Journal of Engineering and Applied Sciences 11 (3): 488-496, 2016
ISSN: 1816-949X

Rough Set Discretize Classification of Intrusion Detection System

[1]Noor Suhana Sulaiman and [2]Rohani Abu Bakar
[1]Faculty of Computer, Media and Technology Management,
Kolej Universiti TATI, 24000 Kemaman, Terengganu, Malaysia
[2]Faculty of Computer Systems and Software Engineering,
Universiti Malaysia Pahang Lebuhraya Tun Razak, 26300 Kuantan,
Pahang, Malaysia

Abstract: Many pattern classification tasks confront with the problem that may have a very high dimensional feature space like in Intrusion Detection System (IDS) data. Rough set is widely used in IDS to overcome the arising issue. In rough set, there are several stage processing including discretization part which is a vital task in data mining, particularly in the classification problem. Two results distinguish showing that the discretization stage is hugely important in both training and testing of IDS data. In proposed framework, analysis should been done to discretization, reduct and rules stage to determine the significant algorithm and core element in IDS data. The classification using standard voting, since it is a rule-based classification.

Key words: Rough set, equal frequency binning, discretization, classiffication, intrusion detection system

INTRODUCTION

Intrusions into a computing system can be defined as any set of actions that attempt to compromise the integrity, confidentiality or availability of a computer system resource. Intrusion Detection System (IDS) is the process of monitoring computer networks and systems for violations of security policy (the set of laws, rules and practices that define the system boundaries and detail exactly what operations are allowed) (Bace, 2000). The vulnerability of present day software and protocols combined with the increasing sophistication of attacks, disgruntled employees, unethical corporations and even terrorist organizations, consequent to as no surprise that network based attacks are on the rise. As many as 30 internet users fall victim to cyber crime daily with fraud and intrusion cases being the most common. Viruses could be laying dormant in smartphones or computers waiting to copy banking passwords, social media accounts connected to public WiFi maybe vulnerable to hacking while others are still falling for old tricks in the cyber-scamming book. In October 2015, CyberSecurity had received 3,752 cases of online fraud and intrusion and a shocking 191, 096 reports of botnet and malware infections by unique IP addresses (Cheng, 2015). According to recent studies, an average of twenty to forty new vulnerabilities in commonly used networking and computer products are discovered every month (Kendall, 1999). Such wide-spread vulnerabilities in software add to today's insecure computing/networking environment. This insecure environment has given rise to the ever evolving field of intrusion detection and prevention. The cyberspace's equivalent to the burglar alarm, intrusion detection systems complement the beleaguered firewall.

In the past years, there have been several attempts to build taxonomies aimed at classifying attacks. One of the most accepted taxonomy is the one proposed by Kendall (1999) in which attacks can be classified into four categories:

Probing: Attacks oriented to gather information about the system for further intrusion. These attacks include network. Traffic sniffing and port/address scanning.

Denial of Service (DoS): Attacks attempting to diminish or totally interrupt the use of a system or a service to their legitimate users.

User to Root (U2R): Attacks that aim to gain superuser access to the system by means of exploiting vulnerabilities in operating systems or software applications. The attacker has a valid account in the system.

Remote to Local (R2L): Attacks oriented to gain local access from outside the network.

IDSs can be categorized based on different purposes of usage. According to the source of acquired

Corresponding Author: Noor Suhana Sulaiman, Faculty of Computer, Media and Technology Management,
Kolej Universiti TATI, 24000 Kemaman, Terengganu, Malaysia

information, an IDS can be classified as host-based or network-based. A host-based IDS obtains information from an individual host, usually operating system audit trails and system logs. It usually employs an agent that resides on each host to be monitored to dissect event logs, critical system files or network traffic records looking for anomalous changes or suspicious patterns of activities. Network-based system collects data by monitoring the traffic across the network to which the hosts are connected. It will analyze the packets from a network and flags those which look suspicious. Audit data from several hosts may be used as well to detect signs of intrusions.

Based on what kinds of attacks that IDSs will monitor they can be categorized into two groups: misuse and anomaly intrusion detection. Misuse IDSs are used to detect the known malicious activities according to the defined security policy. Anomaly (behavior-based) IDSs perform abnormal detection compared with system or user behavior reference model, assuming the deviation of normal activity under attacks. The anomaly detection is to detect novel attacks which cannot be determined by existing patterns. The advantage of anomaly detection is the ability to detect novel attacks against software, variants of known attacks and deviations from normal usage of programs.

Of late, a variety of published algorithms have been applied to an IDS field. The common used algorithms are neural network (Jian *et al*., 2004; Hofmann *et al*., 2003; Golovko and Kochurko, 2007), genetic algorithm (Sung and Mukkamala, 2003; Shazzad and Park, 2005; Luyin *et al*., 2008), support vector machine (Vapnik, 1995; Cortes and Vapnik, 1995; Mukkamala *et al*., 2002; Ambwani, 2003 and particle swarm optimization (Shi *et al*., 1998; De Castro and von Zuben, 2002).

