Intrusion Detection System Based on Decision Tree and Clustered Continuous Inputs

Adel S. Issa
College of Education
University of Duhok, Iraq

Received on: 4 / 4 / 2010 **Accepted on: 25 / 10 / 2010**

ABSTRACT

With the rapid expansion of computer networks during the past decade, security has become a crucial issue for computer systems. Different soft-computing based methods have been proposed in recent years for the development of intrusion detection systems (IDSs). The purpose of this paper is to use ID3 algorithm for IDS and extend it to deal not only with discreet values, but also with continuous ones, by using K_mean algorithm to partition each continuous attribute values to three clusters. The full 10% KDD Cup 99 train dataset and the full Correct test dataset are used. The results of the proposed method show an improvement in the performance as compared to standard ID3 using classical partition method.

Keywords: Intrusion Detection System, Decision Tree, Clustered Continuous Inputs.

نظام كشف التطفل القائم على شجرة القرار والمدخلات المستمرة المعنقدة

عادل عيسى
كلية علوم الحاسوب والرياضيات، جامعة الموصل

تاريخ استلام البحث: 4 / 4 / 2010 **تاريخ قبول البحث: 25 / 10 / 2010**

الملخص

مع التوسع السريع للشبكات خلال العقد الماضي، أصبحت نظم الحماية من أهم مسائل نظم الحاسبات. في السنوات المنصرمة الأخيرة تم اقتراح وتصميم مجموعة من الأفكار اعتمدت على تقنيات الـ Soft-Computing لتصميم نظم كشف التطفل. يهدف البحث استخدام خوارزمية ID3 لبناء نظام كشف التطفل وتطوير هذه الخوارزمية لتتعامل ليس فقط مع القيم المنفصلة (Discreet)، بل لتتعامل أيضا مع القيم المستمرة (Continuous)، وذلك باستخدام خوارزمية K_mean لتقسيم قيم الحقول المستمرة إلى ثلاثة عناقيد. اعتمد البحث على البيانات ألـ 10% KDD Cup 99 الكاملة للتدريب والفحص. أظهرت النتائج كفاءة الخوارزمية المقترحة مقارنة مع خوارزمية ID3 القياسية عند استخدام طريقة التقسيم التقليدية.

الكلمات المفتاحية: نظام كشف التطفل، شجرة القرار ، المدخلات المستمرة المعنقدة.

1. Introduction

Rapid development and expansion of World Wide Web and local network systems have changed the world of computer applications in the last decade. However, this outstanding achievement has an Achilles' heel: The highly connected computing world has also equipped the intruders and hackers with new facilities for their destructive purposes. The costs of temporary or permanent damages caused by unauthorized access of the intruders to computer systems have urged different organizations to increasingly implement various systems to monitor data flow in their networks [1]. These systems are generally referred to as Intrusion Detection Systems (IDSs). There are two main approaches to the design of IDSs. In a misuse detection based IDS, intrusions are detected by looking for activities that correspond to known signatures of intrusions or vulnerabilities. On the other hand, an anomaly detection based IDS detects intrusions by searching for abnormal network traffic. The abnormal traffic pattern can be defined either as the violation of accepted thresholds for frequency of events in a connection or as a user's violation of the legitimate profile developed for his/her normal behavior.

