ISTANBUL TECHNICAL UNIVERSITY ★ INFORMATICS INSTITUTE

# DETECTION OF SOURCES BEING USED ON DDOS ATTACKS

**M.Sc. THESIS**

**Yalda MOTEVAKEL KHOSROSHAHI**

**Department of Applied Informatics**

**Cyber Security Engineering and Cryptography Programme**

**June 2019**

ISTANBUL TECHNICAL UNIVERSITY ★ INFORMATICS INSTITUTE

# DETECTION OF SOURCES BEING USED ON DDOS ATTACKS

M.Sc. THESIS

Yalda MOTEVAKEL KHOSROSHAHI
(707151027)

Department of Applied Informatics

Cyber Security Engineering and Cryptography Programme

Thesis Advisor: Assoc. Prof. Dr. Enver ÖZDEMİR

June 2019

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ BİLİŞİM ENSTİTÜSÜ

# DDOS ATAKLARINDA KULLANILAN KAYNAKLARIN TESPİTİ

YÜKSEK LİSANS TEZİ

Yalda MOTEVAKEL KHOSROSHAHI
(707151027)

BİLİŞİM UYGULAMALARI ANABİLİM DALI

Cyber Security Engineering and Cryptography Pogramme

Tez Danışmanı: Assoc. Prof. Dr. Enver ÖZDEMİR

Haziran 2019

Yalda MOTEVAKEL KHOSROSHAHI, a M.Sc. student of ITU Informatics Institute 707151027 successfully defended the thesis entitled “DETECTION OF SOURCES BEING USED ON DDOS ATTACKS”, which he/she prepared after fulfilling the requirements specified in the associated legislations, before the jury whose signatures are below.

**Thesis Advisor :** **Assoc. Prof. Dr. Enver ÖZDEMİR** ..............................
Istanbul Technical University

**Jury Members :** **Assoc. Prof. Dr. Behçet Uğur TÖREYİN** ..............................
Istanbul Technical University

**Dr. Deniz SARIER** ..............................
TÜBİTAK

**Date of Submission :** **3 May 2019**
**Date of Defense :** **13 June 2019**

## FOREWORD

This thesis is dedicated to my family and specially my spouse, who have always been beside me throughout all seconds of my life.

I would like to express my sincere appreciation to my precious advisor Assoc. Prof. Enver ÖZDEMİR for his patience, technical guidance, support and his warm encouragement that pushes me beyond my boundaries.

June 2019 Yalda MOTEVAKEL KHOSROSHAHI

## TABLE OF CONTENTS

**Page**

## ABBREVIATIONS

| | |
|---|---|
| **DoS** | **:** Denial of Service |
| **DDoS** | **:** Distributed Denial of Service |

## SYMBOLS

| | |
|---|---|
| **H** | **:** Entropy |
| $\eta$ | **:** Normalized Entropy |

## LIST OF TABLES

## LIST OF FIGURES

# DETECTION OF SOURCES BEING USED ON DDOS ATTACKS

## SUMMARY

Distributed Denial of Service (DDoS) attack detection is a challenging topic in cyber defense realm. Detection of this type of attack in the early stages can be beneficial. In this thesis, we propose an entropy-based detection framework by employing Support Vector Machine (SVM) classification method to detect sources being used in DDoS attacks.

This method can prevent Denial of Service (DoS) attack from proceeding in source devices which are involved in a DDoS botnet attack. By intercepting outgoing packets from an Android device, proposed framework extract packet features in a specific time window. Normal and abnormal network behavior of a user will be logged and analyzed using SVM algorithm. The obtained model will be used as a detection system for malicious activities.

# DDOS ATAKLARINDA KULLANILAN KAYNAKLARIN TESPİTİ

## ÖZET

Dağıtık Hizmet Engelleme (DDoS) saldırı tespiti, siber savunma alanında zorlu bir konudur. Bu tür bir saldırının erken aşamalarda tespiti yararlı olabilir. Bu tezde, DDoS ataklarında kullanılan kaynakları tespit etmek için Destek Vektör Makinesi (SVM) sınıflandırma yöntemini kullanarak entropi tabanlı bir algılama çerçevesi öneriyoruz.

Bu yöntem, bir Denial of Service (DoS) saldırısının bir DDoS botnet saldırısında yer alan kaynak cihazlarda ilerlemesini önleyebilir. Bir Android cihazdan giden paketleri arayarak, önerilen çerçeve belirli bir zaman penceresinde paket özelliklerini ayıklamak için önerdi. Bir kullanıcının normal ve anormal ağ davranışları kaydedilir ve SVM algoritması kullanılarak analiz edilir. Elde edilen model, zararlı faaliyetler için bir tespit sistemi olarak kullanılacaktır.

DDoS saldırısı, karlılıkları hizmet kullanılabilirliğine bağlı olan şirketler için bir tehlike türüdür. DDoS'un tespiti son 20 yılda bilgi güvenliği için ciddi bir konu haline geldi. DDoS, sunucuların kaynaklarını tüketmek için Internet'in protokollerinin zayıflığını kullanır ve böylece meşru kullanıcılara hizmet etmelerini önler.

Sunucular tarafında birçok izinsiz giriş tespit yöntemi uygulanmaktadır, ancak bu prosedür zaman alıcı ve pahalıdır ve meşru kullanıcılara cevap olarak gecikmeye neden olabilir. Bir DDoS botnet'inde, bir usta zombi cihazlarına aynı anda hareket etmelerini ve hedefe kötü niyetli paketler göndermelerini emrederek saldırıyı düzenler. Normal erişimin sunucu tarafındaki anormal durumdan ayırt edilmesi, gelen paketlerin çok fazla olması nedeniyle etkili olmayabilir.

Botnet uygulaması, birkaç cihazı tehlikeye atarak bir saldırı ağı oluşturur. Ajanların çoğu, uygun İzinsiz Giriş Tespit Sistemleri (IDS) olmamasından dolayı DDoS saldırısına katılmasının farkında olmayabilir. Bu çalışmanın amacı kaynak cihazlarda DDoS faaliyetlerini tespit etmektir. Uç nokta sistemlerindeki anormal faaliyetler ağ davranışları analiz edilerek tespit edilebilir.

