**Machine Learning Cyber Security System To Detect DOS Attacks**

**Prepared By:**

|  |  |
| --- | --- |
| **Student Name** | **ID** |
|  |  |
|  |  |
|  |  |

**Supervisor:**

**Dr. ..**

**Abstract**

In the Area of Cyber Security, detecting cyber-attacks plays an essential role in the information Security. As the usability of the internet among the users in a wide area is increasing day by day so as the importance of security and to keep the system aware of the malicious activities is also increasing. To keep the system secured, we should ensure the confidentiality, integrity, and availability of the system. In online Services, Denial of Service (DoS) remains as one of the main threats. Attackers can execute DoS by the steps which is easier and with the high efficiency, to slow down services for the user’s access. To detect the DoS attack, supervised machine learning algorithms are used like K-nearest neighbor (KNN), Random Forests and Logistic Regression to detect if the coming traffic is an Attack or Normal traffic. These algorithms are trained on the NSL-KDD dataset which is an effective benchmark dataset set to help researchers compare different intrusion detection methods. The performance of the three models will be compared and the best algorithm is suggested.

1. **Introduction**

In recent years, the attacks that target system availability has widely increased. One of the most effective availability attacks are the DOS attacks (Denial of Service) which result in shutting down the server or the system by flooding the victim with traffic or sending information that triggers a system crash.

Systems can’t detect if the incoming heavy traffic is a normal heavy load traffic due to multiple users sending traffic at same time or it’s a Denial of Service (DOS) attack.

So, by building a DOS classifier model, system can predict if the incoming traffic is an attack or not, and therefore they can stop it or alert the security engineers about an upcoming attack.

Fortunately, there is an available dataset that we will use. NSL-KDD [1] dataset which is a well-known benchmark dataset used for Intrusion Detection Systems, the dataset contains two csv, files one for training (125k records) and the other for testing (22k records).

1. By loading the NSL training dataset, we will make some preprocessing steps on it as follows:

a. Fortunately, the dataset doesn’t contain any null values and any duplicated rows.  
 b. We will convert the attack types into only two classes (0 for normal, 1 for attacks).  
 c. Converting categorical features like (protocol\_type, service, flag) into numerical   
 features using OneHotEncoder from Sklearn library.

2. After the preprocessing step, we will split the training data into two portions (training and validation data), we then will train our algorithms on training part and test its performance on the validation part.

3. Finally, when we make sure that our algorithms perform well on the validation part, we will train the algorithms again but on the whole NSL training dataset (without splitting into train/validation), and then test the models on the NSL testing dataset.

1. **Implementation**

**1. Reading Dataset**

1. First step, is loading the training dataset using pandas library in python, it contains 125973 records with 41 features and 1 label column.

Text

Description automatically generated with medium confidence

2. As the csv file doesn’t contain the column names, we get them from the original NSL-KDD website and added them.

The column names are: [2]

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| duration | wrong\_fragment | su\_attempted | is\_guest\_login | same\_srv\_rate | dst\_host\_same\_src\_port\_rate |
| protocol\_type | urgent | num\_root | count | diff\_srv\_rate | dst\_host\_srv\_diff\_host\_rate |
| service | hot | num\_file\_creations | srv\_count | srv\_diff\_host\_rate | dst\_host\_serror\_rate |
| flag | num\_failed\_logins | num\_shells | serror\_rate | dst\_host\_count | dst\_host\_srv\_serror\_rate |
| src\_bytes | logged\_in | num\_access\_files | srv\_serror\_rate | dst\_host\_srv\_count | dst\_host\_rerror\_rate |
| dst\_bytes | num\_compromised | num\_outbound\_cmds | rerror\_rate | dst\_host\_same\_srv\_rate | dst\_host\_srv\_rerror\_rate |
| land | root\_shell | is\_host\_login | srv\_rerror\_rate | dst\_host\_diff\_srv\_rate | attack |

A screenshot of a computer

Description automatically generated

**2. Preprocessing Steps**

1- The dataset contains 22 types of DoS attacks, these attacks are imbalanced, some of them has a lot of records and others are not, so we combined them into one class (Attack).

Before Combining:

A picture containing chart

Description automatically generated

The Attack Types are: (22 attack and normal)

Text

Description automatically generated with medium confidence

When checking the distribution of these classes, it’s clear that we can’t work on this dataset without combining all attacks into one class.

Chart

Description automatically generated

After Combining them into one class

Graphical user interface, text, application

Description automatically generated

By combining all attack types into one class (1), we will have a dataset with only two classes:

0 🡪 Normal

1 🡪 Attack

Chart, bar chart

Description automatically generated

We can see now that the dataset is well balanced, therefore we can work on it and develop our models.

2- The dataset contains 3 categorical columns (protocol type, service, and flag), since some algorithms can work and learn only from numerical columns, so we need to convert them into numerical form.

