NETWORK INTRUSION DETECTION BASED ON ROUGH SET AND K-NEAREST NEIGHBOUR

Adebayo O. Adetunmbi*, Samuel O. Falaki, Olumide S. Adewale and Boniface K. Alese
Department of Computer Science, Federal University of Technology

ABSTRACT

Increasing numbers of interconnected networks to the internet have led to an increase in cyber attacks which necessitates the need for an effective intrusion detection system. In this paper, two machine learning techniques: Rough Set (LEM2 Algorithm) and k-Nearest Neighbour (kNN) are used for intrusion detection. Rough set is a classic mathematical tool for feature extraction in a dataset which also generates explainable rules for intrusion detection. The experimental study is done on the international Knowledge Discovery and Data mining tools competition (KDD) dataset for benchmarking intrusion detection systems. In the entire experimentations, we compare the performance of Rough Set with k-Nearest Neighbour. The results generated from the experiment reveal that k-nearest neighbour has a better performance in terms of accuracy but consumes more memory and computational time. Rough Sets classifies at relative short time and employs simple explainable rules.

Keywords: Rough set, intrusion detection, nearest neighbour

IJCIR Reference Format:
Adebayo O. Adetunmbi†, Samuel O. Falaki, Olumide S. Adewale and Boniface K. Alese
Network Intrusion Detection based on Rough Set and k-Nearest Neighbour. International Journal of Computing and ICT Research, Vol. 2, No. 1, pp. 60 - 66. http:www.ijcir.org/volume1-number2/article7.pdf.

1. INTRODUCTION

Accompany the benefits of Internet are various techniques to compromise the integrity and availability of the system connected to it due to flaws in its protocols and software widely entrenched. An Intrusion detection system (IDS) is required in addition to the preventive defense mechanisms such as firewall for another layer of protection. Intrusion detection is a process of detecting security breaches by examining events occurring in a computer system.

Intrusion is defined as any set of action that attempt to compromise the integrity, confidentiality or availability of system resources (Adetunmbi *et al*, 2006). Basically, there are two approaches to intrusion detection model as described in (Biswanath *et al,* 1994): Misuse detection model refers to detection of intrusions that follow well-defined intrusion patterns. It is very useful in detecting known attack patterns. Anomaly detection model refers to detection performed by detecting changes in the patterns of utilization or behavior of the system. It can be used to detect known and unknown attack.

IDSs are also classified as network-based or host-based in terms of source of data. The former collect raw network packets as the data source from the network and analyze for signs of intrusions Host-based IDS operates on information collected from within an individual computer system such as operating system audit trails, C2 audit logs, and System logs (Sundaram, 1996; Byunghae *et al,* 2005).

* Author's Address: Adebayo O. Adetunmbi*, Samuel O. Falaki, Olumide S. Adewale and Boniface K. Alese

 International Journal of Computing and ICT Research, ISSN 1818-1139, Vol.2, No.1, pp. 60 - 66, June 2008.

Majority of the IDS entrenched today are either rule-based or expert-system based. Their strengths depend largely on the ability of the security personnel that develops them. The former can only detect known attack types and the latter is prone to generation of false positive alarms. Hence, the need for intelligence techniques known as machine learning techniques which automatically learn from data or extract useful pattern from data as a reference for normal/attack traffic behaviour profile from existing data for subsequent classification of network traffic. Promising researches in this area include among others the work of Lee *et al* (1999), Alan, *et al* (2002); Byunghae, *et al* (2005); Sanjay, *et al* (2005).

Enormous amounts of data are collected from the network for network based intrusion detection. This poses a great challenge. Raw network traffic needs to be summarized into higher-level events, described by some features, such as connection records before feeding the data to a machine learning algorithm. Selecting relevant features is a crucial activity and requires extensive domain knowledge.

