Dynamic Network Traffic Data Classification for Intrusion Detection Using Genetic Algorithm

Rahul Mitra[1], Sahisnu Mazumder[1], Tuhin Sharma[1],
Nandita Sengupta[2], and Jaya Sil[1]

[1] Dept. of Computer Science and Technology,
Bengal Engineering and Science University,
Shibpur, Howrah -711 103, West Bengal, India
[2] University College of Bahrain, Bahrain
{sahisnumazumder,tuhinsharma121,rahulm9999}@gmail.com,
ngupta@ucb.edu.bh, js@cs.becs.ac.in

Abstract. Intrusion Detection System (IDS) classifies network traffic data either ′anomaly′ or ′normal′ to protect computer systems from different types of attacks. In this paper, data mining concepts and genetic algorithm have been applied to classify online traffic data efficiently by developing a rule based lazy classifier. The proposed method updates the rule set dynamically to accommodate the changing pattern in the traffic data in order to attain highest classification accuracy and at the same time maintaining consistency. The classifier is able to detect variants of common network traffic data patterns or modified existing security attacks based on the knowledge gained from its existing training data set with significant classification accuracy.

Keywords: discretization, cut generation, center-spread encoding, nsl-KDDCUP'99 network traffic data set classification accuracy.

1 Introduction

Intrusion Detection System (IDS) [1, 2], a quick evolving domain recently gained its importance in the field of advancement of network security solutions. The current IDSs fail to predict appropriate decision due to problem of 'data overload' resulting classification of normal traffic as malicious (false positive) or vice versa (false negative). Significant advancement has already been done to protect the systems by designing extremely complex security infrastructure including firewalls, identification and authentication systems, access control products, VPNs, encryption products, different antivirus software, scanners and many more. However, yet no system able to handle sufficient and optimum way to the ever changing and evolving innovative security attacks fired intentionally and/or deliberately by a serious attacker in a given network. Moreover, these systems are managed by human beings and so they are highly prone to human error.

Development of an online IDS is a challenging task that would classify network traffic data either *′anomaly′* or *′normal′* by efficiently analyzing the dataset.

B.K. Panigrahi et al. (Eds.): SEMCCO 2012, LNCS 7677, pp. 509–518, 2012.

The proposed system overcomes the limitations of existing classifiers [3, 4]. In the paper sampled training data set has been generated with as much variation as possible using probabilities proportional to size without replacement (PPSWR)[5] to overcome the data overload problem. The objective of the paper is to maximize classification accuracy[3] by building rule based lazy classifier while dealing with the problem of 'false positive' and 'false negative' [2, 3].The proposed classifier can adapt itself with the changing network environment and able to detect network traffic to cope up with variants of common vulnerable security attacks.

2 Proposed Work

The proposed work consists of several procedures to generate the rule set for dynamic classification of the intrusion domain data sets. To develop the classifier a part of the well-known KDDCUP'99 data set [6] has been used for modeling and testing the intrusion detection system[2]. First the data set is discretized by applying cut [7] operation and then intervals are generated considering two successive cut points. As a next step, the intervals are encoded by corresponding centre-spread value [8].Contribution of each interval to classifying the data is evaluated by measuring biasness of an interval to predict the decision class label. Finally, by applying genetic algorithm (GA) [9] best matching rules are learnt, using which maximum classification accuracy is obtained.

2.1 Pre-processing

Network traffic data is preprocessed to obtain discrete attribute values from continuous ones. Using discrete attributes, redundant and less important attributes are removed by applying feature selection algorithm, namely CfsSubsetEval [10] and RankSearch [11] which effectively selects significant attributes and reduces complexity of the system. Instead of all attributes, henceforth only reduced attribute set is used for further processing of data.

2.2 Generation of Intervals

The method generates intervals for each conditional attribute from the set of discrete values as described below.

(i) The set of distinct discrete values for each conditional attribute are sorted in ascending order.
(ii) Each pair of successive values is considered as an interval including the lower value and excluding the upper value.

