Applicability of Rough Set Technique for Data Investigation and Optimization of Intrusion Detection System

Sanjiban Sekhar Roy[1,*], V. Madhu Viswanatham[1], P. Venkata Krishna[1], N. Saraf[1], A. Gupta[1], and Rajesh Mishra[2]

[1] School of Computing Science and Engineering, VIT University, Vellore, India
[2] School of Information & Communication Technology, Gautam Buddha University, Greater Noida, India
{s.roy,pvenkatakrishna,vmadhuviswanatham}@vit.ac.in,
{nkhlsrf,akku.gupta,raj25mis}@gmail.com

Abstract. The very idea of intrusion detection can be perceived through the hasty advancement following the expansion and revolution of artificial intelligence and soft computing**.** Thus, in order to analyze, detect, identify and hold up network attacks a network intrusion detection system based on rough set theory has been proposed in this article. In this paper we have shown how the rough set technique can be applied to reduce the redundancies in the dataset and optimize the Intrusion Detection System (IDS).

Keywords: Intrusion Detection, Network Attacks, Rough Set Theory.

1 Introduction

One of the core necessities of modern day is a closely protected internet service. It enables transmission of huge amounts of data every day without proper security parameters in place. Thus, in recent years unauthorized access to information has been one of the biggest concerns for many big organizations by putting them at risk. Therefore, efficient detection of such threats has become a high priority task.

In the midst of the arrival of intrusion detection technology, the activities of networks can be exemplified by means of uncertainty, intricacy, variety and vibrant tendencies. An intrusion detection system (IDS) scrutinizes all inbound and outbound network activities and identifies mistrustful patterns that may indicate a network or system attack from someone attempting to break into or compromise a system. IDS can be sort out into two kinds: Network-based systems (NIDS) and Host-based systems (HIDS). Within a HIDS, the system analyzes the activity on every being computer or host. In a network-based configuration, the entity packets flowing all the way through a network are examined. NIDS be able to perceive malicious packets

* Corresponding author.

K. Singh, A.K. Awasthi, and R. Mishra (Eds.): QSHINE 2013, LNICST 115, pp. 479–484, 2013.

that are deliberated to be unnoticed by a firewall's basic filtering rules. Here we have concentrated only on NIDS. This paper implements network intrusion detection system using rough set theory. Intrusion detection data sets are generally very large and can result in redundant records which again results in uncertainty, consequently it isn't possible to sift through all the data manually. In order to produce an efficient dataset, reduction techniques of rough sets have been applied. Roy et al. [2] has proposed a technique for IDS, likewise here also KDD'99 Cup Data set has been used for implementation in this paper**.** It is first, analyzed using Rough Sets [1],[3] during which a rule list is generated. Afterwards these rules act as decisive parameters for determining the threat and notify in case of any intrusion. Intrusion data diminution, rule assortment, feature selection by rough set theory is browbeaten to improve detection exactness, preprocess data and trim down false alarm and illusory alarm.

Our article has been prearranged in the following way. Section 1 explains Rough Set theory followed by Intrusion Detection Systems in Section 2. Thereafter Section 2.1 places of interest the KDD'99 Cup Data set that is used. Implementation of the concept is elaborated in Section 2.2 and 2.3, after which the paper is concluded with the advantages of the system implemented and other future prospects.

1.1 An Overview of the Rough Set Theory

The theory of Rough set is a new mathematical tool to deal with intelligent data analysis and data mining [1] proposed by Z Pawlak. Roy et al. [4] has shown that this theoretical framework is based on the concept that every object in the universe is attached with some kind of information. Set theory is a great help to the computer science research and theory of Rough set is an extension to that. It's a mathematical tool to deal with inexact, uncertain or vague data, which are part of artificial intelligent system. It includes algorithms for generation of rules, classification and reduction of attributes. It is hugely used for knowledge discovery [1], [7] and reduction of knowledge. The theory of Rough set has got many important applications.

