*TO DO:*

1. Edit the code to gather the entropy values and PCA values and display it in a graph
2. Make second attack code to attack random hosts (New DDos attack)
3. Run 2 scenarios
   1. Scenario 1
      1. 11 – normal traffic
      2. 152 – DDoS attack
      3. 172 – Double the traffic
   2. Scenario 2 : Destination IP address is chosen random
      1. 8 – normal traffic
      2. 43 – DDoS attack
4. Abstract
5. Proposed system
6. Implementation
7. Result

Notes:

* DDos attack is predicted using entropy and PCA
* Collection of traffic using Wireshark is not feasible since it only collects data from one interface
* Log is used in the codes therefore understanding of log is important
* Learn the formula of entropy
* And how Principal component Analysis (PCA) is calculated
* Search for a better technology to detect DDos in a SDN network

A novel DDos Attack detection scheme for SDN Environment

* **Flow table**: in each switch containing rules for directing packets to their destination, controlled by the controller.
* **Buffer area**: packets which do not match the rules in the flow table are stored here, usually the header of the packet. Memory in less therefore some the existing packets in the buffer are dropped using FIFO and LIFO policy during a case of overflow.
* **Controller**: If there is no match with flow table , the controller sends a PACKET-OUT to all connected switches. If there is no match it will need to wait till timeout, this is taken in to advantage while performing DDos attack to keep the resources of the controller occupied .
* **Low traffic flow**: No matter how heavy the traffic of new flow is,only the first few packets if the flow will be encapsulated in the packet in message and sent to the controller . therefore the attacker eill prefer to use low traffic flow to gain more impact to trigger attack in controller.
* **Heavy** **traffic flow:** heavy traffic that each packet filled with meaningless data to achieve maximum size to consume the space in a switch
* **OD pairs:** Source and destination address of a packet is called Origin-Destination pairs.
* **New DDoS attack:**
  + The destination IP address is chosen at random , no fixed target server is aimed
  + SDN network system is aimed
  + No server detecting been attacked, therefore no server will alarm the attack and therefore harder to be detected and reported.
* **Scapy:** is used to generate traffic
* 0 - normal traffic
* 180 – launch DDos attack (for 20 seconds)
* 200 ­– collect all data
* 1-alpha : confidence level is set to 99%
* Time interval : 1 second
* Number of time intervals : 200
* **Sample entropy**: N ,OD pairs
  + S=sum(ni)Ni=1
  + H(x) = - sum((ni / S)\*log2(ni/2))Ni=1
  + Range of H(x) is (0,log2Nr)
* **PCA:** 
  + PCA identifies DDos attack when the traffic is doubled during the attack, while the value of entropy increases therefore the attack cannot be detected .

Network Intrusion Detection System with help of Machine Learning Algorithms and SND

The base paper for the project is **“Survey on SDN based network intrusion detection system using machine learning approaches”** whichpublished in 2018, my goal is to implement the paper which itself is difficult task as it contains various steps of creating a virtual SDN using NOX which is a OpenFlow controller and to make a classification program which had deep learning capabilities.

SDN: Software- Defined Networking

Software-defined network is an emerging architecture that decouples network control and forwarding functions so that the network control can be directly programmable. The segregation of the control plane from the data plane enables easy network management .This feature of SDN is facilitating. Innovative applications, dictating a new networking paradigm capable of implementing NIDS.

The SDN used in the base paper is NOX which uses OpenFlow protocol to create a programmable network.

NIDS: Network Intrusion Detection System

A system to detect an attack even before it takes is called NIDS, it is very important as NIDS acts as the first layer of protection for a network, Network intrusion detection systems (NIDS) are designed to detect malicious activities including virus, worm, DDoS attacks.

Using Machine Learning, Deep Learning Algorithm:

Machine learning techniques (ML) is applied to develop NIDS to improve detection accuracy and low false alarm rate. As an advanced stream of ML, deep learning (DL) approaches have been adopted in the field of NIDS.

The domain of Machine learning (ML) is dedicated to developing systems that can automatically learn from the data and identify hidden patterns without being explicitly programmed to do so.

**Time-Line:**

1. Create a SDN using NOX
2. Configure the network
3. Gather Traffic information
4. Implement various attacks , main focus is on DDoS attack as it can be identified from the network traffic
5. Implement the classification program and to improve it error rate.

Since this is a time consuming project timeline, I may divert from the time line, First I am planning on creating a network using NOX and in parallel gather a pre-recorded DDoS attack traffic data and classify them using a Deep learning Neural Network Algorithm .

Please let me know if the project is feasible or if any changes needs to be done.

BASE PAPER: **“Survey on SDN based network intrusion detection system using machine learning approaches”** by Nasrin Sultana & Naveen Chilamkurti & Wei Peng & Rabei Alhadad.

**NOTES:**

DL approaches outperformed the ML in logic modeling. Most supervised ML algorithms are good at classification tasks but not in modeling logic. As attacks are unknown unsupervised learning approach such as s**tack autoencode, RNN and hybrid based algorithm** will be the best for NIDS implementation in SDN platform.

**OpenFlow** is the most popular protocol standard that allows the implementation of the SDN concept in both hardware and software environment

**Simulating tools: NS-2 , Mininet , NS-3 , OMNeT++**

**Challenges:**

1. Choosing appropriate feature selection methods that can determine the relevant features of the ID tasks , **optimal number of parameters for the selection**
2. Datasets for evaluation is **KDD Cup 1999** and **NSL-KDD**
3. There is an **existing bottle neck problem** in the NDS controller due to the significant incoming and forwarding traffic , therefore we need to reduce the bottleneck to implement NIDS

“Coates A, Lee H, Ng Andrew Y (2011) An analysis of single-layer networks in unsupervised feature learning. In: Proceedings of the fourteenth international conference on artificial intelligence and statistics,” **paper for feature selection**

DATASETS :

ISCX 2012 Intrusion Detection,

1. **Evaluation Data Set** –“Shiravi A, Shiravi H, Tavallaee M, Ghorbani AA (2012) Toward developing a systematic approach to generate benchmark datasets for intrusion detection.”
2. **CIC DOS Dataset** – “University of New Brunswick (2017) [Online] available http://www.unb.ca/cic/research/datasets/dos-dataset.html.”
3. **ADFA-LD12** – “Creech G, Hu J (2013) Generation of a new IDS test dataset: time to retire the KDD collection. Wirel Commun Netw Conf (WCNC). <https://doi.org/10.1109/WCNC.2013.6555301>”

**NOTES : SDN CISCO**

* The application controls the flow of data using the controller which abstracts the networking devices. Therefore allowing the users the configure the network device without worrying about the OS in the devices
* **APLLICTION LAYER ­- CONTROL LAYER – INFRASTRUCTURE LAYER.**
* The **North bound API** uses : **REST API or JAVA API**
* The **South bound API** uses **OpenFlow API , NETCONF, SNMP**
  + **Openflow** does not update the configuration in the device *(update the OSPF config in the device)* but updates the flow table of the device
* **EAST WEST TRAFFIC :** between controls in the control layer
* **CONTROLLER** : HP SDN controller
* **INFRASTRUCTURE LAYER :** Mininet