Several soft-computing were proposed in recent years for the development of IDS. Arman Tajbakhsh, et al., 2009 [2] proposed a new intrusion detection framework based on classification algorithm using fuzzy association rules for building classifiers.The fuzzy association rulesets are exploited as descriptive models of different classes. The method proposed to speed up the rule induction algorithm. Victor H. et al., 2006 [3] proposed the use of ID3 to Web attack detection. The DT was made to classify a number of not previously considered Web application queries. The results show that the ID3 is an effective means for detecting and classifying web application attack queries. Yacine Bouzida and F. Cuppens 2006 [4] proposed two different techniques for anomaly intrusion namely NN and DT in order to detect new attacks that are not present in the training data set. They improve them for anomaly intrusion detection and test them over the KDD Cup 99 data sets and over real network traffic **i**n real time. Mehdi Moradi and Mohammad Zulkernine 2004 [5], the paper presents a NN approach to intrusion detection. A multi-layer perceptron is used for intrusion detection based on an off-line analysis approach and applying the early stopping validation method on the proposed NN. Rachid Beghdad 2008 [6] aimed to determine which of the NN classifies well the attacks and leads to a higher detection rate of each attack. The paper focused on two classification types of records: a single class (normal, or attack), and a multiclass, where the category of attack is also detected by the NN. Five different types of NNs were tested: Multi-Layer Perceptron (MLP), generalized feed forward (GFF), radial basis function (RBF), self-organizing feature map (SOFM), and principal component analysis (PCA) NN. Yuehui Chen et al., 2007 [7] proposed an IDS model based on a general and enhanced Flexible Neural Tree (FNT). Based on the predefined instruction/operator sets, the framework allows input variables selection. Over layer connections and different activation functions for the various nodes involved.

This paper deals with use and extension of ID3 algorithm for IDS to process both continuous and discrete values. This is accomplished by portioning each continuous attribute values to three clusters by the use of K_mean algorithm.

Besides this introduction, the rest of the paper is organized as follows:
Section 2 presents an introduction to decision trees and ID3 algorithm. Section 3 describes the motivation and the proposed work. The section also deals with the dataset, evaluation criteria, and the features used for classifying network connection records in this study. Section 4 presents the experimental results. Section 5 gives some conclusions.

2. Introduction to Decision Trees and ID3 Algorithm

2.1. Decision Trees (DTs)

DTs fall under the subfield of machine learning within the larger field of artificial intelligence. It is a classifier expressed as a recursive partition of the instance space. The DT consists of nodes that form a rooted tree, meaning it is a directed tree with a node called a "root" that has no incoming edges referred to as an "internal" or "test" node. All other nodes are called "leaves" (also known as "terminal" or "decision" nodes). In the DT, each internal node splits the instance space into two or more sub-spaces according to a certain discrete function of the input attribute values. In the simplest and most frequent case, each test considers a single attribute, such that the instance space is partitioned according to the attributes value [8, 9, 10].

The classical methods of attribute selection, implemented in well-known algorithms ID3 and C4.5 [1], are based on minimizing the entropy or information gain, i.e. the amount of information represented by the clusters of records covered by nodes created upon the selection of the attribute [11, 12, 13]. It should be noted here that only ID3 will be covered in this paper.

2.2. The Interactive Dichotomizer 3 (ID3) Algorithm

ID3 is a basic algorithm developed by Ross Quinlan (1983) [1]. Is is a simple inductive, non-incremental, classification algorithm. Using a top down, greedy search through a fixed set of examples, it builds a decision tree, which is then applied for classifying future samples. Each example has several attributes and belongs to a class. Each non-leaf node of the decision tree is a decision node, while each leaf node corresponds to a class name. ID3 extends the concept learning system algorithm adding a feature selection heuristic [14, 15, 16].

Feature selection is used to identify the attribute that best separates the set of input examples, called the training set. If the selected attribute completely classifies the training set, then the mission is finished. Otherwise, ID3 is recursively applied, in a greedy fashion, to identify the next best attribute. When deciding which attribute is the best, ID3 uses a measure called information gain. Information gain is defined in terms of the amount of information portrayed by an attribute. That amount in information theory is called entropy [15, 16, 17].

3. The Proposed Work

The initial definition of ID3 is restricted in two manners to attributes that take on a discrete set of values. First, the target attribute whose value is predicted by the learned tree must be discrete valued. Second, the attributes tested in the decision nodes of the tree must also be discrete valued. The second restriction can easily be removed so that continuous-valued decision attributes can be incorporated into the learned tree. This can be accomplished by dynamically defining new discrete-valued attributes that partition the continuous attribute values into a discrete set of intervals.

