# A Semi-Supervised IDS Alert Classification Model Based on Alert Context

Haibin Mei, Minghua Zhang
Information College
Shanghai Ocean University
Shanghai, China
nsplum@gmail.com

***Abstract*—How to filtering false positives is a fundamental problem of IDS. Constructing alert classification model is one of efficient methods. However, the high cost of preparing training data and classification feature selection are key points in the problem. This paper gives a semi-supervised alert classification model which makes use of the power of semi-supervised learning. Moreover, four classification features about alert context are introduced to improve classification accuracy. Experiments conducted on the DARPA 1999 dataset show that the use of the alert context properties can increase the classification accuracy by about 3 percent.**

***Keywords-alert classification model; semi-supervised learning; alert context***

## I. INTRODUCTION

Alert classification model based on machine learning is an efficient method to filtering IDS false positives [1-2]. However, current alert classification models are built on the supervised learning techniques which require large amounts of labeled training data. It may need long time and expertise of network security to manually label the alert data.

This paper proposes a semi-supervised alert classification model based on alert context. For classification, semi-supervised learning is a special classification technique [3], which learns from labeled data and unlabeled data and thus can reduce dramatically the number of labeled data required.

The power of semi-supervised learning for building classification model has been demonstrated in many applications such as web pages classification, documents classification, and traffic classification and so on. In the field of network security, semi-supervised learning has also been used [4-5]. However, to the best of our knowledge, we are the first to apply semi-supervised learning to intrusion alert classification.

We have made a study on how to apply semi-supervised learning to alert classification in [6]. Different from it, this paper gives more details of the alert semi-supervised classification model and the pseudo code of EM algorithm. Moreover, this paper introduces alert context information as new classification features to improve the accuracy of classification model. Experiments are conducted to demonstrate the efficiency of the proposed classification model.

The rest of the paper is organized as follows: Section 2 gives the construction of semi-supervised IDS alert classification model, including the classification feature selection and details of the alert classification algorithm. Experiments are conducted in Section 3 with experimental results and analysis. In the end, the whole paper is summarized in Section 4.

## II. SEMI-SUPERVISED IDS ALERT CLASSIFICATION MODEL

In this section, we first describe the feature selection for alert classification. Then we elaborate on the alert classification method based on semi-supervised learning.

### *A. Feature Selection Based on Alert Context*

The performance of a classification model is greatly affected by the selected features used in constructing the model. Usually features used for classification are selected from the inherent property of alert. Some redundant, very specific or general properties are removed by some methods. Although the context information about alerts is not contained in the inherent properties, they also contribute to increase the accuracy of an alert classification model [1, 7]. Considering this, we also use some alert context information to select classification features. Four features on alert context used in this paper are as follows.

(1) Alert history correlation degree. False positives generally tend to be more random and less likely to be correlated than true alerts [8]. This feature is introduced to measure the correlation degree between a given alert with other history alerts.

Given alert $X$ and alert $Y$, their alert correlation degree is calculated by (1).

$$Corr(X,Y) = \sum C(X.p, Y.p) \qquad (1)$$

Where $C(X.p,Y.p)$ is the correlation degree of alert $X$ and alert $Y$ on the property $p$. The type of value $p$ can be *AlertType*, *SrcIP* and *DstIP*, which represent the alert type， the source IP address and the destination IP address respectively.

The correlation degree of two alert types is determined by the logic causal relationship between them, which can be measured by the alert correlation matrix (ACM) [9]. Table 1 illustrates an ACM example of four alert types $a_1$, $a_2$, $a_3$ and $a_4$. The value of cell $C(a_i,a_j)$ in ACM represents the correlation degree of $a_i$ and $a_j$ where $a_j$ arrives after $a_i$. Note that the correlation relationship between two alert types is relative to their arrival order. It satisfies $C(a_i,a_j)\neq C(a_j,a_i)$ where $a_i\neq a_j$.

Published by Atlantis Press, Paris, France.

