Journal of Engineering Sciences

Vol 11, Issue 5,May/2020
ISSN NO:0377-9254

Intrusion Detection Model Using Machine Learning Algorithm on Big Data Environment

V.C.Gayathri[1], K.Vyshnavi[2], U.VishnuPriya[3], K.Jaahnnavi[4], B.Vinay[5]

IT Department, BVRIT, Narsapur, gayathri1252@gmail.com[1]

IT Department, BVRIT, Narsapur, 16211A1245@bvrit.ac.in[2],

IT Department, BVRIT, Narsapur, 16211A1250@bvrit.ac.in[3],

IT Department, BVRIT, Narsapur, 16211A1248@bvrit.ac.in[4],

IT Department, BVRIT, Narsapur, 16211A1217@bvrit.ac.in[5],

Abstract

Recently, the huge amounts of data and its incremental increase have changed the importance of information security and data analysis systems for Big Data. Intrusion detection system (IDS) is a system that monitors and analyzes data to detect any intru- sion in the system or network. High volume, variety and high speed of data generated in the network have made the data analysis process to detect attacks by traditional techniques very difficult. Big Data techniques are used in IDS to deal with Big Data for accurate and efficient data analysis process. This paper introduced Spark-Chi-SVM model for intrusion detection. In this model, we have used ChiSqSelector for feature selection, and built an intrusion detection model by using support vector machine (SVM) classifier on Apache Spark Big Data platform. We used KDD99 to train and test the model. In the experiment, we introduced a comparison between Chi-SVM classifier and Chi-Logistic Regression classifier. The results of the experiment showed that Spark- Chi-SVM model has high performance, reduces the training time and is efficient for Big Data.

EXISTING SYSTEM

There are many types of researches introduced for intrusion detection system. With emerge of Big Data, the traditional techniques become more complex to deal with Big Data. Therefore, many researchers intend to use Big Data techniques to produce high speed and accurate intrusion detection system. In this section, we show some researchers that used machine learning Big Data techniques for intrusion detection to deal with Big Data. Used cluster machine learning technique. The authors used k-Means method in the machine learning libraries on Spark to determine whether the network traffic is an attack or a normal one. In the proposed method, the KDD Cup 1999 is used for training and testing. In this proposed method the authors didn't use feature selection technique to select the related features.

PROPOSED METHOD

Spark Chi SVM

proposed model In this section, the researchers describe the proposed model and the tools and techniques used in the proposed method. Figure 1 shows Spark-Chi-SVM model. The steps of the proposed model can be summarized as follows:

1 Load dataset and export it into Resilient Distributed Datasets (RDD) and Data Frame in Apache Spark.

2 Data preprocessing.

3 Feature selection.

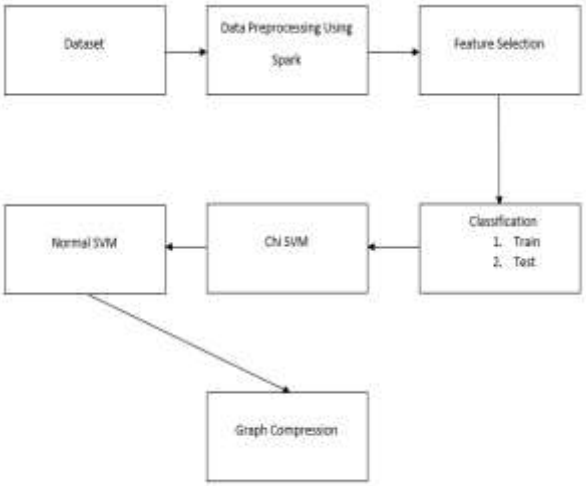
4 Train Spark-Chi-SVM with the training dataset.

5 Test and evaluate the model with the KDD dataset.

DATASET DESCRIPTION

The KDD99 data set is used to evaluate the proposed model. The number of instances that are used are equal to 494,021. The KDD99 dataset has 41 attributes and the 'class' attributes which indicates whether a given instance is a normal instance or an attack. Table provides a description of KDD99 dataset attributes with class labels.

