Feature Selection for Anomaly-Based Intrusion Detection System Using Information Gain and Mutual Correlation

Thuzar Hlaing, May Aye Khine
University of Computer Studies, Yangon
thuzarhlaing85@gmail.com, maya.khine@gmail.com

Abstract

To avoid high computational costs in identifying intrusions by IDSs, the size of a dataset needs to be reduced. Feature selection is considered a problem of global combinatorial optimization in machine learning, which reduces the number of features, removes irrelevant, noisy and redundant data, and results in acceptable classification accuracy. This paper proposes a combine filter method by using IG (information gain) and Mutual Correlation for feature selection in NSL-KDD dataset. IG was used to select important feature subsets from all features in the NSL-KDD dataset. The resulted features set are combined with Mutual correlation to get the optimal reduced features set. Tests are done on NSL-KDD dataset which is improved version of KDD-99 dataset. The results show that the number of selected features is reduced from 41 to 14 and correlated 10 features. The proposed method not only reduces the number of the input features and memory and CPU time but also increases the classification accuracy.

Keywords: Information Gain, NSL-KDD, Feature Selection

1. Introduction

Intrusion detection system (IDS) is known as a critical technology to help protection. Network intrusion detection system (NIDS) performs packet logging, real-time traffic analysis of IP network, and tries to discover if an intruder is attempting to break into the system [10]. Intrusion detection (ID) is a major research problem in network security, where the concept of ID was proposed by Anderson in 1980[9]. The goal of intrusion detection systems (IDS) is to identify unusual access or attacks to secure internal networks [4].

In general, IDSs can be divided into two techniques: misuse detection and anomaly detection [5, 15]. Misuse detection refers to detection of intrusions that follow well-defined intrusion patterns. It is very useful in detection known attack patterns. Anomaly detection refers to detection performed by detecting changes in the patterns of utilization or behavior of the system. It can be used to detect known and unknown attack. The anomaly detection techniques have the advantage of detecting unknown attacks over the misuse detection technique [6]. Anomaly based intrusion detection using data mining algorithms such as decision tree (DT), naïve Bayesian classifier (NB), neural network (NN), support vector machine (SVM), k-nearest neighbors (KNN), fuzzy logic model, and genetic algorithm have been widely used by researchers to improve the performance of IDS [1][13].

The goal of feature selection is to find a feature subset maximizing some performance criterion, such as accuracy of classification. Not only that, selecting important features from input data lead to a simplification of the problem, faster and more accurate detection rates. Thus selecting important features is an important issue in intrusion detection [3].

The rest of the paper is organized as follows: Section 2 reviews related work. Section 3 introduces about the NSL-KDD Dataset. Section 4 describes calculation of Information Gain based on continuous values, data normalization and Mutual Correlation. Section 5 provides a detailed description of proposed

system and the experimental results in section 6. Finally, the paper is concluded with section 7.

2. Related Work

In recent times, intrusion detection has received a lot of interest among the researchers since it is widely applied for preserving the security within a network. Here, some of the techniques used for intrusion detection.

Huang, Pei and Goodman [7], where the general problem of GA optimized feature selection and extraction is addressed. In their paper, Huang, et al. applies a GA to optimize the feature weights of a KNN classifier and choose optimal subset of features for a Bayesian classifier and a linear regression classifier. Experiments in their paper show that the performance of all these three classifiers with feature weighing or selection by a GA is better than that of the same classifiers without a GA. They conclude that performance gain is completely dependent on what kind of classifier is used over what type of data set.

B. Shanmugam and Norbik Bashah Idris [2] have proposed an advanced fuzzy and data mining methods based hybrid model to find out both misuse and anomaly attacks. Their objective was to decrease the quantity of data kept for processing and also to improve the detection rate of the existing IDS using attribute selection process and data mining technique respectively. A modified version of APRIORI algorithm which is an improved fuzzy data mining algorithm utilized for implementing fuzzy rules has enabled the generation of if-then rules that show common ways of expressing security attacks. The DARPA 1999 data set has been used to test and benchmark the efficiency of the proposed model.