TABLE I. ALERT CORRELATION MATRIX OF FOUR ALERT TYPES

| **Alert Type** | **$a_1$** | **$a_2$** | **$a_3$** | **$a_4$** |
|---|---|---|---|---|
| **$a_1$** | $C(a_1,a_1)$ | $C(a_1,a_2)$ | $C(a_1,a_3)$ | $C(a_1,a_4)$ |
| **$a_2$** | $C(a_2,a_1)$ | $C(a_2,a_2)$ | $C(a_2,a_3)$ | $C(a_2,a_4)$ |
| **$a_3$** | $C(a_3,a_1)$ | $C(a_3,a_2)$ | $C(a_3,a_3)$ | $C(a_3,a_4)$ |
| **$a_4$** | $C(a_4,a_1)$ | $C(a_4,a_2)$ | $C(a_4,a_3)$ | $C(a_4,a_4)$ |

The correlation degree of two alert addresses (source address or destination address) is calculated by the formula $C(ip_1,ip_2) = n/32$, where $n$ is the maximum number of high order bits that these two IP addresses match and 32 is the number of bits in an IP address. For example, if $ip_1$ =192.168.0.1 and $ip_2$ =192.168.0.201, then $n$=24 and $C(ip_1,ip_2)$=24/32=0.75.

Alert history correlation degree is the maximum value of all alert correlation degrees between current alert $X$ and the alerts generated in time interval $t$ before $X$. The value is computed as follows:

$$MaxCorr(X,t) = \underset{Y_j \in D_{xt}}{Max}(Corr(X,Y_j)) \qquad (2)$$

Where $D_{xt}$ represents the history alert set consisting of all alerts reported in $t$ time interval before the alert $X$.

(2) The net type of IP address. It represents the type of net which the alert's source or destination IP address belongs to, denoted by *SrcIPnet* and *DstIPnet* respectively. The value of the net type can be *Internet*, *Intranet* and *demilitarized zone* (DMZ).

(3) The operation system of the attacked target. It means the type and version of the operation system running on the victim machine, denoted by *DstIPOS*. The value of this feature can be Windows 98(/NT/XP/2000/vista/7), Linux, UNIX, OS/2, Macintosh, and so on. The reason for selecting this information is that most attacks are only effective to some specific systems or services, which helps to classify alerts.

(4) The device type of the attacked target. It is denoted as *DstIPdev*, and its value can be router, computer, server, workstation, switch, etc.

### *B. Alert Classification Model*

Currently, there are many semi-supervised learning methods. This paper chooses generative models-based approach for its following remarkable advantages: (a) showing comparatively good effectiveness with small training sets, (b) using unlabeled data in parameter estimation, (c) simple to implement, and (d) efficient to train and use.

As to our problem, we give an alert generative model in (3), where $\theta$ is parameter of the mixture model, $c_j \in C \quad C = \{c_1, c_2, ..., c_{|C|}\}$ represents the classes of alert data (only considers two class, i.e. $C$= {"true positive", "false positive"}). $P(c_j|\theta)$ is the class probabilities. $P(x_i|c_j;\theta)$ represents the probability distribution to generate an alert object when the mixture component is selected.

$$P(x_i \mid \theta) = \sum_{j=1}^{|C|} P(c_j \mid \theta) P(x_i \mid c_j; \theta) \qquad (3)$$

The alert generative model is a probability model which explicitly states how the alerts are generated. It can also be considered as a joint probability distribution.

Three basic assumptions about the generative process are: (a) the alert data are generated by a mixture distribution model; (b) it's a one-to-one correspondence between mixture components and alert classes; (c) the features used for classification are independent with each other when the label is known.

