KDD Cup Dataset(Intrusion Detection System)

Surjeet Singh Yadav

1 Abstract of the project

My project is on the KDD CUP dataset. This dataset has been extracted from the network. Dataset can be used for intrusion detection system. On this dataset, we are going to build the model using the neural network. After building the model, we are going to test the accuracy of our system. In the dataset there are forty one features but i have removed some irrelevant features and building our model on twenty six features.

2 Introduction

Instead of increasing the awareness towards the network security , the existing system are not able to protect the system fully against network attack. So to develop an effective intrusion detection system is a challenge which can protect the computer or network. So in recent time intrusion detection system along with the anti-virus software has become an important complement to the security infrastructure of most organization. A significant research work has been made for developing the intelligent based intrusion detection system. Many machine learning methods has been applied for detection of the intrusion. In my project, i am going to apply multilayer perceptron model for detection of the intrusion(intrusion detection system).

2.1 Literature survey

In paper[1] the detailed description of the dataset is given. Description of the dataset contain about the features and class of the dataset. Discussing relevant and irrelevant features and duplicacy of the dataset. In paper[2] the author has describe the technique of reduction of the features and focusing on relevant features. Method of reduction of the features are Mutual information and Chi Square test. In paper[3] the author has used clustering center and nearest neighbor technique and local density for reduction of the features. The resultant features set contain only two features, one is the density and other is the distance. The accuracy of this method is ninety nine,ninety eight, ninety three, sixty seven, and eighty five respectively for the class Normal, Probing, DOS, U2R, R2L respectively. In this paper the author has taken very little dataset around more than one lac. In paper[4] the author has used the mutual information and given the algorithm for reduction of the features. The author has given two algorithm for reduction of the features, by first algorithm the author has get the nineteen features and by second algo he get the seventeen features from forty one features. After that the author has applied the least square support vector machine technique for evaluation of the dataset. The author has achieved the 99.79 and 98.41 accuracy for the first and second algorithm.

3 Resources

Data Source:- http://archive.ics.uci.edu/ml/datasets/KDD+Cup+1998+Data
Tool Use:-Weka

3.1 Work done

This may include the following.

- Description of the data- The dataset used in our project is KDD CUP 1999 dataset. This dataset is extracted from the network. Dataset include forty-one features and twenty two classes but the focus has been done on five classes for the classification.

- Exploration of different neural networks and observation from the same:- I have used multilayer perceptron model for the classification of the data. In first method i have used the one hidden layer at that time the accuracy was 95.93 and in the second method i have increased the hidden layer from one to two the accuracy of the classifier has decreased. In second method the accuracy was 92.72

- Error plot for validation set

- Final architecture:- In our multilayer perceptron model in first method, i have 26 input neuron and 24 hidden neuron and 22 neuron in the output layer.The learning rate has been set 0.3 and momentum was 0.2 .The number of epoch has been used 500. In second method i have increased the number of hidden layer from one to two. All are same as in first method but i have put ten neuron in the second hidden layer.

- Results from different optimization techniques:-In our dataset there are forty-one features, but all are not important, some are irrelevant features. To remove the irrelevant features i have used correlation coefficient. Using the correlation coefficient, i have reduced the features from 41 to 26. I have got the accuracy of 95.93 in the first method when i have used one hidden layer it is shown in Table-1. Summary of the result and Confusion matrix has been shown in table-2 and table-3 respectively. Below i have shown the connectivity from node '0' of input layer to to all other node of hidden layer and thresh-hold value and weight are shown. I have shown this for only one node this dataset is available for all other node. By the second method i have got the accuracy of 92.72 when i have used two hidden layer while other parameter remain same like learning rate, momentum and epoch etc.