Neural network: In previous, several Neural Network (NN) techniques such as Multilayer Perceptron Network (MLPN)/Back Propagation Network (BPN) have been applied in many IDS models and have obtained the corresponding detection performance (Starzyk *et al*., 2000; Skowron and Rauszer, 1992). However, the experiments also revealed many problems such as slow convergence, overfitting, local minimum, difficult to determine the number of hidden layers and hidden nodes as well as the quantity and quality of samples have deep impact on generalization ability and accuracy of neural network. The current improvements of MLPN/BPN include using simulated annealing and genetic algorithms to overcome the local minima; using Levenberg-Marquardt algorithm and conjugate gradient algorithm to speed up the training. Nevertheless to a large number of high-dimensional data, all the above improvements will inevitably result in the structure of the network too large to train and increase contradictory in samples so that the network has poor generalization capability and lower accuracy. All these limitations seriously affected the alert reliability and the performance of the NN improvement. In addition, the key of the anomaly detection is to form a user or system profile of the normal activities. In order to achieve data storage, analysis and sharing of components in the cooperative intrusion detection system, the data must be reduced as much as possible so as to reduce the storage and computational cost (Burges, 1998).

MATERIALS AND METHODS

Genetic algorithm: In a feature selection of intrusion detection, the SNFS (Wroblewski, 1995) algorithm used neural network and support vector machine. CFSSGA (Bazan *et al*., 2000) proposed a hybrid algorithm with Correlation-based Feature Selection (CFS) and employed the SVM and genetic algorithm to achieve the optimization of intrusion detection. While, the FSRGA (Wroblewski, 1995) algorithm is based on rough sets and improved genetic algorithms to improve feature selection. Both SNFS and CFSSGA algorithm need data classification for their each iteration. This produces much more time complexity and do not take care of the combination of characteristics as well as the balance of the number and classification accuracy. However, FSRGA algorithm did not optimize the genetic operation which will easily to make the algorithm be trapped into a local optimal solution (Gunn, 1998).

Support vector machine: Support Vector Machine (SVM) (Wang and Chu, 2005; Bose, 2006) is a new kind of machine learning algorithm proposed recently which is based on structural risk minimization of statistical learning theory. Many researchers verified that SVM performed well in intrusion detection classification (Jian *et al*., 2004; Hofmann *et al*., 2003; Golovko and Kochurko, 2007). However, when applying standard SVM on high dimension and large-scale dataset such as network connection dataset it often suffers memory storage and time consuming problem because a standard SVM solver will solve a dual quadratic optimization problem (Huaping *et al*., 2010; Ambwani, 2003). Decreasing the dimension of training samples by feature selection or attribution reduction method is benefit to help alleviate this problem.

Comparing with traditional ANN, SVR possesses prominent advantages such as excellent properties in learning limited samples, good generalization ability, etc.,

SVR is originally developed for solving classification problems and later extended to solve regression problems, and exhibits good learning performance (Takahashi and Abe, 2003). But there exists a problem in the practical application of SVR. This problem is how to select some of SVR parameters so that the performance of SVR can be brought into the best. In (Hofmann *et al*., 2003) Mukkamala and Sung compare performance of ANN and SVMs in intrusion detection. SVMs outperform ANN in speed and scalability and SVMs are relatively insensitive to the number of data points and the classification complexity does not depend on the dimensionality of the feature space. High dimensions of attribute space and parameters to be optimized are often suffered from by SVM for intrusion detection (Shi and Eberhart, 1998).

Particle swarm optimization: Particle Swarm Optimization (PSO) is robust and has been proven theoretically and empirically to be able to search the optimum solution or near-optimal solution to a complex problem. However, if current optimal position is discovered by certain particle, the other particle will draw close to the optimal position rapidly in the process of running. If this optimal position is local optimal point, particle population will not research in the solution space. So, PSO is easy to sink into local optimized solution that is to say, premature phenomena is appeared. Many scholars resolve problems above by combination with particle swarm optimization and genetic algorithms (Sung and Mukkamala, 2003) or selection inertia weight. Berhart and Shi (Shazzad *et al*., 2005) put forward the strategy of linear decreasing inertia weight which is applied in particle swarm optimization algorithm. Although, this strategy improves the performance of particle swarm optimization it is related with iteration times of PSO and cannot really reflect the algorithms' characteristics of complex and nonlinear in the process of running (Forrest *et al*., 1994).

Rough set discretization theory: The rough sets theory was proposed by Pawlak (1982) to deal with uncertain and fuzzy materials and to simplify knowledge. In the rough sets theory, humans use their general knowledge to classify the world around them as abstract or concrete. Everything is classified according to its characteristics and those with nearly identical characteristics may be put into the same group. This is called indiscernible relation, denoted as Ind and is the basis of rough sets theory.