Example: The set of cut points(S) of a conditional attribute in ascending order is {1, 1.5, 2.5, 4.5, 5.5, 9.5, 10.5, 15.5, 19}

Respective intervals are listed in figure 1.

[1, 1.5)	Interval # 1
[1.5, 2.5)	Interval # 2
[2.5, 4.5)	Interval # 3
[4.5, 5.5)	Interval # 4
[5.5, 9.5)	Interval # 5
[9.5, 10)	Interval # 6
[10.5, 15.5)	Interval # 7
[15.5, 19)	Interval # 8

Fig. 1. Intervals corresponding to the cut-points in set (S)

Intervals	*Centre*	*Spread*
[1, 1.5)	1.25	0.25
[1.5, 2.5)	2	0.5
[4.5, 5.5)	5	0.5
[9.5, 10.5)	10	0.5
[10.5, 15.5)	13	2.5
[16.5, 19)	17.75	1.25

Fig. 2. Encoding of Intervals using Centre-Spread Technique

The next step is to encode these intervals to represent them using discrete value which is used for discretization of the incoming test data.

Centre-Spread encoding technique [8] has been applied in the paper for encoding the intervals. In Centre-Spread encoding method, an interval is denoted as [*lw*,*up*] and encoded as the mid-point of the interval and the span of interval from its mid-point to either side of its end-points. The mid-point of the interval represents discretized value of any continuous value belonging to that interval and the span gives the limit of the continuous values on either side of the mid-point. The intervals of figure 1 are mapped according to the format as given in figure 2.

2.3 Evaluation of Contribution

Once the intervals are generated, as a next step its contribution towards classifying the data is calculated considering training data only.

Contribution of an interval *is calculated using equation* (1) *which refers to a measure based on which one can infer how strongly biased an interval is in predicting a particular decision class.*

$$\boldsymbol{contribution\_interval\_dec\_normal} = \frac{(\boldsymbol{dec_{normal}} \times \mathbf{100})}{(\boldsymbol{dec_{normal}} + \boldsymbol{dec_{anomaly}})} \quad (1)$$

Where dec_{normal}= number of data objects occurring in that interval having decision *normal.*

And dec_{anomaly}= number of data objects occurring in that interval having decision *anomaly*.

Similarly,

$$\boldsymbol{Contribution\_interval\_dec\_anomaly} = \frac{(\boldsymbol{dec\_anomaly} \times \mathbf{100})}{(\boldsymbol{dec\_normal} + \boldsymbol{dec\_anomaly})} \quad (2)$$

Deviation of contribution of a concerned interval

$$= \begin{cases} \text{(Contribution of the interval in predicting decision as "normal" - Contribution of the interval in predicting decision as "anomaly")} & \text{; when Contribution of the interval predicting decision as "normal" is greater than or equals to the contribution of the interval predicting decision as "anomaly".} \\ -\text{(Contribution of the interval in predicting decision as "anomaly")} & \text{; Otherwise} \end{cases} \quad (3)$$

Therefore, for predicting *normal* class label, the deviation is positive while for *anomaly* it is negative with an assumption that the population of data objects having decision *normal* is more than the decision as *anomaly*. So if two intervals having same contribution value while one predicting decision as *normal* and other predicting decision as *anomaly*, then the interval predicting decision as "*anomaly*" have a greater effect. This deviation of contribution of a concerned interval is nothing but the relative measure of strength of that interval on prediction of the decision value as *normal* or *anomaly*. More the value positive, more biased the interval is towards predicting decision as *normal*. Similarly, more the negative value of this measure, more is the interval biased towards predicting decision as *anomaly*.

2.4 Generation of Initial Rule Set

Once the intervals of all conditional attributes in the training dataset and their corresponding contribution towards the prediction of the decision of a particular class are generated, the next task is to generate the initial rule set of the classifier. In the subsequent steps GA [9] is applied to adapt the new test data set.