Tespit teknikleri, uygulama maliyetine ve tespit hızına bağlı olarak değişebilir. DDoS saldırıları, mağdurlara makine kaynaklarını kısa sürede tüketebilir. Böylece, anında algılama, mağdurun tıkanmasını önleyebilir. İşleme aşaması, saldırının ilerlemesini önleyecek kadar hızlı olmalıdır. Bu faktör, tespit prosedüründe kullanılacak üstün bir parametreye ihtiyaç duyar. Bu tez, bir sınıflandırma modeli elde etmek için makine öğrenme yöntemlerini kullanan bir entropi tabanlı algılama çerçevesi önermektedir.

Bu tezin amacı, DDoS saldırılarında kullanılan kaynak cihazları tespit etmektir. Bu yöntem, DDoS saldırılarına bağlı anomalileri tespit etmek için uç nokta sistemlerinde uygulanabilir. Botnet'lerde, bir botmaster savunmasız cihazları bulmak için belirli bir ağı tarar ve bunlardan ödün verir. Seçilen saldırı stratejisine dayanarak, köle cihazları bir hedefe yönelik koordineli bir DDoS saldırısına katılır. Cihazlardaki anormal aktiviteleri tespit etmek için, denetimli bir makine öğrenme modeli kullanarak entropi tabanlı bir tespit yöntemi öneriyoruz. Buna göre, önerilen sistem anormal davranışı tanımlamak için bir cihazdan giden ağ trafiğini analiz eder.

Bu tezde Android cihazlarda TCP SYN saldırı tespiti ele alınacaktır. Bununla birlikte, aynı yöntem UDP sel ve ICMP sel için de optimize edilebilir. TCP taşkınında bir saldırgan, belirli bir kurbana sürekli olarak önemli miktarda TCP SYN paketi gönderir. Hedef sunucu, istemciye SYN + ACK paketi göndererek yanıt verir ve son el sıkışması olan ACK paketini bekler. Yarı açık bağlantılar sonunda sunucunun kaynaklarını tüketecektir. Üç ana DDoS saldırı senaryosu vardır: Sabit hız saldırısı, artan hız saldırısı ve pulsing saldırısı.

Bir cihazın ağ faaliyetlerini analiz etmek için bir Android cihazın ağ paketlerinin yakalanması gerekir. Bu tür cihazlarda yüksek ayrıcalık erişiminin yetersiz olması nedeniyle, paketler doğrudan ağ arayüzünden yakalanamadı. Bu nedenle, istemci cihazında bir Sanal Özel Ağ (VPN) kurulması gerekir. Paketler VPN sunucusuna yönlendirilirken, paket başlıkları yakalanabilir. VPN sunucusu, istemci portundan gelen paketleri sunucunun ağ arayüzüne gönderir ve cevabı istemciye yönlendirir. Bu şekilde, müşterinin trafik bilgisine hem müşterinin hem de sunucunun tarafında erişilebilir. Android kullanıcısı, bir kullanıcı adı ve kayıt aşamasında verilen bir şifre kullanılarak VPN programına giriş yapacaktır.

VPN sunucusuna bir bağlantı kurulduktan sonra, saldırı tespit prosedüründe pratik olan bazı paket başlık alanlarından bazıları aşağıdaki şekilde kaydedilecektir: Günün zaman damgası (saniye olarak), Hedef IP, Hedef Bağlantı Noktası, İletim Protokolü ve Bayrak, Paket Uzunluğu . Göz önünde bulundurulması gereken bir nokta, kaynak IP yerine, verilen kullanıcı adının cihazın log tanımlayıcısı olarak kaydedileceğidir.

Ağ paketlerinin tek tek analiz edilmesi, fazla miktarda trafik çekmesi nedeniyle mümkün değildir. Bu nedenle, özellik çıkarma modülü paketleri $T$ sabit bir zaman penceresi kullanarak toplar. Bir TCP saldırısını algılamak için, paketler protokol türlerine ve bayraklarına göre kategorilere ayrılır. Paket başlık alanlarının entropisi, cihazların ağ davranışının analizinde yardımcı olabilir.

Dağılımların rastgelelik derecesi, entropi, özellik seçimi bileşeni için ölçüm ölçüsü olarak kullanılacaktır. Paketlerin sayısı bir zaman penceresinde değişebildiğinden, normalize edilmiş entropinin karşılaştırılabilir bir metriğe sahip olması için kullanılması gerekir.

Hedef IP'nin normalleştirilmiş entropisi ve seçilen paketlerin portu hesaplanabilir. Ayrıca, SYN paketlerinin iletim hızı $N/T$ olarak hesaplanacaktır. Bu üç parametre bir android cihazda bir ağın normal ve anormal aktivite profilini belirlemek için kullanılacaktır. Tüm SYN paketleri için paket uzunluğunun eşit olduğunu ve standart sapmanın sıfır olacağını belirtmekte fayda var, bu nedenle bu özellik analiz bileşeninde kullanılacak ayırıcı bir parametre olamaz. Bu özellik değerleriyle birlikte en çok istenen hedef IP adresi, port numarası ve oluşum yüzdesi kaydedilecektir. Bu bilgileri kullanarak, hangi IP adreslerinin DDoS saldırısı altında olduğunu onaylayabiliriz.

DoS saldırı davranışına dayanarak, entropi değerleri spesifik olabilir. DoS saldırısının yaygın bir senaryosu olarak, saldırgan sabit bir hedef IP ve port numarasına odaklanır ve TCP paketlerini belirli bir oranda gönderir. Saldırı sırasında, hedef IP ve port numarasının normalleştirilmiş entropisi için beklenen değerler düşük olacak ve iletim hızı özellik değeri daha yüksek olacaktır. Ancak, normal trafik için bu değerler değişebilir ve algılama için tam eşiği bilmek pratik değildir. Bu nedenle, normal ve saldırı verilerini sınıflandırmak için bir makine öğrenme yöntemi uygulanmalıdır.

"Makine Öğrenmesi" bileşeninde, kullanıcının normal ve anormal davranış modelini elde etmemiz gerekir. Özellik çıkarma modülünden elde edilen verileri kullanarak sistemi eğitiyoruz. Bir Destek Modeli oluşturmak için Destek Vektör Makinesi (SVM) kullanılır. SVM, regresyon ve sınıflandırma problemleri için kullanılabilecek denetimli bir algoritmadır.

Sınıflandırılması gereken sadece iki küme vardır: normal ve saldırı. Makine öğrenme modelimizi eğitmek için, kullanıcının davranışını temel alan eğitim veri setini ve etiket girdilerini ikili biçimde hazırlamamız gerekir.

Bir Android kullanıcısının normal ağ etkinlikleri yakalandı ve bu süre zarfında devam eden bir saldırı olmamasını sağladı. Toplanan normal verilerde özellik çıkarma işleminden sonra, girişler normal olan, sıfır olan etiketlenir.

