RESEARCH REVIEW JOURNALS
Volume-04
Issue-04
April -2019

ISSN: 2455-3085 (Online)
RESEARCH REVIEW International Journal of Multidisciplinary
www.rrjournals.com [UGC Listed Journal]

Features we need: A refinement of IDS dataset

[*1]Uzair Bashir & [2]Dr.C.D. Kumawat

[*1]*Mewar University, Ph. D Scholar, Department of Computer Application, Gangrar, Chittorgarh (Rajasthan), India.*
[2]*Mewar University, Professor, Computer Science and Engineering, Gangrar, Chittorgarh (Rajasthan), India.*

ARTICLE DETAILS

Article History
Published Online: 15 April 2019

Keywords
KDD Dataset, Gain ratio, Gini Index

[*]Corresponding Author
Email: ub.cs@uok.edu.in

ABSTRACT

To analyze any network for good or bad traffic, we must have a refined dataset that will help us to train the Intrusion Detection systems in a more efficient manner. The larger the dimension of the data in hand, the longer it will take for any algorithm to make itself aware about the type of data it is dealing with. During our experimental process of designing efficient algorithms, we tried to filter the data in both dimensions. The result was a pruned dataset which contains only those attributes whose values are primarily differentiated during intrusions and can therefore easily lead to identifying an attack.

1. Introduction

The migration of global businesses towards the Internet because of its wider outreach has left no other option but to join alongside others. Over the years, the downsides of being connected over a network have also been pretty clear. It seems though the vantage subjugates its dark side. This has led to a huge amount of data being carried away on the networks. The intrusions also have found a larger turf to act in, therefore, bringing the aspect of security into a much active state. IDS's have come a long way since they began. Expert systems were first introduced to take the burden from system administrators of analyzing huge amounts of data and drawing inferences from it. They proved to be much efficient than their human counterparts, in a way that they never get tired and are less time consuming. Also, the evolution of algorithms that form the core of an IDS have also seen much. Use of artificial intelligence and other machine learning based techniques which try to self-improvise also poses a challenge in the basic structuring of data.

The basic problem in the security however has been unavailability of real time scenario and the data that could help us in understanding the security aspects of this system in a much better way. This has posed limitations in acquiring efficient solutions to security .

In this paper we discuss the KDD dataset that was provided as an experimental resource by DARPA in collaboration with MIT. The dataset contains different scenarios of attacks that have been simulated in an experimental network. This dataset has been around since, and has helped us train many machine based algorithms. We will also analyze the dataset in its second dimension where the attributes of a tuple need to be understood. The later part of the paper tries to classify the attributes according to their importance in the attacks and put the high priority attributes under high cognizance.

2. KDD Data set

The dataset is a matrix of records which represent connections and their attributes arranged in tabular format. The connections can either be normal or they simulate an intrusion. Specifically, four types of attacks have been forces: DOS, U2R, R2L and probes. There are a total of 22 attack types, which are represented as slight variations of these generalized attacks. As (Dhanabal & Shantharajah, 2015) pointed out, the redundancy in the records in both the train and test datasets won't allow the IDS systems to improve and be biased towards specific attack types. The various attack types in the KDD dataset are shown below:

Attack	Attack Type
Back	**DOS**
land	
neptune	
pod	
teardrop	
smurf	
buffer_overflow	**U2R**
loadmodule	
perl	
Rootkit	

ftp_write	**R2L**
guess_passwd	
imap	
multihop	
phf	
spy	
warezmaster	
warezclient	
ipsweep	**Probe**
nmap	
portsweep	
satan	

The dataset itself contains approximately 5 million connection details which have been labelled already. Together

with labelled attribute total of 42 attributes make up a single connection. The values of these attributes are analyzed for differentiating between a good and a bad attack. These features have also been categorized in (Tavallae, Baghe, Lu, & Ghorban, 2009). Besides this, the type of data differentiates data into two types: Numeric and Discrete. Including class labels, there are total of 42 attributes in which 34 are numeric having a particular value at some instant of time and 8 are discrete. If a network administrator has to analyze a connection type, a particular record containing 41 values for each attributes needs to be examined. This problem is discussed in next section which points out the limitations of analyzing these attributes for a connection and how many connections are active during a specific time interval. The list of attributes with their brief description is given in the table below:

