AN ANOMALY DETECTION METHOD BY ASSOCIATIVE CLASSIFICATION

Bum Ju Lee, Heon Gyu Lee, Keun Ho Ryu

Database & Bioinformatics Laboratory, Chungbuk National University,
{bjlee, hglee, khryu}@dblab.chungbuk.ac.kr

ABSTRACT:

For detecting an intrusion based on the anomaly of a user's activities, previous works are concentrated on statistical techniques or frequent episode mining in order to analyze an audit data. But, since they mainly analyze the average behaviour of user's activities, some anomalies can be detected inaccurately. Therefore, we propose an anomaly detection method that utilizes an associative classification for modelling intrusion detection. Finally, we proof that a prediction model built from associative classification method yields better accuracy than a prediction model built from a traditional methods by experimental results.

KEY WORDS: Anomaly Detection, Associative Classification, Class Association Rules, Data Mining

1. INTRODUCTION

The growing dependence of modern society on telecommunication and information networks has become inescapable. The increase in the number of inter-connected networks to the internet has led to an increase in security threats and crimes such as DoS(Denial of Service). Data mining is recognized as a useful tool for extracting regularities in data and thus has been the target of some investigations for its use in intrusion detection. In particular, it promises to help in the detection of previously unseen attacks by establishing sets of observed regularities in network data. These sets can be compared to current traffic for deviation analysis.

In order to analyze audit data, we propose the anomaly detection model using clas-sification method in this paper. To achieve this purpose, we utilize the advantages of associative classification method. This paper first formulates the classification task as discovery of CARs(Class Association Rules). In this task, Apriori algorithm is slightly modified and used to discover CARs efficiently In addition, we use a measure, cohesion to rank CARs and define a new threshold, class support to class label distribution. The pruning techniques, database coverage are also applied to prune uninformative (redandunt) CARs after all CARs are discovered. Finally, the paper describes an experiment and its analysis on tcp-dump data.

2. RELATED WORKS

In this section, we review data mining frameworks including association rules and classification for intrusion detection models[Lee 1999a, 1998b, 1998c] and existing associative classification. The idea of association rules[Agrawal 1994] is to use auditing programs to extract an extensive set of features that present each network connection, and apply data mining programs to learn rules that accurately capture the behavior of intrusion and normal activities. Classification[Quinlan 1993] creates a categorization of data records and it could be used to detect individual attacks. An ideal application in intrusion detection is to gather sufficient normal and abnormal audit data, then apply a classification algorithm to learn a classifier that will determine audit data as belonging to the normal and abnormal class.

Associative classification is a combination of association rule discovery and classification, by taking advantage of employing association rules for classification purpose. The existing associative classifiers mine the training set in an apriori-lilke method[Liu 1998] or use FP-growth method as association rule discovery algorithm[Li 2001]. Although the mining methods are slightly different, all approaches generate the same kind of association rules for classification because they are discovered in the support-confidence framework. The reported several advantages of this approach are as follows. First, differently from most of classifiers as decision trees, association rules consider the simultaneous correspondence of values of different attributes, therefore allowing to achieve better accuracy. Second, it makes association rule techniques applicable to classification. Third, the user can decide to mine both association rules and a classification model in the mining process.

3. CLASS ASSOCIATION RULES DISCOVERY

Associative classification is a combination of two data mining problems, association and classification, that uses association rules to predict a class label. The method is gaining popularity due to several reasons. First, the classifier can handle feature spaces of tens of thousands dimensions, while decision tree classifiers are limited to several hundreds attributes only. Second, the method also solves understandability problem by generating simple rules.

3.1 Concept of the CARs(Class Association Rules)

CARs is a special subset of association rules whose antecedent is an itemsets and consequent are restricted to the classification class label. Let $I = \{i_1, i_2, \ldots, i_m\}$ be a set of all items in D and C to be a set of all class labels.

Definition 3.1: Class Association Rules, CARs is an implication of the form:

$$X \Rightarrow c_i, \text{ where } X \subset I, c_i \subset C.$$

Antecedent of a CARs is called *itemset* and a rule itself is called *ruleItem*.