Various machine learning techniques have been applied to the design of IDS. Among these are neural networks, linear genetic programming, Support vector machines, Bayesian Networks, Multivariate adaptive regression splines, and Fuzzy inference systems (Peddabachigari, *et al*; 2005). Also, many data mining approaches, including discovering association rules, have been applied to intrusion detection (Lee *et al*, 1999). Wang and He (2006) used the hybrid approach of Rough Set theory and Association Rule Mining to improve the accuracy of the intrusion detection. Recently, Andrew and Mukkamala (2003) have used support vector machine and neural network to identify important features for intrusion detection.

In this paper, Rough Set is used in building an intrusion detection model subsequently used for classifying unseen network traffic. Relevance features extracted by Rough Set are then used for classifying Network traffic either as normal or attack. Also, the entire features are used by the k-Nearest neighbour and the results obtained reported.

The rest of the paper is organized as follows. Section 2 provides a brief introduction to the dataset used. In section 3, Basic concepts of rough set theory and Nearest Neighbour are described. The experimental results are reported in Section 4 followed by conclusion in Section 5.

2. INTRUSION DATA SET

The KDD Cup 1999 dataset used for benchmarking intrusion detection problems is used in our experiment. The dataset was a collection of simulated raw TCP dump data over a period of nine weeks on a local area Network. The training data was processed to about five million connection records from seven weeks of network traffic and two weeks of testing data yielded around two million connection records. The training data is made up of 22 different attacks out of the 39 present in the test data. Table 1 shows the different attack types for both training (known) and the additional attack types included for testing (novel) for the four categories. The known attack types are those present in the training dataset while the novel attacks are the additional attacks in the test datasets not available in the training data sets. The attacks types are grouped into four categories:

(1). DOS: Denial of service – e.g. syn flooding
(2). Probing: Surveillance and other probing, e.g. port scanning
(3). U2R: unauthorized access to local super user (root) privileges, e.g. buffer overflow attacks.
(4). R2L: unauthorized access from a remote machine, e.g. password guessing

The training dataset consisted of 494,021 records among which 97,277 (19.69%) were normal, 391,458 (79.24%) DOS, 4,107 (0.83%) Probe, 1,126 (0.23%) R2L and 52 (0.01%) U2R connections. The testing dataset is made up of 311,029 records out of which there were 60,593 (19.48%) normal, 229,853 (73.90%) DOS, 4,166 (1.34%) Probe, 16,189 (5.21%) R2L and 228 (0.07%) U2R. The test and training data are not from the same probability distribution. In each connection are 41 attributes describing different features of the connection (excluding the class attribute), and a label assigned to each either as an attack type or as normal.

DOS	Probe	R2L	U2R
Known			
Back, land, Neptune, Pod, smurf, teardrop	ipsweep, satan, nmap, portsweep	ftp_write, guess_passwd, warezmaster, warezclient, imap, phf, spy, multihop	rootkit, loadmodule, buffer_overflow, perl
Novel			
apache2, udpstorm, processtable, mailbomb	Saint, mscan	named, xlock, sendmail, xsnoop, worm, snmpgetattack, snmpguess	xterm, p.s., sqlattack, httptunnel

Table 1: Known and novel attack types

3. BASIC CONCEPTS OF ROUGH SETS AND K-NEAREST NEIGHBOUR

3.1 Basic Concepts of Rough Set

Rough set theory (RST) is a useful mathematical tool to deal with imprecise and insufficient knowledge, find hidden patterns in data, and reduce dataset size (Pawlak, 1982; Komorowski, *et al,* 1998). Also, it is used for evaluation of significance of data and easy interpretation of results. RST contributes immensely to the concept of reducts. Reducts is the minimal subsets of attributes with the most predictive outcome. Rough Set is a machine learning method which generates rules based on examples contained within an information table. Rough set theory has become well established as a mechanism for solving the problem of how to understand and manipulate imprecise and insufficient knowledge in a wide variety of applications related to artificial intelligence.

Let K = (U,C) be an approximation space, where U is a non-empty, finite set called the universe; A subset of attributes $R \subseteq C$ defines an equivalent on U. Let $[x]_R$ ($x \in U$) denote the equivalence class containing x.