Srinivas and Sung [14] presented the use of support vector machine (SVM) to rank these extracted features, but this method needs many iterations and is very time-consuming. In the research of detection model generation, it is desirable that the detection model be explainable and have high detection rate, but the existing methods cannot achieve these two goals.

3. Introduction of NSL-KDD Dataset

KDDCUP'99 is the mostly widely used data set for the anomaly detection. But researchers conducted a statistical Analysis on this data set and found two important issues which highly affect the performance of evaluated systems, and results in a very poor evaluation of anomaly detection approaches. To solve these issues, they have proposed a new data set, NSL-KDD [12], which consists of selected records of the complete KDD data set.

The data set has 41 attributes for each connection record plus one class label. The data set contains 22 attack types. All these attacks fall into four main categories.

1. **Denial of Service (DOS**): In this type of attacks an attacker makes some computing or memory resources too busy or too full to handle legitimate requests, or denies legitimate users access to a machine.
2. **Remote to User (R2L):** In this type of attacks an attacker who does not have an account on a remote machine sends packets to that machine over a network and exploits some vulnerability to gain local access as a user of that machine.
3. **User to Root (U2R):** In this type of attacks an attacker starts out with access to a normal user account on the system and is able to exploit vulnerability to gain root access to the system.
4. **Probing:** In this type of attacks an attacker scans a network of computers to gather information or find known vulnerabilities. An attacker with a map of machines and services that available on a network can use this information to look for exploits.

Table 1 illustrates a number of attacks falling into four major categories and Table 2 describes the list of the input attributes in data set for each network connections.

Table 1. Different Types of attacks in NSL-KDD Dataset

Denial of Service Attacks	Back,land,Neptune,pod,smurf,teardrop
User to Root Attacks	Buffer_overflow, loadmodule, perl,rootkit
Remote to Local Attacks	Ftp_write, guess_passwd, imap, multihop, phf,spy,warezclient,warezmaster
Probing	Satan, ipsweep, nmap, portsweep

Table 2. Input attributes in NSL-KDD Dataset

No	Input Attribute	Type	No	Input Attribute	Type
1	duration	Con.	22	is_guest_login	Dis.
2	protocol_type	Dis.	23	count	Con.
3	service	Dis.	24	srv_count	Con.
4	flag	Dis.	25	serror_rate	Con.
5	src_bytes	Con.	26	srv_serror_rate	Con.
6	dst_bytes	Con.	27	rerror_rate	Con.
7	land	Dis.	28	srv_rerror_rate	Con.
8	wrong_fragment	Con.	29	same_srv_rate	Con.
9	urgent	Con.	30	diff_srv_rate	Con.
10	hot	Con.	31	srv_diff_host_rate	Con.
11	num_failed_logins	Con.	32	dst_host_count	Con.
12	logged_in	Dis.	33	dst_host_srv_count	Con.
13	num_compromised	Con.	34	dst_host_same_srv_rate	Con.
14	root_shell	Con.	35	dst_host_diff_srv_rate	Con.
15	su_attempted	Con.	36	dst_host_same_src_port_rate	Con.
16	num_root	Con.	37	dst_host_srv_diff_host_rate	Con.
17	num_file_creations	Con.	38	dst_host_serror_rate	Con.
18	num_shells	Con.	39	dst_host_srv_serror_rate	Con.
19	num_access_files	Con.	40	dst_host_rerror_rate	Con.
20	num_outbound_cmds	Con.	41	dst_host_srv_rerror_rate	Con.
21	is_hot_login	Dis.	-	-	-

4. Feature Selection Methods

The increase of data size in terms of number of instances and number of features becomes a great challenge for the feature selection algorithms.

4.1 Information Gain based on continuous value

Information gain (IG) is a feature ranking method based on decision trees that exhibits good classification performance. Information gain used in feature selection constitutes a filter approach. Filter approaches select features using characteristics of individual features. Advantages of the filter-based techniques are that they can easily scale up to high-dimensional datasets and that they are computationally fast and independent of the learning algorithm. Information gain is a measure based on entropy. Entropy is one of the most commonly used discretization measures.

