Journal of King Saud University – Computer and Information Sciences (2012) **24**, 1–6

King Saud University

Journal of King Saud University – Computer and Information Sciences

www.ksu.edu.sa
www.sciencedirect.com

ORIGINAL ARTICLE

Bayesian based intrusion detection system

Hesham Altwaijry *, Saeed Algarny

Computer Engineering Department, King Saud University, P.O. Box 51178, Riyadh 11543, Saudi Arabia

Received 14 December 2010; accepted 24 April 2011
Available online 17 November 2011

KEYWORDS

Intrusion detection system (IDS);
Bayesian filter;
KDD '99

Abstract In this paper an intrusion detection system is developed using Bayesian probability. The system developed is a naive Bayesian classifier that is used to identify possible intrusions. The system is trained a priori using a subset of the KDD dataset. The trained classifier is then tested using a larger subset of KDD dataset. The Bayesian classifier was able to detect intrusion with a superior detection rate.

1. Introduction

As network attacks have increased in number and severity over the past few years, Intrusion Detection Systems (IDSs) have become a necessary addition to the security infrastructure of most organizations. These systems are software or hardware schemes that automate the process of monitoring events that occur in a computer system or network and analyzing them for signs of security problems (Crothers, 2003; Bace and Mell, 2001).

Deploying highly effective IDS systems is extremely challenging. For instance until an IDS is properly tuned to a specific environment, there will be thousands of alerts generated daily. Although most of these alerts are incorrect and thus are false alerts, it is not obvious whether the alert is positive or negative until they have been investigated. There have been many techniques proposed to lessen these false alerts and improve the performance of the system. Agarwal and Joshi (2000) used a two-stage general-to specific framework for learning a rule-based model (PNrule). This model can classify models of a data set that has widely different class distributions in the training data set. Levin (2000) used a data-mining tool for classification of data and prediction of new cases using automatically generated decision trees. In this paper will show that the use of Bayesian probability is very promising in reducing the false positive alert rate.

Bayesian probability is an interpretation of the probability calculus which holds that the concept of a probability can be defined as the degree to which a person (or community) believes that proposition is true. Currently Bayesian theory is used in email spam-filters (Grapham, 2004; Issac et al., 2009; Alkabani et al., 2006), Speech recognition (Chien et al., 2006), Pattern Recognition (Shi and Manduchi, 2003), and Intrusion Detection (Kruegel et al., 2003; Cemerlic et al., 2008; Mehdi et al., 2007).

2. Bayesian theory

Bayesian theory is named after Thomas Bayes (1702–1761), his theory can be explained as follows:

* Corresponding author.
E-mail addresses: twaijry@ksu.edu.sa (H. Altwaijry), sagarny@stc.com.sa (S. Algarny).

Peer review under responsibility of King Saud University.
doi:10.1016/j.jksuci.2011.10.001

ELSEVIER **Production and hosting by Elsevier**

If the events $A_1, A_2, \ldots$ and A_n constitute a partition of the sample space S such that $P(A_k) \neq 0$ for $k = 1,2,\ldots,n$, then for any event B such that $P(B) \neq 0$:

$$P(A_i|B) = \frac{P(A_i \cap B)}{P(B)} = \frac{P(A_i)P(B|A_i)}{\sum_{k=1}^{n} P(A_k)P(B|A_k)} = \frac{P(A_i)P(B|A_i)}{P(B)}$$

In recent years Bayesian networks have been used across a wide range of fields in computer science (Darwiche et al., 2010) because of their ability to obtain a coherent result from probabilistic information about a situation. Additionally there are many efficient algorithms that can be used to derive the results from the information. It is believed that this ability and readily available algorithms would allow one to construct an efficient IDS system.

3. KDD-99 dataset

To test our IDS system we used the *DARPA KDD99 Intrusion Detection Evaluation dataset* (KDD99, 1999). This dataset was created by Lincoln Laboratory at MIT and was used in *The Third International Knowledge Discovery and Data Mining Tools Competition*, which was held in conjunction with *KDD-99 The Fifth International Conference on Knowledge Discovery and Data Mining* (KDD99, 1999). This dataset is one of the most realistic publicly available sets that include actual attacks (Aickelin et al., 2007). Therefore, researchers have been using this dataset to design and evaluate their intrusion detection systems. Additionally a common dataset allows researchers to compare experimental results. The KDD data set was acquired from raw *tcpdump* data for a length of nine weeks. It is made up of a large number of network traffic activities that include both normal and malicious connections. The *KDD99* data set includes three independent sets; "whole KDD", "10% KDD", and "corrected KDD". In our experiments we have used the "10% KDD" and the "corrected KDD" as our training and testing set, respectively. Table 1 summarizes the number of samples in each dataset:

The training set contains a total of 22 training attack types. Additionally the "corrected KDD" testing set includes an additional 17 attack types. Therefore there are 39 attack types that are included in the testing set and these attacks can be classified into one of the four main classes;

- DOS: Denial of Service attacks.
- Probe: another attack type sometimes called Probing.
- U2R: User to Root attacks.
- R2L: Remote to Local attacks.