S. No.	Attribute Name	Description
1	duration	Time for which connection is active (in seconds).
2	protocol_type	Type of protocol used for communication.
3	service	Type of network service running on destination host.
4	flag	Status of Flag during connection.
5	src_bytes	Total no. of bytes sent by the source host.
6	dst_bytes	Total no. of bytes received by the destination host.
7	land	If the address/port no. is from the same host.
8	wrong_fragment	Total no. of wrong fragments.
9	urgent	No. of urgent packets.
10	hot	No of hot packets.
11	num_failed_logins	Total no. of times Login attempt failed.
12	logged_in	Boolean value to represent login status (0/1).
13	num_compromised	Total no. of compromised connections.
14	root_shell	Boolean value to represent access to root shell (0/1).
15	su_attempted	Super user attempted command status (0/1).
16	num_root	Total no. of root connection obtained.
17	num_file_creations	Total no. of file creations attempts made.
18	num_shells	Total no. of shell prompts opened.
19	num_access_files	Total no. of operations on file accesses.
20	num_outbound_cmds	Total no. of outbound commands executed in an FTP.
21	is_host_login	Binary value to represent host logins.
22	is_guest_login	Binary login to represent a guest user (0/1).
23	count	Total no. of connections to the same host.
24	srv_count	Total no. of attempts to the same service on a host in a connection in last 2 seconds.
25	serror_rate	Percentage of connection with SYN errors.
26	srv_serror_rate	Percentage of connection with SYN errors on a particular service.
27	rerror_rate	Percentage of connection with REJ errors.
28	srv_rerror_rate	Percentage of connection with REJ errors on a particular service.
29	same_srv_rate	Percentage of connection to the same service on a host.
30	diff_srv_rate	Percentage of connection to the different service on a host.
31	srv_diff_host_rate	Percentage of connections to different hosts.
32	dst_host_count	Total no. of connections with different hosts.
33	dst_host_srv_count	Count of services having the same destination host.
34	dst_host_same_srv_rate	Percentage of connections having the same destination host and using the same service.
35	dst_host_diff_srv_rate	Percentage of connections having the same destination host and using the different service.
36	dst_host_same_src_port_rate	Percentage of connections on destination host with same port numbers.
37	dst_host_srv_diff_host_rate	Percentage of connections on destination host with different port numbers.
38	dst_host_serror_rate	Percentage of connections to the current host that have an S0 error.
39	dst_host_srv_serror_rate	Percentage of connections to the current host that have an S0 error using a particular service.
40	dst_host_rerror_rate	Percentage of connections to the current host that have an RST error.
41	dst_host_srv_rerror_rate	Percentage of connections to the current host that have an RST error using a particular service.
42	Class	Binary values of class labels denoting the connection type (Normal/Anomaly)

3. Curse of dimensionality

As the task of analyzing moved from humans to ML systems, we saw a considerable efficacy in the behavior of models built upon these algorithms. But to improve the strategy of ML based models, the size of data on which these models train need to be considerably large. This refers to a phenomena where the data under analysis has higher dimensions and each dimension in considerably large in its own linear path. This problem is of significance in security system where dynamic decisions are required to be operated in a small interval of time. A small delay will lead to the system being compromised and loss of confidentiality. The other problems were also discussed by(Zimek, Schubert, & Hans Peter, 2012). Since machine learning algorithms have been introduced to extemporize the task of pointing the attacks or anomalies in a system, there has been a considerable improvement in IDS's. These algorithms, however, need to be trained in order for them to learn the eco system in which they will deliver. We try to train the model with more training instances so as to make these models perform with positive results. (Trunk, 1979)has shown that the increase in the number of training instances with higher dimensions shows a decrease in the performance of these machine learning algorithms. This phenomena is called as Hughes Phenomena (Hughes, 1968) or peaking phenomena (Theodoridis & Konstantinos, 2008)