Since the features are conditionally independent of other features in the same alert when the class label is given, (3) can be further expressed as (4).

$$P(x_i \mid \theta) = \sum_{j=1}^{|C|} P(c_j \mid \theta) \prod_{k=1}^{n} P(a_{x_i,k} \mid c_j; \theta) \qquad (4)$$

Suppose the estimated of parameters $\theta$ is $\hat{\theta}$, for a given alert object $x_i$ the probability that $x_i$ belongs to category $c_j$ can be calculated by (5):

$$P(y = c_j \mid x_i; \hat{\theta}) = \frac{P(x_i \mid c_j; \hat{\theta}) P(c_j \mid \hat{\theta})}{P(x_i \mid \hat{\theta})} \qquad (5)$$

For an unlabeled alert object $x_i$, its estimated class $y_i$ is the one which obtains the maximum value of the posterior probability, that is $y_i = \underset{j=1,...,|C|}{\arg\max}(P(y = c_j \mid x_i; \hat{\theta}))$. Suppose the labeled alert training data is $D_l$ = {$<x_1,y_1>$, $<x_2,y_2>$,…, $< x_{|D_l|}, y_{|D_l|} >$ }, and use the maximum a posteriori (MAP) estimate, we can find that $\hat{\theta} = \arg\max_\theta P(\theta \mid D_l)$. Here, more details of computation formula can refer to [4].

Now given the alert generative model and its estimated parameters calculated from labeled training alerts, it is possible to perform classification on unlabeled alerts. However, when there is only a small labeled alert training set, the classification accuracy will decline because variance in the parameter estimates of the generative model is high. Therefore, this paper proposes an EM algorithm, which utilizes both labeled alert training data and a lot of unlabeled alert data to improve the accuracy of parameter estimates. In order to describe the EM algorithm more convenient, we firstly give some relevant definitions.

Definition 1: Weak labeled dataset. When the elements in unlabeled dataset $D_u$ are assigned labels by classifier, $D_u$ becomes a weak labeled dataset and denoted by $D_p$. Let unlabeled alert dataset $D_u$= $\{x_1^u, x_2^u, ..., x_{|D_u|}^u\}$, then its weak labeled dataset is $D_p$ = {$< x_1^u, y_1^u >$ , $< x_2^u, y_2^u >$,…, $< x_{|D_u|}^u, y_{|D_u|}^u >$}, where $y_i^u$ is the predicted label for the alert object $x_i^u$, where $i = 1, 2, ..., | D_u |$.

Definition 2: Weighted labeled dataset. It consists of triples$<x_i,y_i,p_i>$, where $y_i$ is the label of $x_i$, $p_i$ represents the probability that $x_i$ is assigned the label $y_i$ ( $0 < p_i \le 1$ ). In this paper, $p_i$ is also called the weight of $x_i$ belonging to the class $y_i$. According to this definition, both labeled dataset $D_l$ and weak labeled dataset $D_p$ have weighted labeled dataset. For the labeled dataset $D_l$, we assign every element in it a maximum weight 1. Thus the weighted labeled dataset of $D_l$ is $D_l^w$= $\{<x_1,y_1,p_1>,<x_2,y_2,p_2>,\ldots,< x_{|D_l|}, y_{|D_l|}, p_{|D_l|} >\}$, where $p_i$=1, $i = 1,2,...,| D_l |$. And for the weak labeled dataset $D_p$, its weighted labeled dataset is $D_p^w$ = $\{< x_1^u, y_1^u, p_1^u >, < x_2^u, y_2^u, p_2^u >, \ldots, < x_{|D_u|}^u, y_{|D_u|}^u, p_{|D_u|}^u >\}$, where $p_i$ satisfies $0.5 \le p_i \le 1$, and $i = 1,2,...,| D_u |$.

Definition 3: Extended labeled dataset. It is a subset of weighted labeled dataset, in which the weight is equal or bigger than a given threshold value $\rho$ , i.e. $D_s^w$= $\{<x_i,y_i,p_i> \mid <x_i,y_i,p_i> \in D_p^w \wedge p_i \ge \rho \}$. In the training process of classification model, both the extended labeled dataset and the labeled dataset are utilized.