DoS and Probe attacks are different from the normal traffic data and can be easily separated from normal activities since they come with greater frequency in a short period of time. On the other hand, U2R and R2L attacks are embedded in the data portions of the packets and normally involve only a single connection. It becomes difficult to achieve satisfactory detection accuracy for detecting these two attacks (Lee et al. (1999)).

The KDD-99 network TCP connections have 41-features per connection (record). These features can be divided into four categories (Chou, 2007):

Basis features: Features 1-9 are the basic features that are derived from packet header without inspecting the payload.

Content features: Domain knowledge is used to assess the payload of the original TCP packets. This includes features such as the number of failed login attempts.

Time-based traffic features: These are features that capture properties that mature over a 2-s temporal window. An example is the number of connections to the same host over the 2-s interval.

Host-based traffic features: These features utilize a historical window estimated over the number of connection instead of time. They are designed to assess attacks, which span intervals longer than 2s.

4. Bayesian filter

The Bayesian IDS is built out of a naïve Bayesian classifier. The classifier is anomaly based. It works by recognizing that features have different probabilities of occurring in attacks and in normal TCP traffic. The filter is trained by giving it classified traffic. It will then adjust the probabilities for each feature. After training, the filter will calculate the probabilities for each TCP connection and classify it as either normal TCP traffic or an attack. Therefore our Bayesian filter consists of the following two components:

4.1. Training engine

Fig. 1 shows the block diagram for the Bayesian filter that is constructed for the IDS system. For each input record there is a label describing the type of connection. We use this label to train the engine as following:

- First the number of good records and bad records in the training dataset are calculated.
- Then two hash tables are created; the first one includes the frequency of each attribute for normal records, and the second one includes the frequency of each attribute of the not normal records.
- Finally, a third hash table is created. This table contains each attribute from the normal and not normal records and it is scored using the following formula

where

- B is the frequency of that attribute in the hash table related to not-normal file.
- G the frequency of that attribute in the hash table related to normal file.

Table 1 Basic characteristics of the KDD 99 intrusion detection datasets in terms of number of samples.

Dataset	Normal	DoS	Probing	R2L	U2R	Total
Whl KDD	972,780	3,883,370	41,102	1126	52	4,898,430
10% KDD	97,278	391,458	4107	1126	52	494,020
KDD corr	60,593	229,853	4166	16,189	228	311,029

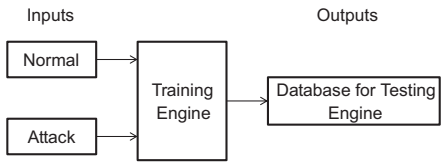

Figure 1 The training engine.

4.2. Testing engine

After training the engine, it is tested by loading the KDD corrected dataset. The following formula is applied to obtain a probability of whether the record is normal or not where

- *n*: number of attributes that we need to use to test the required record
- score(i): the score of the attribute

The record is considered to be an attack if the P(record) is greater than a specified threshold.

5. Experimental setup

Many experiments were conducted until we finally achieved results that are comparable to what has been published in the literature. In this section, the most important experiments will be explained. Experiments differ basically in the training data used to build the database, which accordingly affects the accuracy of the test. Additionally, the number of features and level of threshold used in the testing engine makes a big difference in the results. Therefore, all experiments presented differ due to manipulation of the following inputs: training data, features and threshold.

5.1. Training data

Using the 10% KDD data set we have 494,020 records that can be used to train the training engine. These training records consist of normal (non attack) records and known attack records distributed among the four attacks types: DoS, Probe, U2R and R2L. In all the experiments that we will present we will use the normal records (non attack), adding to them the appropriate not-normal (attack) records. In each experiment, we select one type of not-normal record, except for the first experiment, in which all types of records were used. The objective of varying between not-normal records in each experiment is to see the effect of each different attack on the results. This method of altering the attack type in each experiment shows some interesting results (see experiment 5), as the detection rate of U2R attacks was higher when using R2L records to train the engine than when using U2R (see Experiment 4).