Given $R \subseteq C$ *and* $X \subseteq U$. X can be approximated using only the information contained within R by constructing the R-lower and R-upper approximations of set X defined as:

$$\underline{R}X = \{x \in X \mid [x]_R \subseteq X\}$$
$$\overline{R}X = \{x \in X \mid [x]_R \cap X \neq 0\} \qquad \text{where}$$

$\underline{R}X$ is the set of objects that belong to X with certainty, and $\overline{R}X$ is the set of objects that possibly belong to X. The R-positive region of X is $POS_R(X) = \underline{R}X$, the R-negative region of X is $NEG_R(X) = U - \overline{R}X$, and the boundary or R-borderline region of X is $BN_R(X) = \overline{R}X - \underline{R}X$. X is called R-definable if and only if $\overline{R}X = \underline{R}X$. Otherwise $\overline{R}X \neq \underline{R}X$ and X is rough with respect to R iff $\underline{R}X \neq \overline{R}X$.

The approximation measure $\alpha_R(X)$ is defined as

$$\alpha_R(X) = \frac{|\underline{R}X|}{|\overline{R}X|}$$

where $X \neq \phi$, and $|X|$ denotes the cardinality of set X.

Algorithm LEM2 below developed by Grzymala-Busse (1997) is used in building an intrusion detection model.

LEM2 Algorithm

Input: K set of objects
Output: R set of rules

```
begin
   G = K;
   R = ϕ;
   While G ≠ϕ do
      begin
      C ≠ϕ
      C(G) = {c: [c]∩G≠ϕ};
      While(C ≠ϕ) or (!([C] ⊆ K)) do
        begin
        select a pair c ∈C(G) such that |[c] ∩ G| is maximum;
        if ties, select a pair c ∈C(G) with the smallest cardinality |[c]|;
         if further ties occur, select the first pair from the list;
         C = C ∪ {c}; G = [c] ∩ G;
         C(G) = {c:[c]∩G≠ϕ};
         C(G) = C(G) – C;
         end;
         for each elementary condition c ∈ C do
           if |C – c| ⊆ K then C = C – {c};
          create rule r basing the conjuction C and add it to R;
           G = K – ⋃_{r∈R} | R |
         end;
         for each r ∈ R do
         if ⋃_{s∈R–r} | S |= K then R = R –r
end
```

Figure 1: LEM2 Algorithm

3.2 Nearest Neighbour

Nearest Neighbour (NN) is an easy classification technique, classifies new observations into their appropriate categories by simply searching for similar or closest instances in the well known classified observations (training data set). Closeness is defined in terms of a distance metric, such as Euclidean distance. The Euclidean distance between two points or tuples, say $X_1 = (x_{11}, x_{12},\ldots, x_{1n})$ and $X_2 = (x_{21}, x_{22},\ldots, x_{2n})$, is

$$dist(X_1, X_2) = \sqrt{\sum_{i=1}^{n} (x_{1i} - x_{2i})^2}$$

NN consists of training and testing phases like other supervised learning technique. In the training phase, data points are given in an n-dimensional space with associated labels designating their class. In the testing phase, unlabeled data are given and the algorithm classifies new objects (network traffic) by choosing the class of the nearest neighbour or the most common class in the nearest neighbour (kNN) in the training set as measured by distance metric. Since the introduction of NN by Fix and Hodges (1951), it has been used and improved upon (Bay, 1998), and employed in many domains, such as UCI datasets repository (Hettich and Bay, 1999). We are interested in using the NN classifier to compute all possible distance pairs between all the training data set and the test data set records.

Min-max normalization is used for data transformation of continuous attribute values in the range [0, 1] by computing.

$$v' = \frac{v - \min_A}{\max_A - \min_A}$$

where $\min_A$ and $\max_A$ are the minimum and maximum values of attribute A.

For categorized attributes; a simple method adopted is to compare the corresponding value of the attribute in tuple X_1 with that in tuple X_2. If identical, the difference between the two is taken as zero else as 1.

