Two Level Anomaly Detection Classifier

Azeem Khan
Dublin City University
School of Computing
Dublin, Ireland
raeeska2@computing.dcu.ie

Shehroz Khan *
Department of Information Technology
National University of Galway
Galway, Ireland
s.khan1@nuigalway.ie

Abstract

This paper proposes two-level strategy for building the anomaly detection classifier, namely, macro level and micro level classification. The former intend to classify network data on a broader perspective to predict whether it is normal or a potential attack. The later classifies individual anomalies within each category of known attacks. The paper also investigates various feature selection techniques for choosing relevant features and study its effect on the performance of the anomaly detection classifiers. Experiments suggest that employing feature selection along with the proposed approach give anomaly detection rate of upto 99%.

Keywords: Network anomaly detection, machine learning, feature selection, intrusion detection.

1 Introduction

As computer networks have grown in importance with the widespread use in different organizations, they are becoming target more so than before to malicious attacks. There are defensive mechanisms available such as firewalls that can provide security up to reasonable level by blocking the intrusions. With the advancement in technology, malicious attackers are becoming more sophisticated in intruding across the computer networks. We may not only want to block intrusions but where the attacks are internal to an organization we may want to detect them too. Intrusion Detection Systems (IDSs) should be used to leverage enhanced security to the network [5]. Within a computer network, an IDS monitors the network traffic and raises a alarm if it identifies any anomaly in the network traffic that deviates from the normal behavior.

Generally the detection techniques employed by most of the IDS are signature based, which tries to search for patterns or signatures of the already known malicious attacks [4]. The advantage of such kind of system is that signatures can be developed for known attacks. The signature based IDS process can be faster, however, the disadvantage with this technique is that it can only identify known attacks, that provide a room for the new or unknown attacks to creep inside the network.

Machine Learning (ML) is concerned with design and development of techniques that allow systems to learn from historical data and predict the behavior on unseen samples [14]. Various researcher have used ML techniques for developing IDS [16][6][9]. The advantage of such kind of IDS is that it is not dependent on the databases or signature of already known attacks. ML-based IDS build models based on normal and attack traffic data and tries to classify unseen data as safe or unsafe. Disadvantage of this method can be that it could either do over fitting or if parameters are not properly tuned it may raise false alarms rate.

The rest of the paper is organized as follows. Section 2 discusses the related work on ML based IDS. Section 3 describes the data set used for the experimental study in this paper. Section 4 proposes a two-level anomaly classification strategy, namely macro-level and micro-level. The section also explains the process of feature selection and the classification algorithms used. Experimental results, analysis and evaluations are explained in section 5 . Concluding remarks and future work are presented in section 6.

2 Related work

Machine learning techniques have been used widely by researchers to address anomaly detection problems across the computer networks. ML-based anomaly detection techniques attempt to build a model over the historical normal and anomalous network data and then attempt to predict whether a new data packet is normal traffic or not. EMERALD (Event Monitoring Enabling Responses to Anomalous Live Disturbances) [13] is a technique that uses past records to build model, it then compare the distribution of new data with the model build from past records.Various researchers

*Corresponding Author

have used KDD Cup'99 data set [1] for network anomaly detection (which is refined form of DARPA'98 data set [2]). Panda and Patra [11] used Naive Bayes for anomaly detection and achieved detection rate of 95%. Faroun and Boukelif [9] used Neural Networks with K-mean clustering and showed detection rate of 92%. Gaddam and Phoha [16] proposed a method to cascade clustering and decision tree for classifying anomalous and normal data. Giacinto and Roli [6] performed anomaly detection using three meta-classifiers on the output of neural network. Sabhnani and Serpen [12] compared nine different machine learning algorithm results and cited that no one algorithm can detect all attacks. Mukkamala and Sung [15] addressed the issue related to the importance of feature selection and used Support vector machine algorithm on DARPA'98 data set .

3 Data Sets Constructed for Experimental Evaluation

For the purpose of experimental study we have used the KDD Cup '99 Network Anomaly Data. This data comprises of 20 attacks and normal traffic, with more than 4.9 million instances defined by 41 attributes. We constructed two separate data sets from the original KDD Cup '99 data sets to

- Build a learning data set that contains instances from normal and attack (Macro level classification). The idea here is to develop classifier that can label a new data as normal traffic or attack.
- Build a learning data set that contains instances from the attack data. (Micro-level classification). This kind of analysis would help a network security analyst in designing network attack specific strategies.

Since the original size of KDD Cup'99 data set is very large to scale with our implementation of classifiers and due to limitation of memory resources we used a sampled version of the data set in our experiments. Given below are the details of the two data sets that were constructed:

3.1 Data set one (D1)

This data set consists of 50000 instances of normal and 54695 instances of attack data with total of 104695 instances chosen randomly. This data set is used for macro-level classification in section 5.1.