The pseudo code of EM algorithm based on alert generative model is in Fig. 1:

```
Input: labeled alerts collection $D_l$, unlabeled alerts collection $D_u$;
Output: $\hat{\theta}$ that learned from both labeled alerts and unlabeled alerts;
Let $D_l \Rightarrow D_l^w$; $D_s^w = \varnothing$; $D_t^w = D_l^w$;
Learn an initial $\hat{\theta}$ from $D_t^w$. Use maximum a posteriori parameter estimation to get $\hat{\theta} = \arg\max_\theta P(D_l^w \mid \theta)P(\theta)$.
Repeat {
    Use the current $\hat{\theta}$ to estimate the label of each unlabeled alert in $D_u$.
    Calculate the prediction marked alert dataset $D_p$ and the weighted marked alert dataset $D_p^w$.
    For ($i$ = 1 to $|D_p^w|$) {
        For the element $< x_i^u, y_i^u, p_i^u >$ of $D_p^w$, if ( $p_j^u \ge \rho$ ) then $D_s^w = D_s^w \cup \{< x_i^u, y_i^u, p_i^u >\}$
    }
    Extend the training dataset $D_t^w$, let $D_t^w = D_t^w \cup D_s^w$.
    Use $D_t^w$ to estimate the optimization classifier $\hat{\theta}$, i.e. $\hat{\theta} = \arg\max_\theta P(D_l \mid \theta)P(\theta)$.
} Until ( $D_s^w$ with no significant changes)
```

Figure 1. The EM algorithm based on alert classification model.

## III. Experiments and Analysis

### A. Experimental Datasets

A well-known IDS evaluation dataset, the DARPA 1999 dataset is used to validate the proposed approach. The experimental process consists of preprocessing alert data, building alert classification model, and comparing with other typical methods based on the supervised learning. We use classification accuracy (CA) as evaluation criteria to measure the quality of the alert classification model. The formula is in (6).

$$CA = \frac{|TP| + |TN|}{|TP| + |FP| + |TN| + |FN|} \times 100\% \qquad (6)$$

Where $TP$ is the true positives correctly classified, $TN$ is the false positives correctly classified, $FP$ is the true positives wrongly classified and $FN$ is the false positives wrongly classified.

We adopt Snort to detect attacks and generate alerts by reading the inside tcpdump data files in five weeks. We call these generated alerts as raw alert data and record them into the MS SQL Server 2000 database. The total number of alerts generated by Snort 2.6 with default setting is 85902, and there are 82 different types of attacks among these alerts.

When we label alerts prepared for the training of classification model, the alerts meeting the following criteria are considered as true alerts: matching the source IP address, matching the destination IP address, and alert time stamp in the time window in which the attack has occurred. All remaining alerts are labeled as false alerts.

### B. Results and Analysis

Experiments are carried out to evaluate the effect of alert context properties on the accuracy of classification. Fig. 2 shows results with the typical NB classification model. Fig. 3 illustrates results with the proposed semi-supervised learning classification model. During experiments, we vary the number of labeled alerts used for training, which are randomly chosen from the labeled alert dataset. Each point in the curve represents an average classification accuracy of ten independent experiments. The results in the figures show that no matter which alert classification model is used, the use of the alert context properties can increase the classification accuracy by about 3 percent.

Experimental results about the performance of semi-supervised classification model can refer to [6].