Kötü amaçlı paketler kullanıcının ağına enjekte edilmiştir ve özellik seti edinilecektir. Özellik değerleri, bir saldırı olarak etiketlenecektir.

Test veri seti ayrıca doğru etiketleme ile eğitim veri seti ile aynı şekilde toplanabilir ve modeli test etmek için kullanılabilir.

Ağ paketlerini yakalamak için sırasıyla Java ve C++ 'da bir Android VPN istemci uygulaması ve bir VPN sunucu programı uygulanmaktadır. Android uygulaması, cihaz ile VPN Sunucusu arasında Sanal Özel Ağ oluşturmak için VPN servisini kullanır. Diğer tarafta, VPN sunucusu gelen paketleri mobil cihazdan ağ arayüzüne yönlendirir ve yanıtı istemciye gönderir. Bu arada, paketler sunucudan ayrılmadan önce yakalanabilir ve veritabanı sunucusunun dosya sistemine kaydedilebilir.

Bir Android cihazının ağ etkinliklerini yakalamak için, bir kullanıcı belirtilen bir IP adresini, port numarasını ve giriş bilgilerini kullanarak VPN programına giriş yapmalıdır. Kullanıcı İnternette gezinmeye başladığında, Android cihazındaki VPN portundan giden ağ paketlerinin IP başlığı bilgileri yakalanacaktır. VPN sunucusundaki müşteri portundan gelen paketlerin başlık bilgisi günlük olarak her kullanıcı için virgülle ayrılmış bir dosyaya kaydedilir.

indent DoS saldırı davranışını simüle etmek için, saldırı senaryolarını uygulayan "Paket Gönderen" adlı bir Android uygulaması oluşturduk. Hedef IP adresini ve port numarasını, saldırı süresini, türünü (sabit, artan, pulsing) ve giriş olarak değerlendiriyor. TCP SYN paketlerini, seçilen senaryoda göre $[25,50]$ pps aralığında değişebilen, seçilen orandaki hedefe sürekli gönderir. Paketler yakalanırken, kullanıcı paket gönderen programını açar ve belirli bir IP ve port numarası seçerek, kurbana, belirlenen saldırı oranına göre seçilen senaryoya göre saldırır.

Özellik çıkarma modülü Python'da uygulanır. Hedef bir IP ve port entropisini ve belirli bir zaman aralığında iletim hızıyla birlikte hesaplar. Pencerenin özellik değerleri ve en çok istenen hedef bilgileri eğitim ve test dosyalarına yazılacaktır. Bu testte, zaman penceresi değeri göz önünde bulundurulur, $T = 2$ saniye. Ayarlanan özellik değerleri, "Paket Gönderen" uygulamasında en çok talep edilen IP seçilen hedef IP ise saldırı olarak etiketlenir. Saldırı simülasyonunda en çok talep edilen IP hedef IP değilse, diğer özellik değeri kümeleri normal olarak adlandırılır.

Machine Learning modülü ayrıca Python'da Spyder yazılımı kullanılarak açık bir şekilde tanıtılmak üzere yazılmıştır. Sistemi modellemek için Scikit-learn kütüphanesinin SVM yöntemini kullanır. Doğrusal Çekirdek Destek Vektör Sınıflandırması (SVC) Algoritması, sınıflandırma metodu olarak kullanılır. Program, eğitim verilerine göre SVM Modeline uyacaktır.

Eğitim verilerini sınıflandırmak için iki yaklaşım kullanılmaktadır. İlk senaryoda, verileri sınıflandırmak için veri kümesinin üç özelliğini kullanır. İkinci yaklaşım olarak, özellik sayısını ikiye indirmek için Temel Bileşen Algoritmasını (PCA) kullanır.

## 1. INTRODUCTION

DDoS attack is a type of threat for companies which their profitability depends on their service availability. Detection of DDoS has become a serious topic for information security in the last two decades. DDoS uses the weakness of Internet's protocols to exhaust servers' resources, thus preventing them from serving legitimate users.

Many intrusion detection methods applied in the servers side, but this procedure is time-consuming and expensive and can cause a delay in response to legitimate users. In a DDoS botnet, a master orchestrates the attack by commanding zombie devices to act simultaneously and send malicious packets to the target. Distinguishing the normal access from abnormal in the server side might not be effective due to enormous amount of incoming packets.

Botnet application creates an attack network by compromising several devices. Most of the agents may not be aware of their involvement in the DDoS attack due to lack of proper Intrusion Detection Systems (IDS). The objective of this work is to detect DDoS activities in source devices. Abnormal activities in endpoint systems can be detected by analyzing their network behavior.

Detection techniques can be vary based on the cost of implementation and detection speed. DDoS attacks can exhaust victims machine resources in a short amount of time. So, instant detection can save the victim from clogging. The processing phase should be fast enough to prevent the attack from proceeding. This factor needs a superior parameter to be used in the detection procedure. This thesis proposes an entropy-based detection framework using machine learning methods to obtain a classification model. As structure of thesis, we first introduce related works in the next chapter. Then, in Chapter 3, the proposed attack detection framework process is discussed. In Chapter 4, we give the implementation steps of the proposed framework and results. Finally, concluding remarks are shown and possible future work is presented in Chapter 5.

# 2. RELATED WORK

## 2.1 DoS/DDoS Attacks

A denial of service attack is a type of an attack which prevents the legitimate use of a service [1]. The most frequent DDoS attack is that attackers send a huge amount of packets to a victim server and exhaust it's resources and make it unavailable to legitimate users. The Internet architecture weakness is the reason of DDoS attacks. First of all, the security of the victim depends on the Internet security. Briefly, DDoS can be started with exploiting agent computers by scanning for vulnerable devices in network. Then the attacker injects virus to the agents and force to do actions on a victim machine. There are too many ways to exploit the agent machines e.g. using social engineering techniques like phishing email with malicious attachment. There are totally fours phases in DDoS attacks: Scan, exploit, infect and launch. DDoS attack motivations frequently are personal reasons or reputation. However, sometimes it can be driven by political reasons.

## 2.2 Botnet

Article [2] has a survey on botnet applications and classifies botnet detection techniques. A botnet is a network of bot computers which are compromised by the bot master. A bot is self spreading virus that infects vulnerable devices to create a botnet. These networks employ command and control (C&C) channels with different communication protocols which is classified as:

- IRC-based
- HTTP-based
- DNS-based
- Peer to Peer (P2P)

### 2.2.1 Botnet life cycle

The botnet application phases are: infection, injection, connection, command and update. In first phase, the attacker, scans a target network to find security vulnerabilities. In second phase it exploits vulnerable hosts. Therefore, the infected host executes a shell code according to its operation system (OS) which downloads the bot program. The host turns into a Zombie device and waits for commands from the bot master.