3.2 Data set two (D2)

This data set consists of 54695 instances of attack with 20 categories of attacks that will be used for micro-level classification. For details on data set D2 refer to table 2 section 5.

4 Proposed Two Level Classification

4.1 Two Level Anomaly Classifier

In this section we propose a two-level anomaly classification strategy. We hypothesize that a) various kinds of attacks are different from each other, yet they affect a network in a similar way, and b) they are also dissimilar from the normal traffic in a similar way. The diagrammatic representation of the proposed two level anomaly classification is shown in figure 1. The network traffic which is a mix of normal and attack traffic first passes through stage one, i.e. macro level classifier. The classifier should label the data as normal or attack. In the second stage, the attack data will further pass through the stage two (micro-level classifier) that would classify each of the attacks individually.

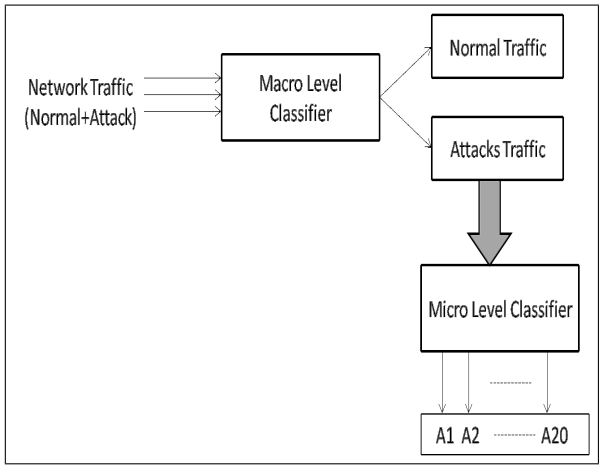

Figure 1. Two Level Classification Strategy

4.2 Feature selection

Feature selection is useful step to reduce dimensionality of data set and scrape out redundant or less informative features from a data set. A good feature selection technique should speed up the computing time and may increase the performance of a classifier [17] (which is detection of anomaly in our case). We used three different methods to study feature redundancy in the data sets and to identify the best performance results.

- Information Gain (*InfoGain*) [17].
- Chi Squared (χ^2) [7].
- Gain Ratio (*GainRatio*) [7].

4.3 Algorithms Used

Machine learning algorithms learn predictive models based on the historical data over a set of predefined categories, and then attempt to predict unlabeled test samples into one of those categories. In this paper we have used *C4.5* - Decision Tree based classification algorithm [8] and Naive Bayes (*NB*) algorithm [10] for detecting anomaly in the network traffic data.

For our implementation of classification algorithms and feature selection, the open source machine learning API - Weka [3] is used.

5 Experiments

The experiments were conducted using the data sets described in section 3. We outline our experiments under two data perspectives:

- Macro Level Classification.
- Micro Level Classification.

As the number of instances of various attacks and normal data are highly imbalanced, True Positive Rate (*TPR*) and False Positive Rate (*FPR*) are used to evaluate the comparative performance of different classification models (instead of Accuracy).

We have also used *Average TPR* and *Average FPR* to assess the overall TP and FP rate for particular classification scheme. They are defined as:

$$AvgTPR = \frac{Sum\ of\ TPR\ of\ each\ category}{total\ number\ of\ category} \quad (1)$$

$$AvgFPR = \frac{Sum\ of\ FPR\ of\ each\ category}{total\ number\ of\ category} \quad (2)$$

5.1 Macro Level Classification

For macro level classification data set D1 is used. Firstly, classification is done using all 41 attributes with both *C4.5* and *NB*. Then we employed feature selection methods as mention in section 4. Top ten ranked attributes are used in our experiment.

By looking at table 1, it can be inferred that decision tree algorithm is able to classify both normal and attack category with 99% *TPR* and 0.1% *FPR*. Using feature selection methods with *C4.5* (*InfoGain* and *Chi squared*) we observe that they did not affect much the classification accuracy as compared to using just *C4.5*, however they reduce the computational time taken to build model.

Nearly 98.6% *TPR* and 2% *FPR* is obtained using *NB* classification and almost similar results are produced by using three feature selection methods. Due to limitation of space we are skipping the details of results for *NB*. It is to be noted that *NB* results (with and without feature selection were worse than using *C4.5* with and without feature selection).

Taking in consideration all three performance measure *TPR, FPR* and time taken to build model (see table 1), it is observed that *C4.5* with *InfoGain* produced best results among all methods studied.

5.2 Micro Level Classification

Data set D2 is used for micro level classification. This data set includes the 20 attacks present in the data set. Both *C4.5* and *NB* algorithm were used for detecting these attacks. Furthermore, we reduce the number of attributes using Feature Selection methods (see section 4.2) to top ten most informative features.

