IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661, p- ISSN: 2278-8727Volume 15, Issue 5 (Nov. - Dec. 2013), PP 107-112
www.iosrjournals.org

Intrusion Detection System Based on K-Star Classifier and Feature Set Reduction

Deeman Y. Mahmood[1], Dr. Mohammed A. Hussein[2]
[1]*(Department of Computer Science, College of science/ University of Sulaimani, Iraq)*
[2]*(Department of Electrical Engineering, College of Engineering/ University of Sulaimani, Iraq)*

Abstract: *Network security and Intrusion Detection Systems (IDS's) is an important security related research area. This paper applies K-star algorithm with filtering analysis in order to build a network intrusion detection system. For our experimental analysis and as a case study, we have used the new NSL-KDD dataset, which is a modified dataset for KDDCup 1999 intrusion detection benchmark dataset. With a split of 66.0% for the training set and the remainder for the testing set a 2 class classifications has been implemented. WEKA which is a java based open source software consists of a collection of machine learning algorithms for Data mining tasks has been used in the testing process. The experimental results show that the proposed approach is very accurate with low false positive rate and high true positive rate and it takes less learning time in comparison with other existing approaches used for efficient network intrusion detection.*

Keywords: *Information Gain, Intrusion Detection System, Instance-based classifier, K-Star, Weka.*

I. INTRODUCTION

During recent years, number of attacks on network has dramatically increased and consequently interest in network intrusion detection has increased among researchers [1]. Intrusion detection systems (IDS) are becoming a very important case of today's network security architectures, where it analyses the network traffic and looks for intrusive activities. Traditionally, intrusion detection techniques come into two categories: Signature detection and anomaly detection [2,3]. Signature or misuse detection searches for well-known patterns of attacks, this system can only detect an attack if there an accurate matching behaviour found against already stored patterns, known as signatures. While anomaly detection is based on establishing a normal activity profile for a system, this technique evolves itself by understanding and gathering the information about the system and determines the behaviour of the system based on it [3]. There are several types of intrusion detection systems and the choice of which one to use depends on the overall risks to the organization and the resources available. There are two primary types of IDS: host-based (HIDS) and network-based (NIDS), HIDS resides on a particular host and looks for indications of attacks on that host while NIDS resides on a separate system that watches network traffic, looking for indications of attacks that traverse that portion of the network [4]. This work aims to design enhanced IDS. The main issue in standard classification problems lies in minimizing the probability of error while performing the classification decision. Hence, the key point is how to choose an effective classification approach to build accurate intrusion detection systems in terms of high detection rate while keeping a low false alarm rate. The proposed work that combines K-Star algorithm classifier with Information Gain as a filtering approach for feature selection; produces better classification accuracy with other existing approaches. We have performed 2 class (attack or normal) classifications and to verify the effectiveness of the proposed IDS system, NSL-KDD dataset has been used. NSL-KDD is a new version of KDDcup99 dataset, which is considered as a standard benchmark for intrusion detection evaluation [5]. The training dataset of NSL-KDD is similar to KDDcup99 and consists of approximately 4,900,000 single connection vectors, each of which contains 41 features and is labelled as either normal or attack type. Every instance in the dataset has 42 features or attributes including target class as shown in Table1.

Sr. No	Feature Name	Sr. No	Feature Name
1	Duration	22	s_guest_login
2	Protocol_type	23	Count
3	Service	24	Srv_count
4	Flag	25	Serror_rate
5	Src_bytes	26	Srv_serror_rate
6	Dst_bytes	27	Rerror_rate
7	Land	28	Srv_rerror_rate
8	Wrong_fragment	29	Same_srv_rate
9	Urgent	30	Diff_srv_rate
10	Hot	31	Srv_diff_host_rate
11	Num_failed_logins	32	Dst_host_count
12	Logged_in	33	Dst_host_srv_count

