Journal of Engineering and Applied Sciences 12 (Special Issue 3): 6514-6518, 2017
ISSN: 1816-949X
© Medwell Journals, 2017

Empirical Analysis of Effective Misuse Intrusion Detection by Trace Classification using Conditional Random Fields

Kyung-Hwan Cha and Dae-Ki Kang
Department of Computer Engineering, Dongseo University, Busan, Korea

Abstract: In intrusion detection systems based on machine learning techniques, most research work prefer N-gram based approaches. There has been insufficient investigation on the application of N-gram based methodologies to intrusion detection. In this study, we consider applying conditional random field method to misuse intrusion detection problems. In order to evaluate the performance of our proposed system, we compare our proposal with Naive Bayes algorithm and support vector machines on host based misuse intrusion detection benchmark datasets. Public host based misuse intrusion detection benchmark datasets are from University of New Mexico. The experimental results on the benchmark datasets such as UNM indicate that CRF generates accurate misuse intrusion detector with comparable performance to support vector machines and Naive Bayes. CRF produces intrusion detection programs with higher or comparable accuracy than the intrusion detectors produced from Naive Bayes with N-gram features. And intrusion detectors generated from CRF exhibits comparable accuracy to the intrusion detectors produced from N-gram featured SVM. For the "denial of service" data, CRF show the highest performance over other algorithms. The experimental results and their analysis have shown that CRF with N-gram will provide comparable prediction accuracy to practical cutting edge machine learning algorithms and will be useful as a component of actual misuse intrusion detectors.

Key words: N-gram, intrusion trace classification, misuse detection, conditional random field, support vector machines, UNM

INTRODUCTION

In intrusion detection systems based on machine learning techniques, most research work prefer N-gram based approaches. N-gram based approaches are notorious for their drawbacks. First of all, it is hard to expand due to insufficient data and memory. Many approaches use N-grams as features and the N-gram words in the system are overlapped each other which causes bias in feature counting. There are many diverse sequence models available in machine learning area however, there has been still insufficient investigation on application of these methodologies to intrusion detection. In this study, we consider applying conditional random field method (Lafferty *et al*., 2001) to misuse intrusion detection problems. In order to evaluate the performance of our proposed system, we compare our proposal with Naive Bayes algorithm and support vector machines on host based misuse intrusion detection benchmark datasets. Public host based misuse intrusion detection benchmark datasets are from University of New Mexico and MIT Lincoln Lab. The experimental results on the benchmark datasets indicate that conditional random field generates accurate misuse intrusion detector with comparable performance to Naive Bayes and support vector machines.

Literature review: Peng and Schuurmans (2003) proposed N-gram augmented Naive Bayes. They applied their algorithms to text classification. Silvescu *et al.* (2004) proposed inter-element dependency models. These models are similar to N-gram-augmented Naive Bayes. Kang and Kang (2012) has applied the inter-element dependency models to misuse intrusion detection task. Rieck and Laskov (2006) used N-gram language models for detecting new network attacks. They basically used a trie architecture (De La Briandais, 1959; Fredkin, 1960) to store the sequences.

Most intrusion detection researches have been attentive to the N-gram approach (Rieck and Laskov, 2006; Warrender *et al*., 1999; Forrest *et al*., 1994; Lee and Stolfo, 1998). It is interesting that there have been a few approaches (Liao and Vemuri, 2002; Kang *et al*., 2005; Liu *et al*., 2005) that try to explore another possibilities such as exploiting system call arguments or treating a bag/set of system calls for intrusion detection tasks. Liao and Vemuri (2002) transformed k-Nearest Neighbor (k-NN) algorithm for text classification method to intrusion detection tasks. Kang *et al.* (2005) have proposed a set/bag of system call features for anomaly and misuse intrusion detection. To the bag/set of system calls representation they have performed various machine algorithms with a view to misuse detection and anomaly

Corresponding Author: Dae-Ki Kang, Computer Engineering Major, Dongseo University, Busan, Korea

detection. From the experimental results, they showed that representing intrusion features as a bag of system calls is sufficient for misuse detection and sometimes effective for anomaly detection. Liu *et al.* (2005) have examined various feature representation of system calls. They concluded that exploiting system call alone is not always sufficient for some attacks including insider threats.

Finally, it is worthwhile to note that Forrest *et al.* (1994) proposed the first N-gram based intrusion detector called "Sequence Time-Delay Embedding (STIDE)". STIDE is basically an N-gram approach with a few thresholds.

