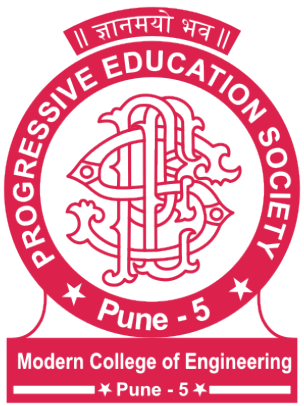
Detection of DDoS in SDN environment using Support Vector Machine, Entropy based discretization and Fuzzy C-Means Clustering

Achyuth Rao - B120314254
Akib Shaikh - B120314257
Arun Pottekat - B120314203
Pranav Tale - B120314249

Guide:- Prof. Mrs. Aparna Junnarkar



PES's Modern College Of Engineering

Contents

Introduction

- SDN separates intelligence from the hardware.
- SDN controller acts as Network Operating System.
- This networking paradigm faces security issues.
- DDoS attack makes the network resources unavailable.

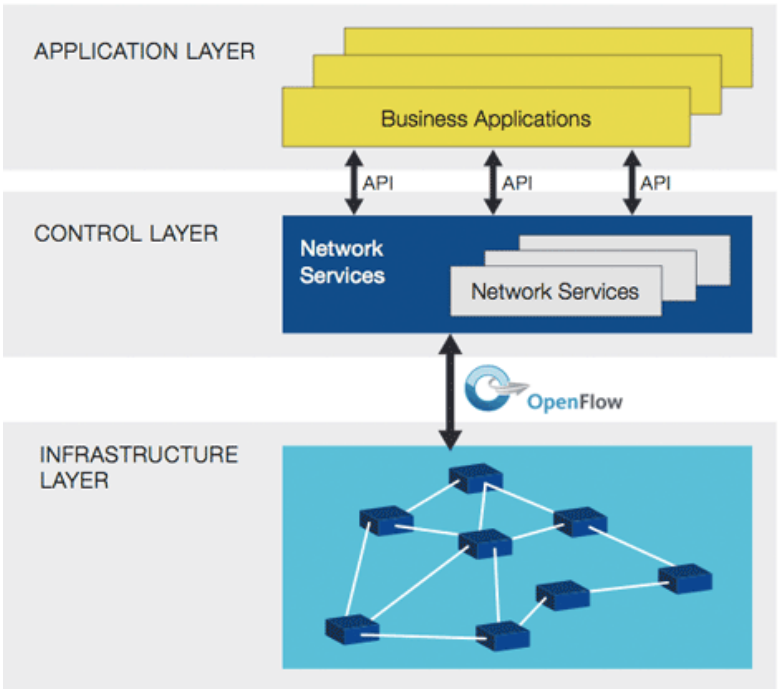


Figure 1: SDN Architecture

Problem Statement

- To develop a solution for the detection of DDoS attack in SDN environment using Support Vector Machine, Entropy Based Discretization, Fuzzy C Means Clustering and monitoring OpenFlow statistics.

Motivation

- Number of cyber attacks is increasing day by day.
- Reluctance to adopt SDN due to lack of security solutions.
- A single DDoS attack can cost an enterprise over $1.6 million.
- SDN market is expected to grow to $56 Billion by 2022.
- Automation of attack detection is required.
- Integration of Machine Learning and Data Mining with SDN.

Objective

- To develop a system to detect DDoS attack in SDN.
- To monitor the network using Elasticsearch, Logstash and Kibana.
- To develop an adaptive solution for change in the network.

Scope

- Set up of SDN environment.
- SVM, Entropy and Fuzzy C Means based DDoS detection method.
- OpenFlow Monitoring application using POX API.

Literature Survey

Title	Author	Journal and Year	Description
DDoS Detection and Analysis in SDN-based Environment Using Support Vector Machine Classifier	Kokila RT, S. Thamarai Selvi, Kannan Govindarajan	IEEE 2014	This paper provides information about DDoS attack in SDN environment using Support Vector Machine to classify the attack.
An Entropy-Based Distributed DDoS Detection Mechanism in Software-Defined Networking	Rui Wang, Zhiping Jia, Lei Ju	IEEE 2015	This paper provides information about DDoS attack in SDN environment using Entropy based mechanism to classify the attack.
Software-Defined Networking:The New Norm for Networks	Open Networking Foundation	ONF White Paper, 2012	Description about Software Defined Networks
Advances in Fuzzy Clustering and its Applications	Jose Valente de Oliveira, Witold Pedrycz	Wiley 2007	This book provides information about the algorithm for Fuzzy C Means Clustering.
OpenFlow Switch Specification	Open Networking Foundation	Version 1.3.2 2013	Description about OpenFlow Protocol