Definition 3.2: Support and Confidence of ruleItem in a transaction set D are the following:

$$\text{Support} = \frac{\text{ruleCount}}{|D|},\ \text{Confidence} = \frac{\text{ruleCount}}{\text{itemsetCount}}$$

where *ruleCount* is the number of items in D that contain the itemset and are labeled with class label and *itemsetCount* is the number of items in D that contain the itemset.

In this paper, we use apriori-like method to discover all *CARs*. On the mining tasks, firstly the algorithm discovers all the *CARs* in dataset D. The algorithm for discovering CARs is given in Figure 1.

CARs algorithm consists of the same number of passes as the Apriori. Initially, L_1 contains all the 1-RuleItems that satisfy the minimum support threshold. This is done by scanning the whole dataset one time. During pass k, the algorithm discovers the set of frequent k-RuleItems and then generates CAR_k that satisfies the minimum confidence. The algorithm terminates when L_k is empty.

```
Input transaction T, minSup, minConf
Output a set of CAR_k

L_1 = {Large 1-RuleItems in T satisfy, minSup}
for (k=2 ; L_{k-1}≠Ø ; k++) {
    C_k=candidate_gen(L_{k-1}); // generate candidate
    for all transaction t∈T {
        C_t=subset(C_k, t);
        for each candidate c∈C_t {
            c.itemsetCount++;
            if c.class = t.class then
                c.ruleCount++;
        }
    }
    L_k={c | c.ruleCount≥minSup}
    CAR_k=gen_rule(L_k, minConf);
}
Output = U_k CAR_k
```

Figure 1. Outline of algorithm for discovering CARs.

4. BUILDING CLASSIFICATION USING CARS

After CARs are discovered, the resulting set of CARs can still be huge and contain many uninformative redundant CARs. All the CARs nedd to be ranked and only the most highly ranked CARs will be inserted into classifier. The rank can be measured by either the cohesion measure itself or by some other criteria. After being sorted in decreasing rank order, the CARs are pruned using database coverage pruning.

4.1 CAR Cohesion Measure and CARs Ranking

CAR Cohesion measure is used for CARs ranking. CARs ranking is needed to select the best CARs in case of overlapping CARs. The proposed CAR Cohesion measure is adapted from the cohesion measure as defined below.

Definition 4.1: For a CAR $(item_1,\ldots,item_n)\rightarrow C$ of length n, CAR Cohesion is a ranking measure defined as

$$\text{Cohesion}(item_1,\ldots,item_n, C) = \sqrt{\frac{\text{Count}(item_1,\ldots,item_n, C)}{\text{Count}(item_1)\cdots\text{Count}(item_n)\,\text{Count}(C)}}$$

where $\text{Count}(item_1,\ldots,item_n, C)$ is a number of transactions where the itemsets and class occur together, $Count(item_i)$, $i=1,\ldots,n$, is a number of transactions containing $item_i$, and $Count(C)$ is a count of transactions classified to class C in a training set.

Cohesion measure is higher if the $item_1,\ldots,item_n$ and C co-occur more frequently in the together and are less frequently encountered separately.

CARs ranking guarantees that only the highest rank CARs will be selected into the classifier. All CARs are ranked according to the following criteria.

Definition 4.2: CARs Ranking, given two rules CAR_i and CAR_j, $CAR_i > CAR_j$ (or CAR_i is ranked higher than CAR_j) if

(1) $Cohesion(CAR_i) > Cohesion(CAR_j)$ or

(2) $Cohesion(CAR_i) = Cohesion(CAR_j)$ but $Frequency(CAR_i) > Frequency(CAR_j)$ or

(3) $Cohesion(CAR_i) = Cohesion(CAR_j)$ and $Frequency(CAR_i) = Frequency(CAR_j)$ but $support(CAR_i) > Support(CAR_j)$

A rule CAR_1: $X\rightarrow C$ is said a general rule w.r.t. rule CAR_2: $X'\rightarrow C'$, if only if X is a subset of X'. First, we need to sort the set of generated CARs for each k-star expression and then given two rules CAR_1 and CAR_2, where CAR_1 is a general rule w.r.t CAR_2, we prune CAR_2 if CAR_1 also has higher rank than CAR_2. The rationale behind the prune is the following: if rule CAR_i covers a case d then sub-rule of CAR_i also covers it, and sub-rule will always be selected since it has higher rank.