13	Num_compromised	34	Dst_host_same_srv_rate
14	Root_shell	35	Dst_host_diff_srv_rate
15	Su_attempted	36	Dst_host_same_src_port_rate
16	Num_root	37	Dst_host_srv_diff_host_rate
17	Num_file_creations	38	Dst_host_serror_rate
18	Num_shells	39	Dst_host_srv_serror_rate
19	Num_access_files	40	Dst_host_rerror_rate
20	Num_outbound_cmds	41	Dst_host_srv_rerror_rate
21	s_host_login	42	Normal or Attack

TABLE 1: FEATURES OF NSL- KDD CUP'99 DATASET

II. RELATED WORKS

This section summarizes some of the techniques that used for designing and developing IDS. In [3] an intrusion detection approach is proposed which has been called FC-ANN. It uses Artificial Neural Network and Fuzzy Clustering with the addition of system restore point (SRP). SRP is a component that allows rolling back of system files, registry keys and installed programs etc. SRP is stored in a cloud server that uses fuzzy clustering technique. In fuzzy clustering technique, the heterogeneous training set is divided to several homogenous subsets. In [5] Chunhua Gu and et al. proposed a system using rough set for attribution reduction and support vector machine for intrusion detection classification. In [6] an Intrusion detection system has been effectively introduced by using Principal Component Analysis (PCA) as an approach to select the optimum feature subset with Support Vector Machines (SVMs) as the system classifier. An architecture proposed by employing a hybrid ANN (Artificial Neural Network) for both visualizing intrusions activities using Kohenen's SOM, and classifying intrusions using resilient propagation neural networks is presented in [7]. In [8] Horeis used self-organizing maps (SOM) and radial basis function (RBF) networks. The system offers better results than IDS based on RBF or SOM networks alone. [9] Shows that the dimension reduction and identification of effective network features for category-based selection can reduce the processing time in an intrusion detection system while maintaining the detection accuracy within an acceptable range.

III. PROPOSED FRAMEWORK FOR IDS

In this section we present the whole framework of the new architecture, and then we discuss the main models, the information gain for feature selection and the K-star classification model. The proposed intrusion detection technique initially filters the given dataset by using Information Gain for feature selection. A feature with the highest information gain is the criteria for the selection of the attributes. First we reduced the features of the data set and then run the proposed architecture. Experimental results show that learning time of the algorithm is obviously decreased without compromising the accuracy of the algorithm, which is desirable feature in any IDS. After selecting the features the data set is passed to the K-Start algorithm for training and testing. A test mode with splitting by 66.0% for training set and the remainder for testing set has been used. The block diagram of the proposed method is given in Figure 1.

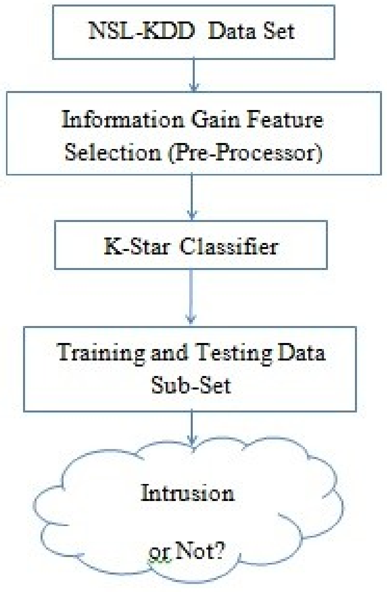
NSL-KDD Data Set
Information Gain Feature Selection (Pre-Processor)
K-Star Classifier
Training and Testing Data Sub-Set
Intrusion or Not?

