

Analysis of Different Machine Learning Classifiers on KDD CUP'99 Dataset

Mohit Kumar Verma
CSE DUAL DEGREE
National Institute of Technology
Hamirpur
Jaipur, India
185502@nith.ac.in

Bhukya Vamshi Rathod
CSE DUAL DEGREE
National Institute of Technology
Hamirpur
Hyderabad, India
185554@nith.ac.in

Nakshtra Goyal
CSE DUAL DEGREE
National Institute of Technology
Hamirpur
Kota, India
185531@nith.ac.in

Abstract— Nowadays information security is very crucial aspect all around the world. All the organizations are trying to secure their confidential information and data from unauthorized access over the network. So, there is a need for analysis of different behaviors over the network for detecting all types of intrusion attacks. This paper presents the Machine Learning methodologies of Intrusion Detection System (IDS) on the network. Based on the source of data, IDS is classified into Host-Based IDS and Network-Based IDS. This paper deals with Network-Based IDS, each packet flowing through the network is analyzed. Principle Component Analysis (PCA) algorithm used to reduce the dimensionality of KDD CUP dataset for classifying in less time. In addition, four classifications used namely Naïve Bayes, K-Means Clustering, Support Vector Machine and Fully Connected Neural Network have been used for effective classification of the dataset. The proposed algorithms enhance the performance of IDS in detecting the attacks with 98.9% accuracy.

I. INTRODUCTION

Billions of Computers have been networked together with very large users over the internet and so security plays a vital role in many organizations such as Industries, Business, Government, Educational Networks. Since, there is a rapid growth of internet communication systems and applications with the availability of tools to intrude the networks, security for network has become indispensable. Most of the users are using computer systems has no knowledge on unauthorized access to their system. IDS systems are used to detect this illegal access to a target application or a computer system over a network.

Network-based IDS's are kept in main areas of network structures in an organization that passively scan and inspect network traffic packets traversing the router or host systems, auditing the packet information and checks its own database of known attack signatures and assign a severity level for each packet. Based on the level assigned, a warning alarm goes-on.

In the past decade, Researchers used statistical approaches and rule-based techniques in IDS development. Due to enormous network traffic leads to oversized large datasets, their methods are now time consuming and not very useful for fast intrusion detection.

Recently, Machine Learning algorithms are providing very promising performance in detecting abnormal behavioral and malicious attacks in the networks.

Machine learning is a subset of artificial intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed. **Machine learning focuses on the development of computer programs** that can access data and use it to learn for themselves. In year 1959, Machine Learning (ML) concept first introduced by Arthur Samuel, defined ML as "field of study that gives computers the ability to learn without being explicitly programmed". It consists in the development of algorithms in order to obtain a predictive analysis from data. There are three kinds of learning based on the nature of the tasks to be performed,

- i. **Supervised Learning:** the machine learns output predictions based on a set of known output for given inputs. In most applications, supervised learning consists in learning the optimal manner to map the inputs to the outputs, by minimizing the value of a loss function representing the difference between prediction and original target.
- ii. **Unsupervised Learning:** the machine learns from input data for which there is no target values. This type of learning method identifies the patterns and features from the inputs, with the aim of extracting new knowledge from the available data.
- iii. **Reinforcement Learning:** at the beginning of the learning, instead of target values feedback about the correctness of the execution is provided after the execution of the task has been completed, thus acting like a reward or punishment, learning reinforcement are often used in dynamic environments.

In this paper, we analyzed KDD CUP dataset using four different Machine Learning techniques namely SVM, Naïve Bayes, K-Means Clustering and Fully connected Neural Network.

A. Support Vector Machine (SVM) :

SVM is a supervised machine learning algorithm that analyzes data for classification and regression analysis. SVM attempts to separate data into multiple classes. It is designed to find the optimal solution to a classification

problem. SVM classifiers produces a better binary classification compared to Decision Tree and Neural Network. Based on the hyper tuning of kernel function and its parameters SVM performance differs. Different types of separating hyperplanes can be achieved by applying a kernel, such as linear, polynomial, Gaussian Radial Basis Function ('rbf'). In datasets, many features are duplicated or less effect in separating datapoints into correct classes. Thus, feature selection should be considered while SVM training. From a set of 41 features of dataset, a subset of features was selected by using feature selection method.