In next phase, connection, the bot virus creates a C&C channel and connects the infected host to the server. Now, The bot master can remotely control the zombie computer. In final phase, bot are ordered to download and update the bot virus in order to avoid getting detected by anti viruses.

### 2.2.2 Botnet detection

To analyze the Botnet technology, a Honeynet project can be used. As another solution to detect botnet, passive network analysis can be employed. These methods are classified as:

- Signature-based
- Anomaly-based
- DNS-based
- Mining-based

In signature-based techniques, with a set of rules on the network traffic, botnets can be detected. This method will be unable to detect new botnet signatures. Anomaly based detection techniques is based on abnormal network activities such as: high volumes of traffic, uncommon access to ports and etc. This method can be used to detect unknown botnets. In order to communicate with zombie devices, botnet master using special DNS information. These queries can be used to detect a botnet network using DNS-based detection methods. Machine learning methods can be used as the mining-based detection. In this approach, captured network traffic will be classified to detect botnet activities.

## 2.3 DDoS Classification

There are several DDoS detection methods which are topics of various research. These mechanisms can be classified into different categories: activity level, cooperation degree, and deployment location. DDoS attacks can be prevented or detected based on the chosen activity level. Detection methods can be autonomous or in a cooperative way. DDoS attacks can also be detected in the victim network, intermediate network and source network [1]. DDoS attack mechanisms can be classified into eight categories.

- Automation Degree
- Weakness exploiting
- Validation od source address
- Characterization
- Attack rate statistics
- Victim’s impact
- Target type
- Agent set persistence

DDoS attack defense mechanisms is classified as:

- Activity level
    - Prevention goal
    - Prevention method
    - Detection strategy
        - Pattern
        - Anomaly
    - Response strategy
- Deployment Location
- Cooperation degree

## 2.4 Source Detection Methods

Due to the effect of the attack on a victim, most service providers are more motivated to implement security defense systems on their side. The intermediate defense can be provided by the infrastructure service providers upon the request. The deployment cost in the victim and intermediate network is high and the attack should be detected as soon as possible without causing a delay in the victim's service [3]. Therefore, it can be beneficial to detect DDoS attacks in the early phase at source network to prevent the attack from proceeding.

As a source detection algorithm, MULTOPS [4] proposes a data-structure detection method based on packet rate statistics. The mechanism can either establish in the victim, or the source network to examine packets. This system can be implemented in router devices.

D-WARD [5] proposes a defense mechanism deployed in a source network and detect attacks by monitoring of incoming and outgoing network traffic. It detects the attack data by comparing to normal traffic model and the flow statistics. The article simulates different attack scenarios injected into legitimate traffic to evaluate the system's detection ability.

The detection systems intercept network traffic to find an abnormal behavior. They can be signature-based or anomaly-based methods. DDoS attacks can be detected based on their signature and rule sets. However, botnet programs update their code eventually, so signature-based methods cannot detect new attacks [2]. Anomaly detection methods can detect an abnormal behavior in comparison to the normal profile which is obtained by observing the normal activities of the network. Some machine learning methods are employed by articles to obtain detection model [6].

Keunsoo Lee [7] proposes a detection method for earlier stage of DDoS attack as well as the attack itself in DMZ network. The system selects detection parameters based on DDoS behavior and employs clustering for attack detection. It computes the entropy of packet header fields on a sample of sequential packets. It classifies clusters by the distance measures of variables using Euclidean distance.

Article [8] proposes a novel approach based on entropy for DDoS attack detection by modeling normal patterns of netwrok flows using cluster analysis methods. This system trains the model with normal traffic and obtains a detection threshold for comparison. After clustering, the system updates its model to have more accurate detection.

The proposed method in [9] detects DDoS attacks by using entropy and chi-square statistics of selected packet fields. This prototype is implemented in Snort router and it is employed to detect anomalies based on threshold comparison.

A DDoS detection model is proposed by [10] based on multiple SVM (Support Vector Machine) and TRA (Traffic Rate Analysis). In the pre-processing stage, it extracts the features from captured network traffic using TRA. In the second stage, the method trains the model using normal and attack training data and classifies the testing data.

Article [11] proposes DDoS detection approach using conditional entropy of statistical features and support vector machine (SVM) classifier. It computes the three conditional entropy of source IP and destination port. These values are used as detection metrics for SVM algorithm and employed to detect DDoS activity.

The reviewed papers implement their detection system model in routers of the network. However, detection modules can interfere with routing functions of the router and cause a delay in forwarding network packets. Hence, DDoS activities should be detected in source devices to prevent coordinated attacks against a victim server. In this thesis, we propose an entropy-based detection framework to identify DoS activities in source devices using SVM. In other terms, the suggested method will detect if an endpoint device is part of a botnet or not.

# 3. PROPOSED FRAMEWORK

The objective of this thesis is to detect source devices being used in DDoS attacks. This method can be implemented in endpoint systems to detect anomalies related to DDoS attacks. In botnets, a botmaster scans a specific network to find vulnerable devices and compromises them. Based on the chosen attack strategy, slave devices participate in a coordinated DDoS attack to a target. In order to detect anomalous activities in devices, we propose an entropy-based detection method using a supervised machine learning model. Accordingly, the proposed system analyzes outgoing network traffic from a device to identify abnormal behavior.

In this thesis TCP flood attack detection in Android devices will be discussed. However, the same method can be optimized for UDP flood and ICMP flood.

## 3.1 TCP Flood

In TCP flood, an attacker continuously sends a significant amount of TCP SYN packets to a specified victim. The target server responses to the client by sending SYN+ACK packet and waits for the final handshake, ACK packet. The half-open connections eventually will exhaust the server's resources. There are three main DDoS attack scenarios [12]:

### 3.1.1 Constant rate attack

The attacker requests a TCP connection to a victim's IP address and the port number by sending SYN packet in a constant rate, e.g. 50 packets per second (pps). This attack can last for a fixed period of time and be stopped by the master's command.

