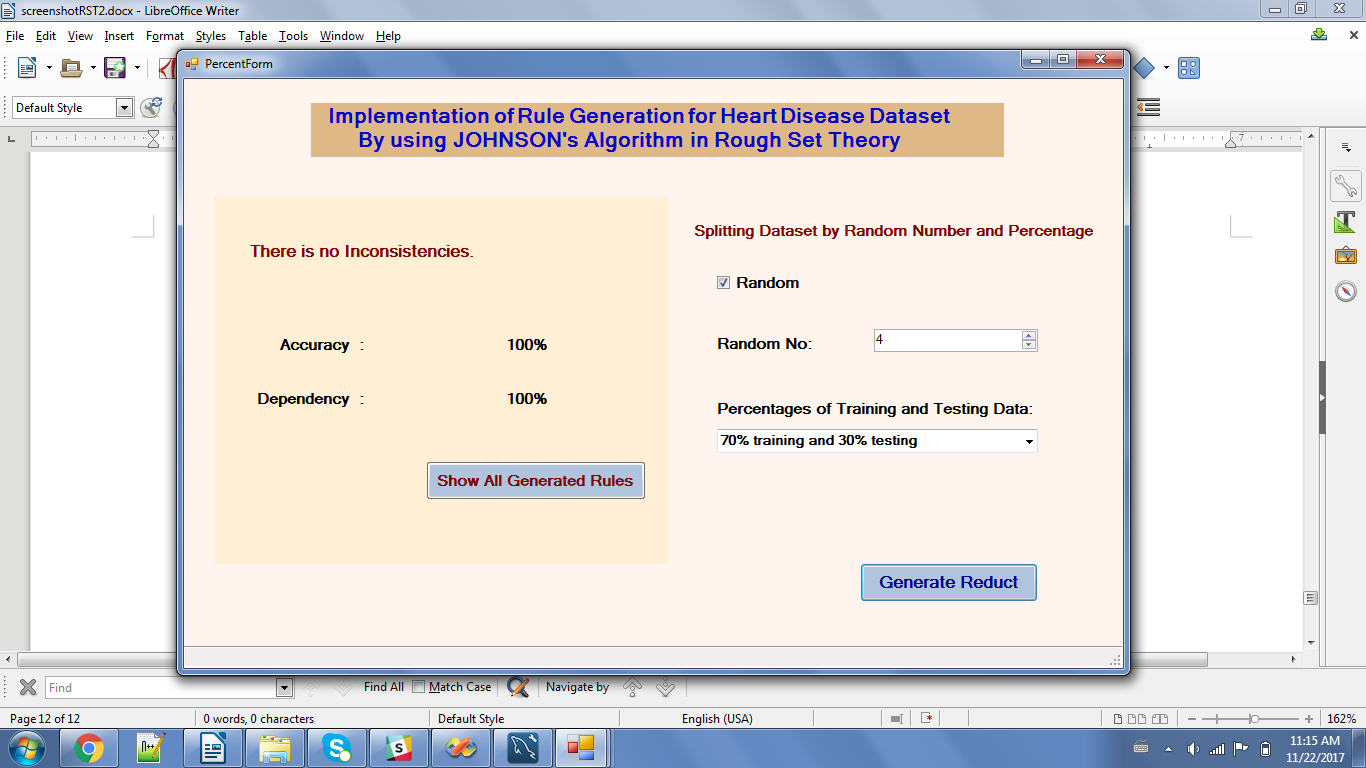


**Figure 4.7 Show Detail of Selected Rule List with Accuracy**

****

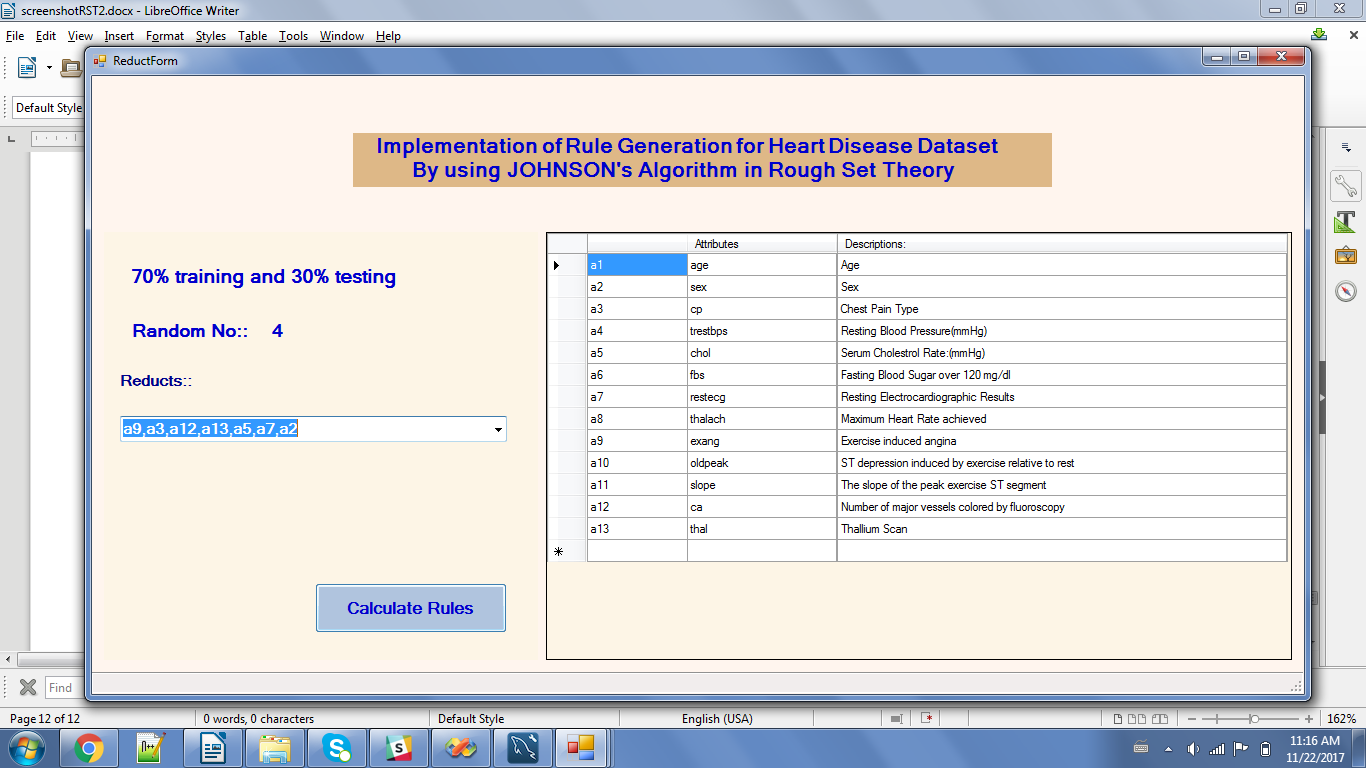
**Figure 4.8 Show Rules with Maximum Accuracy as Suggested Rules**

According to user’s choice, the dataset is randomized with random number 4 and is split 70% of training dataset and 30% of testing dataset in Figure 4.9. The user must click “Generate Reduct” button to get set of important attributes in generating strong rules and to remove unimportant attributes that make waste of time and space.