  Sigmoid Node 0
  Inputs Weights
  Threshold 0.20240647757406782
  Node 22 -10.607811264343017
  Node 23 -8.060498607019074
  Node 24 -3.1865083318149034
  Node 25 -0.22912837565173147
  Node 26 -10.348007485205974
  Node 27 -9.017625391540934
  Node 28 6.927899943162737
  Node 29 2.203048956176762
  Node 30 11.167873554432445
  Node 31 -0.8676080083865616
  Node 32 8.76300126606696
  Node 33 -7.228399051064481
  Node 34 -4.044984666295649
  Node 35 1.062103722220065
  Node 36 -10.60211684073332
  Node 37 -10.05157881925507
  38 -4.853399930052024
  Node 39 -3.7682983981652565
  Node 40 -7.020879050194067
  Node 41 15.154719053574722
  42 7.555349161418893
  Node 43 -13.463234530328034
  Node 44 1.751915775929347
  Node 45 11.4891298845487

Table 1: Accuracy By First Method

Correctly Classified Instances	89519	95.93%
Incorrectly Classified Instances	3790	4.06%

Table 2: Summary of Result from First Method

TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
0.993	0.042	0.851	0.993	0.916	0.898	0.995	0.980	normal
0.999	0.001	1.000	0.999	1.000	0.999	1.000	1.000	dos
0.000	0.000	0.000	0.000	0.000	0.000	0.765	0.025	u2r
0.984	0.001	0.958	0.984	0.971	0.970	0.998	0.977	r2l
0.483	0.000	0.908	0.483	0.631	0.661	0.941	0.781	probe
0.000	0.000	0.250	0.000	0.001	0.009	0.983	0.476	snmpgetattack
0.000	0.000	0.000	0.000	0.000	0.000	0.692	0.000	7d
0.000	0.000	0.000	0.000	0.000	0.000	0.678	0.000	xlock
0.000	0.000	0.000	0.000	0.000	0.000	?	?	xsnoop
0.000	0.000	0.000	0.000	0.000	0.000	0.696	0.000	sendmail
0.812	0.003	0.397	0.812	0.533	0.566	0.873	0.446	saint
1.000	0.000	0.971	1.000	0.985	0.985	1.000	0.989	apache
0.000	0.000	0.000	0.000	0.000	0.000	0.923	0.000	3storm
0.000	0.000	0.000	0.000	0.000	0.000	0.671	0.025	xterm
0.987	0.002	0.622	0.987	0.763	0.783	1.000	0.971	mscan
0.987	0.000	0.996	0.987	0.991	0.991	0.999	0.994	processtable
0.000	0.000	0.000	0.000	0.000	0.000	0.744	0.000	ps
0.000	0.000	0.000	0.000	0.000	0.000	0.809	0.002	3tunnel
0.000	0.000	0.000	0.000	0.000	0.000	0.783	0.000	worm
0.999	0.000	0.981	0.999	0.990	0.990	1.000	0.997	mailbomb
0.000	0.000	0.000	0.000	0.000	0.000	?	?	sqlattack
0.000	0.000	0.000	0.000	0.000	-0.000	0.995	0.755	snmpguess

.

- github location for your code:- https://github.com/surjeetsinghyadav/Feature-reduction-and-normalization
- You may upload the data-set on github if it is developed by you.
- Include figures wherever it is required

3.2 Future work

There are other machine learning technique are available which can also be applied. I have applied only multilayer perceptron model here, we can apply CNN etc.

Table 3: Confusion Matrix

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	<−
18073	12	0	46	23	3	0	0	0	0	0	2	0	0	17	1	0	0	0	26	0	0	a =
8	66870	0	1	2	0	0	0	0	0	0	3	0	0	21	0	0	0	0	2	0	2	b =
9	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	c =
29	0	0	1810	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	d =
11	1	0	0	347	0	0	0	0	0	263	1	0	0	95	0	0	0	0	0	0	0	e =
2327	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	f =
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	g =
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	h =
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	i =
4	0	0	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	j =
8	0	0	0	3	0	0	0	0	0	173	0	0	0	29	0	0	0	0	0	0	0	k =
0	0	0	0	0	0	0	0	0	0	0	231	0	0	0	0	0	0	0	0	0	0	l =
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	m =
3	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	n =
2	0	0	2	0	0	0	0	0	0	0	0	0	0	304	0	0	0	0	0	0	0	o =
1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	230	0	0	0	0	0	0	p =
2	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	q =
9	0	0	18	7	0	0	0	0	0	0	0	0	0	22	0	0	0	0	0	0	0	r =
0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	s =
1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1480	0	0	t =
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	u =
753	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	v =