1. Information/Decision Systems (Tables) :

   IS is a pair (*U, A*). *U* is a non-empty finite set of objects. *A* is a non-empty finite attributes sets such that $a : U \to V_a$ for every $a \in A$. V_a is called the value set of *a*[1].
2. Indiscernibility:

   Here we say, *IS = (U, A)* is an information system, such that $B \subseteq A$, there is an connected equivalence relation:

   $$IND_{IS}(B) = \{(x, x') \in U^2 \mid \forall a \in B, a(x) = a(x')\}$$

   where, $IND_{IS}(B)$ is known the *B*-indiscernibility relation.
3. Set Approximation :

Again, if *IS = (U, A)* and if $B \subseteq A$ and $X \subseteq U$,then we be able to approximate *X* in *B* by constructing the B-lower and B-upper and boundary region of *X,* can be denoted as

$$\underline{B}X = \{x \mid [x]_B \subseteq X\}$$

$$\overline{B}X = \{x \mid [x]_B \cap X \neq \phi\}$$

$$BN_B(X) = \overline{B}X - \underline{B}X \text{ ,.}$$

4. Reduct and Core :

Let B be a subset of A and let a belong to B.

We say that a is dispensable in B if I(B) = I(B− {a}); otherwise a is indispensable in B.

$$Core(B) = \cap Red(B),$$

Where Red (B) is the set off all reducts of *B*.

2 Intrusion Detection System

An intrusion detection system (IDS) scrutinizes [5], [6] all inbound and outbound network activities and identifies mistrustful patterns that may indicate a network or system attack from someone attempting to break into or compromise a system. We have shown a pictorial view of an intrusion detection system.

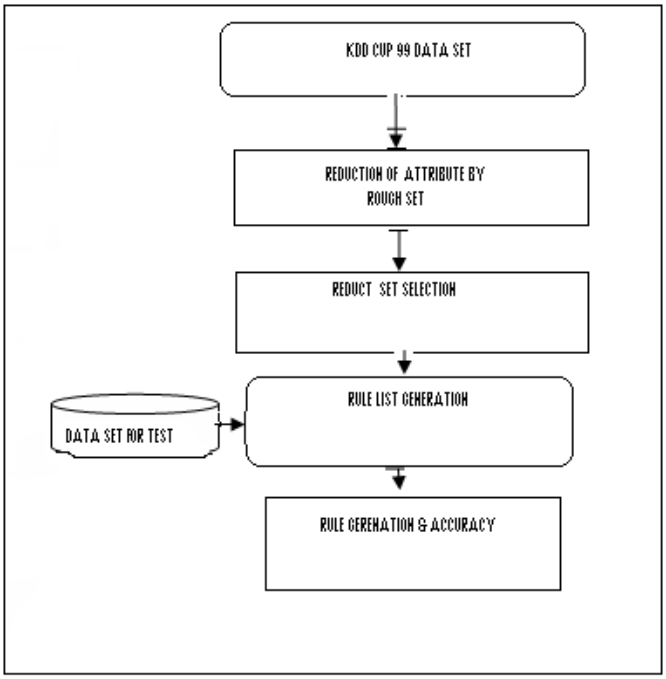

Fig. 1. Intrusion detection system

The following steps are involved in an intrusion detection system.

1. Network data – The network data is collected by analysing and consolidating the network traffic. In this paper, KDD'99 Cup Data set has been used.
2. Attribute reduction and selection of reduct – Since the data set is comparatively huge, it is necessary to filter out the required data from the unclean dataset. This has been done by selecting the core attributes and necessary data from the entire data using rough set reduction theory.
3. Rule generation – Using the reduct and core obtained after applying rough set technique on the data set, decision rules are generated and matched with the already trained history data.
4. Best rule selection – Thus the best rules from the list are selected to form the decision parameters for the system.
5. Alarming – To inform users in case of any intrusion attack or abnormal activity

2.1 KDD'99 Cup Data Set

The tedious job was to create an intrusion detector network, a predictive model having capacity of differentiating between "bad" connections, called intrusions or attacks, and "good" normal connections. This database is consisted of a standard data set required to be audited. This set includes a long range of intrusions replicated within a pro-military environmental network.

The KDD'99 is the most widely used data set for the evaluation of Intrusion Detection Systems. The data set prepared by Stolfo et al. has been populated with about 4 gigabytes of compressed raw tcpdump data captured in DARPA'98 IDS evaluation program. KDD training dataset consists of approximately 4,900,000 single Connection vectors each of which contains 41 features and is labelled as either normal or an attack, with exactly one specific attack type. The simulated attacks fall in one of the following four categories:

1. Denial of Service Attack (DoS): The legitimate user is denied access in this type of attack or the attacker makes the memory/computing resource too busy to respond.