One of the main advantages of rough set theory is that it does not need any preliminary or additional information about data. The main problems that can be approached using rough sets theory include data reduction, discovery of data dependencies, estimation of data significance, generation of decision algorithms from data, approximate classification of data, discovery of patterns in data and discovery of cause-effect relationships (Walczak and Massart, 1999). The following is the concept of rough sets theory (Vapnik, 1995; Cortes and Vapnik, 1995).

Information system: Knowledge can be finished by the information systems, the basic composition of an information system is the set of objects which are to be studied. The knowledge of these objects is described by their attributes and attribute values. The information system is defined as follows:

$$IS = (U, A) \tag{1}$$

where, U is the universe, a finite non-empty set of objects, $U = \{x_1, x_2, ..., x_m\}$ and A is the set of attributes. Each attribute $a \in A$ (attribute a belonging to the considered set of attributes A) defines an information function:

$$f_a : U \rightarrow V_a \tag{2}$$

where, V_a is the set of values of a, called the domain of attribute a. In all attributes, there are decision attributes and condition attributes.

Indiscernible relation: For every set of attributes $B \subseteq A$, an indiscernible relation Ind (B) is defined in the following way: two objects, x_i and x_j are indiscernible by the set of attributes B in A, if $b(x_i) = b(x_j)$ for every $b \in B$. The equivalence class of Ind (B) is called the elementary set in B because it represents the smallest discernible groups of objects. For any element x_i of A, the equivalence class of x_i in relation Ind (B) is represented as $[x_i]_{Ind(B)}$.

Upper and lower approximation: The rough sets approach to data analysis hinges on two basic concepts, namely the lower and the upper approximations of a set, referring to: the elements that doubtlessly belong to the set, and the elements that possibly belong to the set. The definition is shown as follows.

Let X denote the subset of elements of the universe U, the lower approximation of X in B, denoted as $\underline{BX}$ is defined as the union of all these elementary sets which are contained in X. More formally:

$$\underline{BX} = \{x_i \in U \mid [x_i]_{Ind(B)} \subset X\} \tag{3}$$

The above statement is to be read as: the lower approximation of the set X is a set of objects x_i which

belong to the elementary sets contained in X (in the space B), $\underline{BX}$ is called the lower approximation of the set X in B.

The upper approximation of the set X, denoted as $\overline{BX}$ is the union of these elementary sets which have a non-empty intersection with X:

$$\overline{BX} = \{x_i \in U \left| \left[x_i\right]_{Ind(B)} \cap X \neq 0\right.\} \quad (4)$$

The above statement is to be read as: the upper approximation of the set X is a set of objects x_i which belong to the elementary sets that have a non-empty intersection with X, $\overline{BX}$ is called the upper approximation of the set X in B. The difference is called a boundary of X in U:

$$BNX = \overline{BX} - \underline{BX} \quad (5)$$

Discretization: A discretization (De Castro and von Zuben, 2002) replaces value sets of conditional real-valued attributes with intervals. The replacement ensures that a consistent decision system is obtained which assuming a given consistent decision system by substituting original values of objects in the decision table by the unique names of the intervals comprising these values. This substantially reduces the size of the value sets of real-valued attributes. Discrete values have important roles in data mining and knowledge discovery. Rules with discretize values are normally shorter and more understandable and discretization can improve accuracy. A decision table is composed of a 4-tuple as follows:

$$S = <U, Q \{d\} V, f>$$

Where:

U = A finite set of N objects $\{x_1, x_2, ..., x_n\}$

Q = A finite set of n condition attributes $\{q_1, q_2, ..., q_n\}$ (a nonempty set) and d is decision attribute

$$V = U_{q \in Q}$$

where, V_q is a domain of the attribute q. F: U×Q c d 6v is the total decision function called information function such that f (x, q) , V_q for every q , Qcd, x , U. The decision table can be represented as a finite data table in which the columns are labeled by attributes, the rows by objects and the entry in column q_j and row x_i has the valuef (x_i, x_j). Each row in the table describes the information about some objects in S.

Assume $V_q = (l_q, r_q)$dR where R the set of real numbers is and assume that S is consistent decision table. The following notion and description about discretization is referred to reference.

Definition 1: Any pair (q, c) where q0Q and c0R, defines a partition of V_q into left-hand-side and right hand-side interval. The pair (q, c) is called a cut on V_q.

For an attribute q0Q $D_q = \{(q, c^q_1), (q, c^q_2), ..., (q, c^q_{kq})\}$ is composed by all the cuts where k_q0N and $l_q = c^q_0 < c^q_1 < c^q_2 < ... < c^q_{kq} < c^q_{kq+1}$ defines a partition on V_q into subintervals, i.e., $V_q = (c^q_0, c^q_1)$c(c^q_1, c^q_2)c...c(c^q_{kq}, c^q_{kq+1}). Hence, any set of cuts on condition attributes D =cD_a transforms the original decision table $S^D = <U, Q$c{d}, V^D, $f^D>$, into discrete decision table where $f^D(x, q) = I$] f (x, q)0(c^q_1, c^q_{i+1}) and x0U, i {0, 1, ..., k_q}, q0Q.