Architecture Diagram

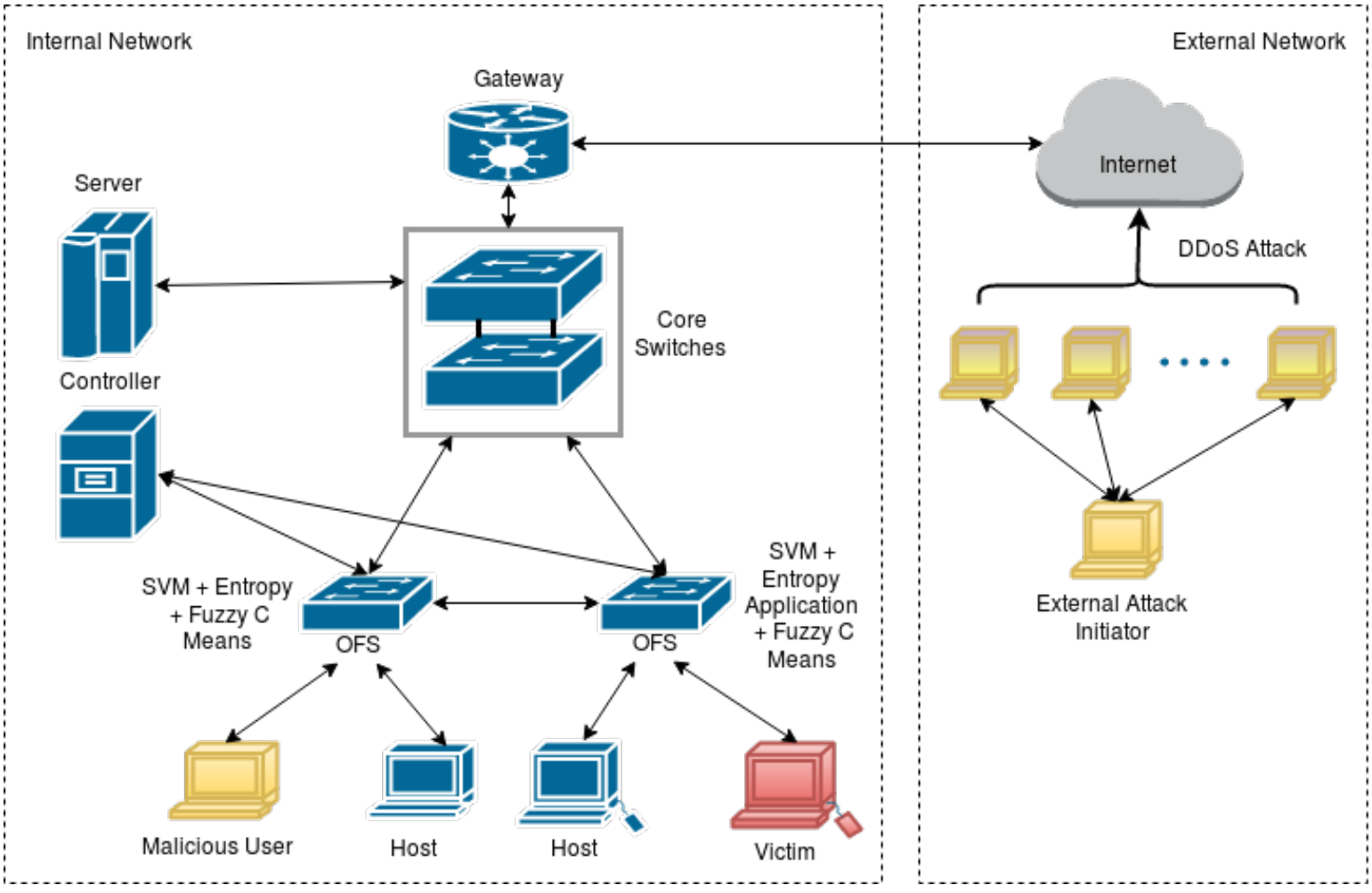


Figure 2: System Architecture

UML Diagrams

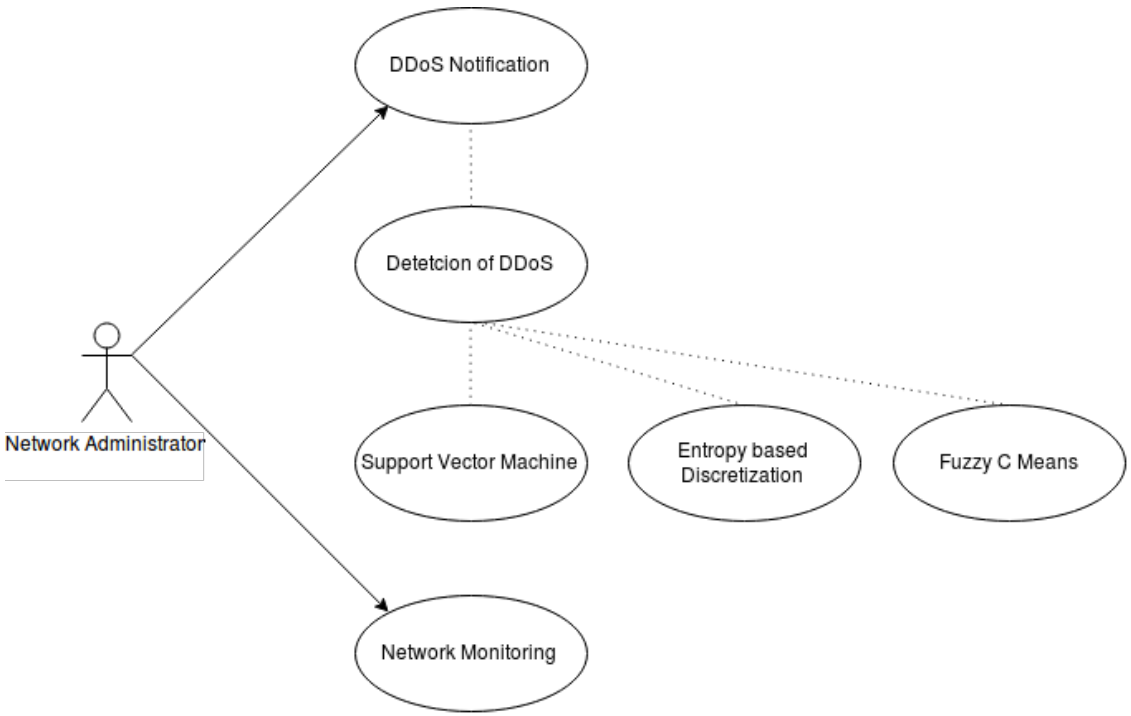


Figure 3: Use Case Diagram

UML Diagrams

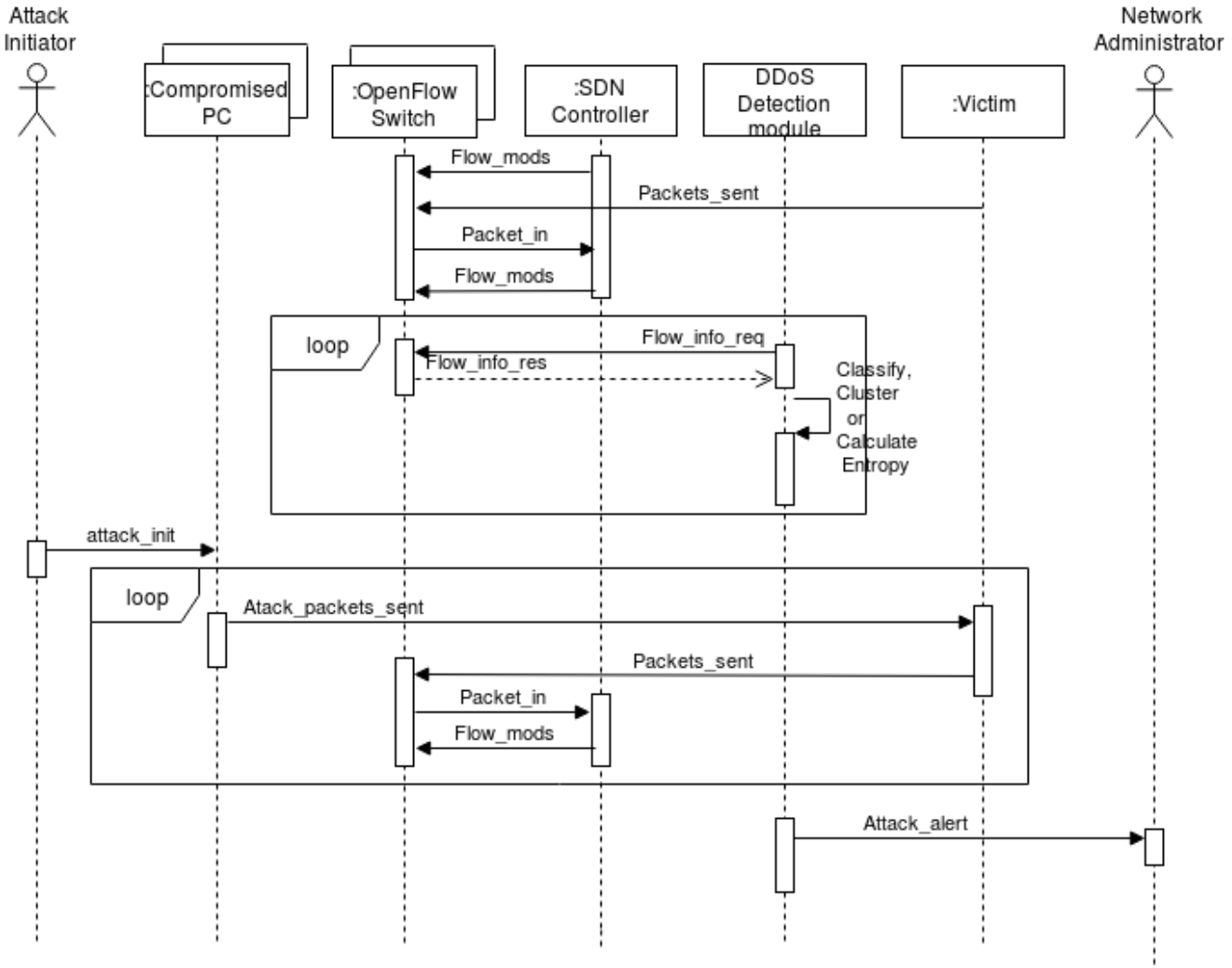


Figure 4: Sequence Diagram

Mathematical Model

$S = \{\{I\}, \{P\}, \{O\}\}$

$I = \{N\}$
where,
$N = \{$ Network Statistics $\}$,
$F = \{F_i \mid F_i \in T, \forall i \ F_i =$ Individual entry $\}$,
$T = \{$ Flow Table $\}$,
$F \subseteq N$