Figure 1. Block diagram for proposed Technique

3.1- Information Gain Feature set Reduction

The dataset which is used as an input for our intrusion detection system consist of a huge amount of data, and each record of data has numerous attributes associated with it which means that needs a lot of processing. A classification process that considers all these attributes needs a lot of processing time and it leads to an increase in the error rate, and decrease in the efficiency of the classification process. The proposed system comes with a solution to overcome this problem; by using a process that is known as feature selection, the features of the dataset are identified as either being significant to the intrusion detection process, or redundant. Redundant features are generally found to be closely correlated with one or more other features. As a result, omitting them from the intrusion detection process does not degrade classification accuracy. In fact, the accuracy may improve due to the resulting data reduction, and removal of noise and measurement errors associated with the omitted features. Therefore, choosing a good subset of features proves to be significant in improving the performance of the system [11]. In Information Gain the features are filtered to create the most prominent feature subset before the start of the learning process. It takes number and size of branches into account when choosing an attribute as it corrects the information gain by taking the intrinsic information of a split into account (i.e. how much information do we need to tell which branch an instance belongs to). The Intrinsic information is the entropy of distribution of instances into branches.

The Information Gain of a set of attributes and set of training example, is calculated as follow:

Let Atr be the set of all attributes and Tx the set of training example, value(x,a) with x ϵ Tx defines the value of a specific example x for attribute a ϵ Atr, H specifies the entropy. The information gain for an attribute a ϵ Atr is defined as follows:

$$IG\,(Tx, a) = H(Tx) - \sum_{v \in values(a)} \left(\frac{|\{x \in Tx | value(x,a) = v\}|}{|Tx|} \cdot H(\{\in Tx | value(x,a) = v\}) \right)$$

Where the entropy of (Tx) for the probability of specific example x, calculated by:

$$\mathrm{H(Tx)} = -\sum_{x} \mathrm{p(x|Tx)} \log_2 \mathrm{p(x|Tx)}$$

WEKA implementation of the Information gain attribute selector (called Info Gain Attribute Eval) [14] is used to determine the effectiveness of the attributes. The attributes are ranked in decreasing order by the information gain values and as shown in Table 2.

Attribute Rank	Sr.No.	Attribute name	Attribute Rank	Sr.No.	Attribute name
0.806777083	5	src_bytes	0.063682209	2	protocol_type
0.672035304	3	service	0.057011737	27	rerror_rate
0.631947382	6	dst_bytes	0.053991071	40	dst_host_rerror_rate
0.519431475	4	flag	0.052479105	28	srv_rerror_rate
0.515902949	30	diff_srv_rate	0.034316178	1	duration
0.507754237	29	same_srv_rate	0.011069506	10	hot
0.47284563	33	dst_host_srv_count	0.009857171	8	wrong_fragment
0.43902981	34	dst_host_same_srv_rate	0.006294468	13	num_compromised
0.412562311	35	dst_host_diff_srv_rate	0.003833062	16	num_root
0.403611513	38	dst_host_serror_rate	0.001968948	19	num_access_files
0.401731617	12	logged_in	0.001131461	22	is_guest_login
0.396138161	39	dst_host_srv_serror_rate	0.000844263	17	num_file_creations
0.390504636	25	serror_rate	0.000755233	15	su_attempted
0.382406912	23	count	0.000265543	14	root_shell
0.377279134	26	srv_serror_rate	0.000151506	18	num_shells
0.268769398	37	dst_host_srv_diff_host_rate	0.000000263	7	land
0.194957487	32	dst_host_count	0	20	num_outbound_cmds
0.192532029	36	dst_host_same_src_port_rate	0	21	is_host_login
0.144081691	31	srv_diff_host_rate	0	9	urgent
0.093588834	24	srv_count	0	11	num_failed_logins
0.088691095	41	dst_host_srv_rerror_rate	Target class	42	Normal or Attack

Table 2 : Information Gain of the all attributes

3.2- K-star classification algorithm:

K-star or K* is an instance-based classifier. The class of a test instance is based on the training instances similar to it, as determined by some similarity function. It differs from other instance-based learners in that it uses an entropy-based distance function. Instance-based learners classify an instance by comparing it to a database of pre-classified examples. The fundamental assumption is that similar instances will have similar classifications. The question lies in how to define "similar instance" and "similar classification". The corresponding components of an instance-based learner are the distance function which determines how similar two instances are, and the classification function which specifies how instance similarities yield a final classification for the new instance [13]. The K-star algorithm uses entropic measure, based on probability of transforming an instance into another by randomly choosing between all possible transformations. Using entropy as a meter for an instance distance is very beneficial and information theory helps in computing the distance between the instances. The complexity of a transformation of one instance into another is actually the distance between instances. This is achieved in two steps. First define a finite set of transformations that will map one instance into another. Then transform one instance (a) to (b) with the help of the program in a finite sequence of transformations starting at (a) and terminating at (b).