After discretization, the original decision table is replaced with the new one. And different sets of cuts will construct different new decision table. It is obvious that discretization process is associated with loss of information. Usually the task of discretization is to determine a minimal set of cuts from a given decision table and keeping the discernibility. The selected cuts can be evaluated by the following criteria:

Consistency of D: For any objects u, v0U they are satisfying if u, v are discerned by Q, then u, v are discerned by D.

Irreducibility: There is no DdD, satisfying the consistency.

Optimality: For any D satisfying consistency it follows card (D)#card (D), then D is optimal cuts.

INTRUSION DETECTION SYSTEM DATASET

This study employed KDD Cup dataset which is an IDS benchmark dataset, prepared by the 1998 DARPA intrusion detection evaluation program by MIT Lincoln Lab. Since 1999, KDD CUP 99 has been the most commonly used data set for the testing and training of anomaly detection methods. This data set is constructed based on the data captured in the DARPA 98 IDS evaluation program. DARPA 98 is about 4 gigabytes of compressed raw (binary) tcpdump data of 7 weeks of network traffic which can be processed into about 5 million connection records. KDD training dataset consists of approximately 4,900,000 single connection vectors each of which is made up of 41 features and is labeled as either normal or an attack with only one specific attack type. An example of KDD cup'99 is shown in Fig. 1. There are several text words in the dataset. The system will transform text into numeric values in advance. KDD 99 features can be classified into three main groups:

Fig. 1: The original data of KDD cup'99

Basic features: This category binds all the characteristics that can be extracted from a TCP/IP connection.

Traffic features: This category includes the features that are calculated with respect to a window of interval and is divided into two groups.

Same host features: These examine only the connections in the last 2 sec that have the same destination host as the current connection and calculate statistics dealing with protocol behavior, service, etc.

Same service features: Examine only the connections in the last 2 sec that have the same service as the current connection.

Content features: Unlike most of the DoS and probing attacks, the R2L and U2R attacks do not display any intrusion frequent sequential patterns. This is so because the DoS and Probing attacks involve many connections to some particular host (s) in a very minute period of time; but the R2L and U2R intrusions are embedded in the data field of the packets and normally involve only a single connection. To detect these kinds of attacks its need some features to be able to depict the suspicious behavior in the data field such as number of failed login attempts. These features are called content features.

RESULTS AND DISCUSSION

As a new framework, rough set theory is applied to feature selection and classification of IDS data. It is based on the concept of an upper and a lower approximation of a set, the approximation space and models of set. The rough set approach consists of several steps leading towards the final goal of generating rules from information or decision system. In KDD'99 dataset there are 32 continuous attributions in its condition attributions set. Since, rough set theory is proposed based on set theory, those continuous attributions must be discrete. Discretization amounts to search for cuts points that determine intervals. All values that lie within

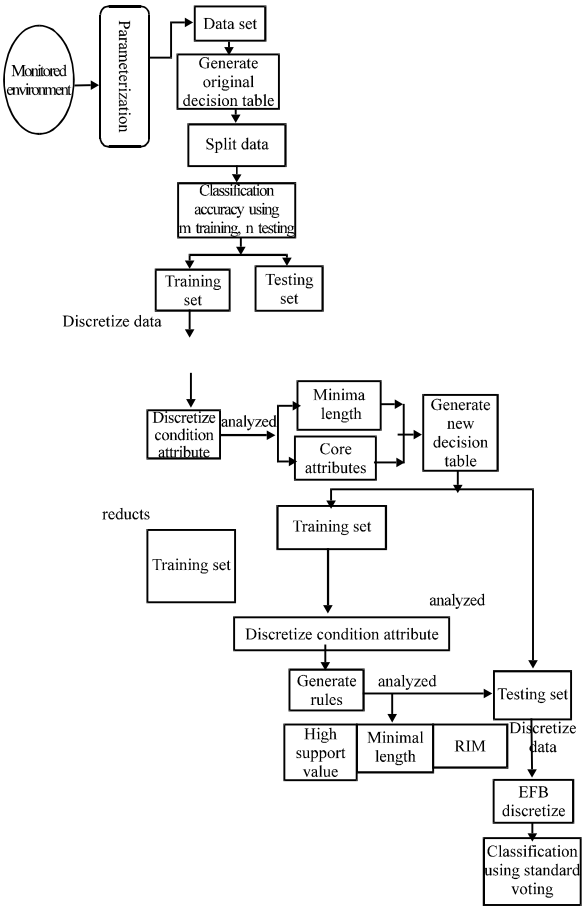

Fig. 2: Rules generation and classification process

each interval are then mapped to the same value. In rough set data analysis, it is done by computing the minimal numbers of attributes; generation of all reducts and generates set of rules. The best reduct are identified if there are two or more reducts with the same number of attributes. The prediction of the new objects is dependent on the type of rules generated. Subsequently, the generated rules and the distance function are adopted for classification of the new object. The general process of rules generation, analysis of rough set reducts and classification for both IDS data is illustrated in Fig. 2.