Algorithm. *Initial Rule Set Generation*

Step 1: The conditional attribute (A_{max}) with maximum number of intervals is obtained.

Step 2: For each interval (*I*) in A_{max} create a rule with
(i) Attribute A_{max} consists of interval *I*.
(ii)Rest of the attributes consist interval having closest contribution deviation value to *I*.
(iii) The intervals which are selected removed from the corresponding attributes. If an attribute contains no more intervals, then all the intervals are added back.

Step 3: Repeat Step 2 until A_{max} does not contain any more intervals.

2.5 Generation of the Best Matching Rule

For each incoming test data, the rule which matches with the maximum number of attribute values is obtained using the concepts of genetic algorithm.

Algorithm. *best matched rule generation*

Step 1: Copy the original rule set into a temporary rule set (T).
Step 2: Select a rule (R1) from T having maximum number of attribute matching with that of the test data set.
Step 3: Remove R1 from T.
Step 4: Select a rule (R2) at random from T.
Step 5: Create a new rule with attributes of R1 and R2, matches with that of the incoming data.
Step 6: Assign the newly created rule as R1. Remove R2 from T.
Step 7: Repeat Step 4 to Step 6 until T is empty.

The resultant rule has maximum number of attribute value, matching with that of the test data.

An instance rule set is shown in figure 3.

Rule no.	Attr-1	Attr-2	Attr-3	Attr-4	Attr-5	Attr-6	Attr-7	Attr-8	Attr-9	Attr-10
1	1.5	2	2	3.5	0.5	1	2	0	2	3.5
2	1.5	0	3	4	1.5	0.5	3	0.5	3	2.5
3	1	0.5	2.5	3	2	0.5	4	1.5	2.5	2
4	0	2	3	3.5	1.5	1	0	0.5	2.5	3.5
5	2	0.5	3.5	2.5	1.5	0	1	0.5	1	3

Fig. 3. Sample population of the Initial Rule set

The incoming test data is given in figure 4.

Attr-1	Attr-2	Attr-3	Attr-4	Attr-5	Attr-6	Attr-7	Attr-8	Attr-9	Attr-10
1.5	0	2	3	2	0.5	0	0	1.5	3.5

Fig. 4. Instance of Incoming Test Data

After applying the *best matched rule generation* algorithm we get the best matching rule, given in figure 5.

Attr-1	Attr-2	Attr-3	Attr-4	Attr-5	Attr-6	Attr-7	Attr-8	Attr-9	Attr-10
1.5	0	2	3	2	0.5	0	0	3	3.5

Fig. 5. Best Matching Rule

2.6 Predicting Decision

For predicting decision of an incoming test data, the proposed classifier employs the decision making procedure as described below.

Step 1: For each attribute in the *reduct* set, the interval which matches with the test data, scale the absolute value of contribution deviation of the interval **(*abs*)** from (0 ~ 100) to (0 ~ 20).

Step 2: The effect of this interval towards prediction of the decision is calculated using equation (4).

$$\boldsymbol{effect(e)} = \mathbf{2}^{(\boldsymbol{abs})} \tag{4}$$

Step 3: The prediction is made based on the net effect of all the attributes matching the test data, calculated using equation (5).

$$\begin{aligned} &\boldsymbol{Net\ Effect} = \\ &\boldsymbol{Net\ Effect} + \boldsymbol{e};\ \textit{if contribution deviation of the interval is positive} \\ &\boldsymbol{Net\ Effect} = \\ &\boldsymbol{Net\ Effect} - \boldsymbol{e};\ \textit{if contribution deviation of the interval is negative} \end{aligned} \tag{5}$$

Step 4: If Net Effect is positive or zero then the decision predicted is *normal* else *anomaly*.

Note: Here an interval having modulus of contribution deviation close to 100 have an effect which is exponentially greater than those having a value close to 0.