System Architecture



Modules:

- **DATA COLLECTION**
- **DATA PRE-PROCESSING**
- **FEATURE EXTRATION**
- **EVALUATION MODEL**

DATA COLLECTION

Data used in this paper is a set of records. This step is concerned with selecting the subset of all available data that you will be working with. ML problems start with data preferably, lots of data (examples or observations) for which you already know the target answer. Data for which you already know the target answer is called *labelled data.*

DATAPRE-PROCESSING

Organize your selected data by formatting, cleaning and sampling from it.

Three common data pre-processing steps are:

1. Formatting
2. Cleaning
3. Sampling

Formatting: The data you have selected may not be in a format that is suitable for you to work with. The data may be in a relational database and you would like it in a flat file, or the data may be in a proprietary file format and you would like it in a relational database or a text file.

Cleaning: Cleaning data is the removal or fixing of missing data. There may be data instances that are incomplete and do not carry the data you believe you need to address the problem. These instances may need to be removed. Additionally, there may be sensitive information in some of the attributes and these attributes may need to be anonym zed or removed from the data entirely.

Sampling: There may be far more selected data available than you need to work with. More data can result in much longer running times for algorithms and larger computational and memory requirements. You can take a smaller representative sample of the selected data that may be much faster for exploring and prototyping solutions before considering the whole dataset.

FEATURE EXTRATION

Next thing is to do Feature extraction is an attribute reduction process. Unlike feature selection, which ranks the existing attributes according to their predictive significance, feature extraction actually transforms the attributes. The transformed attributes, or features, are linear combinations of the original attributes. Finally, our models are trained using Classifier algorithm. We use classify module on Natural Language Toolkit library on Python

We use the labelled dataset gathered. The rest of our labelled data will be used to evaluate the models. Some machine learning algorithms were used to classify pre-processed data. The chosen classifiers were Random forest. These algorithms are very popular in text classification tasks.

EVALUATION MODEL

Model Evaluation is an integral part of the model development process. It helps to find the best model that represents our data and how well the chosen model will work in the future. Evaluating model performance with the data used for training is not acceptable in data science because it can easily generate overoptimistic and over fitted models. There are two methods of

evaluating models in data science, Hold-Out and Cross-Validation .

To avoid over fitting, both methods use a test set (not seen by the model) to evaluate model performance. Performance of each classification model is estimated base on its averaged. The result will be in the visualized form. Representation of classified data in the form of graphs. Accuracy is defined as the percentage of correct predictions for the test data. It can be calculated easily by dividing the number of correct predictions by the number of total predictions.

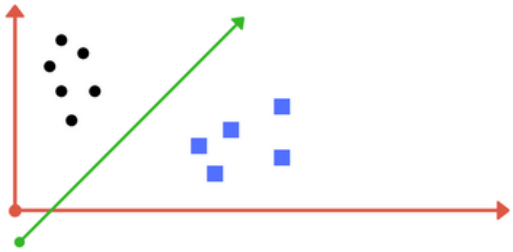
Algorithm:

1. **SVM:SUPPORT VECTOR MACHINE**

A Support Vector Machine (SVM) is a discriminative classifier formally defined by a separating hyperplane. In other words, given labeled training data (*supervised learning*), the algorithm outputs an optimal hyperplane which categorizes new examples. In two dimensional space this hyperplane is a line dividing a plane in two parts where in each class lay in either side.

Suppose you are given plot of two label classes on graph as shown in image *(A)*. Can you decide a separating line for the classes?

You might have come up with something similar to following image (*image B*). It fairly separates the two classes. Any point that is left of line falls into black circle class and on right falls into blue square class. ***Separation of classes. That's what SVM does.*** It finds out a line/ hyper-plane (in multidimensional space that separate outs classes). Shortly, we shall discuss why I wrote multidimensional space.



Kerenal

The learning of the hyperplane in linear SVM is done by transforming the problem using some linear algebra. This is where the kernel plays role.