C4.5 TPR results (see table 2 and 3) show that most of the attacks categories are classified except for three which have less number of instances in the data set. *FPR* is zero for most of the categories which means very few misclassifications. If we compare *C4.5* and *C4.5* with feature selection methods we can trace out that *C4.5* with *InfoGain* shows best classification results, and least time taken to build model among all classification methods.

Due to limitation of space we are unable to show the results of *NB* for micro level classification. *NB* was able to classify almost all attacks except for two categories attacks which have less number of instances. *NB FPR* shows small percentage of misclassification with almost half of the categories of attacks.

Taken into account all three performance measures i.e with, *TPR, FPR,* time taken to build model and comparing all techniques of classification with *C4.5* and *NB*, *C4.5* with *InfoGain* produce best results among all methods.

5.3 Comparison between one stage and two stage classification

A one stage classification method can be defined as employing classification algorithm on full data and try to separate out normal and each of the attacks in one go. As opposed to one stage classification approach, in our proposed two-stage classification approach, we first classify whether the data is normal or not at macro level and if not normal than classify the attacks into individual categories at micro level. By doing so we can reduce the number of comparisons any unknown sample has to undergo for being classified (provided high *TPR* and low *FPR* at stage 1 and 2 of the proposed classifier). To compare the performance of two stage classification with one stage classification we did few experiments (see tables 4 and 5).

Classification Algorithm	Normal TPR	Attack TPR	Normal FPR	Attack FPR	Avg TPR	Avg FPR	Time taken to build model in sec
C4.5	0.999	0.999	0.001	0.001	0.999	0.001	67.81
C4.5+InfoGain	0.999	0.999	0.001	0.001	0.999	0.001	6.77
C4.5+Chi squared	0.999	0.999	0.001	0.001	0.999	0.001	8.72
C4.5+GainRatio	0.999	0.997	0.003	0.001	0.998	0.002	9.19

Table 1. C4.5 TPR/FPR and Time taken to build model for macro level classification

Attacks	*Number of Instances*	*C 4.5 TPR*	*C4.5 + InfoGain TPR*	*C4.5 + Chi squared TPR*	*C4.5 + GainRatio TPR*	*C 4.5 FPR*	*C4.5 + InfoGain FPR*	*C4.5 + Chi squared FPR*	*C4.5 + GainRatio FPR*
satan	10903	0.999	0.996	0.996	0.999	0	0.003	0.008	0.680
ipsweep	10144	0.998	0.970	0.954	0	0.007	0.007	0.007	0
smurf	10000	1	1	1	0	0	0	0	0
septune	9994	1	1	0.999	1	0	0.001	0.002	0.018
portsweep	8609	0.999	0.985	0.951	0.001	0	0	0.004	0
nmap	2000	0.846	0.850	0.852	0.124	0	0.006	0.006	0
back	2000	1	1	0.999	0.997	0	0	0	0
teardrop	681	0.997	1	1	0.991	0	0	0	0
pod	202	1	1	1	0.985	0	0	0	0
guess passwd	53	0.962	0.962	0.962	0.943	0	0	0	0
buffer overflow	25	0.960	0.840	0.680	0.520	0	0	0	0
land	21	0.905	0.857	0.905	0	0	0	0	0
warezmaster	20	0.750	0.700	0.800	0	0	0	0	0
imap	12	0.833	0.500	0.500	0	0	0	0	0
ftp write	8	0.375	0.500	0.625	0	0	0	0	0
multihop	6	0	0	0	0	0	0	0	0
phf	3	0.333	0.667	1	1	0	0	0	0
perl	2	0	0.500	0.500	0	0	0	0	0
loadmodule	9	0.667	0.889	0	0	0	0	0	0
rootkit	3	0	0	0	0	0	0	0	0

Table 2. C4.5 TPR and FPR for micro level classification

Classification Algorithm	Time taken to build model in sec
C4.5	29.02
C4.5+InfoGain	3.91
C4.5+ Chi squared	4.9
C4.5+GainRatio	4

Table 3. Time taken to build model by C4.5 for micro level classification

Classification Algorithm	*C4.5*	*C4.5 + InfoGain*	*C4.5 + Chi squared*	*C4.5 + GainRatio*
Avg TPR	0.7439	0.7721	0.7486	0.4075
Avg FPR	0.0003	0.0008	0.0013	0.0333

Table 4. Two stage classification

Classification Algorithm	*C4.5*	*C4.5 + InfoGain*	*C4.5 + Chi squared*	*C4.5 + GainRatio*
Avg TPR	0.7000	0.6489	0.6763	0.4412
Avg FPR	0.0002	0.00038	0.0006	0.0211

Table 5. One stage classification

Results shows increase in detection rate of *C4.5* and its feature selection methods in case of the proposed two stage classification method. Similarly, in case of *NB* two stage classification methods shows better detection rate over one stage method. This lead to the conclusion that, doing classification in two stage can increase the anomaly detection rate. We also note that the combined time taken to build the classification model for macro and micro stage is also less than the one stage classification method.