Adel Sabry [20] uses the classical partition method to partition continuous values to three intervals as follows:

- First: determining the maximum and minimum values for each continuous item.
- Second: partitioning the item domain to three parts depending on maximum and minimum value as shown below:

Part 1: $Part\_1 \leq [min + (max – min) / 3]$
Part 2: $[min + (max – min)]/ 3] < Part\_2 \leq [max – (max – min) / 3]$
Part 3: $Part\_3 > [max - (max – min)/3]$

where max and min represent respectively, maximum and minimum values extracted from attributes domain.

This method cannot represent data accurately because data may not be necessarily similarly distributed in this three interval. For this reason K_mean algorithm is used here to partition each continuous attribute to three groups. By this way data will be represented more accurately than that in classical method. For example: attribute A = {2, 3, 5, 7, 10, 20, 30, 40, 100, 120}, if the classical method is used, then:

Part_1 = {2, 3, 5, 7, 10, 20, 30, 40}.
Part_2 = { }.
Part_3 = {100, 120}.

On the other hand, using K_mean algorithm with k = 3, then:

Group A = {2, 3, 5, 7, 10}.
Group B = {20, 30, 40}.
Group C = {100, 120}.

As can be seen, when using K_mean algorithm, the groups represent the nature of data more accurately.

Functionality of the proposed system is divided into four phases:

- Input Data and Partition process.
- Labeling continuous values.
- Training.
- Classification.

In the first phase, the continuous attributes will be partitioned to 3 groups (A, B, C) by applying K_mean algorithm on input data. In the second phase, the data is converted into suitable input data by assigning each continuous value to one of three groups (A, B, C) so that the input is given to ID3 algorithm. In the training phase, the system gathers knowledge about the normal and attacks from the preprocessed input data, and store the acquired knowledge. In classification phase, the system detects normal behavior or specific attack based on the knowledge, which is achieved during the training phase. The main task is to generalize and classify each connection record to one of the five classes considered in the KDD Cup 99 dataset (Normal, Probing, DoS, U2R and R2L). Figure (1) describes the block diagram of the proposed system.

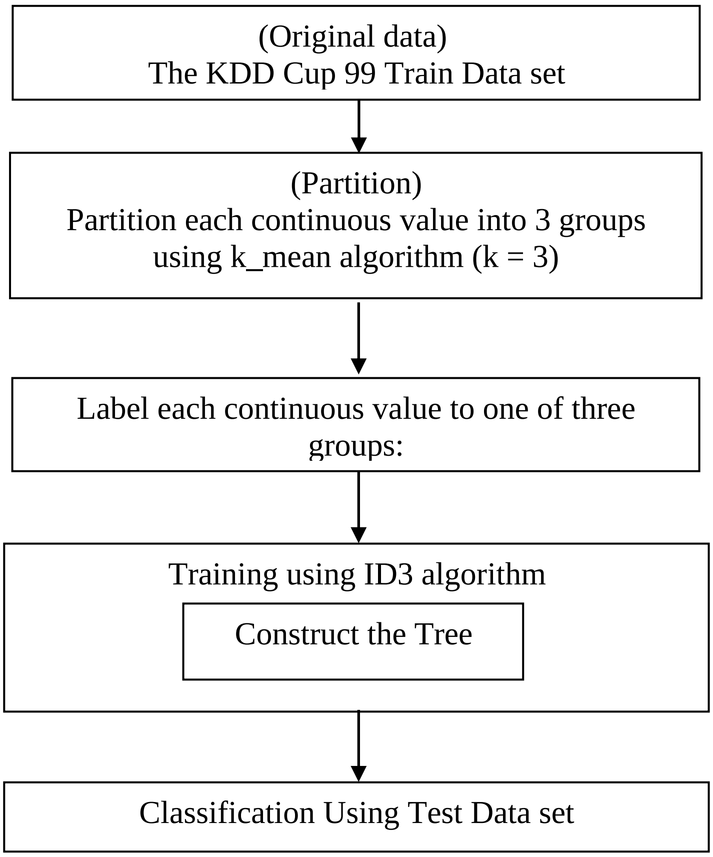
(Original data)
The KDD Cup 99 Train Data set
(Partition)
Partition each continuous value into 3 groups using k_mean algorithm (k = 3)
Label each continuous value to one of three groups:
Training using ID3 algorithm
Construct the Tree
Classification Using Test Data set