For **linear kernel** the equation for prediction for a new input using the dot product between the input (x) and each support vector (xi) is calculated as follows:

f(x) = B(0) + sum(ai * (x,xi))

This is an equation that involves calculating the inner products of a new input vector (x) with all support vectors in training data. The coefficients B0 and ai (for each input) must be estimated from the training data by the learning algorithm.

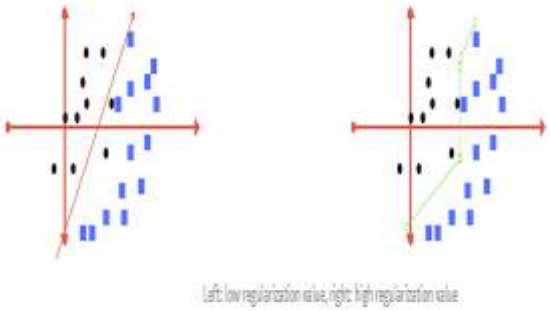
Regularization

Regularization parameter (often termed as C parameter in python's sklearn library) tells the SVM optimization how much you want to avoid misclassifying each training example.

For large values of C, the optimization will choose a smaller-margin hyperplane if that hyperplane does a better job of getting all the training points classified correctly. Conversely, a very small value of C will cause the optimizer to look for a larger-margin separating hyperplane, even if that hyperplane misclassifies more points.

The images below (same as image 1 and image 2 in section 2) are example of two different regularization parameter. Left one has some misclassification due to lower regularization value. Higher value leads to results like right one.

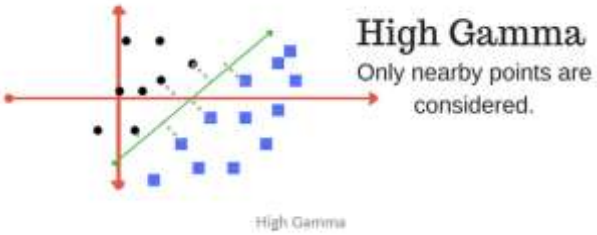
Journal of Engineering Sciences
Vol 11, Issue 5,May/2020
ISSN NO:0377-9254



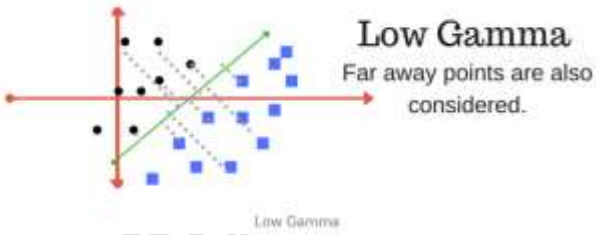
Left: low regularization value, right: high regularization value

Gamma

The gamma parameter defines how far the influence of a single training example reaches, with low values meaning 'far' and high values meaning 'close'. In other words, with low gamma, points far away from plausible separation line are considered in calculation for the separation line. Whereas high gamma means the points close to plausible line are considered in calculation.



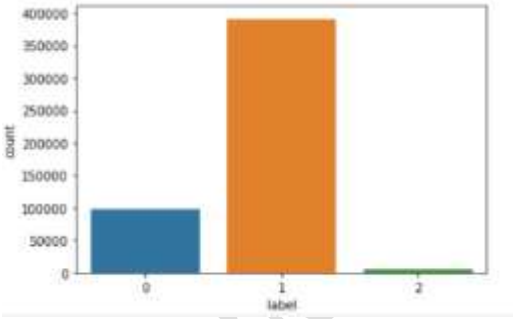
High Gamma



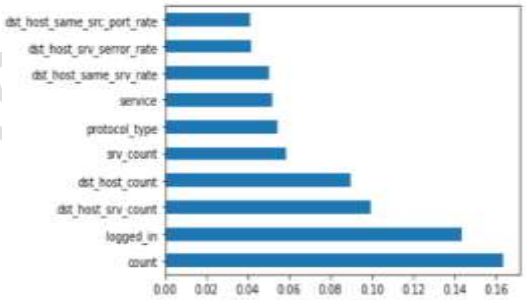
Low Gamma

Results:

- **Figure1:**



- **Figure2:**



- **Conclusion:**

Intrusion detection is aviable and practical approach I for providing a different notion of security in our huge and existing infrastructure of (possible inse- cure) computer and network systems. Intrusion detec- tion systems are based on host-audit-trail and network traffic analysis, and their goal is to detect attacks, preferably in real time. Anumber of prototype intru- sion detection systems have been built, and this con- cept has been proven to be extremely promising. In the future, it is expected that the current proto- typeswill be developed further in order to turn them into production-quality systems. Benchmarking mech- anisms in order to test the effectiveness of IDSs should be developed. Accurate approaches for represent- ing attacks and misuse (including development of mod- elsfor new attackmethods) as well as new and more- effective detection strategies must be investigated. In addition, much more research

is expected to be conducted, e.g., how can the intrusion-detection con- cept be extended to arbitrarily large networks (e.g., the worldwide Internet), how can the IDS itself be protected from attackers, etc.

- **REFERENCES**

[1] A. Alazab, M. Hobbs, J. Abawajy, and M. Alazab, "Using feature selection for intrusion detection system," 2012 Int. Symp. Commun. Inf. Technol., pp. 296–301, 2012.

[2] M. P. K. Shelke, M. S. Sontakke, and A. D. Gawande, "Intrusion Detection System for Cloud Computing," Int. J. Sci. Technol. Res., vol. 1, no. 4, pp. 67–71, 2012.

[3] S. Suthaharan and T. Panchagnula, "Relevance feature selection with data cleaning for intrusion detection system," 2012 Proc. IEEE Southeastcon, pp. 1–6, 2012.

[4] S. Suthaharan and K. Vinnakota, "An approach for automatic selection of relevance features in intrusion detection systems," in Proc. of the 2011 International Conference on Security and Management (SAM 11), pp. 215-219, July 18-21, 2011, Las Vegas, Nevada, USA.

[5] L. Han, "Using a Dynamic K-means Algorithm to Detect Anomaly Activities," 2011, pp. 1049-1052.

[6] R. Kohavi, et al., "KDD-Cup 2000 organizers report: peeling the onion," ACM SIGKDD Explorations Newsletter, vol. 2, pp. 86-93, 2000.

[7] I. Levin, "KDD-99 Classifier Learning Contest: LLSoft s Results Overview," SIGKDD explorations, vol. 1, pp. 67-75, 2000.

[8] J. McHugh, "Testing intrusion detection systems: a critique of the 1998 and 1999 darpa intrusion detection system evaluations as performed by lincoln laboratory," ACM Transactions on Information and System Security, vol. 3, no. 4, pp. 262–294, 2000. [9] M. Tavallaee, E. Bagheri, W. Lu, and A. Ghorbani, "A Detailed Analysis of the KDD CUP 99 Data Set," Submitted to Second IEEE Symposium on Computational Intelligence for Security and Defense Applications (CISDA), 2009.

[10] KDD 99 dataset, Accessed December 2015, http://kdd.ics.uci.edu/databases/kddcup99

[11] NSL KDD dataset, Accessed December 2015, https://github.com/defcom17/NSL_KDD

[12] P. Ghosh, C. Debnath, and D. Metia, "An Efficient Hybrid Multilevel Intrusion Detection System in Cloud Environment," IOSR J. Comput. Eng., vol. 16, no. 4, pp. 16–26, 2014.

[13] Dhanabal, L., Dr. S.P. Shantharajah, "A Study on NSL_KDD Daaset for Intrusion Detection System Based on Classification Algorithms," International Journal of Advanced Research in Computer and Communication Engineering, vol. 4, issue 6, pp. 446-452, June 2015

[14] C. F. Tsai, et al., "Intrusion detection by machine learning: A review," Expert Systems with Applications, vol. 36, pp. 11994-12000, 2009.