B. Naïve Bayes(NB) :

This approach is based on the theorem called Bayes' principle with robust independence assumptions among the features. Naïve Bayes produce the result of the probability of a particular kind of attack is occurring, given the observed system activities. Naïve Bayes relies on features with different probability of what happens to attacks and normal behavior. Naïve Bayes is the most powerful classification model in IDS due to its ease of use and calculation efficiency, both of which are taken from its conditional independence assumption property. But, in KDD'99 dataset, the IDS do not operate well if this independence assumption is not valid, as the dataset has the complex attribute dependencies.

C. K-Means Clustering :

The K-means method is one of the best common integration techniques of clustering analysis that aims to separate 'n' data objects into 'k' clusters in which each data object is chosen with nearest mean in the cluster. K means is a centroid-based iterative clustering algorithm that provides the highest value for every iteration. The main idea is to reduce the sum of the distances between the datapoints and their respective centroids, it does not need to calculate the distance between all combinations of data as it is a distance-based clustering technique. It uses a Euclidean metrics as a similarity measure, the number of clusters is mentioned by the user in advance if not by default it is 8. Thus, K means clustering is a better approach to classify the data using unsupervised methods for intrusion detection.

D. Fully-Connected Neural Network :

Fully Connected Neural Networks is a feedforward standard neural network architecture applied in mainly basic neural network applications. Fully connected refers to an individual neuron in the earlier layer is linked to every neuron in the sub sequent layer. A fully connected neural network consists of a series of fully connected layers. A fully connected layer is a function from \mathbb{R}^m to \mathbb{R}^n . Each output dimension depends on each input dimension. Let $x \in \mathbb{R}^m$ represent the input to a fully connected layer. Let $y_i \in \mathbb{R}$ be the i -th output from the fully connected layer. Then $y_i \in \mathbb{R}$ is computed as follows:

$$y_i = \sigma(w_{i1}x_1 + \dots + w_{im}x_m)$$

Here, σ is a nonlinear function and the w_i are learnable parameters in the network. The full output y is then

$$y = \sigma(w_{11}x_1 + \dots + w_{1m}x_m) : \sigma(w_{n1}x_1 + \dots + w_{nm}x_m)$$

For a layer of neurons, it is often convenient for efficiency purposes to compute y as a matrix multiply: $y = \sigma(w x)$ where σ is a matrix in $\mathbb{R}^{n \times m}$ and the nonlinearity σ is applied componentwise.

II. METHODOLOGY

In this research, we developed a network-based IDS since we are differentiating the intrusions packets from normal flow traffic based on the underlying network behavior.

A. Dataset Used

There are so many popular datasets such as KDD Cup'99, NSL-KDD'99, UNSWL-NB-15. In this paper, we used most widely adopted for NIDS analysis is KDD CUP'99 dataset for analysis.

KDD CUP'99 is a dataset used for detection of abnormal behaviour of network from normal connections. The dataset is extracted from DARPA dataset in 1999, which contains records of military network environment with injected attacks. KDD'99 consists of 4,898,430 records that is larger than other data sets. Attacks fall into four major categories, they are

- i. Denial of Service (DoS),
- ii. Remote to User (unauthorized access from a remote machine),
- iii. User to Remote (unauthorized access to Root), and
- iv. Probing.

KDD CUP'99 consists of 41 features along with class label. The features are classified into four types, they are

- i. Basic features of individual TCP connections (#1 to #9),
- ii. Content features within a connection suggested by domain knowledge (#10 to #22),
- iii. Traffic features computes using a two-second time window (#23 to #31),
- iv. Host based traffic features (#32 to #41)

B. Pre-processing

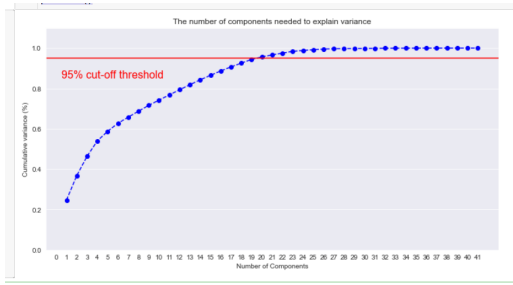
In order to perform our experiments, we need to preprocess the raw data files. In KDD CUP'99 dataset has no noise or null values; thus, it is a clear dataset. It has both categorical and numerical features and numerical data dominates text attributes, it will slow down the training and complicate the process. And the text values cannot be processed in the operations of machine learning. Hence, the dataset needs to be pre-processed. The pre-processing in this model is classified into three main steps:

- i. Normalization: some numerical features are having large number of unique values and scaling ranges differs from each attribute which may reduce the prediction accuracy. So, in this model we used Standard Scaler technique to normalize numerical features to lessen its values and reduce the training processes.