MATERIALS AND METHODS

Proposed work

Problem definition: Firstly, we formally describe intrusion detection problem. We define $\Sigma = (s_1, s_2, s_3 \ ... \ , s_m)$ to be a set of system calls. Here, m (= $|\Sigma|$) is the number of the system calls. We define data set $D = \{\langle Z_i, c_i\rangle | \ Z_i \in \Sigma^*, c_i \in \{0, 1\}\}$ as a set of labeled program traces. Here, $Z_i = z_1, z_2, z_3, ..., z_l$ is an input sequence and c_i is its corresponding class label. The value of the class label c_i is 0 and 1. 0 is for "normal" state and 1 is for "intrusive" state. Therefore, given the data set D, an intrusion detection generation algorithm will try to induce a hypothesis (i.e., intrusion detector) $D = \{\langle Z_i, c_i\rangle | \ Z_i \in \Sigma^*, c_i \in \{0, 1\}\}$ by optimizing the given criteria. False positive rate, detection rate, f-1 measure and accuracy can be those criteria.

If an intrusion detector h is viewed as a probabilistic graphical model, then it will be used to estimate the probability P_h for a sequence Z. Its detailed algorithmic description can be formulated as follows: For each class denoting intrusion or normal state c_i, the algorithm calculates the probabilities $P_h(c_i)$ using all the training sequences Z coupled with the class c_i. For a new sequence Z, the algorithm tries to decide the most probable class c_h as follows:

$$c_h = \text{agrmax}_{c\in\{0,1\}} P_h(Z = z_1, z_2, ..., z_l | c) P_h(c)$$

Naive Bayes classifier: Note that Naive Bayes is one typical example of probabilistic graphical models. Naive Bayes classifier is simple and effective for text classification and other simple machine learning tasks. It has been shown that a bag of system calls can be coupled with Naive Bayes to generate effective misuse intrusion detector (Kang *et al*., 2005). However, this Naive Bayes classifier based intrusion detector will have an unrealistic assumption that each system call of the sequence is conditionally independent of the other system calls given the attack type. As for Naive Bayes, the classification of a new input trace will be formulated as follows:

$$C_{NB} = \text{agrmax}_{c\in\{0,1\}} P_h(c) \prod_i P_h(z_i | c)$$

NB N-gram (Naive Bayes with N-gram feature input): In N-gram based features we map sequences with variable length into a finite n-dimensional feature vector. In case of host based intrusion detection that monitors a program's behavior, this sequence is actually a program's trace. This means that we apply a sliding window of length n to program traces to generate N-gram features. The sliding window will move from the beginning of the program trace to the end. At each step, the window will move by one system call. After the completion, the sliding window will generate a bag of N-gram features from the trace. For these N-gram features, the probabilistic graphical model can be estimated as follows:

$$C_{NB\ n\text{-}gram} = \text{agrmax}_{c\in\{0,1\}} P_h(c) \prod_{i=1}^{l-n+1} P_h(z_i, z_{i+1}, ..., z_{i+n-1} | c)$$

Here, l is the length of the sequence. We explain one problem of NB N-gram approach. Note that a sliding window is scanned over an original program trace to generate N-gram features. This means that the sliding window based feature generator will consider one system call in the program trace at most n times in the resulting N-gram features. It is because the generator moves by one system call at each step. This N-gram feature generation systematically conflicts with the independence assumption in Naive Bayes.

Support vector machines: We explain Support Vector Machines (SVM) algorithm here. Before deep learning, SVM was one of the most efficient and accurate machine learning algorithms for data classification. It was first proposed by Vapnik (2000). Two strengths of SVM algorithm are margin maximizer and kernel trick. Using kernel trick, SVM maps the inputs into higher-dimension. Let, the training data to be described as:

$$(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}),..., (x^{(l)}, y^{(l)}), x \in R^n, y \in \{+1, -1\}$$

SVM training algorithm tries to calculate the maximum margin hyper plane. Note that this obtained hyper plane separates the two classes of data points which is obtained by maximizing the distance (i.e., margin) between the generated classifiers and the closest data points. These data points are called support vectors. From these support vectors, a hyper plane can be formulated as $w \cdot x + b = 0$; where denotes the dot product of wand the data point $x^{(i)}$. Note that w is the normal vector to the

hyperplane. We want to maximize the distance among the data point x and the hyperplane w·x+b = 0 which can be formulated by:

$$d(x) = \frac{|w \times x + b|}{\sqrt{\|w\|_2^2}} = \frac{|w \times x + b|}{\sqrt{\sum_{i=1}^{d} w_i^2}}$$

We are concerned on the nearest data points to the hyper plane because they actually affect the hyperplane, which is the decision boundary. Therefore, we can formulate the hyperplane and the minimum distance from the data points:

$$margin = \min_{x \in d} d(x) = \min_{x \in d} \frac{|w \times x + b|}{\sqrt{\sum_{i=1}^{d} w_i^2}}$$