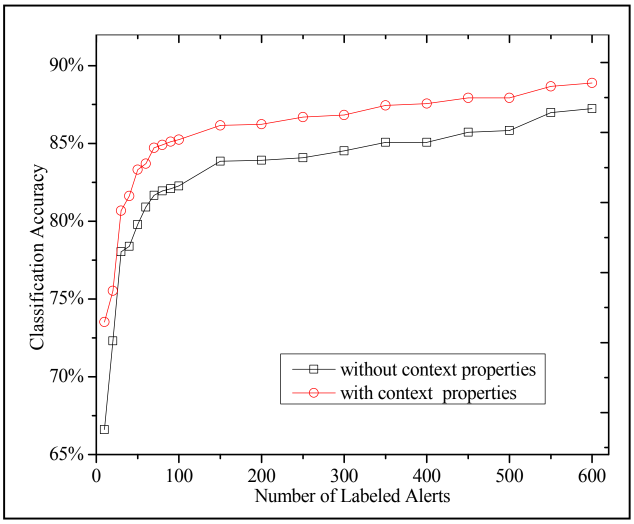

Figure 2. Classification accuracy of Naïve Bayes classification model

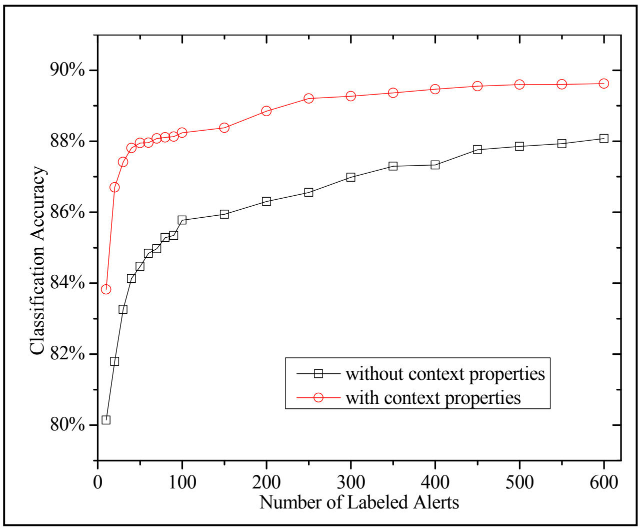

Figure 3. Classification accuracy of semi-supervised classification model

## IV. Conclusion

Alert classification model based on machine learning is an efficient method to filtering IDS false positives. This paper proposes a semi-supervised alert classification model based on alert context to use both labeled data and unlabeled data and thus reduce the cost of manually labeling the alert data. Moreover, classification features on alert context are introduced to improve classification accuracy. Experimental results with the DARPA 1999 dataset show that the use of the alert context properties can increase the classification accuracy by about 3 percent.

## Acknowledgment

This research is supported by Shanghai Ocean University Startup Fund for Doctors (A-2400-11-0210). The authors would like to thank anonymous reviewers for their perspicacious comments.

## References

[1] T. Pietraszek, “Using Adaptive Alert Classification to Reduce False Positives in Intrusion Detection,” Proc. the 7th Int. Symposium on Recent Advances in Intrusion Detection, 2004, pp. 102-124.

[2] T. Subbulakshmi, G. Mathew, and S. M. Shalinie, “Real Time Classification and Clustering of IDS Alerts Using Machine Learning Algorithms,” International Journal of Artificial Intelligence & Application(IJAIA), vol.1, no.1, 2010, pp.1-9.

[3] O. Chapelle, B. Scholkopf, and A. Zien, “Semi-Supervised Learning”. Cambridge: MIT Press, 2006.

[4] H. Li, Z. Hu, Y. Wu, and F. Wu, “Behavior Modeling and Abnormality Detection Based on Semi-Supervised Learning Method,” Journal of Software, vol.18, no.3, 2007, pp.527−537.

[5] Y. Tian, G. M. Weiss, and Q. Ma, “A Semi-Supervised Approach for Web Spam Detection Using Combinatorial Feature-Fusion,” Proc. of the ECML/PKDD 2007 Graph Labelling Workshop and Web Spam Challenge, 2007, pp. 16-23.

[6] M. H. Zhang and H. B. Mei. “A New Method for Filtering IDS False Positives with Semi-supervised Classification,” Proc. ICIC2012, Lecture Notes in Computer Science, vol. 7389, 2012, pp. 513-519.

[7] B. Morin, L. Mé, H. Debar, and M. Ducassé, “M2D2: A Formal Data Model for IDS Alert Correlation,” Proc. of the 5th Int. Symposium on Recent Advances in Intrusion Detection, 2002, pp. 115-137.

[8] P. Ning, Y. Cui, D. S. Reeves, and X. Dingbang, “Techniques and Tools for Analyzing Intrusion Alerts,” ACM Transactions on Information and System Security, vol.7, no.2, 2004, pp.274-318.

[9] B. Zhu and A. A. Ghorbani, "Alert Correlation for Extracting Attack Strategies,” International Journal of Network Security, vol.3, no.3, 2006, pp.244-258.