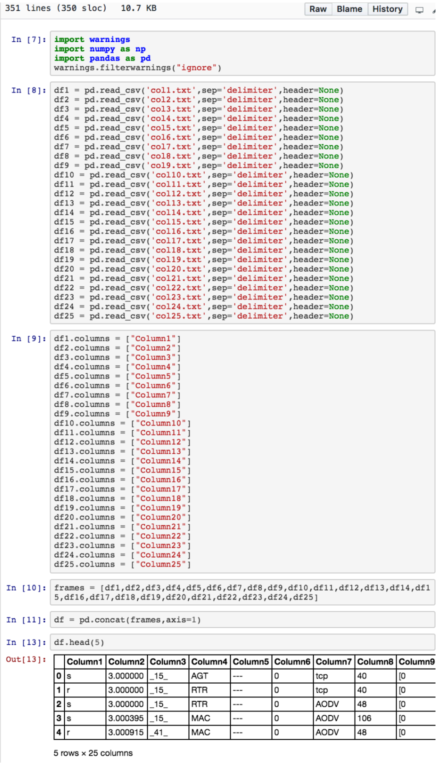
DOS Attack Detection Using Machine Learning

INTRODUCTION :

One of the major security flaws experienced by WSNs is denial of service (DoS) which can even lead to the breakdown of the complete system or to wrong decisions being made by the system that can cause adverse results.Our project aims to detect the the probability of a DoS attack by using Machine Learning techniques.A sensor is an object used to gather information about a physical object or the occurrence of events. Together, many sensors can be used to collect data and communicate wirelessly to a processing station. A Wireless Sensor Network (WSN) is formed when these sensors are deployed cooperatively to monitor large physical environments. Major constraints for WSN include: security, energy (where sensor nodes are powered through either batteries or solar power), memory, computational capability and communication bandwidth.

IMPLEMENTATION :

We generated the dataset by using ns-2 which simulates a DoS attack. In the simulator we take the normalized key parameters as features for our model. We will take 100 training examples and 30 testing examples as our train-test split (70-30) and build a model with it.Primarily, we focus on doing a comparative study of models such as SVM,Neural Networks and XGBoost and build a model with the highest accuracy.



```
In [7]: import warnings
        import numpy as np
        import pandas as pd
        warnings.filterwarnings("ignore")

In [8]: df1 = pd.read_csv('col1.txt',sep='delimiter',header=None)
        df2 = pd.read_csv('col2.txt',sep='delimiter',header=None)
        df3 = pd.read_csv('col3.txt',sep='delimiter',header=None)
        df4 = pd.read_csv('col4.txt',sep='delimiter',header=None)
        df5 = pd.read_csv('col5.txt',sep='delimiter',header=None)
        df6 = pd.read_csv('col6.txt',sep='delimiter',header=None)
        df7 = pd.read_csv('col7.txt',sep='delimiter',header=None)
        df8 = pd.read_csv('col8.txt',sep='delimiter',header=None)
        df9 = pd.read_csv('col9.txt',sep='delimiter',header=None)
        df10 = pd.read_csv('col10.txt',sep='delimiter',header=None)
        df11 = pd.read_csv('col11.txt',sep='delimiter',header=None)
        df12 = pd.read_csv('col12.txt',sep='delimiter',header=None)
        df13 = pd.read_csv('col13.txt',sep='delimiter',header=None)
        df14 = pd.read_csv('col14.txt',sep='delimiter',header=None)
        df15 = pd.read_csv('col15.txt',sep='delimiter',header=None)
        df16 = pd.read_csv('col16.txt',sep='delimiter',header=None)
        df17 = pd.read_csv('col17.txt',sep='delimiter',header=None)
        df18 = pd.read_csv('col18.txt',sep='delimiter',header=None)
        df19 = pd.read_csv('col19.txt',sep='delimiter',header=None)
        df20 = pd.read_csv('col20.txt',sep='delimiter',header=None)
        df21 = pd.read_csv('col21.txt',sep='delimiter',header=None)
        df22 = pd.read_csv('col22.txt',sep='delimiter',header=None)
        df23 = pd.read_csv('col23.txt',sep='delimiter',header=None)
        df24 = pd.read_csv('col24.txt',sep='delimiter',header=None)
        df25 = pd.read_csv('col25.txt',sep='delimiter',header=None)

In [9]: df1.columns = ["Column1"]
        df2.columns = ["Column2"]
        df3.columns = ["Column3"]
        df4.columns = ["Column4"]
        df5.columns = ["Column5"]
        df6.columns = ["Column6"]
        df7.columns = ["Column7"]
        df8.columns = ["Column8"]
        df9.columns = ["Column9"]
        df10.columns = ["Column10"]
        df11.columns = ["Column11"]
        df12.columns = ["Column12"]
        df13.columns = ["Column13"]
        df14.columns = ["Column14"]
        df15.columns = ["Column15"]
        df16.columns = ["Column16"]
        df17.columns = ["Column17"]
        df18.columns = ["Column18"]
        df19.columns = ["Column19"]
        df20.columns = ["Column20"]
        df21.columns = ["Column21"]
        df22.columns = ["Column22"]
        df23.columns = ["Column23"]
        df24.columns = ["Column24"]
        df25.columns = ["Column25"]

In [10]: frames = [df1,df2,df3,df4,df5,df6,df7,df8,df9,df10,df11,df12,df13,df14,df1
         5,df16,df17,df18,df19,df20,df21,df22,df23,df24,df25]

In [11]: df = pd.concat(frames,axis=1)

In [13]: df.head(5)
```

Out[13]:

	Column1	Column2	Column3	Column4	Column5	Column6	Column7	Column8	Column9
0	s	3.000000	_15_	AGT	---	0	tcp	40	[0
1	r	3.000000	_15_	RTR	---	0	tcp	40	[0
2	s	3.000000	_15_	RTR	---	0	AODV	48	[0
3	s	3.000395	_15_	MAC	---	0	AODV	106	[0
4	r	3.000915	_41_	MAC	---	0	AODV	48	[0

5 rows × 25 columns

THINGS DONE :

- Generation of Dataset using Ns-2
- Used awk to extract the texts
- Data Preprocessing is still ongoing using Python frameworks.

THINGS TO DO:

- Finish Preprocessing
- Use Feature Extraction.
- Train and test models and compare their accuracy.

REFERENCES AND WORKS :

Security Enhancement in Wireless Sensor Networks using Machine Learning by Aswathy B. Raj.

The paper achieved an accuracy of 97% . We plan to improve the accuracy by our Machine Learning Models.