4.2 Database Coverage Pruning

After the CARs are ranked and sorted in decreasing rank order, the highest ranked CARs are selected into classifier, while the others are pruned using database coverage pruning. We use coverage pruning like that in [Li 2001]. Database coverage pruning ensures that every CAR selected into the classifier classifies correctly at least one data case and that each training data case is covered by several CARs of the highest possible rank. The Figure 2 shows the process of database coverage pruning.

```
Input a training data set D, a set of CARs R, coverage threshold ε .
Output a set of CARs SR used by classifier, a default rule DR.

(1) DR={the majority class in D, C_m};
(2) for each case d_i ⊆D cover count dcover_i=0;
(3) SR=Ø;
(4) for ( training data set D=Ø or R=Ø ) do {
(5)     for each CAR_i ⊆R do {
(6)         find a set D_cover ⊆D of cases covered by CAR_i and D_cover ∈CAR_i
(7)         if at least one d⊆D_cover is correctly classified by CAR_i then {
(8)             SR = SR∪CAR_i;
(9)             for each case d_j ⊆D_cover and do {
(10)                dcover_j ++;
(11)                if dcover_j =ε then
(12)                    delete d_j from D;
(13)            }
(14)        }
(15)    }
(16) }
```

Figure 2. Database coverage pruning algorithm.

First for each case in D, set its cover-count to 0. If CAR can correctly classify at least one case then select CAR and increase the dcover of those cases matching CAR by 1. A case is removed if its dcover passes coverage threshold. A default rule is also selected. It has an empty rule and predicts a class label, which is a majority class among the case left in D or if none is left is the majority class in the original database. After a set of rules is selected for classifier, we classify new cases. We discover the rules matching case in the selected one and classify class labels of the discovered rules. If all the rules matching the new case have same class label, the new case is assigned to that label.

5. EXPERIMENTAL RESULTS

In this section, we evaluate our experiments of the proposed anomaly detection model on the dataset from the DARPA'98 Intrusion Detection Evaluation. The DARPA program that was prepared and managed by MIT Lincoln Labs[DARPA 1998] has provided the dataset. A standard set of data to be audited, which includes a wide variety of intrusions simulated in a military network environment, was provided.

The DARPA training data consists of seven weeks of network-based attacks. Attacks are labeled in the training data. To preprocessing the training data, we use connections as the basic granule, obtaining the connection from the raw packet data of the audit trail. We preprocess the packet data in the training data files in the following schema :

R=(*Src_IP, Src_Port, Dest_IP, Dest_Port, Class*)

where *Src_IP* and *Src_Port* denote source address and port number respectively, while *Dest_IP* and *Dest_Port*, represent the destination address and Port number. The attribute *Class* is class labels that are two categories: *attack* and *normal*. Figure 3 depicts the preprocessing results *TCP-dump* data from raw packet data of audit trail.

#	Start Date	Start Time	Duration	Service	Src Port	Dest Port	Src IP address	Dest IP address	Attack Score/Name
1	07/03/1998	08:00:01	00:00:01	http	1106	80	192.168.001.005	192.168.001.001	0 -
2	07/03/1998	08:00:01	00:00:02	domain/u	53	53	172.016.112.020	192.168.001.001	0 -
3	08/03/1998	08:00:01	00:00:01	smtp	1026	25	172.016.113.084	194.007.248.153	0 -
8383	07/03/1998	11:46:39	00:00:26	telnet	20504	23	197.218.177.069	172.016.113.050	loadmodule
9966	07/03/1998	11:49:39	00:00:01	tcpmux	1234	1	205.160.208.190	172.016.113.050	portsweep

preprocessing

Src_ip	Src_port	Dest_ip	Dest_port	Intrusion
192.168.001.005	1106	192.168.001.001	80	normal
172.016.112.020	53	192.168.001.001	53	normal
172.016.113.084	1026	194.007.248.153	25	normal
197.218.177.069	20504	172.016.113.050	23	attack
205.160.208.190	1234	172.016.113.050	1	attack

Figure 3. Preprocessing of the training dataset.

Our experiment focused on DoS and Probe attacks since there appear to be no sequential patterns that are frequent in records of R2L and U2R

We have two important thresholds for the classifier performance(*minSup, Database coverage*). These thresholds control the number of frequent CARs selected for constructing classifier in our experiments. We carried out experiments using different thresholds over random samples of training dataset, 10% of total size. Figure 4 show classifier average classification error according to support and database coverage respectively where *minConf*=0.6.