Let D be the set of n instances and *C* be the set of m classes. $P(C_i, D)$ represents the fraction of the example in D that has class C_i. Then, the expected information from this class membership is given by [8]:

$$Info(D) = -\sum_{i=1}^{m} P(C_i, D)\, log P(C_i, D) \qquad (1)$$

For continuous-valued attribute for A, the best split-point for A must be determined, where the split-point is a threshold on A. Firstly, the values of A in increasing order must be sorted. Typically, the midpoint between each pair of adjacent values is considered as a possible split-point. Therefore, given v values of A, then v-1 possible splits are evaluated. The midpoint between the values a_i and a_{i+1} of A is:

$$\frac{a_i + a_{i+1}}{2} \qquad (2)$$

For each possible split-point for A, $Info_A(D)$ must be calculated, where the number of partitions is two, that is v=2 (or j=1, 2). The point with the minimum expected information requirement for A is selected as the split-point for A. D_1 is the set of tuples in *D* satisfying A ≤ split-point and D_2 is the set of tuples in *D* satisfying A > split-point. It is given by:

$$Info_A(D) = \frac{|D_1|}{|D|} Entropy(D_1) + \frac{|D_2|}{|D|} Entropy(D_2) \qquad (3)$$

The entropy function for a given set is calculated based on the class distribution of the tuples in the set. For example, given m classes, C_1, C_2… C_m, the entropy of D_1 is:

$$Entropy(D_1) = -\sum_{i=1}^{m} P_i log_2 (P_i) \qquad (4)$$

Where P_i is the probability of class C_i in D_1, determined by dividing the number of tuples of class C_i in D_1 by $|D_1|$, the total number of tuples in D_1.Therefore,when selecting a split-

point for attribute A, we want to pick the attribute value that gives the minimum expected information requirement (i.e.,min($Info_A(D)$)). This would result in the minimum amount of expected information (still) required to perfectly classify the tuples after partitioning by A≤ split point and A>split point. The value of Entropy (D_2) can be computed similarly as in Equation (4).

Then, the difference between $Info(D)$ and $Info_A(D)$ provides the information gained by partitioning S according to the test A:

$$Gain(A) = Info(D) - Info_A(D) \quad (5)$$

4.2 Data Normalization

Data normalization is an essential step of data preprocessing for most anomaly detection algorithms that learns the statistical characters of attributes extracted from the audit data.

Min-max normalization performs a transformation on the original data. Suppose that min_A and max_A are the minimum and maximum values of an attribute, A. Min-max normalization maps a value, v, of A to v□ in the range [new min_A, new max_A] by computing:

$$v' = \frac{v - \min_A}{\max_A - \min_A}(new\_\max_A - new\_\min_A) + new\_\min_A \quad (6)$$

Min-max normalization preserves the relationships among the original data values.

4.3 Feature Selection based on Mutual Correlation

Correlation is a well known similarity measure between two random variables. If two random variables are linearly dependent, then their correlation coefficient is close to ±1. If the variables are uncorrelated the correlation coefficient is 0. The correlation coefficient is invariant to scaling and translation. Hence two features with different variances may have same value of this measure. The p-dimensional feature vectors X_i of N number of instances is given by:

$$X_i = [\ {}^{i}x_1, \ldots, {}^{i}x_p] \qquad i=1,\ldots.., N$$

The mutual correlation for a feature pair x_i and x_j is defined as

$$r_{x_i, x_j} = \frac{\sum_k {}^{k}x_i\, {}^{k}x_j - N\,\bar{x_i}\bar{x_j}}{\sqrt{(\sum_k {}^{k}x_i^2 - N\bar{x_i}^{-2})(\sum_k {}^{k}x_j^2 - N\bar{x_i}^{-2})}} \quad (1)$$