Figure (1). The Block Diagram of the Proposed System

3.1. Input Data (KDD Cup 99 data set)

As mentioned before, KDD Cup 99 dataset [18] is used to evaluate the proposed framework for intrusion detection. This dataset is a common benchmark for evaluation of intrusion detection techniques. '10%KDD Cup 99' dataset is used for the purpose of training and the so-called 'Corrected' dataset is used as a test set. Several new and novel never-before-seen attacks have been used in 'Corrected' in order to assess the generalization ability of ID systems. Statistical details of the two KDD components used here are summarized in Table (1).

From Table (1), Normal (97,277; 60,593) mean that the number of normal connection record in train dataset is (97277) records, while the number of normal connection record in test dataset is (60593) records.

Table (1). The Different Attack Types and their Corresponding Occurrence Number Respectively in the Training and Test Data sets

Normal(97,277; 60,593)	
Probing (4, 107; 4, 166)	DoS(391, 458; 229, 853)
ipsweep(1, 247; 306), mscan(0; 1, 053), nmap(231; 84), portsweep(1, 040; 364), saint(0; 736), satan(1, 589; 1, 633).	apache2(0; 794), back(2, 203; 1.098), land(21; 9), mailbomb(0; 5, 000), neptune(107, 201; 58, 001), pod(264; 87), processtable(0; 759), smurf(280, 790; 164, 091), teardrop(979; 12), udpstorm(0; 2).
U2R(52; 228)	R2L(1, 126; 16, 189)
buffer overflow(30, 22), httptunnel(0; 158), guess passwd(53; 4, 367), loadmodule(9; 2), perl(3; 2), perl(3; 2), ps(0; 16), rootkit(10; 13), sqlattack(0; 2), xterm(0; 13).	ftp write(8; 3), imap(12; 1), multihop(7; 18), named(0; 17), phf(4; 2), sendmail(0; 17), snmpgetattack(0; 7, 741), snmpguess(0; 2, 406), spy(2; 0), warezclient(1, 020; 0), warezmaster(20; 1, 602), worm(0; 2), xlock(0; 9), xsnoop(0; 4).
Total Train data set = 494020 Total Test data set = 311028	

Each record contains values of 41 independent variables (fields) describing the different features of the connection, and the value of the dependent variable labeled as either normal, or as an attack, with exactly one specific attack type. The sample of four connection record corresponding to the attack types is shown for each type of attack as:

```
0,tcp,http,SF,334,1684,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,1,9,0.00,0.00,0.00,0.00,1.00,0.00,0
.33,0,0,0.00,0.00,0.00,0.00,0.00,0.00,normal.

0,tcp,private,S0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,271,13,1.00,1.00,0.00,0.00,0.05,0.07,
0.00,255,13,0.05,0.07,0.00,0.00,1.00,1.00,0.00,0.00,neptune.

0,icmp,ecr_i,SF,1032,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,511,511,0.00,0.00,0.00,0.00,1.00,
0.00,0.00,255,255,1.00,0.00,1.00,0.00,0.00,0.00,0.00,0.00,smurf.

0,udp,private,SF,28,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,0.00,0.00,0.00,0.00,1.00,0.00,0.
0,73,1,0.01,0.05,0.01,0.00,0.00,0.00,0.00,0.00,teardrop.
```

3.2. Evaluation Criteria

To rank the different results a cost matrix *C* is defined [19]. Given the cost matrix illustrated in Table (2) and the confusion matrix obtained subsequent to an empirical testing process, a cost per test (CPT) is calculated by using the following formula:

Table (2). The Cost Matrix

	Normal	Probing	DoS	U2R	R2L
Normal	0	1	2	2	2
Probing	1	0	2	2	2
DoS	2	1	0	2	2
U2R	3	2	2	0	2
R2L	4	2	2	2	0

$$CPT = \frac{1}{N}\sum_{i=1}^{m}\sum_{j=1}^{m} CM(i,j) * C(i,j)$$

where *CM* and *C* are, respectively confusion matrix and cost matrix, *N* represents the total number of test instances, and *m* is the number of the classes in classification.