$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}}$$

$$z = \frac{x - \mu}{\sigma}$$

Where A refers to the Z-score normalization, X refers to the feature values, μ indicates to the mean of that feature column and S indicates the standard deviation of the attribute.



- ii. One-Hot Encoding: Since categorical values cannot handle the operations, we convert the text attributes to numerical values using One-Hot encoding technique. After executing one-hot encoder on the dataset, the number of features has been increased to 125 features rather than 41 features.
- iii. Principle Component Analysis: PCA is a very useful mathematical technique to obtain the patterns in higher dimensional dataset. The goal of the PCA is to reduce the dimensionality of the data while retaining as much as possible of the variation present in the original dataset. It is a way of identifying patterns in data and expressing data in such a way that highlights their similarities and difference. As the number of features increased to 125 using PCA method reduced the dimension to 21 features with 95% as cut off accuracy.
- iv. Standard scalar: It transforms the data in such a manner that it has mean as 0 and standard deviation as 1. In short, it standardizes the data. Standardization is useful for data which has negative values. It arranges the data in a standard normal distribution. It is more useful in classification than regression. The idea behind StandardScaler is that it will transform your data such that its distribution will have a mean value 0 and standard deviation of 1. In case of multivariate data, this is done feature-wise (in other words independently for each column of the data).

III. EXPERIMENTAL RESULTS & DISCUSSION

The analysis on proposed NIDS model has been implemented using Jupyter Notebook on Anaconda navigator, which is an open-source platform based on python. KDD CUP'99 dataset is divided into training and testing sets in a ratio of 67% and 33%, respectively. Further training set is divided with 10 fold for cross validating the trained model.

Experiment Setup

This experiment is executed in Jupyter Notebook (python interface) on Dell G3 3590 laptop with configuration Intel(R) Core (TM) i7-9750H CPU @ 2.60GHz 2.59 GHz, 16GB memory without GPU usage, windows 10 64-bit operating system.

A. Evaluation Metrics

Evaluating a model is a major part of building an effective machine learning model to rank the different results. This paper uses sci-kit learn metrics such as Accuracy, Precision, Recall, F1 score to measure the Binary and multi class classification.

- True Positive (TP) : Observation is positive, and is predicted to be positive.
- False Negative (FN) : Observation is positive, but is predicted negative.
- True Negative (TN) : Observation is negative, and is predicted to be negative.
- False Positive (FP) : Observation is negative, but is predicted positive.
- Accuracy :- accuracy is the fraction of predictions our model got right.

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}}$$

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

- Precision :

precision (also called positive predictive value) is the fraction of relevant instances among the retrieved instances

$$\text{Precision} = \frac{TP}{TP + FP}$$

- Recall

recall (also known as sensitivity) is the fraction of relevant instances that were retrieved. Both precision and recall are therefore based on relevance.

$$\text{Recall} = \frac{TP}{TP + FN}$$

- F1-score

The F measure (F1 score or F score) is a measure of a test's accuracy and is defined as the weighted harmonic mean of the precision and recall of the test.

$$\text{F1 Score} = \frac{2 \times (\text{Precision} \times \text{Recall})}{\text{Precision} + \text{Recall}}$$

- False Positive Rate

The false positive rate is calculated as the ratio between the number of negative events wrongly categorized as positive (false positives) and the total number of actual negative events (regardless of classification).

$$\frac{FP}{N} = \frac{FP}{FP + TN}$$

FP = number of false positives

TN = number of true negatives

N = total number of negatives

FOR SINGLE CLASS CLASSIFICATION

1. Naive Bayes

Metrics

Accuracy	98.7014 %
Precision	99.6698 %
Recall	98.7104 %
F1 score	99.1878 %
False Positive Rate	1.34 %

Confusion Matrix

	Predicted : No	Predicted : Yes
Actual : No	38353	519
Actual : Yes	2047	156689

2. K-Means Clustering

Metrics

Accuracy	99.9574 %
Precision	99.9729 %
Recall	99.9741 %
F1 score	99.9735 %
False Positive Rate	1.1 %

Confusion Matrix

	Predicted : No	Predicted : Yes
Actual : No	38829	43
Actual : Yes	41	158695

3. SVM (Support Vector Machine)

Metrics

Accuracy	99.7069 %
Precision	99.8682 %
Recall	99.7669 %
F1 score	99.8175 %
False Positive Rate	5.4 %