In this study, KDD CUP 99 contain randomly generated 29,995 records having 41 features are selected. The data are divided into two parts; training and testing group. The training group is split into 70% which equal to 20, 996 records while the testing group is accounted for 30% which equal to 8, 999 records. The significant discretization algorithms are analyzed suits to IDS data. The classification is implemented using standard voting classifier. Figure 2 missing values of the dataset have

Table 1: List of generated reduct

Reduct	Support	Length
{service, flag, src_bytes, dst_bytes, hot, count, srv_count, srv_diff_host_rate, dst_host_count, dst_host_same_srv_rate, dst_host_diff_srv_rate, dst_host_same_src_port_rate, dst_host_srv_diff_host_rate}	100	13
{service, src_bytes, dst_bytes, hot, count, srv_count, srv_diff_host_rate, dst_host_count, dst_host_same_srv_rate, dst_host_diff_srv_rate, dst_host_same_src_port_rate, dst_host_srv_diff_host_rate, dst_host_rerror_rate}	100	14
{service, src_bytes, dst_bytes, hot, count, srv_count, srv_rerror_rate, srv_diff_host_rate, dst_host_count, dst_host_same_srv_rate, dst_host_diff_srv_rate, dst_host_same_src_port_rate, dst_host_srv_diff_host_rate, dst_host_srv_serror_rate}	100	14
{service, src_bytes, dst_bytes, hot, count, srv_count, serror_rate, rerror_rate, srv_diff_host_rate, dst_host_count, dst_host_same_srv_rate, dst_host_diff_srv_rate, dst_host_same_src_port_rate, dst_host_srv_diff_host_rate}	100	14
{service, src_bytes, dst_bytes, hot, count, srv_count, srv_serror_rate, srv_rerror_rate, srv_diff_host_rate, dst_host_count, dst_host_same_srv_rate, dst_host_diff_srv_rate, dst_host_same_src_port_rate, dst_host_srv_diff_host_rate}	100	14
{service, src_bytes, dst_bytes, hot, logged_in, count, srv_count, serror_rate, srv_diff_host_rate, dst_host_count, dst_host_same_srv_rate, dst_host_diff_srv_rate, dst_host_same_src_port_rate, dst_host_srv_diff_host_rate}	100	14
{service, src_bytes, dst_bytes, hot, logged_in, count, srv_count, srv_diff_host_rate, dst_host_count, dst_host_same_srv_rate, dst_host_diff_srv_rate, dst_host_same_src_port_rate, dst_host_srv_diff_host_rate, dst_host_srv_serror_rate}	100	14
{service, src_bytes, dst_bytes, hot, count, srv_count, srv_rerror_rate, srv_diff_host_rate, dst_host_count, dst_host_same_srv_rate, dst_host_diff_srv_rate, dst_host_same_src_port_rate, dst_host_srv_diff_host_rate, dst_host_serror_rate}	100	14
{service, src_bytes, dst_bytes, hot, count, srv_count, serror_rate, srv_rerror_rate, srv_diff_host_rate, dst_host_count, dst_host_same_srv_rate, dst_host_diff_srv_rate, dst_host_same_src_port_rate, dst_host_srv_diff_host_rate}	100	14
{service, src_bytes, dst_bytes, hot, logged_in, count, srv_count, srv_serror_rate, srv_diff_host_rate, dst_host_count, dst_host_same_srv_rate, dst_host_diff_srv_rate, dst_host_same_src_port_rate, dst_host_srv_diff_host_rate}	100	14
{service, src_bytes, dst_bytes, hot, count, srv_count, srv_serror_rate, rerror_rate, srv_diff_host_rate, dst_host_count, dst_host_same_srv_rate, dst_host_diff_srv_rate, dst_host_same_src_port_rate, dst_host_srv_diff_host_rate}	100	14
{duration, service, src_bytes, dst_bytes, hot, count, srv_count, rerror_rate, srv_diff_host_rate, dst_host_count, dst_host_same_srv_rate, dst_host_diff_srv_rate, dst_host_same_src_port_rate, dst_host_srv_diff_host_rate}	100	14
{service, src_bytes, dst_bytes, hot, count, srv_count, rerror_rate, srv_diff_host_rate, dst_host_count, dst_host_same_srv_rate, dst_host_diff_srv_rate, dst_host_same_src_port_rate, dst_host_srv_diff_host_rate, dst_host_srv_serror_rate}	100	14
{service, src_bytes, dst_bytes, hot, logged_in, count, srv_count, srv_diff_host_rate, dst_host_count, dst_host_same_srv_rate, dst_host_diff_srv_rate, dst_host_same_src_port_rate, dst_host_srv_diff_host_rate, dst_host_serror_rate, dst_host_srv_rerror_rate}	100	15
{duration, service, src_bytes, dst_bytes, hot, logged_in, count, srv_count, srv_diff_host_rate, dst_host_count, dst_host_same_srv_rate, dst_host_diff_srv_rate, dst_host_same_src_port_rate, dst_host_srv_diff_host_rate, dst_host_serror_rate}	100	15
{duration, service, src_bytes, dst_bytes, hot, logged_in, count, srv_count, srv_diff_host_rate, dst_host_count, dst_host_same_srv_rate, dst_host_diff_srv_rate, dst_host_same_src_port_rate, dst_host_srv_diff_host_rate, dst_host_srv_rerror_rate}	100	15