$P = \{P_{EBD}, P_{SVM}, P_{FCM}\}$

$O = \{O_{EBD} \cup O_{SVM} \cup O_{FCM}\}$

Mathematical Model

P_{EBD} (I_{EBD}, O_{EBD})
{

- $I_{EBD} = \{U_i \mid U_i = (\text{ Dest. Addr., Count}),\ U_i \subset F_i\}$
- $P_i = \frac{C_i}{N}$
- $\varepsilon = \sum_{i=0}^{n} -P_i log P_i$
- $\varepsilon_n = \frac{\varepsilon}{N}$
- $(\lambda < \varepsilon_n) \rightarrow (\beta = 0)$
  $(\lambda > \varepsilon_n) \rightarrow (\beta = 1)$
- $O_{EBD} = \{\beta \mid \beta \in (0, 1)\}$

}

P_{SVM} (I_{SVM}, O_{SVM})
{

- $I_{SVM} = \{V_i \mid V_i = (\text{ Src. Addr., Dest. Addr., Time, Prot.})\}$
- $y = \overline{\omega} * x + b$
- $(y \leq -1) \rightarrow (\alpha = 1)$
  $(y \geq 1) \rightarrow (\alpha = 0)$
- $O_{SVM} = \{\alpha \mid \alpha \in (0, 1)\}$

}

Mathematical Model

P_{FCM} (I_{FCM}, O_{FCM})
{

- $I_{FCM} = \{W_i \mid W_i = (\text{ Time, Dest. Addr., X })\}$
- $P_I(z_i) = \sum_{j=1}^{X} \epsilon^{-\alpha||z_i - z_I||^2}$
- $u_{ij} = \dfrac{d_{ij}^{-\frac{2}{m-1}}}{\sum_{l=1}^{c} d_{lj}^{-\frac{2}{m-1}}}$
- $c_{ij} = \dfrac{\sum_{j=1}^{X} u_{ij}^m C_j}{\sum_{j=1}^{X} u_{ij}^m}$
- $(y \leq u_{attack}) \rightarrow (\gamma = 0)$
  $(y \geq u_{attack}) \rightarrow (\gamma = 1)$
- $O_{FCM} = \{\gamma \mid \gamma \in (0, 1)\}$