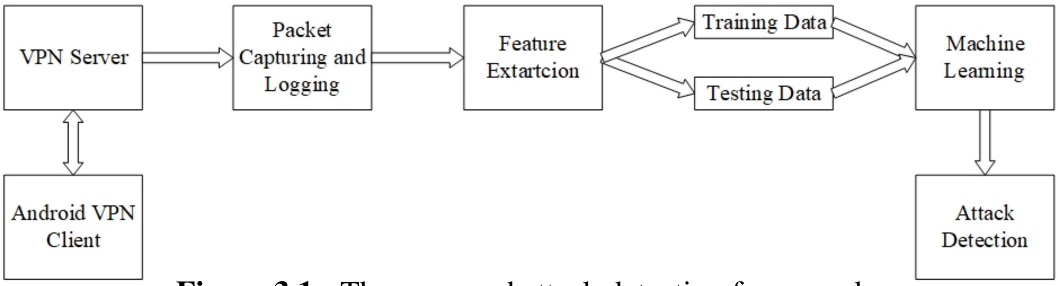


**Figure 3.1** : The proposed attack detection framework.

### 3.1.2 Increasing rate attack

The compromised device attacks to a victim by constantly sending SYN packets to an IP address and a port number with an increasing rate. The attack starts with the lowest rate and continues to proceed gradually over time until it reaches the highest rate.

### 3.1.3 Pulsing attack

In this scenario, the attacker sends TCP SYN packets to a victim in a constant rate in a pulsing manner for a known amount of time. The active and inactive time is equal in this approach.

The suggested framework will attempt to detect possible attack scenarios in source devices by components shown in Fig. 3.1. Briefly, it captures outgoing Internet traffic of an Android device and logs the packet header information. Then, the machine learning component uses training data from the feature extraction module to train its detection model. Finally, testing data will be classified to show the accuracy of the method. We set up a simulation test-bed environment which is shown in Fig. 3.2 to implement the proposed framework.

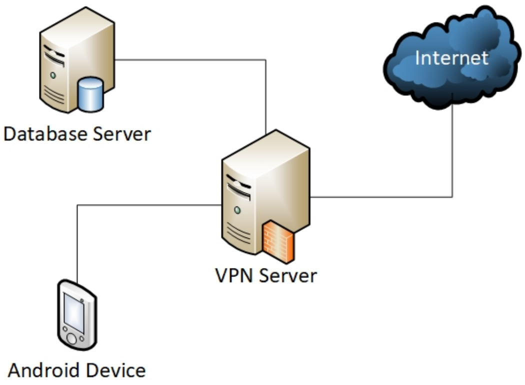


**Figure 3.2** : Simulation environment.

### 3.2 Packet Capturing

In order to analyze network activities of a device, network packets of an Android device should be captured. Due to lack of high privilege access in these types of devices, packets couldn't be captured directly from its network interface. Hence, A Virtual Private Network (VPN) needs to be set up in the client device. While packets are being redirected to VPN server, the packet headers can be captured. VPN server sends incoming packets from the client port to the server's network interface and redirects the response to the client. In this way, the client's traffic information can be accessible in both client and server's sides. The Android user will be logged into the VPN program using a username and a password which are given in the registration phase.
After establishing a connection to the VPN server, some of the packet header fields which are practical in attack detection procedure will be logged as follows:

- Timestamp of the day (in seconds)
- Destination IP
- Destination Port
- Transmission Protocol and Flag
- Packet Length

A related point to consider is that instead of source IP, the given username will be saved as the device's log identifier.

### 3.3 Feature Extraction

Analyzing network packets individually is not feasible due to the high amount of outgoing traffic. Therefore, feature extraction module aggregates packets using a fixed time window, $T$. To detect a TCP attack, the packets are categorized by their protocol type and flags. The entropy of packet header fields can assist in the analysis of devices network behavior.

### 3.3.1 Shannon entropy

Degree of the randomness of distributions, entropy will be employed as the measurement metric for the feature selection component. The entropy $H$ is defined as [13]:

$$H(X) = -\sum_{i=1}^{N} p_i(X)\log_2 p_i(X) \tag{3.1}$$

where $N$ is the number of packets in a window and $p_i(X)$ represents the probability of the particular feature of $X$. As the entropy gets closer to its minimum value which is zero, it means that there is less discrepancy in sample data. In contrast, when it has the maximum value which is $\log_2 N$, it means that the data is in maximum uncertainty.

### 3.3.2 Normalized entropy

Since the number of packets can vary in a time window, the normalized entropy should be employed to have a comparable metric which is defined as follows:

$$\eta(X) = -\sum_{i=1}^{N} \frac{p_i(X)\log_2 p_i(X)}{\log_2 N} \tag{3.2}$$

In this manner, the $\eta(X)$ of a specific feature will be in the range of $[0,1]$. By using logged packets info, there will be three selected features as follows:

- $\eta$(Destination IP)
- $\eta$(Destination Port)
- Transmission Rate

The normalized entropy of the destination IP and port of selected packets can be computed. Also, the transmission rate of the SYN packets will be calculated as $N/T$. These three parameters will be used to determine a network's normal and abnormal activity profile in an android device. It's worth mentioning that packet length for all the SYN packets is equal and the standard deviation will be zero, so this feature can not be a discriminant parameter to be employed in analysis component. Along with these feature values, the most requested destination IP address, port number and the occurrence percentage will be saved. Using these information we can confirm that which IP address is under DDoS attack. Also, they will be used in labeling process which will be discussed in later sections.

Based on DoS attack behavior, entropy values can be specific. As a common scenario of DoS attack, an attacker focuses on a fixed target IP and port number and sends TCP packets in a specific rate. During the attack, the expected values for normalized entropy of destination IP and port number will be lower and transmission rate feature value will be higher. However, these values for normal traffic can vary and it's impractical to know the exact threshold for detection. Therefore, a machine learning method should be applied to classify normal and attack data.

## 3.4 Machine Learning

We need to obtain a model of normal and abnormal behavior of the user. We train the system using the obtained data from the feature extraction module. The Support Vector Machine (SVM) is employed to create a detection model. SVM is a supervised algorithm which can be used for regression and classification problems [14].

There are only two clusters that need to be classified: normal and attack. In order to train our machine learning model, we need to prepare the training dataset and label inputs based on the user's behavior in binary format.

### 3.4.1 Normal dataset

Normal network activities of an Android user has been captured and ensured that there is no ongoing attack on that time. After feature extraction process on collected normal data, the inputs will be labeled as normal which is zero.

### 3.4.2 Attack dataset

The malicious packets has injected to the user's network and the feature set will be acquired. The features values will be labeled as the attack which is one.