been removed by incorporating the incomplete process. Then dataset are split into 70% of training and 30% of testing records. Training dataset has gone through the equal frequency binning discretization. The discretization data generates the discretize condition attribute which is a reducts. The rules are produced for classification process. In stage of reduct, an analysis and evaluation of reduct are done based on choosing minimal cardinality and core attributes in the generated reduct are analyzed. Consequently, a new Decision Table (DT) is constructed based on the attributes consist of reducts with minimal cardinality and core attributes. Second phase of analysis and evaluation is in the rules stage which rules with highest support values, rules with less length and rules with highest percentage of Rule Importance Measure (RIM) are favored. Discretization process by using EFB is done to testing set of new DT. Finally, the classification process is done. Results of classification are compared in terms of classification accuracy of original DT and new DT. Based on Table 1, the core attributes are:

> {service, src_bytes, dst_bytes, hot, count, srv_ count, srv_diff_host_rate, dst_host_count, dst_ host_same_srv_rate, dst_host_diff_srv_rate, dst_ host_same_src_port_rate, dst_host_srv_diff_ host_rate}

The reduct with minimal cardinality also contribute to the connotation reduct in generating the significant rule. It will consider the reduct with minimal cardinality of minimal length. In this experiment the reduct with minimal cardinality is {infectious endocarditis} with length of 13. Based on these core attributes and attributes with minimal cardinality, new decision table are mapped. Subsequently, the rules generated from this new table are analyzed for better classification compared to original KDD Cup 99 dataset without prior analysis on reduct and rules generated.

Original KDD Cup 99 data have 42 attributes including the decision attributes. Nevertheless, the new decision table based on the analyzed reducts reveals 12 condition attributes and 1 decision attribute. These new rules derivation are analyzed based on the rough set benchmark and measurement for better classification than original decision table of KDD Cup 99 dataset. Table 2, only 5 significant rules of KDD Cup 99 Dataset are used to the next classification stage, rather than 18,632 number of rules derivation in old decision table.

Testing data is discretized using EFB algorithm to obtain an equal number of objects into each interval. Based on the generated Equal Frequency Binning algorithm there are 8 bins allocated to the value of numeric numbers of IDS data (Table 3).

Table 2: Significant rules generation of KDD Cup 99 Dataset

Rules	LHS/RHS support	Accuracy	LHS/RHS coverage	LHS/RHS length
service(ecr_i) AND src_bytes([355, *)) AND dst_bytes([*, 281)) AND hot([*, 1)) AND count([17, *)) AND srv_count([21, *)) AND srv_diff_host_rate(0) AND dst_host_count([255, *)) AND dst_host_same_srv_rate(1) AND dst_host_diff_srv_rate(0) AND dst_host_same_src_port_rate([0.09, *)) AND dst_host_srv_diff_host_rate([*, 0.01)) => type_attack(smurf.)	1.1	1.0	1	12.1
service(finger) AND src_bytes([*, 239)) AND dst_bytes([*, 281)) AND hot([*, 1)) AND count([*, 4)) AND srv_count([*, 5)) AND srv_diff_host_rate(0) AND dst_host_count([*, 80)) AND dst_host_same_srv_rate(1) AND dst_host_diff_srv_rate(0) AND dst_host_same_src_port_rate([0.09, *)) AND dst_host_srv_diff_host_rate([0.03, *)) => type_attack(land.)	1.1	1.0	1	12.1
service(imap4) AND src_bytes([355, *)) AND dst_bytes([1696, *)) AND hot([*, 1)) AND count([*, 4)) AND srv_count([*, 5)) AND srv_diff_host_rate(0) AND dst_host_count([255, *)) AND dst_host_same_srv_rate(0) AND dst_host_diff_srv_rate(0) AND dst_host_same_src_port_rate([*, 0.01)) AND dst_host_srv_diff_host_rate([*, 0.01)) => type_attack(imap.)	1.1	1.0	1	12.1
service(http) AND src_bytes([*, 239)) AND dst_bytes([1696, *)) AND hot([1, 3)) AND count([*, 4)) AND srv_count([*, 5)) AND srv_diff_host_rate(1) AND dst_host_count([255, *)) AND dst_host_same_srv_rate(1) AND dst_host_diff_srv_rate(0) AND dst_host_same_src_port_rate([*, 0.01)) AND dst_host_srv_diff_host_rate([*, 0.01)) => type_attack(phf.)	4297.0	0.2	1	12.1
service(telnet) AND src_bytes([239, 355)) AND dst_bytes([281, 1696)) AND hot([1, 3)) AND ` count([*, 4)) AND srv_count([*, 5)) AND srv_diff_host_rate(0) AND dst_host_count([*, 80)) AND dst_host_same_srv_rate(1) AND dst_host_diff_srv_rate(0) AND dst_host_same_src_port_rate([0.09, *)) AND dst_host_srv_diff_host_rate([0.03, *)) => type_attack(loadmodule.)	4297.0 1.1	12.1	1	12.1

