**Intrusion Detection System using various Machine Learning Algorithms**

An intrusion Detection System can also be built using various Machine Learning Algorithms. In the section, we will build Intrusion Detection System using 6 machine learning algorithms and discuss their results. Intrusion Detection System using Machine Learning is a type of classifier problem in which the trained ML model is meant to classify a connection between a good (no intrusion detected) and a bad connection (possible intrusion). We may classify the attacks into four different categories as follows:

1. DOS: Denial of Service Attacks. E.g.: SYN – Flooding
2. R2L: Unauthorized access to the system from a remote machine.
3. U2R: Unauthorized access to root user from a remote machine or connection.
4. Probing: Surveillance and other types of similar attacks. E.g.: port scanning.

For this purpose, we used the KDD Cup 1999 dataset. Below is a brief description of the dataset.

Files in the dataset:

1. Kddcup.names : This file contains a list of features.
2. Training\_attack\_types: This file contains list of all the types of attacks.
3. Kddcupdata.gz: Contains the full dataset.

The data set contained 42 columns which represents features and 494021 columns. Some of the sample features are as below:

1. Duration: time period for which connection was active
2. Protocol Type: category of the protocol. Eg : TCP, UDP
3. Service : destination’s network service
4. Src\_Bytes: total bytes from source to destination
5. Dst\_bytes: total bytes from destination to source.
6. Flag : error flag set or not.
7. Land: if connection is from the same host or not ( 0 or 1 value)
8. Wrong\_format: total number of wrong formats
9. Urgent: total number of urgent packets
10. Hot : total number of “hot” indicators
11. Num\_of failed\_attempts: total number of failed login attempts.

The figure below is a heatmap of all the features which shows a correlation between them via the intensity of the color as per the scale shown on the right side of the figure. With the help of heatmap we can remove highly correlated features otherwise they might lead to the problem of overfitting in machine learning and our model might perform well on the testing data but not perform well in the actual scenarios. The following features were removed from the data set as they were highly correlated with some other features.

1. Num\_root
2. Srv\_error\_rate
3. Srv\_rerror\_rate
4. Dst\_host\_srv\_serror\_rate
5. Dst\_host\_serror\_rate
6. Dst\_host\_rerror\_rate
7. Dst\_host\_srv\_rerror\_rate
8. Dst\_host\_same\_srv\_rate

Chart

Description automatically generated

We have used the following machine learning models for building the intrusion detection system from the above data set:

1. Decision Trees
2. Gaussian Naïve Bias
3. Gradient Descent
4. Logistic Regression
5. Random Forest
6. Support Vector Classifier

All of these are classifier algorithms that can classify an incoming connection into a good or bad connection. We have relied upon the accuracy of these models to compare them. Accuracy can be described as how good a certain model is in classifying or identifying a type correctly. More specifically, accuracy is the ratio of a total number of correct classifications to a total number of classifications done in the training data. The accuracy of the various Machine Learning models used is described below:

|  |  |
| --- | --- |
| **Algorithm** | **Accuracy** |
| Decision Trees | 99.0523% |
| Gaussian Naïve Bias | 87.9038% |
| Gradient Descent | 99.7181% |
| Logistic Regression | 99.3528% |
| Random Forest | 99.9687% |
| Support Vector Classifier | 99.8791% |

From the above accuracies, Random Forest is the most accurate of all followed by the Support Vector Classifiers. The Gaussian Naïve Bias performed the worst of all the with an accuracy of less than 90%.

**REFERENCES**

1. By: IBM Cloud Education, “What is gradient descent?,” *IBM*. [Online]. Available: https://www.ibm.com/cloud/learn/gradient-descent#:~:text=Gradient%20descent%20is%20an%20optimization,each%20iteration%20of%20parameter%20updates. [Accessed: 14-Aug-2022].
2. mudgalabhay57@mudgalabhay57, “Intrusion detection system using machine learning algorithms,” *GeeksforGeeks*, 14-Jan-2022. [Online]. Available: https://www.geeksforgeeks.org/intrusion-detection-system-using-machine-learning-algorithms/. [Accessed: 14-Aug-2022].
3. Nidhirastogi, “Network intrusion detection using python,” *Kaggle*, 15-Dec-2018. [Online]. Available: https://www.kaggle.com/code/nidhirastogi/network-intrusion-detection-using-python. [Accessed: 14-Aug-2022].
4. S. M. Othman, F. M. Ba-Alwi, N. T. Alsohybe, and A. Y. Al-Hashida, “Intrusion detection model using machine learning algorithm on Big Data Environment,” *Journal of Big Data*, vol. 5, no. 1, 2018.
5. T. Saranya, S. Sridevi, C. Deisy, T. D. Chung, and M. K. A. A. Khan, “Performance analysis of machine learning algorithms in Intrusion detection system: A review,” *Procedia Computer Science*, vol. 171, pp. 1251–1260, 2020.
6. T. Yiu, “Understanding random forest,” *Medium*, 29-Sep-2021. [Online]. Available: https://towardsdatascience.com/understanding-random-forest-58381e0602d2. [Accessed: 14-Aug-2022].