}

Algorithmic Strategies : Support Vector Machine

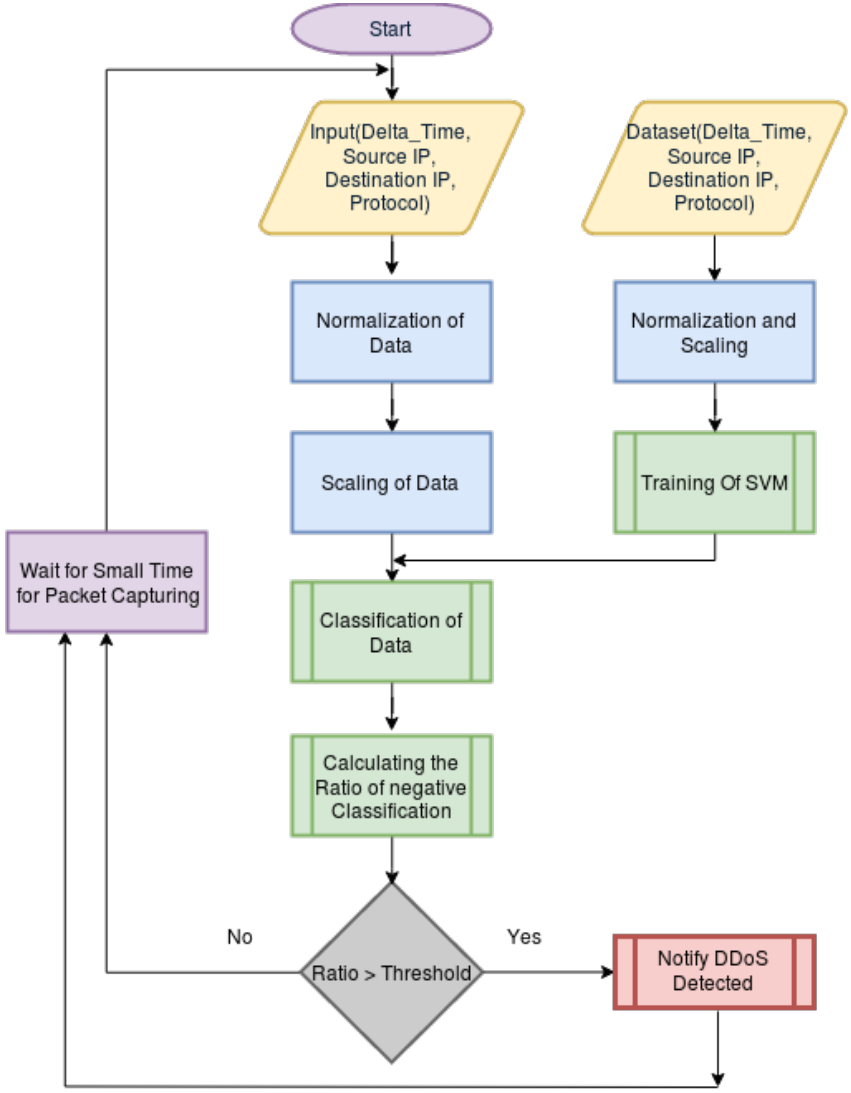


Figure 5: Flowchart: Support Vector Machine Algorithm

Algorithmic Strategies : Support Vector Machine

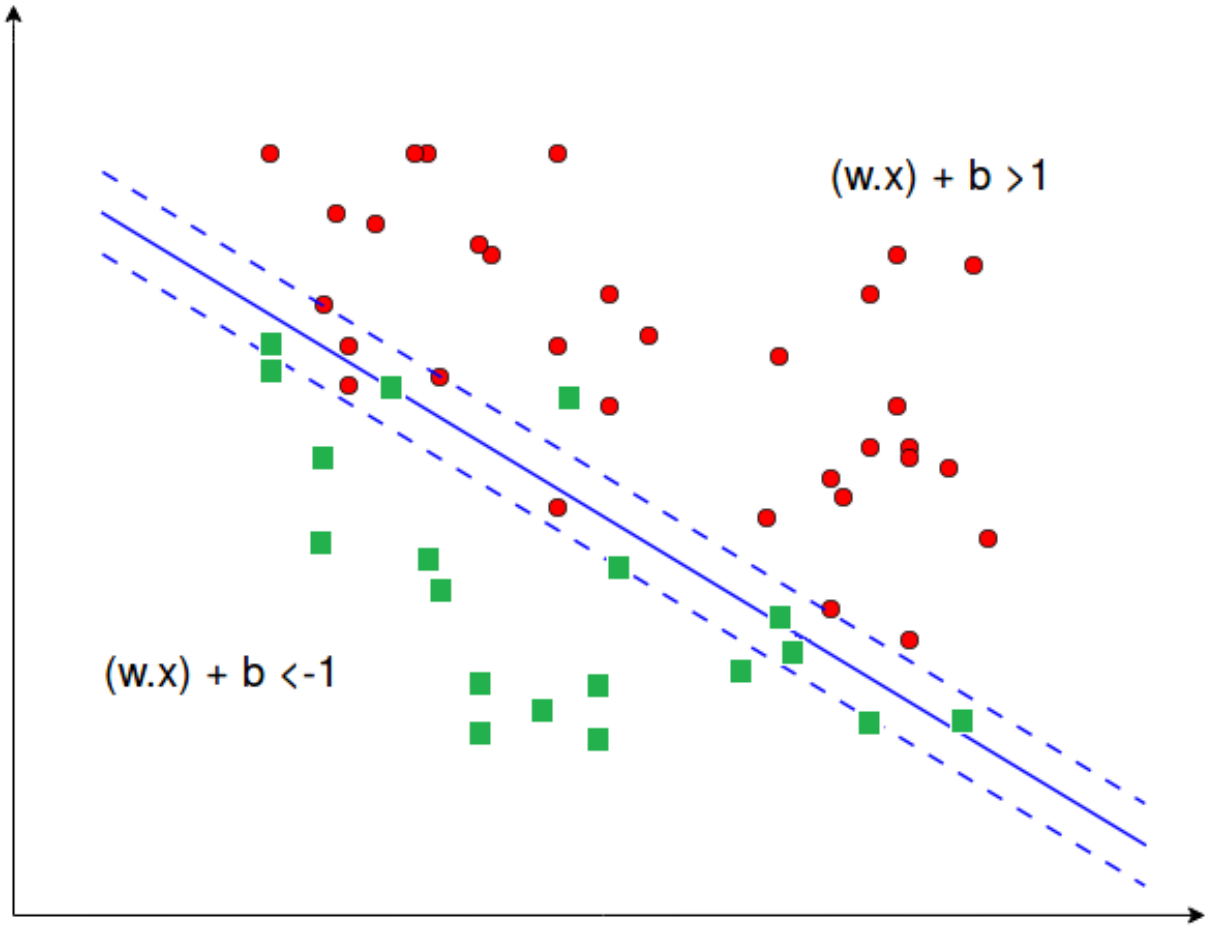


Figure 6: Support Vector Machine Graph

Algorithmic Strategies : Entropy Based Discretization

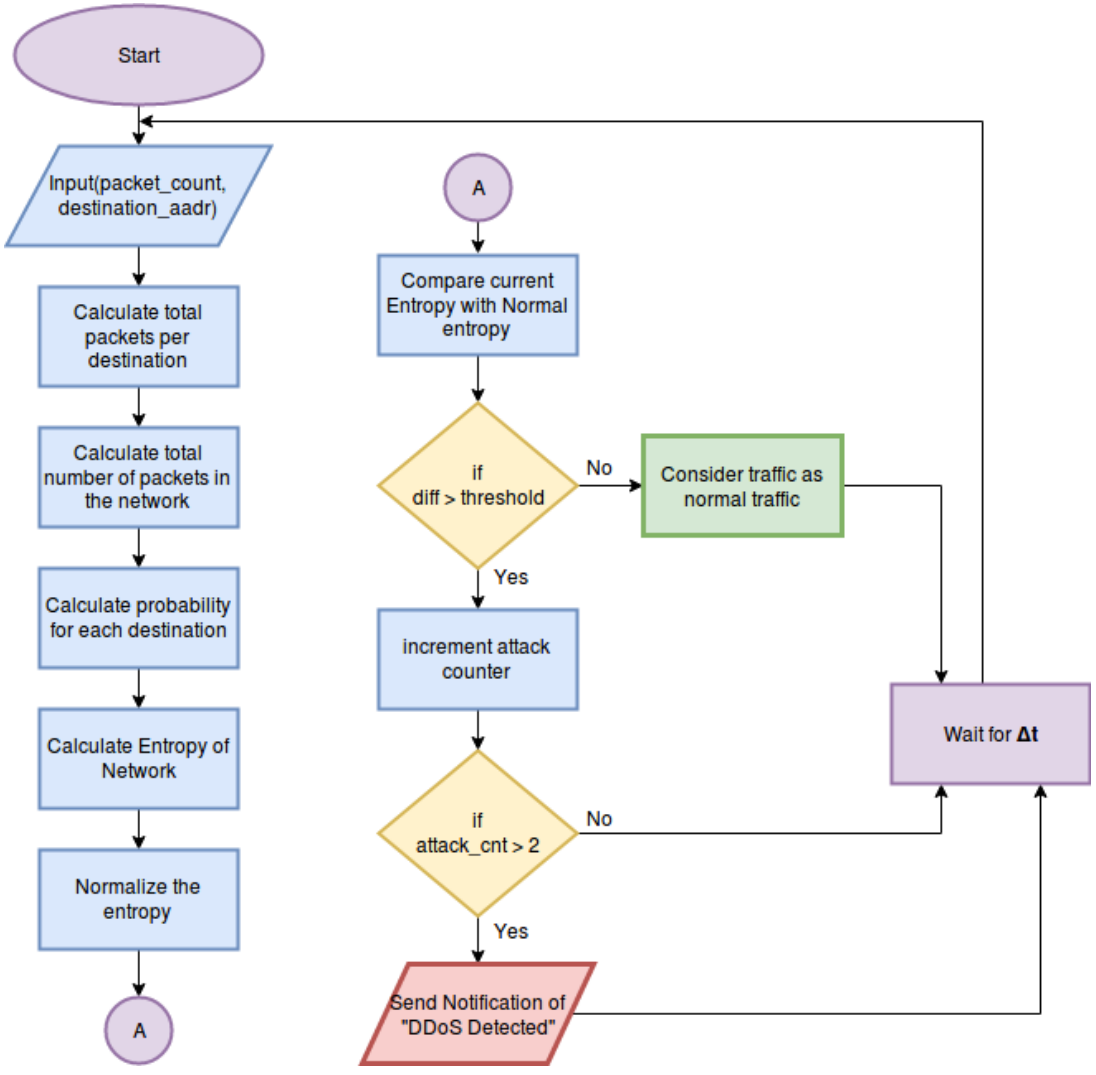


Figure 7: Flowchart: Entropy Based Discretization Mechanism

Algorithmic Strategies : Fuzzy C-Means Clustering

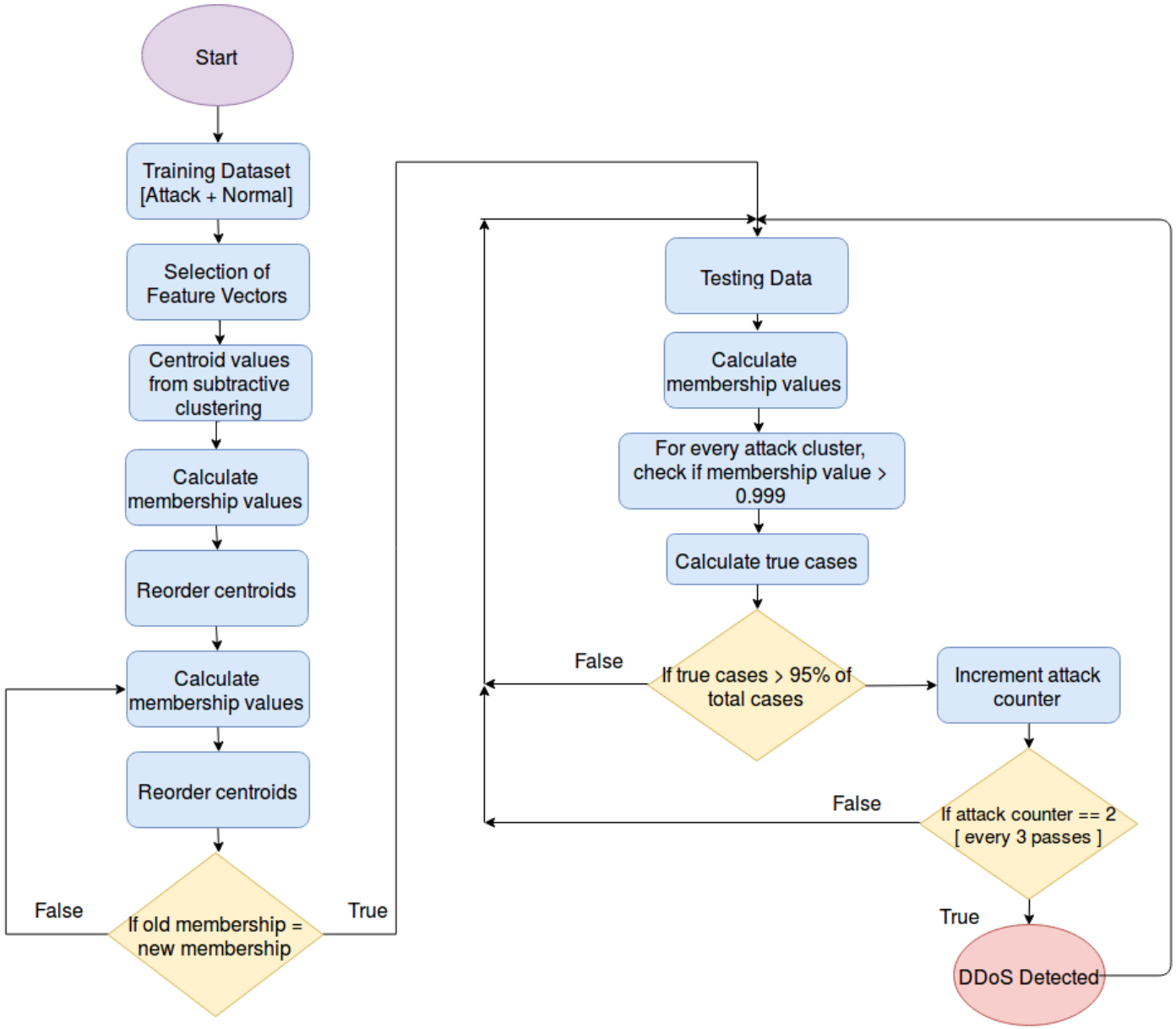


Figure 8: Flowchart: Fuzzy C Means Clustering

Software Specifications

- Linux based Operating System.
- Open vSwitch (OVS).
- Oracle VirtualBox.
- POX Controller.
- Python 2.7 or above.
- Mininet 2.2.1
- Flask
- Numpy, Pandas [Data analysis tools]
- tshark [CLI version of wireshark]