The testing dataset can also be collected in the same manner as the training dataset with correct labeling and be used to test the model. The implementation of the proposed framework and detailed data labeling process will be discussed in the next chapter.

# 4. IMPLEMENTATION AND RESULTS

An Android VPN client application and a VPN server program are implemented in Java and C++, respectively to capture network packets. The Android application uses VPN service to establish the Virtual Private Network between the device and the VPN Server. On the other side, the VPN server redirects incoming packets from the mobile device to the network interface and sends the response to the client. Meanwhile, the packets can be captured before leaving the server and saved in the database server's file system.

## 4.1 Data Collection

To capture network activities of an Android device, a user should log into the VPN program using a specified IP address, port number ,and login information. As the user starts browsing on the Internet, the IP header information of outgoing network packets from the VPN port in the Android device will be captured. The header information of incoming packets from the client port in VPN server will be logged in a comma separated file for each user on a daily basis.

In order to simulate DoS attack behavior, we created an Android application called "Packet Sender" which implements the attack scenarios discussed in Chapter 3. It takes Target IP address and port number, attack duration, type(constant, increasing, pulsing) and rate as the input. It continuously sends TCP SYN packets to the target in selected rate which can vary in the range of $[25, 50]$ pps based on the chosen scenario. While the packets are being captured, the user opens the packet sender program and selects a specific IP and a port number and attacks the victim based on the chosen scenario with specified attack rate.

Feature extraction module is implemented in Python. It computes the entropy of destination IP and port along with the transmission rate within a specific time window. The feature values of the window and most requested destination information will be written in training and testing files. In this test, the time window value is considered,

$T = 2$ seconds. The feature values set will be labeled as attack if its most requested IP is the selected target IP in "Packet Sender" application. The other feature value sets are called normal if their most requested IP is not the target IP in the attack simulation.

## 4.2 Attack Detection

Machine Learning module is also written in Python using Spyder software to have a clear demonstration. It uses the SVM method of Scikit-learn library [15] for modeling the system. The Linear Kernel Support Vector Classification (SVC) Algorithm is employed as the classifier method. The program will fit the SVM Model according to the training data.

Two approaches are employed in order to classify the training data. In the first scenario, it uses the three features of the dataset to classify the data. As the second approach, it uses Principal Component Algorithm (PCA) [16] to reduce the number of features to two.

### 4.2.1 Three feature

All three features of the training dataset, $\eta$(Destination IP), $\eta$(Destination Port) and transmission rate are used as the input features and classified by Linear SVC algorithm. It predicts the labels for the testing dataset attributes.

Fig. 4.1 depicts the three-dimensional plot of the result in the classification process. The result shows the decision hyperplane which classifies normal and the attack traffic of the monitored device. According to the classifier, DoS activities of an Android user can be detected in source end.

### 4.2.2 Two features

PCA is applied to training and testing dataset and the number of features reduced to two. The two components, $PC1$ and $PC2$ will fit the Linear SVC classifier. The decision region boundary of the classified testing dataset is demonstrated in Fig. 4.2. The right side of the boundary line indicates attack instances and left of the line shows normal data. Hence, based on the decision line, network packets in a time window can be classified.

For the purpose of the model validation, we evaluate the performance of the classifier

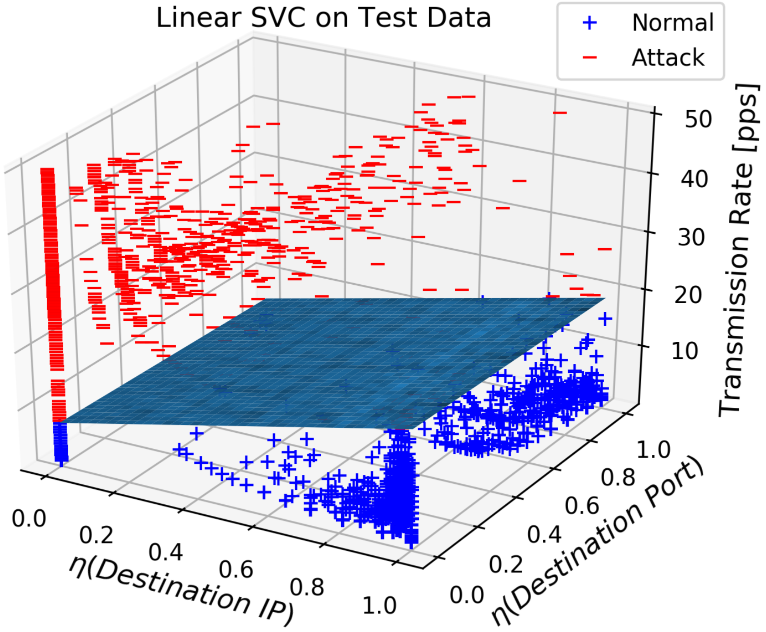


**Figure 4.1** : SVM on three features.

using a confusion matrix for testing data. The detection accuracy, precision and error rates are demonstrated in Table 4.1.

**Table 4.1** : Detection rates.

| Model | Accuracy Rate | Precision Rate | Error Rate |
|---|---|---|---|
| 3D SVM | 0.985 | 0.971 | 0.014 |
| 2D SVM | 0.988 | 0.984 | 0.011 |

We also examine the proposed framework employing "k-fold cross validation". Accuracy of the SVM algorithm is compared to Logistic Regression (LR), Linear Discriminant Analysis (LDA), K-Nearest Neighbors (KNN), Decision Tree Classifier (CART) and Gaussian Naive Bayes (NB) algorithms. The box and whisker plot of algorithms comparison is illustrated in Fig. 4.3. The mean and standard variations of accuracy rate of the algorithms can be seen in Table 4.2 for 10-fold cross validation.

In order to examine the reliability of SVM algorithm, accuracy of SVM algorithm with the linear kernel is compared to Radial Basis Function (RBF), degree two polynomial (d=2) and sigmoid kernels. The box and whisker plot of SVM kernels comparison is shown in Fig. 4.4. The mean and standard variations of the accuracy rates of the kernels can be seen in Table 4.3 for 10-fold cross validation.