4. Attribute Selection Measures

In real life applications, the data which is collected in raw form contains numerous attributes which may not be relevant to the specific problem at an instant of time(James, 2013). Also to improve the models built upon such data it is required to refine the data(Bermingham, 2015) in order to achieve the points that were discussed in previous section. The process of selecting the optimal attributes that will contribute to the learning of ML based models is known as attribute selection measures or feature selection.

5. Algorithm for Selection Highly sensitive features

Input:Data, D, which is a set of training tuples and their associated class labels;
Attribute list, the set of candidate attributes.
Output: A file containing the attributes with their respective information gain, gain ratio and gini index.
Method:
(1) calculate*Information Gain* for the class attribute$\boldsymbol{C}$.
(2) for each attribute $\boldsymbol{A}$
(3) calculate probability for unique values in the attribute$\boldsymbol{A}$.
(4) calculate the *SplitInfo* of the attribute$\boldsymbol{A}$.
(5) for each unique_value of attribute
(6) calculate probability of class values of that unique value of attribute A.
(7) calculate the *Information Gain.*
(8) calculate the *Gini Index.*
endfor
(9) calculate*Gain* for attribute $\boldsymbol{A}$.
(10) calculate*Gain Ratio* for attribute $\boldsymbol{A}$.
endfor
return information_gain, gain_ratio, gini_index

The algorithm uses a KDD dataset with 81125 connections and values for each of the attributes discussed in the earlier sections. It then calculates the information ratio, gini index and gain ratio for each of the attributes and prepares a list of highly sensitive features. We then choose high priority features from each of the selection measures and we take a set of attributes that are common in all the techniques of feature selection.

$$(\boldsymbol{A} \cap \boldsymbol{B} \cap \boldsymbol{C}) = [\{\boldsymbol{A} \cap \boldsymbol{B}\} \cap \{\boldsymbol{B} \cap \boldsymbol{C}\}] \cap \{\boldsymbol{C} \cap \boldsymbol{A}\}$$

The list of attributes with their values is given below:

S. No.	Information Gain	Gain Ratio	Gini Index
1.	Land	Logged_in	Src_bytes
2.	urgent	Srv_serror_rate	Service
3.	Num_outbound_cmds	Flag	Dst_bytes
4.	Is_host_login	Serror_rate	Flag
5.	Num_failed_logins	Dst_host_srv_serror_rate	Diff_srv_rate
6.	Num_shells	Dst_host_serror_rate	Same_srv_rate
7.	Root_shell	Diff_srv_rate	Dst_host_srv_count
8.	Is_guest_login	Same_srv_rate	Dst_host_same_srv_count
9.	Su_attempted	Service	Dst_host_diff_srv_count
10.	Num_file_creations	Src_bytes	Logged_in
11.	Num_access_files	Dst_host_srv_diff_host_rate	Count
12.	Num_root	Wrong_fragment	Dst_host_serror_rate
13.	Num_compromised	Dst_host_diff_srv_rate	Serror_rate
14.	Wrong_fragment	Dst_bytes	Dst_host_srv_serror_rate
15.	Hot	Dst_host_same_srv_rate	Srv_serror_rate
16.	Duration	Dst_host_srv_rerror_rate	Dst_host_srv_diff_host_rate