That the support vectors of the two categories are lied down on the lines with the two hyper planes. The hyper planes are w·x+b = +1 and w·x+b = -1. The margin of the hyper planes is represented as follows:

$$\arg\max_{w, b} margin(w, b, D) =$$

$$\arg\max_{w, b} \min_{x_i \in d} d(x_i) = \min_{x_i \in D} \frac{|w \cdot x_i + b|}{\sqrt{\sum_{i=1}^{d} w_i^2}}$$

We can transform the problem of maximizing the margin to minimizing the following:

$$\min \|w\| \text{ subject to } y^{(i)}(w^{(T)} \times x^{(i)} + b) \geq 1, i = 1, 2, \ldots, m$$

We reformulate minimizing ||w|| as minimizing ½||w||:

$$\text{Min} \frac{1}{2} \|w\|^2 \text{ subject to } y^{(i)}(w^{(T)} \times x^{(i)} + b) \geq 1, i = 1, 2, \ldots, m$$

With this term, we can perform Quadratic Programming (QP) optimization using the lagrange multiplier α for the minimization problem above. So, far we have assumed that the data we are handling is linearly separable. However, actual data can contain anomalies. SVM uses a regularization mechanism to avoid generating predictive models that do not efficiently predict novel data. Even if the data can be linearly separable, a single outline value can disturb the decision boundary or a hyper plane with malfunctioning of regularization.

Regularization is applied to SVM to make it less sensitive to outliers. C in the regularization term is related with the number of training errors. The number of training errors is then weighted with the distances of training error points to their right class. This means that C is basically the regularization parameter of SVM. Support vector machines algorithm uses l_1 regularization. The problem of maximizing the hyper plane margin can be considered as a problem of minimizing ||w||. And the minimization problem can be solved as a QP optimization problem with the Lagrange multipliers. Therefore regularization should be applied to SVM and it makes SVM less sensitive to outliers. We introduce a tradeoff parameter C and multiply the distance of misclassified points from the margin. Basically C is the SVM's regularization parameter. Support vector machine algorithm in this study uses l_1 normalization.

Conditional Random Field (CRF): CRF is widely used statistical modeling methods. It is used for structural prediction in machine learning and pattern recognition. Generally classifiers do not consider neighboring features when they perform prediction from a sample label. In contrast, CRF considers adjacent features when performing prediction. In Natural Language Processing (NLP), linear chain based CRF is frequently applied for the prediction of a set of labels on a sequence of input samples. CRF can be considered a differential unidirectional graph which encrypts the known relationships of observable things and builds a consistent interpretation. It can also be used to predict and analyze labels of documents written in natural language, biological sequences and computer vision images. As an alternative to hidden Markov Models, CRF can be used for gene search and retrieval, image segmentation, object recognition and Part-of-Speech (POS) analysis.

Formally, CRF (Y, X) is an undirected graph G = (V, E) or can be considered as a Markov random field. If the conditional independence can be assumed in the input sequence, the theoretical structure of the graph can represent various forms. However, in general, Y nodes in an application often are represented as a simple chain (Lafferty *et al*., 2001). CRF has the advantage that the parameter independence condition is not needed compared with the Hidden Markov Model (HMM). In addition, CRF has the advantage of having no bias in comparison with the Maximum Entropy Markov Model (MEMM) (Lafferty *et al*., 2001). The following is the definition of a conditional random field.

Let G = (V, E) be a graph structure and $Y = (Y_v)_{v \in V}$ where Y represents the vertex of the graph G. And let E is the edge in the graph G. If a random variable Y_v for a random variable X shows a Markov property in the graph, i.e., $p(Y_v|X, Yw, w \neq v) = p(Y_v|X, Yw, w \sim v)$, then (X, Y) becomes a conditional random field (here w~v means w and v are neighbors each other).

Table 1: The number of instances for UNM denial of service attack

Attack	Positive	Negative	Total
Denial of service (stide)	105	13.726	13.831

Table 2: Accuracy (A) and False Positivity (FP) generated by CRF, NB n-gram and SVM n-gram from UNM denial of service data

	CRF		NB N-gram		SVM N-gram	
N	A	FP	A	FP	A	FP
1	98.69	0.00	98.69	0.92	99.98	0.01
2	99.76	0.00	99.24	0.04	99.99	0.00
3	99.81	0.00	99.06	0.69	N/A	N/A
4	99.87	0.00	99.18	0.68	N/A	N/A
5	99.91	0.00	99.24	0.65	N/A	N/A
6	99.99	0.00	99.32	0.63	N/A	N/A
7	99.99	0.00	99.40	0.59	N/A	N/A
8	99.99	0.00	99.52	0.47	N/A	N/A