Table 3: Value of bin by Equal Frequecy Binning (EFB)

No. of bin	Value of bin
1	(241.5, 367.5)
2	(274.5, 1701.5)
3	(0.5, 3)
4	(3.5, 16.5)
5	(5.5, 22.5)
6	(83.5, 254.5)
7	(0.5, 8.5)
8	(0.5, 2.5)

Table 4: Classification performance of KDD Cup 99 data

Decision table	Rule set	Overall accuracy (%)
Original decision table	All rules	74.3
New decision table	Selected rules	95.4

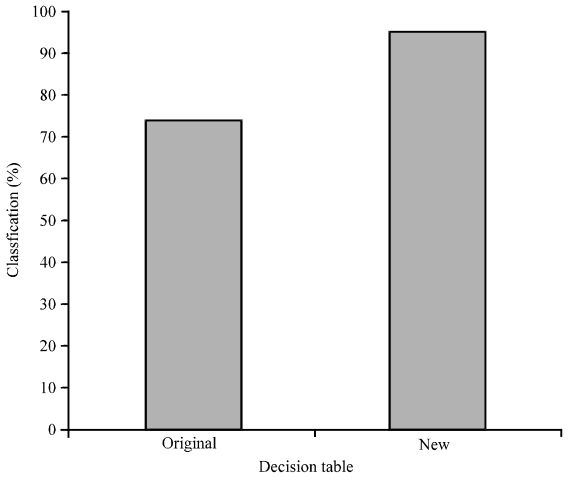

Fig. 3: Accuracy % of original DT and new DT for KDD Cup

In previous analysis it reveals that better classification percentage has been achieved. Based on proposed framework, IDS do not need to use all the attributes and rules to diagnosis the pattern of attacks detected, since these will incur long processing time and no guarantee of better performance. The discretization part is important stage in training and testing part, consequent to better accuracy. Thus, the core attributes and the significant rule are preferred for quick decision making in determining better result for classification. Table 4 shows the result of classification performance of original and new decision table of KDD Cup 99 dataset. Figure 3 illustrates the classification accuracy for original decision table and new decision table of KDD Cup 99 dataset.

CONCLUSION

Main conclusions summarized from the analytical experiments performed in previous are in the following. The classification discretization algorithm have been done in both training and testing data, followed with all empirical phase in rough set, thus proven to have better classification accuracy compared to the results which employ all attributes, reduct and generated rules.

Technical effort has been made in this study to explore the discretization impact in classification technique suit to both training and testing IDS data. Proposed framework including a several analyses of significant attribute, reduct and rules are well determined to probe a better percentage accuracy of IDS classification. The process also involving a set of procedure principally for eliminates missing value and redundant features. Consequently, as a result, high percentage result of new DT classification had been achieved comparing to old DT of IDS data.

Aforementioned, it is a huge influences of using discretization algorithm in the training and testing course of IDS classification. The significant attributes, reduct and rule are preferred for quick decision making in determining better result of classification.

REFERENCES

Ambwani, T., 2003. Multi class support vector machine implementation to intrusion detection. Proceedings of the International Joint Conference on Neural Networks, Volume 3, July 20-24, 2003, Portland, Oregon, USA., pp: 2300-2305.

Bace, R.G., 2000. Intrusion Detection. Macmillan Technical Publishing, Indianapolis, USA.

Bazan, J., H.S. Nguyen, S.H. Nguyen, P. Synak and J. Wroblewski, 2000. Rough Set lgorithms in Classification Problem. In: Rough Set Methods and Applications, Polkowski, L., S. Tsumoto and T.Y. Lin, (Eds.). Physica-Verlag, Heidelberg, New York, pp: 49-88.

Bose, I., 2006. Deciding the Financial Health of Dot-Coms Using Rough Sets. School of Business, University of Hong Kong Hong Kong.