The accuracy is based on the Percentage of Successful Prediction (PSP) on the test data set, which is given by:

$$PSP = \frac{number\ of\ successful\ instance\ classification}{number\ of\ instances\ in\ the\ test\ set} * 100$$

Higher values of PSP and Lower of CPT show better classification for the intrusion detection system. In this paper, the Detection Rate (DR), PSP and CPT measures are used to rank the different results. Table (3) illustrates the confusion matrix for the winner on KDD Cup 99 [19].

Table (3). Present the Confusion Matrix Related to the Best Percentage of Successful Predication for the Winner on KDD Cup 99

Predicted Actual	Normal	Probing	DoS	U2R	R2L	%DR
Normal(60,593)	**60262**	243	78	4	6	99.5
Probing (4,166)	511	**3471**	184	0	0	83.3
DoS (229,853)	5299	1328	**223226**	0	0	97.1
U2R (228)	168	20	0	**30**	10	13.2
R2L (16,189)	14527	294	0	8	**1360**	8.4
***PSP* = 92.71%**				***CPT* = 0.2331**		

4. Experimental Results

The system of Figure (1) is implemented under Visual Studio.NET 2008 environment using Visual C# language. It requires 1 Megabyte RAM memory for execution. Tables (4) and (5) present the confusion matrix related to the DR, PSP, and CPT obtained using ID3 by both: classical partition method and K_mean algorithm, respectively.

Table (4). The DR for each Classification type, PSP and CPT using ID3 Algorithm with Classical partition method.

Predicted Actual	Normal	Probing	DoS	U2R	R2L	%DR
Normal(60,591)	**60223**	243	109	9	5	99.4
Probing (4,166)	601	**2862**	700	0	3	68.7
DoS (229,853)	7124	300	**222431**	0	0	96.77
U2R (228)	191	0	0	**36**	1	15.8
R2L (16,189)	15646	13	514	11	**5**	0.03
***PSP* = 91.811%**				***CPT* = 0.2613**		

Table (5). The DR for each Classification type, PSP and CPT using ID3 Algorithm with K-mean for partition.

Predicted Actual	Normal	Probing	DoS	U2R	R2L	%DR
Normal(60,591)	**59560**	921	68	4	6	98.3
Probing (4,166)	367	**3259**	379	1	160	78.23
DoS (229,853)	6071	842	**222940**	0	2	96.99
U2R (228)	59	7	17	**143**	2	62.72
R2L (16,189)	14995	242	3	8	**941**	5.81
***PSP* = 92.224%**				***CPT* = 0.2451**		

From Tables (4) and (5), it can be seen that the DT technique gives better accuracy when using K_mean algorithm for Probing, U2R and R2L class compared to the result when using classical partition method. For Normal and Dos class, there is only a small difference in the accuracy between these two techniques. For PSP and CPT the result show that when using K_mean algorithm is better performance than when using classical partition method.

5. Conclusions

An approach for a K_mean algorithm based partition system to split continuous data to three groups instead of classical partition method has been proposed. The implementation of intrusion detection system using ID3 algorithm to classify the normal, attack patterns and the type of each attack has been presented. The results show that the performance of the proposed K_mean partition method has been improved when compared with standard ID3 using classical partition method. One may check that if more than three groups give better performance. It is worth mentioning that partition method can be used with other techniques, such as neural network and data mining association rules for designing IDS.

REFERENCES

[1] Steven L. Salzberg, Book Review: C4.5: "Programs for Machine Learning by J. ross Quinlan", Morgan Kaufmann Publishers, Inc., 1993, Kluwer Academic Publisher, Boston. Manufatured in the Netherlands, Machine Learning, Volume. 16, pages 235-240, 1994.