- Elasticsearch, Logstash, Kibana(ELK Stack)
- Watcher or ElastAlert
- sFlow-RT and hsflowd

Hardware Specifications

- Any Enterprise/ Data Center/ Campus Network Topology with 1000+ Mbps.
- Manageable switches that support OpenFlow Protocol / Whitebox Switches.
  - HPE Altoline 6900 48G ONIE AC Switch.
  - Pica8 P-3297 48 X 1Gbe.
  - HP 2920 Switch Series.
- Server Running the Controller
  - Dell PowerEdge R720

Dataset Specifications

- Dataset prepared using packet capturing tool tshark during both training and prediction phases.
- Training datasets include scenarios for both attack as well as normal nature of network.

Time Interval	Src. IP.	Dst. IP	Protocol
0.025412000	192.168.1.11	192.168.1.13	1
0.037555000	192.168.5.1	192.168.5.2	6
0.024478000	192.168.1.11	192.168.1.13	1

Table 1: Normal Traffic Dataset

Time Interval	Src. IP.	Dst. IP	Protocol
0.000001000	192.168.1.14	192.168.1.11	1
0.000002000	192.168.1.14	192.168.1.11	1
0.000001000	192.168.1.14	192.168.1.11	1

Table 2: Attack Traffic Dataset

Test Cases

Id	Description	Scenario	Expected Output
1	Time span between attack detection and alert generation	Attack has occured	Instantaneous alert generation
2	Normal Network Traffic, Log file not altered and attack is not detected	SDN functioning in normal mode	Alert not generated

Scenario: Attack Traffic

Id	Description	Input	Expected Output
1	Simple DoS attack	Large no. of same type of packets	Attack detected
2	UDP flood attack	Large no. of UDP packets	Attack detected
3	Varying DDoS attack bandwidth	Large no. of packets	DDoS attack should be detected only when bandwidth exceeds normal threshold traffic.