3 Experimental Results and Comparisons

3.1 Comparisons with Original KDD Data Set

Comparison with respect to classification accuracy of the common existing classifiers of different categories with the proposed classifier is demonstrated here. We have taken 10000 data objects from NSL-KDD Data set [12, 13] for network-based intrusion detection and used 10 fold cross validation technique [3] for measuring classification accuracy [3] of the different classifiers. We have used the modules provided by the WEKA tool [14, 15] to run different classifiers. The accuracy of the proposed classifier is **94.9**% on an average, as shown in figure 6.

The proposed classifier is primarily a predictive model and falls under lazy classifiers, comparison with different lazy and probability based classifiers are shown in figure 7.

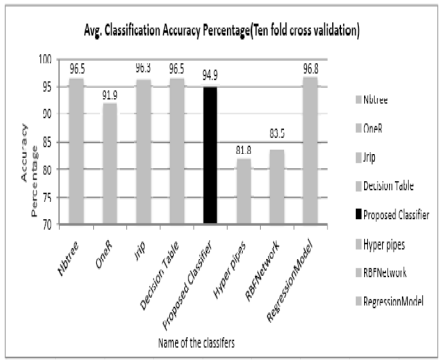

Fig. 6. Comparison of the Proposed Classifier with different types of well-known Classifiers

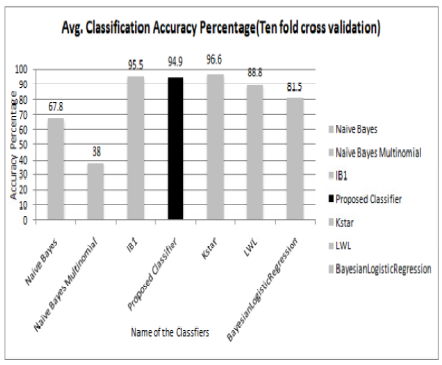

Fig. 7. Comparisons of the proposed classifier with lazy and probabilistic classifiers

By analyzing figure 6 and 7, one can observe that the proposed classifier provides slightly less classification accuracy from that of the well-known classifiers (like Nbtree, Jrip, Decision table and Regression Model as shown in figure 6), when applied on original NSL-KDD dataset (considering both training and test data set taken from the original one). This is due to the fact that we have not considered the concept of correlation between different conditional attributes in building the classifier while all of the well-known classifiers (giving better performance than our one) consider the correlation between different attributes during learning. However, we have tried to ensure that the proposed classifier gives better classification accuracy when it runs on a test dataset which is a variant of the original one and has less correlation.

3.2 Comparisons Using Evolutionary Test Dataset

The classification accuracy of different classifiers are obtained considering data set created using evolutionary method. To generate the data set concepts of crossover and mutation operations of genetic algorithm are applied. This new data set is a variant of the original network traffic data set and represents modified forms of intrusions or attacks that can be encountered by the IDS.