Where k=1,….., N

If two features x_i and x_j are independent then they are also uncorrelated, i.e. r_{x_i, x_j}=0. Let us evaluate all mutual correlations for all feature pairs and compute the average absolute mutual correlation of a feature over δ features

$$r_{j,\delta} = \frac{1}{\delta}\sum_{i=1, i\neq j}^{\delta} \left| r_{x_i, x_j} \right| \quad (2)$$

The feature which has the largest average mutual correlation

$$\alpha = \arg\max_j r_{j,\delta} \quad (3)$$

will be removed during each iteration of the feature selection algorithm. When feature x_α is removed from the feature set, it is also discarded from the remaining average correlation, i.e.

$$r_{j,\delta-1} = \frac{\delta r_{j,\delta} - |r_{x_\alpha, x_j}|}{\delta - 1} \quad (4)$$

Algorithm 1: Feature Selection based on mutual correlation

Input: Original features set X of size N x p
Output: Reduced feature set of size N x D (D<<p)
Method:

1. Initialize δ =p
2. Discard features Xα for α determined by Equation (3)
3. Decrement δ= δ-1, if δ<D return the Resulting D dimensional feature set and stop otherwise.
4. Recalculate the average correlations by using Equation (4).
5. Go to step 2.

5. Proposed Framework

In the first level, the proposed approach is applied on p original feature to obtain the reduced D features using information gain value. In the next level algorithm 1 is applied on the reduced using the concept of mutual correlation to get the further reduced feature set.

Algorithm

Input: Original Dataset X of size N x p, class label
Output: Reduced feature set of size N x D (D<<p)
Method:
Level1: Compute the information gain (IG) value of 34 features, define threshold value and select high IG values under threshold value to obtain the reduced set size N x f.
Level 2: Apply Algorithm 1 on the reduced feature set of size N x f obtained from Level 1 to get the further reduced feature set of size N x D.

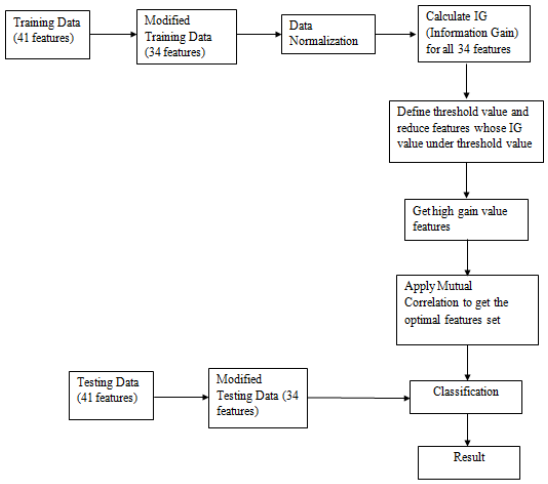

Figure.1. Overview of the Proposed Framework

The detailed analysis of NSL- KDD data is given in section 3. Based on the analysis, the NSL-KDD data contains four types of attacks and normal behavior data with 41 attributes that have both continuous and discrete attributes. The proposed system is designed only for the continuous attributes because the major attributes in NSL-KDD data are continuous in nature. Therefore, the proposed system have taken only the continuous attributes for instance, 34 attributes from the input dataset by removing discrete attributes. Then, the system calculated the information gain value (IG values) for 34 attributes. A feature with a higher information gain value indicates higher discrimination of this feature useful information for classification.