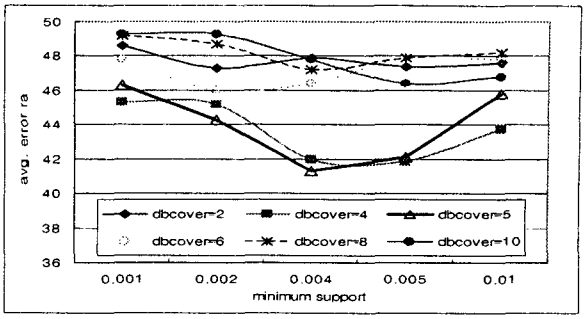

Figure 4. Classifier accuracy on training data.

Based on Figure 4, for our algorithm, the minimum support is set to 0.004 and the database coverage is set to 5. The minimum confidence and minimum frequency are set to 0.5 and 0.6 respectively. We compare accuracy of our algorithm, CMAR [Li 2001] and CBA[Liu 1998]. The result is shown on Figure 5.

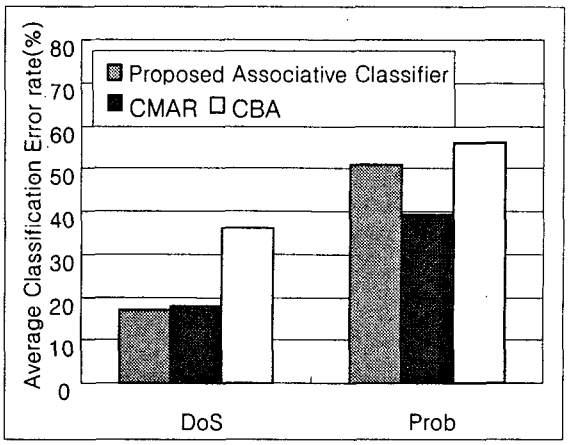

Figure 5. Comparison of our algorithm, CMAR and CBA accuracy.

6. CONCLUSIONS

The purpose of this paper was to develop accurate and efficient mining algorithm to automatic audit data classification. The proposed method is associative classification that is a combination of two data mining problems, Association rules mining and Bayesian classification. The proposed classification method used 1988 DARPA Intrusion Detection Evaluation data and the results were equivalent or more accurate than the existing static algorithms. Although the proposed approach was not more accurate and efficient than CMAR, we expect that the results using associative classification should be better, if we use real network packet data.

Our classification method can be applied not only to the problem of network-based attack classification but also to other classification tasks as well. The pattern pruning technique can be used association rules mining and classification based on association rules.

Acknowledgements

This works was supported by the Regional Research Centers Program of Ministry of Education & Human Resources Development in Korea.

References from Books:
J. Quinlan, 1993, "C4.5: Programs for Machine Learning," Morgan Kaufmann, San Mateo, CA.

References from Other Literature:
W. Lee, S. J. Stolfo and K. W. Mok, 1999a, "A Data Mining Framework for Building Intrusion Detection Models," In Proceedings of the IEEE Symposium on Security and Privacy, pages 120-132.

W. Lee, S. J. Stolfo and K. W. Mok, 1998b, "Mining audit data to build intrusion detection models," In Fourth International Conference on Knowledge Discovery and Data Mining(KDD'98), pages 66-72, New York, NY, August.

W. Lee, S. J. Stolfo, 1998c, "Data mining approaches for intrusion detection," In Proceedings of Seventh USENIX Security Symposium, pages 66-72, San Antonio, TX, January.

R. Agrawal and R. Srikant "Fast algorithms for mining association rules," In Proceedings of the 20th VLDB Conference, pages 487-499, Santiago, Chile, 1994.

B. Liu, W. Hsu and Y. Ma "Integrating classification and association rule mining," In Pro-ceedings of the 4th International Conference Knowledge Discovery and Data Mining, 1998.

W. Li, J. Han and J. Pei "CMAR: Accurate and Efficient Classification Based on Multiple Association Rules," In Proceedings of 2001 International Conference on Data Mining, 2001.

References from websites:
DARPA Intrusion Detection Evaluation Data, homepage URIs:http://www.ll.mit.edu/IST/ideval/data/data_index.html.(1998)