5.2. Features

Since the data record consists of 41 features, we can select between them and perform a very large number of combinations. We have selected the features as follows:

1. Using specific features like basic features (1–9), content features (10–22) and traffic features (23–31)
2. Using all the 41 features.
3. Using selected features by inspecting the score map.

The experiments performed show that the first method does not yield good results compared to the second method. However, the results from using all features are worse than results previously published. Consequently, it was necessary to find key features that can lead to better results. To do so, we analyzed the score map that was built by the training engine to see the highest value that can result in a score that is above the threshold so that the detection rate maybe increased. Among the many features and after many experiments, we ended up with three features that raised the detection rate of R2L attack to 85% as will be explained in Experiment 5.

5.3. Threshold

This is the level that we used to distinguish between normal records and attacks. The threshold value was adjusted between the experiments to increase the detection rate.

After performing each experiment, we analyzed the results based upon the number of normal and not-normal records that the testing engine succeeded or failed in classifying. We use the following expressions to analyze the data:

True negative (TN): The percentage of valid records that are correctly classified.

True positive (TP): The percentage of attack records that are correctly classified.

False positive (FP): The percentage of records that were incorrectly classified as attacks whereas in fact they are valid activities.

False negative (FN): The percentage of records that were incorrectly classified as valid activities whereas in fact they are attacks.

6. Results

Each of the following five experiments consists of five sub-experiments. However, we categorize them into five main experiments since we use the same training engine to perform the five sub-experiments. For example, using the normal and not-normal records in the training engine for the first group of experiments, we test not-normal records, DOS, Probing, U2R and R2L. Each test has its own setup in terms of features and threshold. Table 2 summarizes the records used in training the Bayesian filter for all five experiments

6.1. Experiment 1: using all attack records and normal (non-attack) records

Table 3 shows what was used for each test in this experiment, after the Bayesian filter had been trained using 65,593 normal

Table 2 Number of records used in experiments.

Experiment	Type	Number	Type	Number
Experiment 1	Normal	65,593	All Attacks	401,195
Experiment 2	Normal	92,827	DOS	280,504
Experiment 3	Normal	92,827	Probing	4107
Experiment 4	Normal	92,827	U2R	188
Experiment 5	Normal	92,827	R2L	1126

Table 3 Experiment 1 testing environment.

Testing data	No of records	Features	Threshold
Normal	60,593	All	0.9
Not Normal	250,436	All	0.9
DOS	165,299	All	0.9
Probing	4166	All	0.9
U2R	188	All	0.9
R2L	16,180	All	0.9

Table 4 Test results percentages.

Test	TN	TP	FN	FP	DR	CR
All attacks	99.03	89.70	0.97	10.30	89.70	94.37
DOS	99.03	99.36	0.97	0.64	99.36	99.20
Probing	99.03	57.15	0.97	42.85	57.15	78.09
U2R	99.03	0.00	0.97	100.00	0.00	49.52
R2L	99.03	0.00	0.97	100.00	0.00	49.52

Table 5 Experiment 2 testing environment.

Testing data	No of records	Features	Threshold
Normal	60,593	All	0.9
All attacks	250,436	All	0.9
DOS	165,299	All	0.9
Probing	4166	23,24,31	0.6
U2R	188	23,24,31	0.6
R2L	16,180	23,24,31	0.6

Table 6 Experiment 2 test result percentages.

Test	TN	TP	FN	FP	DR	CR
All attacks	99.60	65.50	0.40	34.50	65.50	82.55
DOS	99.60	99.24	0.40	0.76	99.24	99.42
Probing	99.60	17.73	0.40	82.27	17.73	58.67
U2R	99.60	0.00	0.40	100.00	0.00	49.80
R2L	99.60	0.00	0.40	100.00	0.00	49.80

Table 7 Experiment 3 testing environment.

Testing data	No of records	Features	Threshold
Normal	60,593	All	0.9
All attacks	250,436	All	0.9
DOS	165,299	All	0.9
Probing	4166	All	0.9
U2R	188	23,24,31	0.6
R2L	16,180	23,24,31	0.6

records and 401,195 attack records. We then performed five tests on the trained network to see the effects on each attack type. Normal records were used in all tests and each test used one attack at a time. For example, the first test in the experiment used 60,593 normal records and 250,436 not-normal records.

From Table 4, we can see that normal records were detected with high accuracy = 99.03%, while the best detection rate was for the DOS attack type which had 99.36% accuracy while the worst results were for U2R, and R2L attacks with 0%. The overall detection rate did not reflect the actual accuracy of the filter if the TP rate was low, as in U2R and R2L.