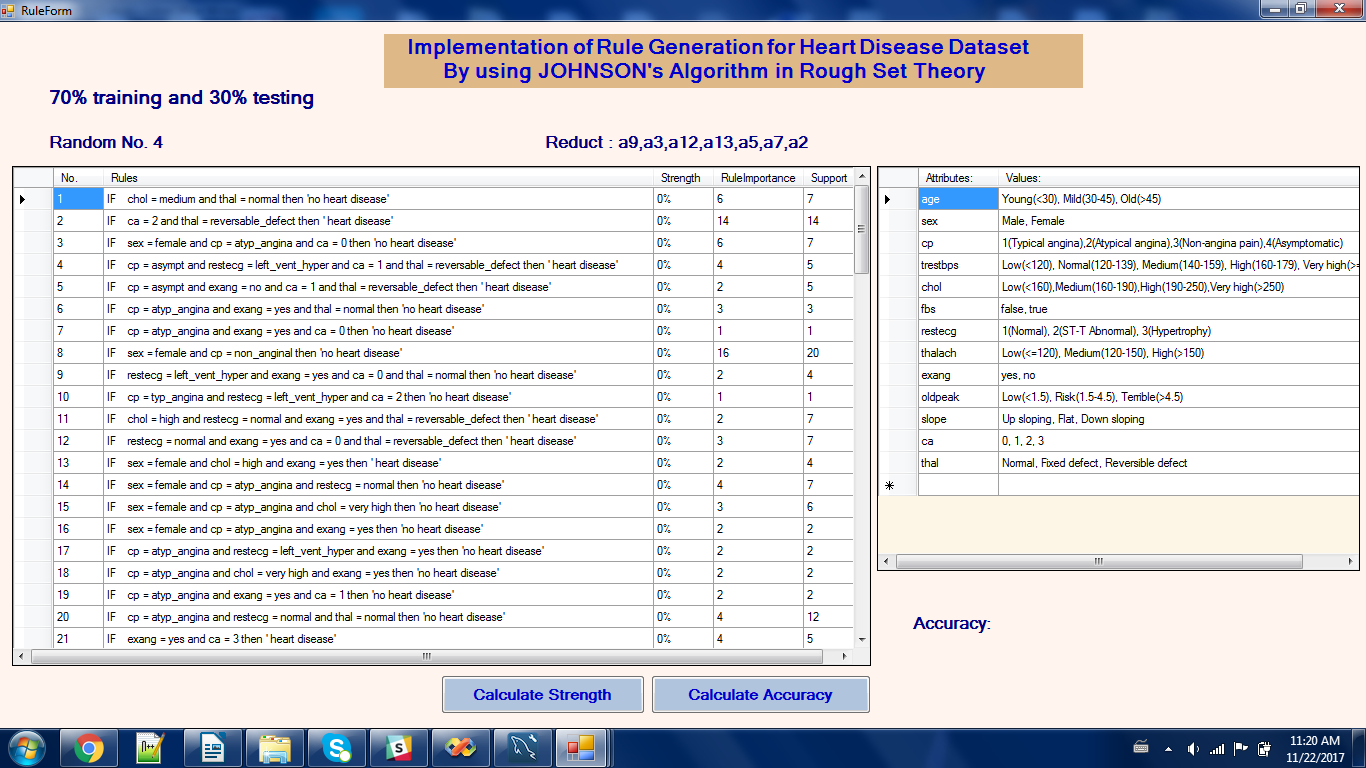
**Figure 4.9 Select Random Number and Percentage**

Reducts are generated from training dataset by using JOHNSON’s algorithm in Figure 4.10. Reduct is the set of important features from dataset to classify the decision class and can be available from reducing unimportant attributes and values from training dataset. Reducts can be generated more than one. At the right side of the Figure 4.10, the system shows detailed explanation of Attributes and its Description. The user can choose reduct and click “Calculate Rules” button to generate rule sets that use only attributes in selected reduct.