3.3 Preprocessing

Since Rough set only works on discrete features just like other machine learning techniques like nearest neighbour and decision trees among others; continuous features in the dataset are discretized based on Entropy, a supervised splitting technique (Shannon, 1948; Jiawei and Micheline, 2006). Entropy is used to determine how informative a particular input attribute is about the output attribute for a subset. It is computed as follows:

i. Let D be a set of training data set defined by a set of attributes with their corresponding class labels.

ii. A split-point for an attribute *A* within the set can partition the tuples in D into two subsets satisfying the conditions $A \leq$ split_point and $A >$ split_point respectively.

iii. $Info_A(D) = \frac{|D_1|}{D} Entropy(D_1) + \frac{|D_2|}{D} Entropy(D_2)$ where D_1 and D_2 correspond to the tuples in D satisfying the conditions $A \leq$ split_point and $A >$ split_point respectively.

iv. $Entropy(D_1) = -\sum_{i=1}^{m} P_i \log_2(P_i)$, where P_i is the probability of C_i in D_1, determined by dividing the number of tuples of C_i in D_1 by $|D_1|$, the total number of tuples in D_1.

In selecting a spilt-point for attribute A, pick an attribute value that gives the minimum information required. This process is performed recursively on an attribute until the information requirement is less than a small threshold.

4. EXPERIMENTAL RESULTS

4.1 Experiments using Rough Set Theory

This section presents experimental results obtained when directly applying the two methods discussed in section 3 on the testing dataset, made up of 311,029 records. Here, we are only interested in knowing to which category (normal, DOS, R2L, U2R, Probing) a given connection belongs. The accuracy of each experiment is based on percentage of successful classification (PSC) on the test dataset, where

$$PSC = \frac{number\ of\ correctly\ classified\ instances\ in\ the\ testset}{number\ of\ instances\ in\ the\ testset}$$

Table 2 presents the confusion matrix of the first experiment performed when rules induced by LEM2 algorithm (Rough Set) are used on the test dataset. The information system is first discretized by the Entropy described in Section 3.3 followed by building a classification model for all the attack types and normal (rule induction) LEM2 Algorithm. The performance of the classifier is then measured using the test data set.

Predicted as Actual	Normal	Probing	DOS	U2R	R2L
Normal(60593)	**99.24%**	0.75%	0.00%	0.005%	0.005%
Probing(4166)	42.68%	**55.83%**	1.25%	0.24%	0.00%
DOS(229853)	6.32%	0.06%	**93.61%**	0.00%	0.01%
U2R(228)	90.35%	1.32%	0.00%	**8.33%**	0.00%
R2L(16189)	99.88%	0.00%	0.00%	0.04%	**0.08%**
PSC = 89.34%					

Table 2 Confusion Matrix

Table 2: Confusion matrix obtained with Rough Set classification model

From Table 2, we can see that the last two classes R2L and U2R are not well detected. The low presence of these two classes of attacks in the training dataset accounts for the poor detection. The percentage of R2L and U2R in the dataset are 0.23% (1,126 records) and 0.01 (52 records) respectively.

4.2 Experiments using k-Nearest Neighbour

Predicted as Actual	Normal	Probing	DOS	U2R	R2L
Normal(60593)	**99.47%**	0.24%	0.29%	0.00%	0.00%
Probing(4166)	13.12%	**74.10%**	12.28%	0.00%	0.50%
DOS(229853)	2.81%	0.15%	**97.04%**	0.00%	0.00%
U2R(228)	39.96%	18.80	32.01	**6.60**	2.63
R2L(16189)	95.55%	2.76%	0.20%	0.24%	**1.25%**
PSC = 92.63%					

Table 3 Confusion Matrix With k-nearest Neighbours

Table 3 shows the confusion matrix obtained with k-nearest neighbour using the whole 41attributes and the value of k equals 3.

kNN also records poor detection in the last two classes. In the two experiments, kNN outperform Rough set algorithm in the detection of all attack types except U2R. kNN algorithm has overall accuracy of 92.63% against 89.34% using Rough set.

