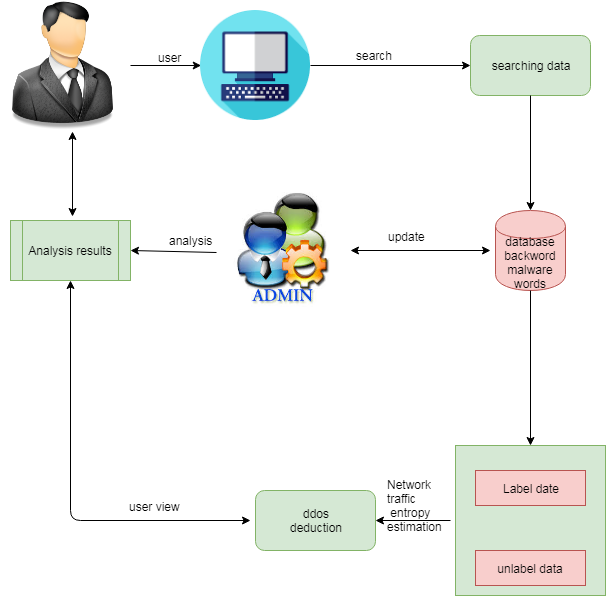
**SEMI-SUPERVISED MACHINE LEARNING APPROACH FOR DDOS DETECTION**

**ABSTRACT:**

The appearance of malicious apps is a serious threat to the Android platform. Most types of network interfaces based on the integrated functions, steal users' personal information and start the attack operations. In this paper, we propose an effective and automatic malware detection method using the text semantics of network traffic. In particular, we consider each HTTP flow generated by mobile apps as a text document, which can be processed by natural language processing to extract text-level features. Later, the use of network traffic is used to create a useful malware detection model. We examine the traffic flow header using N-gram method from the natural language processing (NLP). Then, we propose an automatic feature selection algorithm based on chi-square test to identify meaningful features. It is used to determine whether there is a significant association between the two variables. We propose a novel solution to perform malware detection using NLP methods by treating mobile traffic as documents. We apply an automatic feature selection algorithm based on N-gram sequence to obtain meaningful features from the semantics of traffic flows. Our methods reveal some malware that can prevent detection of antiviral scanners. In addition, we design a detection system to drive traffic to your own-institutional enterprise network, home network, and 3G / 4G mobile network. Integrating the system connected to the computer to find suspicious network behaviors.

**Index Terms**—Malware detection, HTTP flow analysis, text semantics, machine learning.

**ARCHITECTURE:**

****

**EXISTING SYSTEM:**

The first phase of their approach consists of dividing the incoming network traffic into three type of protocols TCP, UDP or Other. Then classifying it into normal or anomaly traffic. In the second stage a multi-class algorithm classify the anomaly detected in the first phase to identify the attacks class in order to choose the appropriate intervention. Two public datasets are used for experiments in this paper namely the UNSW-NB15 and the NSL-KDD Several approaches have been proposed for detecting DDoS attack. Information theory and machine learning are the The performances of network intrusion detection approaches, in general, rely on the distribution characteristics of the underlaying network traffic data used for assessment. The DDoS detection approaches in the literature are under two main categories unsupervised approaches and supervised approaches. Depending on the benchmark datasets used, unsupervised approaches often suffer from high false positive rate and supervised approach cannot handle large amount of network traffic data and their performances are often limited by noisy and irrelevant network data. Therefore, the need of combining both, supervised and unsupervised approaches arises to overcome DDoS detection issues.

**DISADVANTAGES**:

* The datasets above are split into train subsets and test subsets using a configuration of 60% and 40% respectively. The train subsets are used to fit the Extra-Trees ensemble classifiers and the test subsets are used to test the entire proposed approach. Before fitting the classifiers the train subsets are normalized using the *MinMax* method
* This section presents the details of the proposed approach and the methodology followed for detecting the DDoS attack. The proposed approach consists of five major steps: Datasets preprocesing, estimation of network traffic Entropy, online co-clustering, information gain ratio
* The aim of splitting the anomalous network traffic is to reduce the amount of data to be classified by excluding the normal cluster for the classification. For DDoS detection normal traffic records are irrelevant and noisy as the normal behaviors continue to evolve. Most of the time the new unseen normal traffic instances cause the increase of the false positive rate and the decrease of the classification accuracy. Hence, excluding some noisy normal instances of the network traffic data for classification is beneficial in terms of low false positive rates and classification accuracy. Assuming that after the network traffic clustering one cluster contains only normal traffic, a second one contains only DDoS traffic and a third one contains both DDoS and normal traffic.

**PROPOSED SYSTEM:**

This sections introduces our methodology to detect the DDoS attack. The five-fold steps application process of data mining techniques in network systems discussed in characterizes the followed methodology. The main aim of combining algorithms used in the proposed approach is to reduces noisy and irrelevant network traffic data before preprocessing and classification stages for DDoS detection while maintaining high performance in terms of accuracy, false positive rate and running time, and low resources usage. Our approach starts with estimating the entropy of the FSD features over a time-based sliding window. When the average entropy of a time window exceeds its lower or upper thresholds the co-clustering algorithm split the received network traffic into three clusters. Entropy estimation over time sliding windows allows

to detect abrupt changes in the incoming network traffic distribution which are often caused by DDoS attacks. Incoming network traffic within the time windows having abnormal entropy values is suspected to contain DDoS traffic. The focus only on the suspected time windows

allows to filter important amount of network traffic data, therefore only relevant data is selected for the remaining steps of the proposed approach. Also, important resources are saved when no abnormal entropy occurs. In order to determine the normal cluster, we estimate the

information gain ratio based on the average entropy of the FSD features between the received network traffic data during the current time window and each one of the obtained clusters. As discussed in the previous section during a DDoS period the generated amount of attack traffic is largely bigger than the normal traffic. Hence, estimating the information gain ratio based on the FSD features allows to identify the two cluster that preserve more information about the DDoS attack and the cluster that contains only normal traffic. Therefore, the cluster that produce lower information gain ratio is considered as normal and the remaining clusters are considered as anomalous. The information gain ratio is computed for each cluster as follows:

**3.2.1 ADVANTAGE:**

* Where *subsetw* represents the received subset of network data during the time window *w*, *Ci (i* = 1*,* 2*,* 3*)* are the obtained clusters from *subsetw* and |*Ci* | is the size of the *ith* cluster. *avgH(subset)* is the average entropy of the FSD features of the input *subset* and |*subset* | represents the size
* The clustering of the incoming network traffic data allows to reduce important amount of normal and noisy data before the preprocessing and classification steps. More than 6% of a whole traffic dataset can be filtered .

**MODULES:**

There are three modules can be divided here for this project they are listed as below

* User Apps
* DDOS Attack Deduction
* Classifications of DDOS attack
* Graphical analysis

From the above four modules, project is implemented. Bag of discriminative words are achieved

1. **User Apps**

User handling for some various times of smart phones ,desktops laptops and tablets .If any kind of devices attacks for some unauthorized Malware softwares .In this Malware on threats for user personal dates includes for personal contact, bank account numbers and any kind of personal documents are hacking in possible.

1. **DDOS Attack Deduction**

User search the any link Notably, not all network traffic data generated by malicious apps correspond to malicious traffic. Many malware take the form of repackaged benign apps; thus, Malware can also contain the basic functions of a benign app. Subsequently, the network traffic they generate can be characterized by mixed benign and malicious network traffic. We examine the traffic flow header using Co-clustering algorithm from the natural language processing (NLP).

3.**Classifications of DDOS Attack:**

Here, we compare the classification performance of Co-clustering algorithm with other popular machine learning algorithms. We have selected several popular classification algorithms. For all algorithms, we attempt to use multiple sets of parameters to maximize the performance of each algorithm. Using Co-clustering algorithm algorithms classification for malware bag-of-words weightage.

**4. Graphical analysis**

The graph analysis is done by the values taken from the result analysis part and it can be analyzed by the graphical representations. Such as pie chart, pyramid chart and funnel chart here in this project.

**ALGORITHM**

Co-clustering algorithm performs a simultaneous clustering of rows and columns of a data matrix based on a specific criterion . It produces clusters of rows and columns which represent sub-matrices of the original data matrix with some desired properties. Clustering simultaneously rows and columns of a data matrix yields three major benefits: Dimensionality reduction, as each cluster is created based on a subset of the original features. More compressed data representation with preservation of information in the original data. Significant reduction of the clustering computational complexity. The co-clustering computational complexity is *O(mkl* + *nkl)* which is much smaller than that of the traditional Kmeans algorithm *O(mnk)* . Where *m* is the number of rows, *n* is the number of columns, *k* is the number of clusters and *l* is the number of columnclusters.

**REQUIREMENT ANALYSIS**

The project involved analyzing the design of few applications so as to make the application more users friendly. To do so, it was really important to keep the navigations from one screen to the other well ordered and at the same time reducing the amount of typing the user needs to do. In order to make the application more accessible, the browser version had to be chosen so that it is compatible with most of the Browsers.

**REQUIREMENT SPECIFICATION**

**Functional Requirements**

* Graphical User interface with the User.

**Software Requirements**

For developing the application the following are the Software Requirements:

1. Python
2. Django
3. MySql
4. MySqlclient
5. WampServer 2.4

**Operating Systems supported**

1. Windows 7
2. Windows XP
3. Windows 8

**Technologies and Languages used to Develop**

1. Python

**Debugger and Emulator**

* Any Browser (Particularly Chrome)

**Hardware Requirements**

For developing the application the following are the Hardware Requirements:

* Processor: Pentium IV or higher
* RAM: 256 MB
* Space on Hard Disk: minimum 512MB

**CONCLUSION:**

Android is a new and fastest growing threat to malware. Currently, many research methods and antivirus scanners are not hazardous to the growing size and diversity of mobile malware. As a solution, we introduce a solution for mobile malware detection using network traffic flows, which assumes that each HTTP flow is a document and analyzes HTTP flow requests using NLP string analysis. The N-Gram line generation, feature selection algorithm, and SVM algorithm are used to create a useful malware detection model. Our evaluation demonstrates the efficiency of this solution, and our trained model greatly improves existing approaches and identifies malicious leaks with some false warnings. The harmful detection rate is 99.15%, but the wrong rate for harmful traffic is 0.45%. Using the newly discovered malware further verifies the performance of the proposed system. When used in real environments, the sample can detect 54.81% of harmful applications, which is better than other popular anti-virus scanners. As a result of the test, we show that malware models can detect our model, which does not prevent detecting other virus scanners. Obtaining basically new malicious models VirusTotal detection reports are also possible. Added, Once new tablets are added to training samples, we will Please re-train and refresh and update the new malware