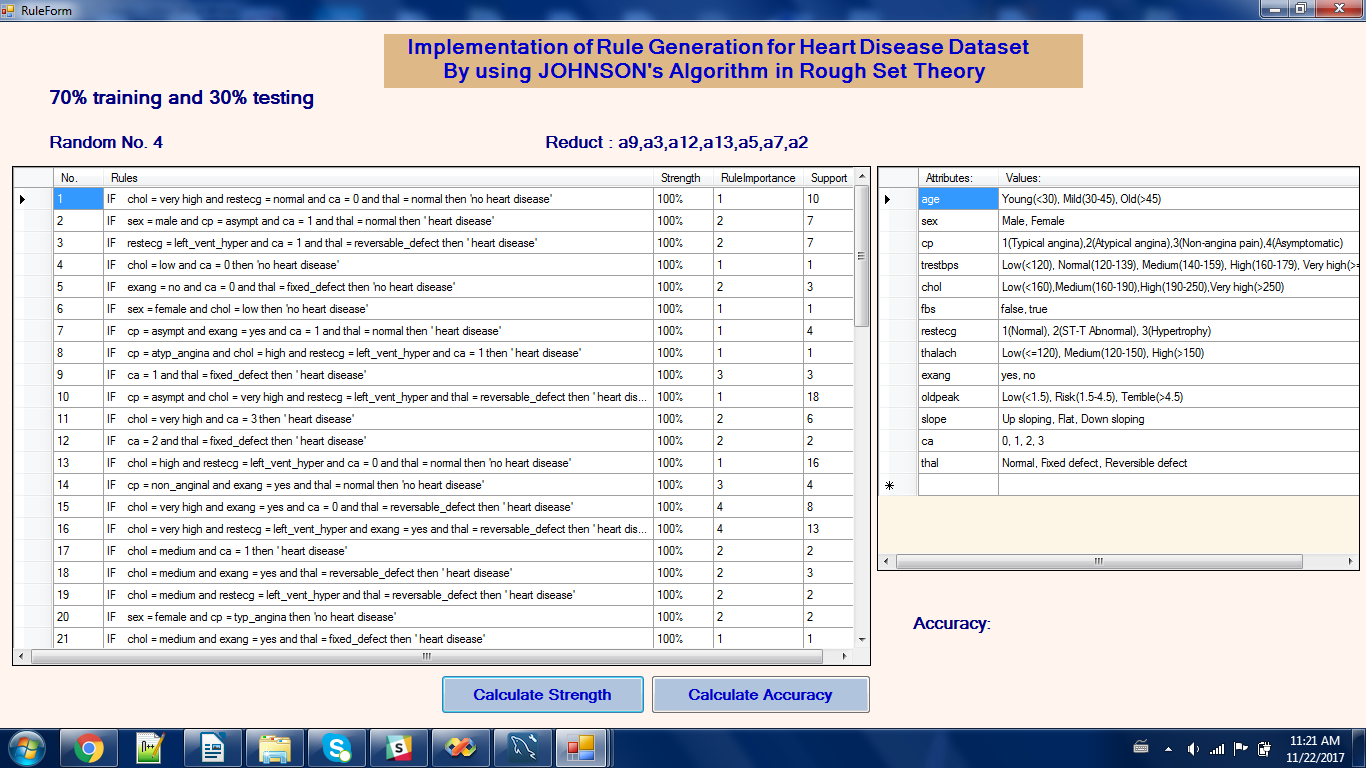
**Figure 4.10 Generate Reducts**

The set of rules are generated in Figure 4.11 according to random number, percentages of training and testing dataset and reduct which the user chooses. And then rule importance and support of each rule can be seen in Figure 4.11. At the right side of the Figure 4.11, the system shows detailed explanation of Attributes and its Values. Furthermore, the user can calculate strength of each rule by clicking “Calculate Strength” button and can see accuracy of the whole rule set by clicking “Calculate Accuracy” button. The strength of rule is calculated to know how many percentages of importance and correctness are in each rule. After calculating strength for all rules, the system assigns predicted decision values into all objects in 30% of testing dataset. So that the accuracy of rule set can be estimated and the rule set with maximum accuracy can be known among generated rules.