5. CONCLUSION

In this paper, we presented the performance of rough set and k-nearest neighbour algorithms on intrusion detection for comparisons. The two algorithms performed poorly on U2R and R2L due to their few representations in the training dataset. However, the attribute values in a training data set completely differ from the attribute values from the test dataset mostly for these two attack types. This leads to wrong classification because these instances are not learned in the training phase.

Also pertinent is that rough set is three times faster during classification than k-nearest neighbour. As our future work, we intend to explore other machine learning techniques, supervised or unsupervised, to improve the detection accuracy; most especially for the last two classes (R2L and U2R).

REFERENCES

ADETUNMBI A..O., ZHIWEI S., ZHONGZHI S., AND ADEWALE O.S. 2006. Network Anomalous Intrusion Detection using Fuzzy-Bayes, in IFIP International Federation for Information Processing, Volume 228, Intelligent Information Processing III, Eds. SHI Z., SHIMOHARA K., FENG, D., (Boston: Springer) pp. 525 – 530.

ANDREW H. S. AND MUKKAMALA, S. 2003. Identifying important features for intrusion detection using support vector machines and neural networks". IEEE Proceedings of the Symposium on

Application and the Internet (SAINT ‘ 03).
BISWANATH, M., TODD L.H., AND KARL, N.L. 1994. Network Intrusion Detection. IEEE Network, 8(3): 26-41.
BYUNGHAE-CHA, K.P. AND JAITTYUN, S. 2005. Neural Networks Techniques for Host anomaly Intrusion Detection using Fixed Pattern Transformation. ICCSA 2005, LNCS 3481 pp. 254-263.
ESKIN, E., ARNOLD, A., PRERAU, M., PORTNOY, L AND STOLFO, S. 2003. A Geometric framework for unsupervised anomaly detection: Detecting intrusions in unlabeled data. Applications of Data Mining in Computer security.
FIX E. AND HODGES J.L. 1951. Discrimination analysis: Non parametric discrimination: Consistency properties. Technical report 21-49-004, USAF School of Aviation Medecine, Randoff Field, Texas.
GRZYMALA-BUSSE J.W. 1997. A New Version of the Rule Induction System LERS. Fundamenta Informaticae, 31(1) pp. 27-39.
HETTICH, S. AND BAY, S.D. 1999. The UCI KDD Archive. Available at http://kdd.ics.uci.edu.
JIAWEI, H. AND MICHELINE, K. 2006. Data Mining Concepts and techniques, second edition, China Machine Press, pp. 296 -303.
KDD CUP 1999 DATASET: http://kdd.ics.uci.edu/databases/kddcup99/
KOMOROWSKI, J., POKOWSKI, L. AND SKOWRON, A. 1998. Rough Sets: A Tutorial citeseer.ist.psu.edu/komorowski98rough.html
LEE, W., STOLFO, S.J. AND MOK, K. 1999. Data Mining in work flow environments: Experiments in intrusion detection. In Proceedings of the 1999 Conference on Knowledge Discovery and Data Mining.
PAWLAK, Z. 1991 Rough Sets: Theoretical Aspects of Reasoning About Data. Kluwer Academic Publishing, Dordreiht.
PEDDABACHIGARI, S., ABRAHAM, A., GROSAN, C., AND THOMAS, J. 2005. Modelling Intrusion detection using hybrid Intelligent systems, Journal of Network and Computer Applications, Elsevier science.
SANJAY, R., GULATI, V.P. AND ARUN, K.P. 2005. A Fast Host-Based Intrusion Detection System Using Rough Set Theory in Transactions on Rough Sets IV, LNCS 3700, 2005, pp. 144 – 161.
SHANNON, C.E. (1948). A mathematical theory of communication, bell System technical Journal 27: 379-423 and 623 – 656.. http://cm.bell-labs.com/cm/ms/what/shannonday/paper.html
SUNDARAM, A. 1996. An Introduction to Intrusion detection ftp.cerias.purdue.edu/pub/doc/intrusion_detection/Intrusion-Detection-Intro.ps.Z.
WANG, X. AND HE, F. 2006. Improving Intrusion Detection Performance Using Rough Set Theory and Association Rule Mining, IEEE International Conference on Hybrid Information Technology.