Given a set of infinite points and set of predefined transformations T, let t be a value of the set T.

This t will map t: I→I. To map instances with itself σ is used in T (σ (a) =a). σ terminates P, the set of all prefix codes from T*. Members of T* and of P uniquely define a transformation on I.

$\bar{t}(a) = t_n(t_{n-1}(\cdots t_1(a)\cdots))$ Where t = t1...tn

P is a probability function on T*. It satisfies the following properties:

$$0 \leq \frac{p(\bar{t}u)}{p(\bar{t})} \leq 1$$

$$\sum_{u} p(\bar{t}u) = p(\bar{t})$$

$$p(\Lambda) = 1$$

As a consequence, it satisfies the following:

$$\sum_{\bar{t}\epsilon p} p(\bar{t}) = 1$$

The probability function P* is defined as the probability of all paths from instance a, to instance b:

$$P*\left(\frac{b}{a}\right) = \sum\nolimits_{\bar{t}\epsilon p:\bar{t}(a)=b} P(\bar{t})$$

It is easily proven that P* satisfies the following properties.

$$\sum_{b} P*\left(\frac{b}{a}\right) = 1$$

$$0 \leq P*\left(\frac{b}{a}\right) \leq 1$$

The K* function is then defined as:

$$K*\left(\frac{b}{a}\right) = -\log_2 P*\left(\frac{b}{a}\right)$$

IV. Experiments And Result

4.1- Performance measurement:

In the metric used for evaluating of our proposed architecture the following terms have been used: True positive (TP) for correctly identified, true negative (TN) for correctly rejected, and false positive (FP) for incorrectly identified, Precision, Recall, F-Measure, and Accuracy. Achieving very high accuracy is very easy by carefully selecting the sample size but if we use accuracy as a measure for testing the performance of the system, the system can be biased and can attain very high accuracy. However, Precision and Recall are not dependent on the size of the training and the test samples. These metrics are derived from a basic data structure known as the confusion matrix. A sample confusion matrix for two class case can be represented as shown in Table3.

	Predicted Class		
Actual Class	Activity	Attack	Normal
	Attack	TP	FN
	Normal	FP	TN

Table 3: Confusion Matrix

These metrics are defined as follows:

$$Precision = \frac{tp}{tp + fp}$$

$$Recall = \frac{tp}{tp + fn}$$

Recall in this context is also referred to as the True Positive Rate or Sensitivity, and **precision** is also referred to as Positive predictive value (PPV); other related measures used in classification include True Negative Rate and Accuracy. True Negative Rate is also called **Specificity**.

$$True\ negative\ rate = \frac{tn}{tn + fp}$$

Accuracy is the most basic measure of the performance of a learning method. This measure determines the percentage of correctly classified instances. From the confusion matrix, we can state that:

$$Accuracy = \frac{tp + tn}{tp + tn + fp + fn}$$

F-measure is a measure of a test's accuracy. It considers both the precision and the recall of the test to The F-measure can be interpreted as a weighted average of the precision and recall, where F-measure reaches its best value at 1 and worst score at 0.

The traditional F-measure is the harmonic mean of precision and recall:

$$F - measure = \frac{2 * Precision * Recall}{Precision + Recall}$$

4.2 – Result

The K-Star classification algorithm is used in two ways with the dataset. First the classification model is implemented by using all the features of the dataset. The results of this evaluation are summarized in Table 4.

Parameter	Value
Accuracy	98.1 %
Error Rate	1.82 %
Average True Positive Rate	0.97
Average False Positive Rate	0.018
Average Precision	0.97
Average Recall	0.98
Average F-Measure	0.97
Learning Time	14.03 sec.