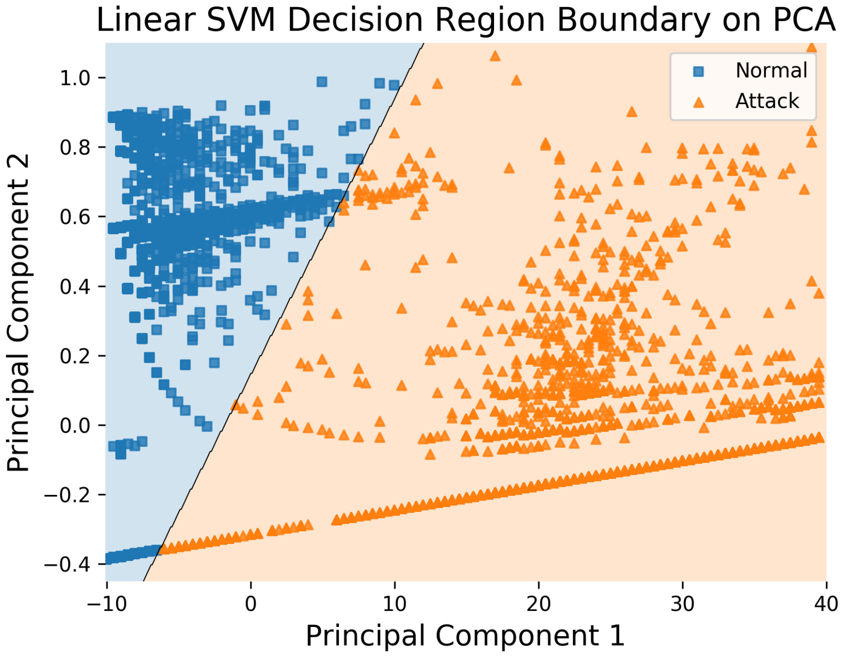


**Figure 4.2** : Two component PCA.

**Table 4.2** : Algorithms comparison results.

| Algorithm | Mean | Standard deviation |
|---|---|---|
| SVM | 0.989 | 0.014 |
| LR | 0.987 | 0.015 |
| LDA | 0.984 | 0.016 |
| KNN | 0.988 | 0.014 |
| CART | 0.987 | 0.013 |
| NB | 0.982 | 0.017 |

### 4.3 Comparison

MULTOPS [4] and D-WARD [5] propose a DDoS detection method in source network based on the packet rate statistics which should be implemented in source network router devices. Packets rate feature might not be adequate for detecting DDoS activities. Also, the implementation of these techniques in routers may cause problems

**Table 4.3** : SVM comparison results.

| Kernel | Mean | Standard deviation |
|---|---|---|
| Linear | 0.988 | 0.014 |
| RBF | 0.989 | 0.013 |
| Polynomial d=2 | 0.982 | 0.019 |
| Sigmoid | 0.974 | 0.023 |

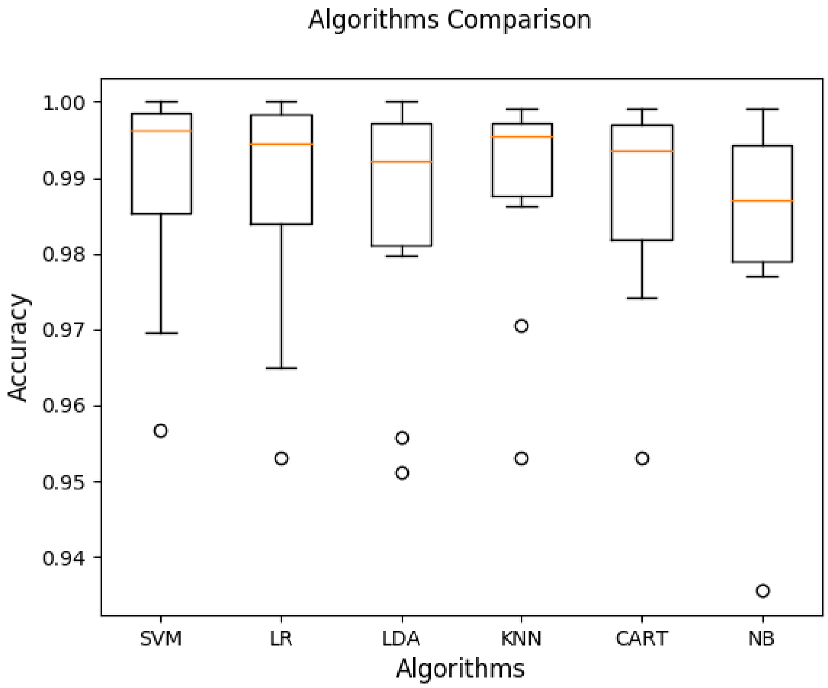


**Figure 4.3** : Machine learning algorithms comparison on PCA.

in routing functions and packet loss. Our proposed framework employs more features and implements an entropy-based detection system using SVM. The method should be implemented in source end devices which have more computation power and resources. Additionally, detecting attacks in the first place is more essential and attacks can be easily blocked without causing a problem in network flows of any other users.

In article [7], the author uses nine features from IP packet headers and clustering analysis to classify different phases of a DDoS attack. Using more features may increase the computational cost of the system and cause a delay in detection. This technique should be implemented in high-level network layer to access to the entire networking activities of a DMZ network. By employing our proposed framework, DDoS activities can be detected in source end devices using fewer features. Thereby, with a small amount of computation, attacks can be prevented in the first place.

Unlike article [8] which uses the entropy of network flows, we use normalized entropy of aggregated packets IP header information in a specified time window. It detects attacks using threshold comparison, while our proposed framework uses a machine learning method. It's impractical to select a proper threshold value for features due to the various DDoS scenarios. Hence, utilizing machine learning methods in DDoS detection can automatically find the classifier line between normal and attack classes.

Proposed algorithm in [11] uses source IP conditional entropy and SVM classifier to