Text

Description automatically generated

By using OneHotEncoder function from Sklearn library, we encoded them into numerical columns.

**3. Train / Validation Datasets**

Splitting the data into training and validation dataset, so we can build and train our algorithms on the training part and validate its performance using the validation part.

Graphical user interface, text, application

Description automatically generated

**4. Training Machine Learning Algorithms**

By training three different classification algorithms on the training data and then comparing their performance on the validation dataset, we can predict which algorithm will perform best on the test data.

1- Logistic Regression Model: [3]

Logistic regression is a statistical analysis method to predict a binary outcome, such as yes or no, based on prior observations of a data set.

A logistic regression model predicts a dependent data variable by analyzing the relationship between one or more existing independent variables.

Text, letter

Description automatically generated



Chart, treemap chart

Description automatically generated

87% Training and validation accuracy are not bad for a baseline model, and also aren’t very good to be a classifier for this dataset, so we will try another family of classifiers.

2- Random Forest Classifier: [4]

Random Forest is a meta estimator that fits several decision tree classifiers on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting.

Text

Description automatically generated



Chart, treemap chart

Description automatically generated

This algorithm gives a very good results with 99.9 Accuracy on both training and validation datasets.

3- K-Nearest Neighbor Classifier (KNN): [5]

K-Nearest Neighbor is one of the simplest Machine Learning algorithms based on Supervised Learning technique.

K-NN algorithm stores all the available data and classifies a new data point based on the similarity. This means when new data appears then it can be easily classified into a well suite category by using K- NN algorithm.

Text, letter

Description automatically generated



Chart, treemap chart

Description automatically generated

This algorithm performs well also on the dataset with accuracy 99.6%.

**By comparing the three results, we can try both KNN and RandomForest classifiers on the test dataset.**

**5. Evaluating on Test Dataset**

From the previous comparison between the three models (Logistic Regression, Random Forest and KNN classifiers) we can note that both Random Forest and KNN performs well on the dataset.

At first, we will load the test dataset and make the same preprocessing steps we made on the training dataset on it.

Then we will fit the chosen models on the whole training dataset and evaluate them on the testing dataset.

1- Random Forest Classifier



Graphical user interface, text, application

Description automatically generated

We can notice that our model is overfitting the training dataset, it has 99.9% accuracy on the training dataset with very poor performance on the testing dataset, so we tried to solve the overfitting by increasing number on estimators (n\_estimators) to 200 and decreasing number of features per sample (max\_features) to 20%.

A picture containing text

Description automatically generated



We can see that the testing accuracy increased to 78%, but this still is a very poor performance on the testing dataset.

2- KNN Classifier

Text

Description automatically generated



This algorithm also overfits the training data with 99.7% accuracy on training data and 76% on the testing dataset.

After researching for the reasons that makes our algorithms overfitting the NSL-KDD.  
We found that Overfitting is common in this domain due to the lack of highly representative training data points from real-world applications corresponding to the true data distribution [6].

**III. Comparing Results of Related Works**

According to this paper [7]:

Table

Description automatically generated

We can see that our preprocessing steps and algorithms performed better than some results in this paper and improved the testing accuracy.

**IV. Conclusion**

1- Availability of Systems, is one of the main concerns in the field of Cyber Security, so many systems are built to ensure System’s availability and to protect them from attacks.

2- One of the major and easy attacks that target system availability are the Denial of Service attacks, that flood system with heavy packets, leading the system to crash or to reboot, therefore clients can’t access the site for some times.

3- Machine Learning systems are developed to help to protect these systems from these (DoS) attacks, by training algorithms on different types of packets traffic, so the algorithm can later predict if the upcoming traffic is an attack or not.

4- We developed and trained three algorithms on the benchmark NSL-KDD dataset, that contains about 125k records for normal and different types on DoS attacks.

5- RandomForest and KNN classifiers gave a very good performance when validated on the training dataset, about 99% accuracy.

6- The problem occurred when trying these algorithms on the testing NSL-KDD dataset, giving about 78% accuracy, which means that these algorithms have overfitted the training dataset.

7- This was due to the large amount of dos attack types and ways, and the training dataset is so small, so our algorithms were not trained effectively on all types of attacks provided in the testing dataset.

**REFERENCES:**

[1] https://www.unb.ca/cic/datasets/nsl.html

[2] https://medium.datadriveninvestor.com/did-you-know-the-famous-data-set-called-nsl-kdd-293b39420c74?gi=b68daf39f0fd

[3] https://www.techtarget.com/searchbusinessanalytics/definition/logistic-regression

[4] https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

[5] https://www.javatpoint.com/k-nearest-neighbor-algorithm-for-machine-learning

[6] https://www.researchgate.net/figure/Description-of-NSL-KDD-dataset\_tbl2\_339213807

[7]https://www.researchgate.net/publication/340687919\_An\_Explainable\_Machine\_Learning\_Framework\_for\_Intrusion\_Detection\_Systems