Table 8 Experiment 3 test result percentages.

Test	TN	TP	FN	FP	DR	CR
Not normal	99.40	71.00	0.60	29.00	71.00	85.20
DOS	99.40	70.30	0.60	29.70	70.30	84.85
Probing	99.40	81.90	0.60	18.10	81.90	90.65
U2R	99.40	91.50	0.60	8.50	91.50	95.45
R2L	99.40	61.50	0.60	38.50	61.50	80.45

Table 9 Experiment 4 testing environment.

Testing data	No of records	Features	Threshold
Normal	60,593	All	0.9
Not normal	250,436	23,24,31	0.6
DOS	165,299	All	0.6
Probing	4166	23,24,31	0.6
U2R	188	23,24,31	0.6
R2L	16,180	23,24,31	0.6

Table 10 Experiment 4 test result percentages.

Test	TN	TP	FN	FP	DR	CR
Not normal	99.70	76.50	0.30	23.50	76.50	88.10
DOS	99.70	0.02	0.30	99.98	0.02	49.86
Probing	99.70	71.60	0.30	28.40	71.60	85.65
U2R	99.70	93.00	0.30	7.00	93.00	96.35
R2L	99.70	61.50	0.30	38.50	61.50	80.60

6.2. Experiment 2: using DOS records and normal (non-attack) records

Table 5 shows what was used to test the Bayesian filter for the second experiment. While in Table 6, we can see that normal records were detected with high accuracy: TN = 99.6%. The best DR was for the DOS attack type, which was 99.24%, as expected since it was trained using only DOS attacks, and the worst result was for U2R and R2L attacks with 0%. The overall DR did not reflect the actual accuracy of the filter if the TP rate was low as in U2R and R2L.

6.3. Experiment 3: using Probing records and normal records

Table 7 shows what was used to test the Bayesian filter for the third experiment. While Table 8, shows that normal records are detected with high accuracy: TN = 99.4%.The best accuracy was for the U2R attack type, which was 91.5% although the training data was using PROBING attacks.

By analyzing the features, it is found that U2R and R2L attacks can be detected with a better rate if selected features are chosen. These features are 23(count), 24(serror_rate) and 31(Srv_diff_host_rate).

Table 11 Experiment 5 testing environment.

Testing data	No of records	Features	Threshold
Normal	60,593	All	0.9
Not normal	250,436	All	0.9
DOS	165,299	All	0.9
Probing	4,166	23,24,31	0.6
U2R	188	23,24,31	0.6
R2L	16,180	23,24,31	0.6

Table 12 Experiment 5 test result percentages.

Test	TN	TP	FN	FP	DR	CR
Not normal	68.03	10.00	31.97	90.00	10.00	39.02
DOS	68.03	2.00	31.97	98.00	2.00	35.02
Probing	68.03	63.60	31.97	36.40	63.60	65.82
U2R	68.03	96.30	31.97	3.70	96.30	82.17
R2L	68.03	85.35	31.97	14.65	85.35	76.69

Table 13 Detection rate for various algorithms.

Algorithm	DOS	Probe	U2R	R2l
KDD cup winner	97.10	83.30	12.30	8.40
SOM map	95.10	64.30	22.90	11.30
Gaussian classifier	82.40	90.20	22.80	9.60
K-means clustering	97.30	87.60	29.80	6.40
Nearest clustering	97.10	88.80	2.20	3.40
Radial basis	73.00	93.20	6.10	5.90
C4.5 decision tree	97.00	80.80	1.8	4.6
PN-rule			6.60	10.70
Linear GP	96.70	85.70	1.30	9.30
Online k-means	69.81	99.62	49.45	6.48
SVM	99.90	67.31	0.00	29.09
KMO + SVM	75.76	99.61	49.45	22.24
Backpropagation	97.23	96.63	87.71	30.97

6.4. Experiment 4: using U2R records and normal records

Table 9 shows what was used to test the Bayesian filter for the fourth experiment. While Table 10, shows that normal records were detected with high accuracy: TN = 99.7%. The best DR was for the U2R attack type which was 93.0% and the worst result was for DOS attacks with 0.02%. Also in this experiment, like the previous one, attacks show sensitivity if using features 23, 24 and 31.

6.5. Experiment 5: using L2R records and normal records

Table 11 shows what was used to test the Bayesian filter for the fifth experiment. While Table 12, shows that normal records accuracy decreased to 68.03% since the threshold value was decreased to improve the TP for R2L. The best DR was for the U2R attack type which was 96.3% although the training data used the R2L type. The worst result was for DOS attacks with 2%. Also in this experiment, like the previous two experiments, showed sensitivity to using features 23, 24 and 31.