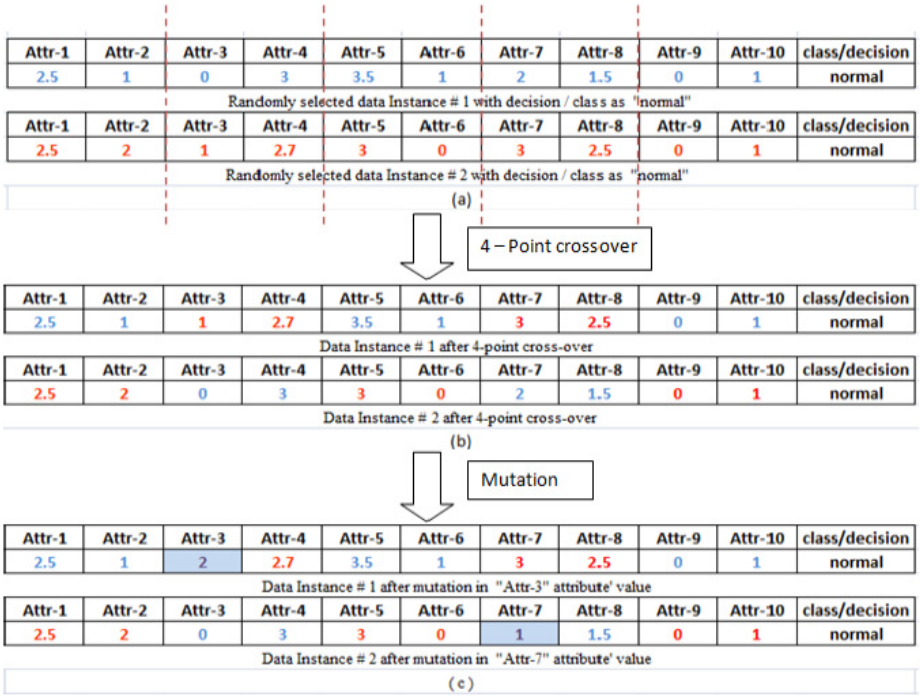

Fig. 8. Generation of evolutionary data instances

For creation of each data instance in this new data set, two data instances are sampled from the original data set having same class label(figure 8(a)) and performed 4 point cross over to generate the new instances, shown in figure 8(b). This new instance has the same class label as that of its parent chromosome. After performing crossover, mutation is applied by replacing the value of any randomly chosen conditional attribute by a value which lies in the domain of that conditional attribute, represented in figure 8(c).

The new data set maintains the ratio of number of *normal* instances to the number of *anomaly* instances same as that of the original data set. Here the correlation between different attributes in newly generated evolutionary data set has been shown in the figure 9(b),different from the correlation values in the original data set as shown in the figure 9(a). In figure9, we have shown top 4 attributes of rank as determined by rank search in both (a) and (b).

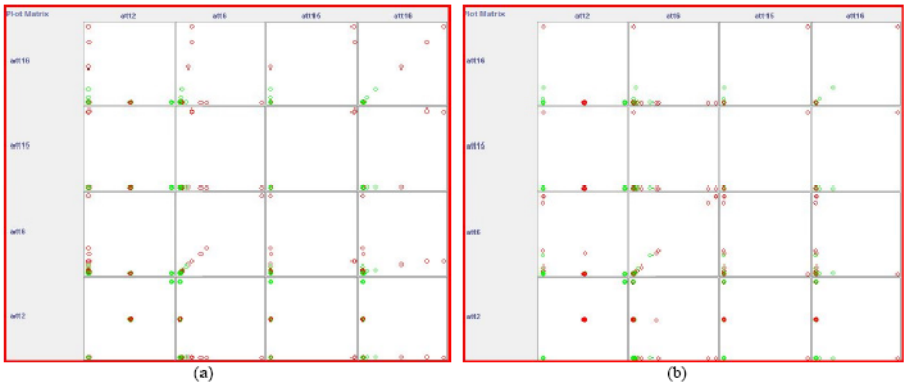

Fig. 9. (a) Correlation between different conditional attributes in original KDD data set; (b) Correlation between different conditional attributes in evolutionary (test) data set obtained from original NSL-KDD data set by crossover and mutation