5.4 Analysis of Features

In order to investigate the important features that might be of interest for a Network Security Analyst, we did an analysis to find common features among the studied feature selection methods. The criterion we choose to select an attribute as 'important' is: If an attribute is occurring in two or more feature selection algorithms then it is deemed as more important.

Table 6 shows that results obtained by using common features are similar to the best results obtained in macro and micro level classification. The important features for macros level were: *protocol type, service, flag, source bytes, destination bytes, count, srv count,dst host srv count, dst host diff srv rate, dst host same src port*. For micro level classification common features were: *service, flag, source bytes, wrong fragment, hot, compromised, count, srv count, dst host srv count*. Hence we believe that these are some of the most informative features that may provide interesting insight to the analyst to further investigate the system.

Result	*C4.5 Avg TPR*	*C4.5 Avg FPR*
Macro level	0.9990	0.0001
Micro level	0.6967	0.0014

Table 6. Analysis of common features in macro and micro level

6 Conclusion and Future work

In the present paper we proposed two-level anomaly classification strategy to discern the network data first as normal or attack traffic (macro level) and then classify the anomaly (micro level) on the basis of attacks present in the data. To study the impact of feature selection on the performance of anomaly classification, we used three feature selection methods: *Information Gain, Chi Squared* and *Gain Ratio*. *C4.5* and Naive Bayes classification algorithms were used to classify data at macro and micro level and to construct the anomaly detection classifiers with and without feature selection methods to classify the network anomaly data at macro and micro level and to study the feature relevance. For the proposed anomaly classification strategy, results show that reducing the features does not affect the *TPR* and *FPR* rate much however, time taken to build model goes down drastically. We also did comparison of classifying data with one stage and our proposed two stage classification method and our results shows better detection rate and less time to build model for the proposed strategy. We also did feature relevance analysis and deduce that some of the attributes are worth looking at for classification of data and can assist a network security analyst to develop intrusion prevention strategies. In future, we would like to extend this work by employing other classification algorithms and feature selection methods and to study their best performance results. We also plan to study our proposed two-level strategy in further detail to build a real time network anomaly classification system.

References

[1] KDD Cup'99 Data `http://www.sigkdd.org`.

[2] DARPA'98 Data `http://www.ll.mit.edu/mission/communications/ist/corpora/ideval/data/1998data.html`.

[3] Open Source WEKA Project `http://www.cs.waikato.ac.nz/ml/weka/`.

[4] A.Lazarevic, A.Ozgur, L.Ertoz, J.Srivastava, and V.Kumar. A comparative study of anomaly detection schemes in network intrusion detection. *Proceedings of the SAIM International Conference on Data Mining*, 2003.

[5] C.P.Pleeger. *Security in Computing 2nd Edition*. Prentice Hall, 1997.

[6] Giacinto and G.F.Roli. Intrusion detection in computer networks by multiple classifier systems. *ICPR*, 2002.

[7] H.Liu and R.Setiono. Feature selection and discretization of numeric attributes. *Knowledge and Data Engineering*, 1995.

[8] J.R.Quinlan and R.Rivest. Inferring decision trees using the minimum description length principle. *Information and Computation*, 1989.

[9] K.M.Faroun and A.Boukelif. Neural network learning improvement using k-means clustering algorithm to detect network intrusions. *IJCI*, 2006.

[10] T. Mitchell. *Machine Learning*. McGraw Hill, 1997.

[11] M.Panda and M.R.Patra. Network intrusion detection using naive bayes. *IJCSNS*, 2006.

[12] M.Sabhnani and G.Serpen. Application of machine learning algorithms to kdd intrusion detection dataset within misuse detection context. *Proceedings of the Intelligent Data Analysis*, 2004.

[13] P.G.Neumann and P.A.Porras. Experience with emerald to date. *In First USENIX Workshop on Intrusion Detection and Network Monitoring, Santa Clara*, 1999.

[14] S.J.Russell and P.Norvig. Artificial intelligence: A modern approach. *International Edition, Pearson US Imports and PHIPEs*, 2002.

[15] S.Mukkamala and A.H.Sung. Feature selection for intrusion detection using neural networks and support vector machines. *IEEE Computer Society*, 2003.

[16] S.R.Gaddam, V.V.Phoha, and K.S.Balagani. Means+id3 a novel method for supervised anomaly detection by cascading k-means clustering and id3 decision tree learning methods. *IEEE Transactions on Knowledge and Data Engineering*, 2007.

[17] T.Ganchev, P.Zervas, N.Fakotakis, and G.Kokkinakis. Benchmarking feature selection techniques on the speaker verification task. *CSNDSP*, 2006.