Table 4: Results of classification model (K-Star) with all attribute

Then the classification algorithm (K-Star) is evaluated on the dataset by using feature reduction using the Information Gain measure. The results of this test are summarized in Table 5.

Parameter	Value
Accuracy	99.47 %
Error Rate	0.5254
Average True Positive	0.995
Average False Positive	0.005
Average Precision	0.995
Average Recall	0.995
Average F-Measure	0.995
Learning Time	4.3 sec.

Table 5: Results of classification model (K-Star) with feature reduction (Information Gain)

V. CONCLUSION

In this work, the K-Star classifier for Intrusion Detection System and its high accuracy for classifying traffics to either normal or attack with NSL-KDD dataset [15] have been implemented; The dataset has been used in two ways with the same classifier, first using all the dataset features and then in a reduced form (using Information Gain of the attributes). The results show that there is a significant decrease in learning time of the algorithm and an increase in the accuracy, and based on the experiments done in this work using Weka tool [16], and their corresponding results, it could emphasized that information gain is the suitable technique for feature reduction and K-Star classification algorithm is convenient and effective methodology which can be used in the field of intrusion detection.

REFERENCES

[1] Peyman Kabiri and Ali A. Ghorbani, “Research on Intrusion Detection and Response:A Survey” International Journal of Network Security, Vol.1, No.2, PP.84–102, Sep. 2005.

[2] Bhavin Shah and Bhushan H Trivedi, “Artificial Neural Network based Intrusion Detection System: A Survey” International Journal of Computer Applications (0975 – 8887) Volume 39– No.6, February 2012.

[3] Gaikwad, Sonali Jagtap, Kunal Thakare, and Vaishali Budhawant “Anomaly Based Intrusion Detection System Using Artificial Neural Network and Fuzzy Clustering” International Journal of Engineering Research & Technology (IJERT) Vol. 1 Issue 9, November- 2012, ISSN: 2278-0181.

[4] Sandip Sonawane , Shailendra Pardeshi, and Ganesh Prasad “A survey on intrusion detection techniques” World Journal of Science and Technology 2012, 2(3):127-133, ISSN: 2231 – 2587.

[5] Chunhua Gu and Xueqin Zhang,” A Rough Set and SVM Based Intrusion Detection Classifier”, Second International Workshop on Computer Science and Engineering, 2009.

[6] Heba F. Eid, Ashraf Darwish, Aboul Ella Hassanien, and Ajith Abraham” Principle Components Analysis and Support Vector Machine” based Intrusion Detection System”, IEEE 2010.

[7] Jirapummin, C., Wattanapongsakorn, N., and Kanthamanon, P. “Hybrid neural networks for intrusion detection system”, Proceeding of ITCCSCC, PP. 928-931, 2002.

[8] Horeis, T, "Intrusion detection with neural network - Combination of self-organizing maps and redial basis function networks for human expert integration", a Research report 2003. Available in hap://ieee-cis.org/Jiles/ EA C-Research-2003-Report-Horeis.pdf

[9] Zargar, G. R. “Category Based Intrusion Detection Using PCA”, International Journal of Information Security (October 2012), 3, 259-271.

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

[11] Zubair A. Baig, Abdulrhman S. Shaheen, and Radwan AbdelAal, "One-Dependence Estimators for Accurate Detection of Anomalous Network Traffic", International Journal for Information Security Research (IJISR), Volume 1, Issue 4, December 2011.

[12] http://en.wikipedia.org/wiki/Information_gain_ratio.

[13] John G. Cleary, Leonard E. Trigg: “K*: An Instance-based Learner Using an Entropic Distance Measure”, 12th International Conference on Machine Learning, 108-114, 1995.

[14] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, Ian H. Witten “The WEKA Data Mining Software: An Update “ ; SIGKDD Explorations, Volume 11, Issue 1 2009.

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

[16] http://www.cs.waikato.ac.nz/ml/weka/