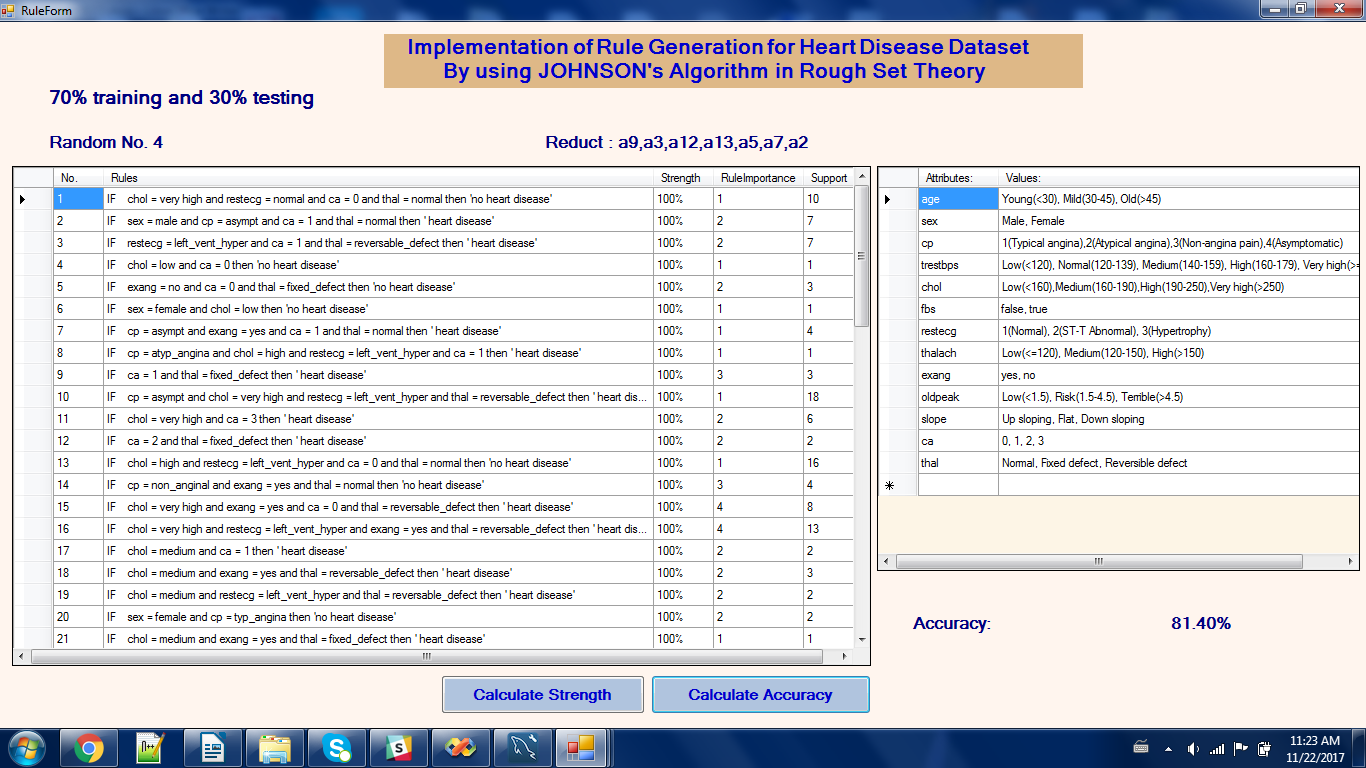


**Figure 4.11 Generate Rules**

The user can see strength of each rule in Figure 4.12 and can see accuracy of the whole rule set in Figure 4.13.



**Figure 4.12 Calculate Strength**



**Figure 4.13 Calculate Accuracy**

Many rule sets with its own accuracy can be generated according to random numbers, percentages of training and testing data.