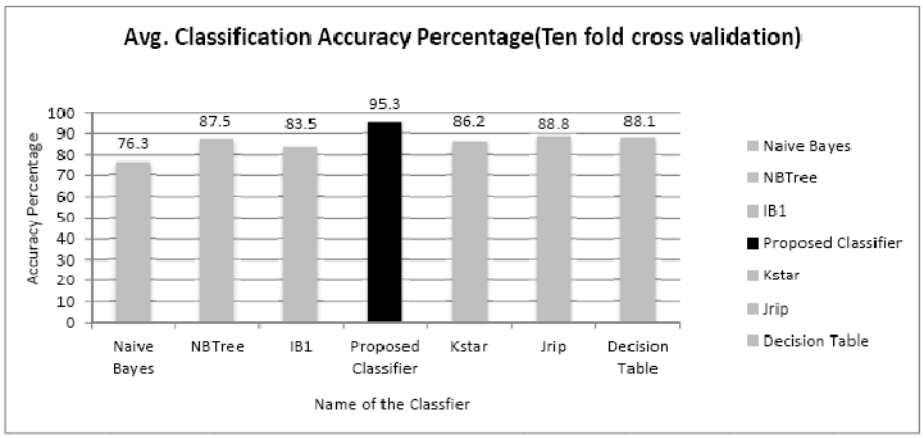

Fig. 10. Comparisons while detecting variants of common network traffic data patterns

Experiments with new test data set and comparing classification accuracies of different well-known efficient classifiers and our method, following results are obtained, shown in the figure10. It has been concluded that the proposed classifier outperforms all others giving the best average classification accuracy as **95.3**% while detecting variants of common network traffic data patterns.

4 Conclusions

In this paper, we have built a classifier which does not take correlation into account while predicting decisions of the incoming test data. The proposed classifier outperforms all the classifiers while detecting test data which are variants of the original data set representing modified form of common network security attacks. Since these test data patterns have less correlation than the original training data set, rule based and decision tree based classifiers gives lower classification accuracy compared to that of our proposed classifier obtained by 10 fold cross validation. Moreover, the proposed classifier shows impressive performance while detecting normal behavioral traffic data patterns compared to other efficient existing classifiers as discussed in the previous section. Therefore, our proposed classifier can efficiently meet the needs for designing an up-to-date intrusion detection system with providing solutions to the limitations of existing ones operating in modern dynamic network environment.

References

1. SANS Institute InfoSec Reading Room site: Understanding Intrusion Detection Systems
2. Scarfone, K., Mell, P.: Guide to Intrusion Detection and Prevention Systems (IDPS), Recommendations of the National Institute of Standards and Technology
3. Han, Kamber, M.: Data Mining: Concepts And Techniques. MorganKaufmann Publishers, San Francisco (2001)
4. Wu, X., Kumar, V., Ross Quinlan, J., Ghosh, J., Yang, Q., Motoda, H., McLachlan, G.J., Ng, A., Liu, B., Yu, P.S., Zhou, Z.-H., Steinbach, M., Hand, D.J., Steinberg, D.: Top 10 algorithms in data mining. Knowl. Inf. Syst. 14, 1–37 (2008), doi:10.1007/s10115-007-0114-2
5. Rosen, B.: On sampling with probability proportional to size. Journal of Statistical Planning and Inference 62, 159–191 (1997)
6. KDD Cup (1999), `http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html`
7. Komorowski, J., Polkowski, L., Skowron, A.: Rough Set: A tutorial
8. Wyatt, D.: Applying the XCS Learning Classifier System to Continuous-Valued Data-mining Problems, Technical Report UWELCSG05-001 (September 2004)
9. Goldberg, D.E.: Genetic algorithms in search, optimization and machine earning. Addison-Wesley, Reading (1989)
10. Selvakuberan, K., Indradevi, M., Rajaram, R.: Combined Feature Selection and classification–A novel approach for thecategorization of web pages
11. Data Mining Algorithm In R/Dimensionality Reduction/Feature Selection, `http://www.wikipedia.org/`

12. Nsl-kdd data set for network-based intrusion detection systems (March 2009), http://nsl.cs.unb.ca/NSL-KDD/
13. Tavallaee, M., Bagheri, E., Lu, W., Ghorbani, A.A.: A Detailed Analysis of the KDD CUP 99 Data Set
14. Weka 3: Data Mining Software in Java, http://www.cs.waikato.ac.nz/ml/weka/
15. Weka User Manual, http://www.gtbit.org/downloads/dwdmsem6/dwdmsem61man.pdf, http://kent.dl.sourceforge.net/project/weka/documentation/3.6.x/WekaManual-3-6-2.pdf