RESULTS AND DISCUSSION

Experimental setup and results

Data set: The dataset we have tested is from University of New Mexico (UNM). UNM has provided a number of system call data sets. Each data set is generated from a specific exploit or attack. We tested "Denial of Service (DoS)" attack data (Table 1). UNM system call trace is a single program output. Sometimes there are several processes in one trace. When there are several processes in a trace, you have to create as many sequences as there are processes in the original trace. Therefore, if there are multiple processes in the input trace, multiple system call sequences are created in one trace. Table 2 shows the accuracy and false positive rate of the three algorithms (CRF, NB N-gram and SVM N-gram) on "denial of service" data. CRF shows the best performance when n is between 6 and 8 over Support Vector Machines (SVMs) and NB N-gram.

CONCLUSION

We discussed a method for classifying coherent sequences by applying a conditional random field model to the N-grams. We evaluated the performance of Support Vector Machines (SVMs) and Naive Bayes with N-gram functionality through experiments on intrusion detection benchmark data sets.

RECOMMENDATIONS

In future studies, we regard the system call arguments for accurate intrusion detection. Also it will be worthwhile to consider the structure of the audit record data (i.e., system calls and their arguments).

ACKNOWLEDGEMENT

Research was supported by Dongseo University, "Dong seo Frontier Project" Research Fund of 2015.

REFERENCES

De La Briandais, R., 1959. File searching using variable length keys. Proceedings of the Conference on Western Joint Computer, March 3-5, 1959, ACM, San Francisco, California, pp: 295-298.

Forrest, S., A.S. Perelson, L. Allen and R. Cherukuri, 1994. Self-nonself discrimination in a computer. Proceedings of the IEEE Computer Society Symposium on Security and Privacy, May 16-18, 1994, Oakland, CA, USA., pp: 202-212.

Fredkin, E., 1960. Trie memory. Commun. ACM., 3: 490-499.

Kang, D.K. and P. Kang, 2012. Intrusion trace classification using inter-element dependency models with k-truncated generalized suffix tree. Intl. J. Secur. Appl., 6: 385-390.

Kang, D.K., D. Fuller and V. Honavar, 2005. Learning classifiers for misuse and anomaly detection usin a bag of system calls representation. Proceeding of the 6th Annual IEEE SMC Workshop o Information Assurance (IAW'05), June 15-17, 2005, IEEE, West Point, New York, ISBN:0-7803-9290-6, pp: 118-125.

Lafferty, J., A. McCallum and F. Pereira, 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. Proceedings of the Eighteenth International Conference on Machine Learning ICML, June 28-July 2, 2001, ACM, San Francisco, California, USA, ISBN: 1-55860-778-1, pp: 282-289.

Lee, W. and S. Stolfo, 1998. Data mining approaches for intrusion detection. Proceedings of the 7th USENIX Security Symposium, January 26-29, 1998, USENIX Association, Berkeley, CA., USA., pp: 79-94.

Liao, Y. and V.R. Vemuri, 2002. Using text categorization techniques for intrusion detection. Proceedings of the 11th Symposium on USENIX Security Vol. 12, August 5-9, 2002, USENIX Association, San Francisco, California, pp: 51-59.

Liu, A., C. Martin, T. Hetherington and S. Matzner, 2005. A comparison of system call feature representations for insider threat detection. Proceedings from the 6th Annual IEEE SMC Workshop on Information Assurance (IAW'05), June 15-17, 2005, IEEE, West Point, New York, ISBN:0-7803-9290-6, pp: 340-347.

Peng, F. and D. Schuurmans, 2003. Combining Naive Bayes and n-gram language models for text classification. Proceedings of the 25th European Conference on IR Research (ECIR'03) Vol. 2633, April 14-16, 2003, Springer, Pisa, Italy, pp: 335-350.

Rieck, K. and P. Laskov, 2006. Detecting unknown network attacks using language models. Proceedings of the 3rd International Conference on Detection of Intrusions and Malware and Vulnerability Assessment, July 13-14, 2006, Springer, Berlin, Germany, pp: 74-90.

Silvescu, A., C. Andorf, D. Dobbs and V. Honavar, 2004. Inter-element dependency models for sequence classification. MSC Thesis, University of Io2wa, Iowa, USA.

Vapnik, V.N., 2000. The Nature of Statistical Learning Theory. 2nd Edn., Springer, New York, USA., ISBN: 9780387987804, Pages: 314.

Warrender, C., S. Forrest and B.A. Pearlmutter, 1999. Detecting intrusions using system calls: Alternative data models. Proceedings of the Symposium on Security and Privacy, May 9-12, 1999, Oakland, CA., USA., pp: 133-145.