Burges, C.J.C., 1998. A tutorial on support vector machines for pattern recognition. Data Mining Knowl. Discov., 2: 121-167.

Cheng, N., 2015. More than 30 Malaysians fall prey to cyber crime daily. Star Online.

Cortes, C. and V. Vapnik, 1995. Support-vector networks. Mach. Learn., 20: 273-297.

De Castro, L.N. and F.J. von Zuben, 2002. Learning and optimization using the clonal selection principle. IEEE Trans. Evol. Comput., 6: 239-251.

Forrest, S., A.S. Perelson, L. Allen and R. Cherukuri, 1994. Self-nonself discrimination in a computer. Proceedings of the IEEE Computer Society Symposium on Security and Privacy, May 16-18, 1994, Oakland, CA, USA., pp: 202-212.

Golovko, V. and P. Kochurko, 2007. Intrusion Recognition Using Neural Networks. Proceedings of the IEEE Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, September 6-8, 2007, Bulgaria, Dortmund, Germany, pp: 108-111.

Gunn, S.R., 1998. Support vector machines for classification and regression. Technical Report, University of Southamptonn.

Hofmann, A., C. Schmitz and B. Sick, 2003. Rule extraction from neural networks for intrusion detection in computer networks systems. Proeedings of the IEEE International Conference on Systems, Man and Cybernetics, October 5-8, 2003, IEEE Inc., CA., New York pp: 1259-1265.

Huaping, L., J. Yin and L. Sijia, 2010. A new intelligent intrusion detection method based on attribute reduction and parameters optimization of SVM. Proceedings of the 2nd International Workshop on Education Technology and Computer Science, March 6-7, 2010, Wuhan, China, pp: 202-205.

Jian, L., G. Zhang and G. Gu, 2004. The research and implementation of intelligent intrusion detection system based on artificial neural network. Proceedings of the 3rd International Conference on Machine Learning and Cybernetics, August 26-29, 2004, Shanghai, China, pp: 3157-3159.

Kendall, K., 1999. A database of computer attacks for the evaluation of intrusion detection systems. Masters Thesis, Massachusetts Institute of Technology, Cambridge, USA.

Luyin, C., J. Qingshan and C. Lifei, 2008. A feature selection method for network intrusion detection. Comput. Res. Dev. Suppl., 45: 156-160.

Mukkamala, S., G. Janoski and A. Sung, 2002. Intrusion detection using neural networks and support vector machines. Proceedings of IEEE International Joint Conference on Neural Network, May 12-17, Honolulu, HI, USA., pp: 1702-1707.

Pawlak, Z., 1982. Rough sets. Int. J. Comput. Inform. Sci., 11: 341-356.

Shazzad, K.M. and J.S. Park, 2005. Optimization of intrusion detection through fast hybrid feature selection. Proceedings of the 6th International Conference on Parallel and Distributed Computing Applications and Technologies, December 5-8, 2005, Dalian, China, pp: 264-267.

Shi, Y. and R. Eberhart, 1998. A modified particle swarm optimizer. Proceedings of the World Congress on Computational Intelligence and IEEE International Conference on Evolutionary Computation, May 4-9, 1998, Anchorage, AK., pp: 69-73.

Skowron, A. and C. Rauszer, 1992. The Discernibility Matrices and Functions in Information Systems. In: Intelligent Decision Support: Handbook of Applications and Advances of the Rough Sets Theory, S³owinski, R. (Ed.). Kluwer Academic Publisher, Dortrecht, The Netherlands, ISBN: 0792319230, pp: 331.

Starzyk, J.A., D.E. Nelson and K. Sturtz, 2000. A mathematical foundation for improved reduct generation in information systems. Knowl. Inf. Syst., 2: 131-146.

Sung, A.H. and S. Mukkamala, 2003. Identify important features for intrusion detection using support vector machines and neural networks. Proceedings of the 2003 Symposium on Application and the Internet, January 27-31, 2003, IEEE Computer Society Washington, DC, USA., pp: 209-216.

Takahashi, F. and S. Abe, 2003. Decision-tree-based multiclass support vector machines. Proceed. Int. Conf. Neural Inf. Proc., 3: 1418-1422.

Vapnik, V.N., 1995. The Nature of Statistical Learning Theory. Springer-Verlag, New York, ISBN-10: 0387945598.

Walczak, B. and D.L. Massart, 1999. Rough sets theory. Chem. Intell. Lab. Syst., 47: 1-16.

Wang, T.C. and H.L. Chu, 2005. Applying rough sets theory to bank evaluations of loan applicants credit. Proceedings of the 11th International Conference on Industrial Engineering and Engineering Management, April 23-25, 2005, Northeastern University, Shenyang, China, Pp: 726-730.

Wroblewski, J., 1995. Finding minimal reducts using genetic algorithms. Proceedings of the 2nd Annual Join Conference on Information Sciences, September 28-October 1, 1995, Wrightsville Beach, NC., pp: 186-189.