The previous five experiments showed that by tweaking selected features it is possible to achieve high accuracy for all four types of attacks.

The results obtained by using Bayesian filters are comprable to what has been presented in Chou's PhD thesis (Chou, 2007) where he reported the results of most algorithms (Table 13). However, Bayesian filters were able to achieve superior results for in detecting U2R and R2L attacks.

7. Conclusions

Since the goal of this research was to improve the accuracy of the R2L attack using Bayesian methods, we have succeeded in achieving our target by using the Bayesian method as an engine to classify the data accordingly. We achieve results superior to Chou in his PhD dissertation (Chou, 2007), where he achieved a DR of 69.82% for the R2L. Our research results show that we could have better results for R2L attack with a DR of 85.35% by using the three features: 23, 24 and 31 and a threshold value of 0.6. However, the CR which equals 76.69% is considered low comparing to Chou result because we used a low threshold value which reduces the accuracy of detection of normal records (TN) but increases the DR for R2L attack. To improve the accuracy of an IDS system we propose that we should use several Bayesian filters in parallel with each filter optimized to detect one type of record; this can be a good subject for further research in this field.

References

Agarwal, R., Joshi, M.V., 2000. "PNrule: A New Framework for Learning Classifier Models in Data Mining (A Case-Study in Network Intrusion Detection)," Department of Computer Science, University of Minnesota TR 00-015.

Aickelin, U., Twycross, J., Hesketh-Roberts, T., 2007. Rule generalization in intrusion detection systems using SNORT. International Journal of Electronic Security and Digital Forensics 1 (1), 101–116.

Alkabani, Y.M., El-Kharashi, M.W., Bedor, H.S., 2006. Hardware/software partitioning of a Bayesian spam filter via hardware profiling. In: IEEE International Symposium on Industrial Electronics, Montreal, Canada, pp. 3264–3269.

Bace, R., Mell, P., 2001. NIST Special Publication on Intrusion Detection Systems. National Institute of Standards and Technology.

Cemerlic, A., Yang, L., Kizza, J.M., 2008. Network intrusion detection based on Bayesian networks. In: Proceedings of the Twentieth International Conference on SOftware Engineering and Knowledge Engineering (SEKE'2008), San Francisco, CA, USA, pp. 791–794.

Chien, J.-T., Huang, C.-H., Shinoda, K., Furui, S., 2006. Towards optimal Bayes decision for speech recognition. In: IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP, Toulouse.

Chou, T.S., 2007. Ensemble fuzzy belief intrusion detection design. Florida International University, Paper AAI3299199, 2007.

Crothers, T., 2003. Implementing Intrusion Detection Systems: A Hands-on Guide for Securing the Network. Wiley.

Darwiche, A., 2010. Bayesian networks. Communications of the ACM 53 (12), 80–90.

Grapham, P., 2004. Hackers and Painters: Big Ideas from the Computer Age. O'Reilly.

Issac, B., Jap, W.J., Sutanto, J.H., 2009. Improved Bayesian anti-spam filter implementation and analysis on independent spam corpuses. In: International Conference on Computer Engineering and Technology, ICCET, Singapore, pp. 326–330.

KDD'99 archive: The Fifth International Conference on Knowledge Discovery and Data Mining. [Online]. <http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html>.

Kruegel, C., Mutz, D., Robertson, W., Valeur, F., 2003. Bayesian event classification for intrusion detection. In: 19th Annual

Computer Security Applications Conference (ACSAC). IEEE Computer Society, Las Vegas, NV, USA, pp. 14–23.

Lee, W., Stolfo, S.J., Mok, K.W., 1999. A data mining framework for building intrusion detection models. In: Proceedings of the 1999 IEEE Symposium on Security and Privacy, Oakland, CA, USA, 1999, pp. 120–132.

Levin, I., 2000. KDD-99 Classifier Learning Contest LLSoft's Results Overview. ACM SIGKDD Explorations I (2), 67–75.

Mehdi, M., Zair, A., Anou, A., Bensebti, M., 2007. A Bayesian networks in intrusion detection systems. Journal of Computer Science 3 (5), 259–265.

Shi, X., Manduchi, R., 2003. A study on Bayes feature fusion for image classification. In: Conference on Computer Vision and Pattern Recognition Workshop, CVPRW, Madison, Wisconsin, USA, pp. 95–95.