Table 3. Information Gain values of 34 Attributes

No.	Attributes	IG (Information Gain) values
A1	duration	0.004225
A2	src_bytes	0.000242
A3	dst_bytes	0.000026
A4	wrong_fragment	0.170263
A5	urgent	0.000007
A6	hot	0.001867
A7	num_failed_logins	0.000060
A8	num_compromised	0.001833
A9	root_shell	0.000324
A10	su_attempted	0.000514
A11	num_root	0.003716
A12	num_file_creations	0.000918
A13	num_shells	0.000104
A14	num_access_files	0.002154
A15	num_outbound_cmds	0.000000
A16	count	0.331054
A17	srv_count	0.018277
A18	serror_rate	0.371727
A19	srv_serror_rate	0.373764
A20	rerror_rate	0.150893
A21	srv_rerror_rate	0.049753
A22	same_srv_rate	0.009607
A23	diff_srv_rate	0.007523
A24	srv_diff_host_rate	0.091057
A25	dst_host_count	0.150730
A26	dst_host_srv_count	0.398228
A27	dst_host_same_srv_rate	0.395404
A28	dst_host_diff_srv_rate	0.339649
A29	dst_host_same_src_port_rate	0.098760
A30	dst_host_srv_diff_host_rate	0.150247
A31	dst_host_serror_rate	0.377187
A32	dst_host_srv_serror_rate	0.386163
A33	dst_host_rerror_rate	0.048581
A34	dst_host_srv_rerror_rate	0.173213

After calculating the information gain values for all features, a threshold for the results was established. Since the results show that most IG values are nearly zero after the computation process, not many features have an influence on the category in a data set, signifying that these features are irrelevant for classification. In this paper, the proposed system was defined 0.09 threshold value over the 34 attributes by experimenting. If the information gain value of the feature was higher than the threshold, the feature was selected; if not, the feature was not selected. After the application of IG, the 14 values that were above this threshold value (A4,

A16, A18, A19, A20, A25, A26, A27, A28, A29, A30, A31, A32, and A34) were used.

The correlated 10 features are resulted from the remaining 14 features by calculating with Mutual Correlation as shown in Table 4.

Table 4. Number of correlated Features by Mutual Correlation

No.	Correlated Attributes
A4	wrong_fragment
A16	count
A18	serror_rate
A20	rerror_rate
A25	dst_host_count
A26	dst_host_srv_count
A28	dst_host_diff_srv_rate
A29	dst_host_same_src_port_rate
A30	dst_host_srv_diff_host_rate
A34	dst_host_srv_rerror_rate

6. Experimental Result

NSL-KDD dataset [13] contains 125973 records in train set and 22544 records in the test set. Each connection record contains 41 features and two classes are labeled as either normal or an attack. Most of the existing IDS use all 41 features in the network to evaluate and look for intrusive pattern some of these feature are redundant and irrelevant. The drawback of this approach is time-consuming detection process and degrading the performance of IDS system. Firstly, training and testing is applied on all 41 features, modify 34 features and 14 features and correlated 10 features. After that training and testing is done with reduced features and the results of classification is calculated and shown in Table 5.

All experiments were performed using a 2.20GHZ Dual-Core Processor and 2GB of RAM running windows7.The proposed system used WEKA (Waikato Environment for Knowledge Analysis) software [16] for Naïve Bayesian. The various results of this experiment are given below.

Table 5. Performance of Proposed System using Naïve Bayesian Classifier

Feature	Classification Rate (%)	Time (secs)
41 features	90.4178 %	10.02 secs
Modified 34 features	90.1209 %	9.11 secs
After features Reduction (14 features) by IG	91.5569 %	4.38 secs
10 correlated features by Mutual Correlation	91.2267%	3.29 secs

I

In this paper, the NSL-KDD dataset is divided into four section datasets and the selected features and experimental results of each section are shown in Table 6. After reducing to 14 features and correlated 10 features, the accuracy and time is more efficient than that of both 41 features and 34 features. Each dataset contains nearly 20550 connections in NSL-KDD dataset.

Table 6. Performance of Four Sections NSL_KDD Dataset using Naïve Bayesian

Features	41 Features		Modified 34 Features		After features Reduction (14 features) by IG		10 correlated features by Mutual Correlation	
Accuracy	Classification Rate (%)	Time (secs)	Classification Rate (%)	Time (secs)	Classification Rate (%)	Time (secs)	Classification Rate (%)	Time (secs)
NSL_KDD (1)	89.4349	1.26	88.2654	0.94	91.3415	0.43	91.1744	0.31
NSL_KDD (2)	89.5318	1.56	88.7308	1.11	91.5717	0.39	91.3379	0.28
NSL_KDD (3)	89.475	1.25	88.7376	1.04	91.3627	0.43	90.9448	0.36
NSL_KDD (4)	90.4251	1.29	90.002	1.12	90.8925	0.4	90.7025	0.28