17.	Srv_rerror_rate	Dst_host_srv_count	Dst_host_count
18.	Dst_host_rerror_rate	Srv_diff_host_rate	Dst_host_same_src_port_rate
19.	Rerror_rate	Srv_serror_rate	Srv_diff_host_rate
20.	Protocol_type	Protocol_type	Srv_count
21.	Dst_host_srv_rerror_rate	Rerror_rate	Dst_host_srv_rerror_rate
22.	Srv_count	Su_attempted	Protocol_type
23.	Srv_diff_host_rate	Dst_host_same_src_port_rate	Rerror_rate
24.	Dst_host_same_src_port_rate	Count	Dst_rerror_rate
25.	Dst_host_count	Num_compromised	Srv_serror_rate
26.	Dst_host_srv_diff_host_rate	Num_root	Duration
27.	Srv_serror_rate	Hot	Hot
28.	Count	Dst_host_count	Wrong_fragment
29.	Serror_rate	Duration	Num_compromised
30.	Dst_host_srv_serror_rate	Num_access_files	Num_root
31.	Logged_in	Num_file_creations	Num_access_files
32.	Dst_host_serror_rate	Dst_host_rerror_rate	Is_guest_login
33.	Dst_host_diff_srv_rate	Num_shells	Num_file_creations
34.	Dst_host_same_srv_rate	Urgent	Su_attempted
35.	Dst_host_srv_count	Srv_count	Root_shell

The final selected list of attributes that we consider is:

S. No.	Attribute
1.	Num_root
2.	Num_compromised
3.	Hot
4.	Srv_rerror_rate
5.	Rerror_rate
6.	Protocol_type
7.	Dst_host_same_src_port_rate
8.	Dst_host_count
9.	Srv_serror_rate
10.	Count
11.	Serror_rate
12.	Dst_host_srv_serror_rate
13.	Wrong_fragment
14.	Dst_host_srv_diff_host_rate
15.	Srv_diff_host_rate
16.	Dst_host_srv_rerror_rate
17.	Logged_in
18.	Dst_host_serror_rate
19.	Dst_host_diff_srv_rate
20.	Dst_host_same_srv_rate
21.	Dst_host_srv_count
22.	Su_attempted
23.	Num_file_creations
24.	Num_access_files
25.	Duration
26.	Dst_rerror_rate
27.	Srv_count

6. Conclusion

As the number of connections is increasing every second, the size of data to be analyzed sees an exponential growth. It also becomes difficult for the machine learning algorithms to analyze data with large dimensions. Our paper discusses the problems that we have to deal with in order to train these ML models. We have made an attempt to refine the dimensions of the data so that the nature of network security which requires a real time decision to whether the connection is good or bad. Any further delay in reacting to a bad connection may lead to unimaginable loss of confidentiality and integrity of the system. The proposed solution may seem to compromise the efficiency by taking lesser features of data into consideration. However, we tend to provide a model that will act in minimum time and try to prevent the system from being compromised.

References

1. Bermingham, M. (2015). Application of high-dimensional feature selection: evaluation for genomic prediction in man. Scientific Reports, 10312.
2. Dhanabal, L., & Shantharajah, S. P. (2015). A Study on NSL-KDD Dataset for Intrusion Detection System Based on Classification Algorithms. International Journal of Advanced Research in Computer and Communication Engineering.
3. Hughes, G. (1968). On the mean accuracy of statistical pattern recognizers. IEEE transactions on information theory, 55-63.
4. James, G. (2013). An introduction to statistical learning. Newyork: Springer, 18.
5. Tavallae, M., Baghe, E., Lu, W., & Ghorban, A. A. (2009). A Detailed Analysis of the KDD CUP 99 Data Set. A detailed analysis of the KDD CUP 99 data set. In 2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications (pp. 1-6). IEEE.
6. Theodoridis, S., & Konstantinos, K. (2008). Pattern recognition. IEEE Transactions on Neural Networks , 376.
7. Trunk, G. (1979). A problem of dimensionality: A simple example. IEEE Transactions on Pattern Analysis & Machine Intelligence, 306-307.
8. Zimek, A., Schubert, E., & Hans Peter, K. (2012). A survey on unsupervised outlier detection in high-dimensional numerical data. Statistical Analysis and Data Mining: The ASA Data Science Journal , 363-387.