Results of SVM based Method

- Normal and attack traffic datasets are generated at run time using tshark.
- Normal to Total Ratio: Total normal classifications / Total number of packets

Traffic Rate (pkts/second)	**SVM Ratio Values**
x	0.33
50x	0.004
200x	0.00002

Table 3: SVM Ratio Values for different traffic rates

Results of Entropy Based Discretization

- Switch configuration: Machine with Ubuntu 14.04, 8GB RAM, 4 logical cores
- Host configuration: Virtual Machine with Ubuntu 14.04, 1.5GB RAM, single core.
- POX controller.
- Flow Table Entries
- Normal traffic Rate: x pkts/second

Traffic Rate (pkts/second)	**Entropy Values**
x	0.854
50x	0.633
200x	0.511

Table 4: Entropy Values for different traffic rates

Results of Fuzzy C-Means Clustering

- Normal and attack traffic datasets are generated at run time using tshark.
- Normal Traffic: x pkts/second
  Normal Centroids: 0.00146, 2.1613
- Attack Traffic: 200x pkts/second
  Attack Centroid: 0.000014

Traffic Rate(pkts/second)	**% attack packets in attack cluster**
x	56%
50x	78%
200x	99%

Table 5: Accuracy with different parameters

Performance Analysis

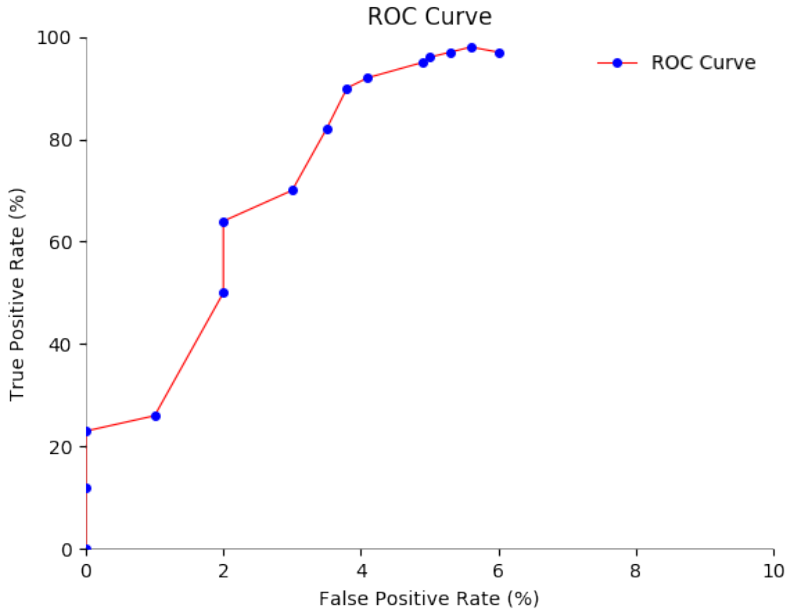


Figure 9: Reciever Operating Characteristic Curve

The performance analysis factors for the system are:

- Accuracy: 95.33%
- Detection Rate: 97.43%
- False Positive Rate: 6.9%

Conclusion

- Hence, we build a solution which can be used for detecting flooding type of DDoS attacks. For this purpose we use three algorithms operating during runtime that utilize less resources with high accuracy and high detection rate. Thus taking a step towards solving the security issues and accelerating the adoption of SDN.

References

- "DDoS Detection and Analysis in SDN-based Environment Using Support Vector Machine Classifier" - Kokila RT, S. Thamarai Selvi, Kannan Govindarajan - 2014 Sixth International Conference on Advanced Computing(ICoAC) - Department of Computer Technology, Anna University (MIT Campus), Chennai.
- "An Entropy-Based Distributed DDoS Detection Mechanism in Software-Defined Networking" - Rui Wang, Zhiping Jia, Lei Ju - 2015 IEEE Trustcom/BigDataSE/ISPA - School of Computer Science and Technology Shandong University Jinan, China.
- "Software-Defined Networking:The New Norm for Networks and Open Networking Foundation" - Open Networking Foundation - ONF White Paper April 13, 2012.
- "Advances in Fuzzy Clustering and its Applications" - Jose Valente de Oliveira, Witold Pedrycz Wiley 2007
- T.Subbulakshmi , Dr. S. Mercy Shalinie, V.GanapathiSubramanian, K.BalaKrishnan, D. AnandK, K.Kannathal - IEEE-ICoAC 2011 - Department of CSE, TCE Madurai, India.
- "OpenFlow Switch Specification" - Open Networking Foundation - Version 1.3.2 2013.

Thank You. . .

Demo