7. Conclusion

Feature subset selection is an important problem in knowledge discovery, not only for the insight gained from determining relevant modeling variables, but also for the improved scalability, understandability ,and possibly accuracy of the resulting models. The proposed system used NSL-KDD dataset which is a new

dataset for the evaluation of researches in network intrusion detection system. First, the proposed system reduces from all 41 features to 14 features using information gain over the modified 34 features which are continuous values in NSL-KDD dataset. Second, by using mutual correlation method, that is removed uncorrelated features. In this way, our proposed system can greatly reduce the redundant, least important features and computational cost in the process of intrusion detection.

References

[1] Barbara, Daniel, Couto, Julia, Jajodia, Sushil, Popyack, Leonard, Wu, and Ningning, "ADAM: Detecting intrusion by data mining," IEEE Workshop on Information Assurance and Security, West Point, New York, June 5-6, 2001.

[2] B. Shanmugam, N. Bashah Idris, "Improved Intrusion Detection System Using Fuzzy Logic for Detecting Anamoly and Misuse Type of Attacks", in Proceedings of the International Conference of Soft Computing and Pattern Recognition, pp: 212-217, 2009.

[3] B. J. Kim and I. K. Kim, "Machine Learning Approach to Real time Intrusion Detection System." In Proceedings of 18th. Australian Join Conference
on Artificial Intelligence, 2005, Sydney, Australia. Vol. 3809. Pp. 153-163.

[4] C. Tsai, Y. Hsu, C. Lin and W. Lin,"Intrusion detection by machine learning: A review", Expert Systems with Applications, vol.36, pp.11994-12000, 2009.

[5] E. Biermann, E. CloeteandL.M.Venter,"A comparison of intrusion detection Systems", Computer and Security, vol.20, pp.676-683, 2001.

[6] E. Lundin and E. Jonsson, "Anomaly-based intrusion detection: privacy concerns and other problems", Computer Networks, vol.34, pp.623-640, 2002.

[7] Huang, Z., Pei, M., Goodman, E., Huang, Y., and Li, G.Genetic algorithm optimized feature transformation: a comparison with different classifiers. In Proc. GECCO 2003, pp. 2121-2133.

[8] Jiawei Han and Micheline Kamber, "Data Mining Concepts and techniques", 2nd ed., Morgan Kaufmann Publishers, San Francisco, CA, 2007.

[9] J.P. Anderson, "Computer security threat monitoring and surveillance", Technical Report, James P. Anderson Co., Fort Washington, PA, April1980.

[10] Krasser, S., Grizzard, J., Owen, H., and Levine. J., "The use of honey nets to increase computer network security and user awareness," Journal of Security Education, vol. 1, 2005, pp. 23-37.

[11] Mahbod Tavallaee, Ebrahim Bagheri, Wei Lu, and Ali, A. Ghorbani, " A Detailed Analysis of the KDD CUP 99 Data Set", proceeding of IEEE symposium on computational Intelligence in security and defence application, 2009.

[12]NSL-KDD dataset: http://nsl.cs.unb.ca/NSL-KDD

[13] Shon T., Seo J., and Moon J., "SVM approach with a genetic algorithm for network intrusion detection," In Proc. of 20th International Symposium on Computer and Information Sciences (ISCIS 2005), Berlin: Springer-Verlag, 2005, pp. 224-233.

[14] Srinivas, M., Sung, A., "Feature Ranking and Selection for Intrusion Detection". Proceedings of the International Conference on Information and Knowledge Engineering, 2002.

[15] T.Verwoerd and R.Hunt,"Intrusion detection techniques and approaches", Computer Communications, vol.25, pp.1356-1365, 2002.

[16]WEKA: Software machine learning, the University of Waikato, Hamilton, New-Zealand.