[2] Arman Tajbakhsh, Mohammad Rahmati and Abdolreza Mirzaei, "Intrusion detection using fuzzy association rules", Applied Soft Computing ASOC-509, Elsevier B.V, 2008.

[3] Victor H. Garcia, Raul Monroy and Maricela Quintana, "Web Attack Detection Using ID3", In, John Debenham, editor, 2nd IFIP International Symposium on Professional Practice in AI, WCC 2006, IFIP, Volume 219, pages 323-332, Santiago, Chile, 2006.

[4] Yacine Bouzida, and Frederic Cuppens, "Neural networks vs. decision trees for intrusion detection", IEEE, IST Workshop on Monitoring, Attack Detection and Mitigation MonAM2006 Tuebingen, Germany, September 2006.

[5] Mehdi Moradi, and Mohammad Zulkernine, "A Neural Network Based System for Intrusion Detection and Classification of Attacks", International Conference on Advances in Intelligent Systems, Theory and Applications, Luxembourg, Kirchberg, Luxembourg, IEEE, November 2004.

[6] Rachid Beghdad, "Critical Study of Neural Networks in Detection Intursions", Press, Computer and Security, Elsevier, June 2008.

[7] Yuehui Chen, Ajith Abraham, and Bo Yang, "Hybrid Flexible Neural-Tree-Based Intrusion Detection Systems", International Journal of Intelligent Systems, Volume 22, pp 337-352, 2007.

[8] Gary Stein, Bing Chen, Annie S. Wu, and Kien A. Hua, "Decision Tree Classifier For Network Intrusion Detection With GA-based Feature Selection", 43rd ACM Southeast Conference, Kennesaw, GA, March 18-20, 2005.

[9] Yacine Bouzida, and Frederic Cuppens, "Neural networks vs. decision trees for intrusion detection", IEEE, IST Workshop on Monitoring, Attack Detection and Mitigation MonAM2006 Tuebingen, Germany, September 2006.

[10] Sandhya Peddabachigari,Ajith Abraham, Crina Grosan, and Johnson Thomas, "Modeling intrusion detection system using hybrid intelligent systems", Elsevier, 28 June 2005.

[11] Radim Belohlavek, Bernard De Baet, Jan Outrata, and Vilem Vychodil, "Inducing decision trees via concept lattices", International Journal of General Systems, Volume 38, Issue 4, pp 455 – 467, May 2009.

[12] Francisco Azuaje AND Joaquín Dopazo, "Data analysis and visualization in genomics and proteomics", Wiley, ISBN-10, 0470094397, ISBN-13, 9780470094396, 2005.

[13] Ben Coppin, "Artificial intelligence illuminated", Jones & Bartlett Publishers, ISBN-10, 0763732303, ISBN-13, 9780763732301, 2004.

[14] Nala Ben Amor, Salem Benferhat, and Zied Elouedi, "Qualitative Classification

with Possibilistic Decision Trees", International Conference on Information Processing of Uncertainty in Knowledge Based System IPMU,2004, Perugia Italie, 2004.

[15] Utgoff P. E, "Incremental Induction of Decision Trees", Machine Learning, NEC, Volume 4, pp 161-186, 1989.

[16] Yehuda Lindell, and Benny Pinkas, "Secure Multiparty Computation for Privacy-Preserving Data Mining", The Journal of Privacy and Confidentiality, Volume 1, No. 1, pp 59-98, 2009.

[17] Hla Hla Htay, G. Bharadwaja Kumar, Kavi Narayana Murthy, "Constructing English-Myanmar Parallel Corpora", ICCA, 2006.

[18] Sandhya Peddabachigari,Ajith Abraham, Crina Grosan, and Johnson Thomas, "Modeling intrusion detection system using hybrid intelligent systems", Elsevier, 28 June 2005.

[19] Charles Elkan, "Results of the KDD’99 Classifier Learning", SIGKDD Explorations, ACM SIGKDD, Issue 2, Volume 1, Page 67, January 2000.

[20] Adel Sabry Issa, "A Comparative Study among Several Modified Intrusion Detection System Techniques", B.Sc., Computer Science, Duhok University, 2009.