2. User to Root Attack (U2R): the attacker logs in as a normal user initially. However,, he gradually gains root access to the system thus making it vulnerable.

3. Remote to Local Attack (R2L): an attacker who though having the ability to send packets to a machine over a network tends to exploit susceptibility of that machine for achieving local access as a user, for not having an account on that machine.

4. Probing Attack: a step taken to collect information in order to sabotage the network related security.

2.2 Experimental Result

10% of the KDD Data set has been used for implementation to help form the decision algorithm.

Table 1. We have used three types of attributes and they are as follows

Feature name	Description	Type
Duration	length (number of seconds) of the connection	continuous
Protocol type	type of the protocol, e.g. tcp, udp, etc.	discrete
Service	network service on the destination, e.g., http, telnet, etc.	discrete
Source bytes	Number of data bytes from source to destination	continuous
Destination bytes	Number of data bytes from destination to source	continuous
Flag	normal or error status of the connection	discrete

The Decision Attribute of the dataset was selected to be the FLAG attribute.

It can be classified into two values a) Successful b) Rejected based on the intrusion in the network.

Table 2. Contains the Reduct Set

Size	Pos. Reg.	SC	Reducts
5	0.96	1	{duration, protocol_type, service, dst_host_rerror_rate, count}
3	0.96	1	{ dst_host_rerror_rate, src_bytes, count }
3	0.96	1	{ service, src_bytes, count }
4	0.96	1	{ protocol_type, dst_host_rerror_rate, dst_bytes, count }
4	0.96	1	{ protocol_type, service, dst_bytes, count }

2.3 Decision Algorithms

1. If (service = http) & (dst_host_rerror_rate = 0) then (flag = SF) [34 matches]
2. If (protocol_type = udp) then (flag = SF) [32 matches]
3. If (dst_host_rerror_rate = 1) then (flag = REJ) [27 matches]
4. If (count = 2) then (flag = SF) [15 matches]
5. If (src_bytes = 105) then (flag = SF) [13 matches]
6. If (dst_bytes = 147) then (flag = SF) [7 matches]
7. If (protocol_type = tcp) & (service = private) & (count = 1) then (flag = REJ) [3 matches]
8. If (protocol_type = tcp) & (protocol_type = tcp) & (dst_bytes = 0) & (count = 1) then (flag = REJ) [3 matches]

3 Conclusion

The most essential requirement in these days of globalization is the maintenance of a well guarded internet service as that helps in transmitting huge amounts of data every day. Now, as recently unchecked access to information has turned out to be one of the biggest concerns, so development of an efficient technique to detect such threats has become very necessary. Hence, to analyze, detect, identify and hold up network attacks in an effective manner a network intrusion detection system based on rough set theory has been proposed in this article.

Acknowledgment. We would like to thank to the higher ranking management of VIT University for their kind help and encouragement towards our research.

References

1. Pawlak, Z.: Rough Sets. International Journal of Computer and Information Sciences 11, 341–356 (1982)
2. Roy, S.S., Viswanatham, V.M., Krishna, P.V.: Intrusion Detection Data Analysis Using Dominance Based Rough Set Annals. Computer Science Series 10 (2012)
3. Saquer, J., Deogun, J.S.: Concept approximations based on rough sets and similarity measures. International Journal of Applied Mathematics and Computer Science 11, 655–674 (2001)
4. Roy, S.S., Rawat, S.S.S.: Core generation from phone calls data using rough set theory. Annals Computer Science Series 10, 29–32 (2012)
5. Cheng, X., Xiang, B., Zhang, Y.L.: Attribute Reduction Method Applied to IDS, Information engineering Institute, Jingdezhen Ceramic Institute. In: International Conference on Communications and Mobile Computing (2010)
6. Zainal, A., Maarof, M.A., Shamsuddin, S.M.: Feature Selection Using Rough Set in Intrusion Detection. In: IEEE Region 10 Conference, TENCON (2006)
7. Roy, S.S., Gupta, G., Sinha, A., Ramesh, R.: Cancer data investigation using variable precision Rough set with flexible classification. In: Proceedings of the Second International Conference on Computational Science, Engineering and Information Technology, ACM Digital Library, pp. 472–575 (2012)