**CHAPTER 5**

**CONCLUSION**

Many organizations may become to keep increasing data for their daily business operations as long as the time taken. They also need to make decision for new issues by reviewing previous problems and solutions. As the data become increased day by day, it is difficult to search helpful information and accurate knowledge to support decision making process. Decision support system (DSS) are prevalent information systems for decision making. Decision making process is related to quality of information systems. Approaches to decision making are not able to handle the huge amount of real data and the processes are very slow.

Data mining and knowledge discovery is considered as the main module in development of advanced Decision Support System. Data mining is the process of posing various queries and extracting useful information, patterns, and trends often previously unknown from large quantities of data possibly stored in a database. There are many tools in data mining process. Rough set theory is powerful tool for data mining in supporting decision making process. It concentrates on individual objects than population of the objects.

Rough set theory (RST) is an effective tool for mining deterministic rules from a database. It is considered one of the first non-statistical approaches in data analysis. With various concepts of RST like lower and upper approximation, information system etc., objects can be placed in different decision classes as per their features. Reducts can be defined as the information obtained by omitting or neglecting the unwanted data from the system.

The main objective of attribute reduction algorithm is to get reducts from the information system by eliminating redundant information and at the same time have minimum time and space complexity. The need of finding reducts is to remove redundancy and incompatible attributes to obtain key information and thus make decision rules. If these reducts are not removed, then not only the time and space complexity of rule discovery increases but also the quality of the rules discovered degrades. The cost of finding the reducts depends on the size of object and attributes set. Many attribute reduction algorithms have been proposed till now like heuristic algorithm, which based on discernibility matrix, unsupervised and supervised quick reduct algorithm, algorithm on 0-1 integer programming, etc. JOHNSON’s algorithm is used as the attribute reduction algorithm using discernibility matrix. It generates rule forms of knowledge from previous large data by reducing unnecessary attributes and values. This generated knowledge is a useful and meaningful form in helping decision making for users.

**5.1 Limitations**

The system generates decision rules and estimates accuracy of rule set with about 300 patients’ records of heart disease dataset. This is not enough data for the system to classify data and get more effective results.

**5.2 Further Extension**

The number of training dataset may increase to get more effective rule set as efficient system in real world environment. And then the number of attributes may also increase in estimating accuracy of the generating rules.

**REFERENCES**

[1] A Kusiak, Rough Set Theory: A Data Mining Tool for Semiconductor manufacturing, IEEE Transaction on Electronics Packing Manufacturing, Vol.24, No.1, January 2001.

[2] A. hrn. Discernibility and Rough Sets in Medicine: Tools and Applications. PhD thesis, Department of Computer Science and Information Science, Norwegian University of Science and Technology, Trondheim, Norway, February 2000.

[3] Efraim Turban, Jay E. Aronson, Ting-Peng Liang (2008). "Decision Support Systems and Intelligent Systems". p. 574.

[4] J. Błaszczyński, R.Słowiński, M., Szelag "Sequential covering rule induction algorithm for variable consistency rough set approaches” Information Sciences, 181 (2011), pp. 987–1002.

[5] Jensen, Richard, and Qiang Shen. "Rough set based feature selection: A review Rough Computing (2007).

[6] M. Bjanger. Vibration Analysis in Rotating Machinery using Rough Set Theory and ROSETTA. Technical report, Department of Computer Science and Information Science, Norwegian University of Science and Technology, Trondheim, Norway, April 1999.

[7] "State of the Art" in Byte Magazine October 1995, Contains three articles on Data Mining and Data Warehousing.

[8] T.Munakata, Z.Pawlak, Rough Control Application of Rough Set Theory to Control, Fourth European Congress on Intelligent Techniques and Soft Computing, Aachen, Germany, September 2 – 5, 1996, Proceedings, Volume 1, PP. 209 – 218.

[9] W.Frawley, G.Piatetsky, C.Matheus, Knowledge Discovery in Databases: An Overview, AI Magazine, Fall 1992, PP.213-228.