Confusion Matrix

	Predicted : No	Predicted : Yes
Actual : No	38663	209
Actual : Yes	370	158366

4. Fully Connected Neural Network

Metrics

Accuracy	99.90 %
Precision	99.86 %
Recall	99.78 %
F1 score	99.82 %
False Positive Rate	0.22 %

Confusion Matrix

	Predicted : No	Predicted : Yes
Actual : No	4510647	9780
Actual : Yes	6484	99446639

FOR MULTI CLASS CLASSIFICATION

1. Naive Bayes

Metrics

Accuracy	89.9270 %
Precision	47.7873 %
Recall	62.7815 %
F1 score	44.8814 %

Confusion Matrix

2. K-Means Clustering

Metrics

Accuracy	99.9509 %
Precision	72.9829 %
Recall	74.7004 %
F1 score	73.3674 %

3. SVM (Support Vector Machine)

Metrics

Accuracy	99.4833 %
Precision	53.8200 %
Recall	54.9822 %
F1 score	53.3663 %

4. Fully Connected Neural Network

Metrics

Accuracy	99.80 %
Precision	99.67 %
Recall	98.88 %
F1 score	99.27 %
False Positive Rate	0.33 %

Confusion Matrix

	Predicted : No	Predicted : Yes
Actual : No	9492279	31578
Actual : Yes	107539	211172624

B. Analysis of Experimental Results

IV. CONCLUSION

In this paper, we proposed two models (Binary and Multi-Class Classification) using multiple machine learning techniques for detecting network attacks. The Accuracy, Precision, Recall, False Positive Rate scores are compared with different ML algorithms. We can achieve highest accuracy by using fully connected neural network as compared to other ML methods. But this approach is a time consuming for training the model instead we can use other techniques for faster intrusion detection.

Our experimental results showed that the fully connected neural network shows better accuracy with 99% accuracy

score. SVM model able to detect and classify with 97.9% accuracy while K means, and NB predicts with 96.6% accuracy score.

Our future work is to detect intrusions instantly by increasing the time efficiency and to reduce the training process rate for fully connected neural network. And we compare modern datasets with more new features and intrusion attacks with the existing analysis.

REFERENCE

1. Kemmerer, D., Vigna, G.: Intrusion detection: A brief history and overview. *Computer* 35(4), 27–30 (2002)
2. Y. Liu, Shengli Liu and Xing Zhao, "Intrusion Detection Algorithm Based on Convolutional Neural Network," ICETA, 2017
3. H. Belkhiri, A. Messai, M. Belaoued, and F. Haider, "Security in the internet of things: recent challenges and solutions," in *International Conference on Electrical Engineering and Control Applications*, pp. 1133–1145, Constantine, Algeria, 2019.
4. Abdelouahid Derhab, Arwa Aldweesh, Ahmed Z. Emam, Farrukh Aslam Khan, "Intrusion Detection System for Internet of Things Based on Temporal Convolution Neural Network and Efficient Feature Engineering", *Wireless Communications and Mobile Computing*, vol. 2020, Article ID 6689134, 16 pages, 2020.
5. Nadia Chaabouni. Intrusion detection and prevention for IoT systems using Machine Learning. *Systems and Control [cs.SY]*. Université de Bordeaux, 2020. English.
6. Ahmad, M., Riaz, Q., Zeeshan, M. et al. Intrusion detection in internet of things using supervised machine learning based on application and transport layer features using UNSW-NB15 data-set. *J Wireless Com Network* 2021, 10 (2021).
7. UCI: KDD Cup 1999 Data. University of California, Irvine (1999)
8. Intrusion Detection Systems for IoT: opportunities and challenges offered by Edge Computing Pietro Spadaccino and Francesca Cuomo, Senior Member, IEEE.
9. A. Khraisat, I. Gondal, and P. Vamplew, An Anomaly Intrusion Detection System Using C5 Decision Tree Classifier. Springer International Publishing, 2018
10. Thakkar, A., Lohiya, R. A Review on Machine Learning and Deep Learning Perspectives of IDS for IoT: Recent Updates, Security Issues, and Challenges. *Arch Computat Methods Eng* 28, 3211–3243 (2021)
11. Rasheed Ahmad, Izzat Alsmadi, Machine learning approaches to IoT security: A systematic literature review , *Internet of Things*, Volume 14, 2021, 100365, ISSN 2542-6605,
12. .M. Hasan et al. Attack and anomaly detection in IoT sensors in IoT sites using machine learning approaches *Int. Things*