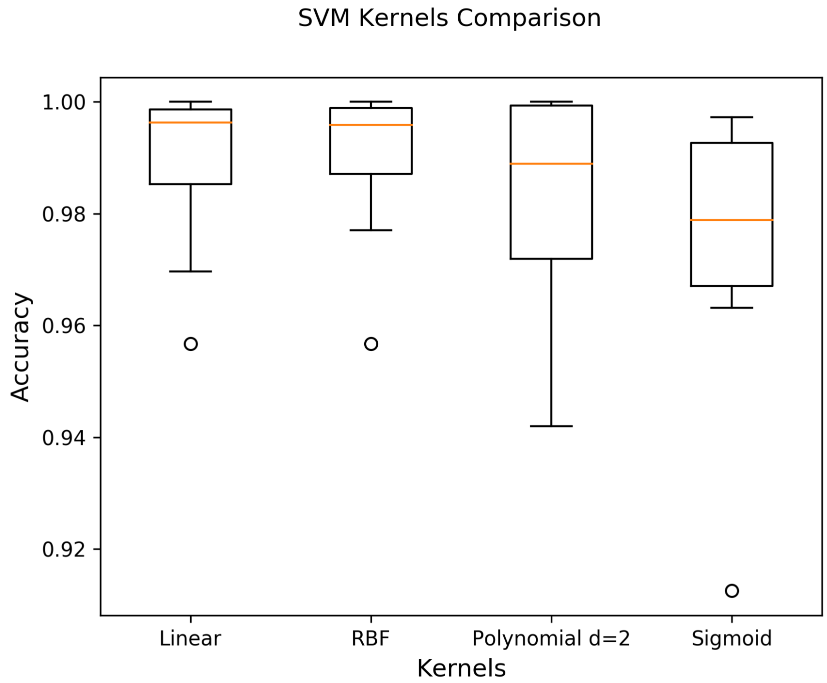


**Figure 4.4** : SVM kernels comparison on PCA.

detect DDoS activities in a network. This method may not be effective in the case of source IP spoofing. Our proposed method is implemented in source end devices and only destination information would be enough to detect DoS activities.

The related works which are discussed in this work mostly propose DDoS attack detection methods based on collected network data from a network. Accordingly, more features need to be used as detection metrics and it might take longer processing time to detect an attack. However, our proposed method can detect DDoS activity in the early stage with fewer features and prevent the attack from proceeding.

## 5. CONCLUSION

In this thesis, we proposed an entropy-based detection framework using the SVM algorithm for detecting the sources being used in DDoS attacks. The network activities of an Android user are captured via VPN service. The entropy of IP header fields are calculated and the feature value set is prepared.

Using the extracted features, we train the model to classify normal and attack network data of the user using SVM machine learning method with two and three feature. The model is tested on the testing dataset to test the performance of the classifier. The model succeeds in detecting abnormal activity in the Android device with an accuracy of 0.98.

As for the future work, the proposed framework will be implemented to detect and prevent DoS activities in real time. The framework will be tested on all of the DDoS attack types including UDP, ICMP and etc.

# REFERENCES

[1] **Mirkovic, J. and Reiher, P.** (2004). A taxonomy of DDoS attack and DDoS defense mechanisms, *ACM SIGCOMM Computer Communication Review*, *34*(2), 39–53.

[2] **Feily, M.**, **Shahrestani, A. and Ramadass, S.** (2009). A survey of botnet and botnet detection, *2009 Third International Conference on Emerging Security Information, Systems and Technologies*, IEEE, pp.268–273.

[3] **Douligeris, C. and Mitrokotsa, A.** (2004). DDoS attacks and defense mechanisms: classification and state-of-the-art, *Computer Networks*, *44*(5), 643–666.

[4] **Gil, T.M. and Poletto, M.** (2001). MULTOPS: A Data-Structure for Bandwidth Attack Detection., *USENIX Security Symposium*, pp.23–38.

[5] **Mirkovic, J.**, **Prier, G. and Reiher, P.** (2002). Attacking DDoS at the source, *Network Protocols, 2002. Proceedings. 10th IEEE International Conference on*, IEEE, pp.312–321.

[6] **Suresh, M. and Anitha, R.** (2011). Evaluating machine learning algorithms for detecting DDoS attacks, *International Conference on Network Security and Applications*, Springer, pp.441–452.

[7] **Lee, K.**, **Kim, J.**, **Kwon, K.H.**, **Han, Y. and Kim, S.** (2008). DDoS attack detection method using cluster analysis, *Expert systems with applications*, *34*(3), 1659–1665.

[8] **Qin, X.**, **Xu, T. and Wang, C.** (2015). DDoS attack detection using flow entropy and clustering technique, *Computational Intelligence and Security (CIS), 2015 11th International Conference on*, IEEE, pp.412–415.

[9] **Feinstein, L.**, **Schnackenberg, D.**, **Balupari, R. and Kindred, D.** (2003). Statistical approaches to DDoS attack detection and response, *null*, IEEE, p.303.

[10] **Seo, J.**, **Lee, C.**, **Shon, T.**, **Cho, K.H. and Moon, J.** (2005). A new DDoS detection model using multiple SVMs and TRA, *International Conference on Embedded and Ubiquitous Computing*, Springer, pp.976–985.

[11] **Liu, Y.**, **Yin, J.**, **Cheng, J. and Zhang, B.** (2010). Detecting DDoS attacks using conditional entropy, *2010 International Conference on Computer Application and System Modeling (ICCASM 2010)*, volume 13, IEEE, pp.V13–278.

[12] **Yuan, J. and Mills, K.** (2005). Monitoring the macroscopic effect of DDoS flooding attacks, *IEEE Transactions on Dependable and secure computing*, *2*(4), 324–335.

[13] **Shannon, C.E. and Weaver, W.** (1963). The mathematical theory of communication. 1949, *Urbana, IL: University of Illinois Press*.

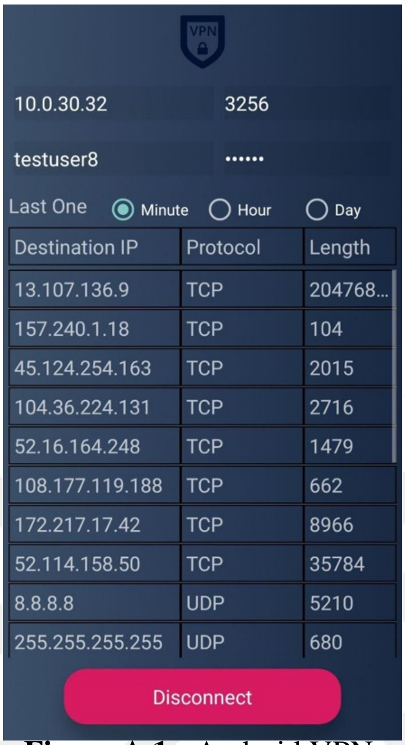
[14] **Gunn, S.R.** *et al.* (1998). Support vector machines for classification and regression, *ISIS technical report*, *14*(1), 5–16.

[15] **Pedregosa, F.**, **Varoquaux, G.**, **Gramfort, A.**, **Michel, V.**, **Thirion, B.**, **Grisel, O.**, **Blondel, M.**, **Prettenhofer, P.**, **Weiss, R.**, **Dubourg, V.** *et al.* (2011). Scikit-learn: Machine learning in Python, *Journal of machine learning research*, *12*(Oct), 2825–2830.

[16] **Wold, S.**, **Esbensen, K. and Geladi, P.** (1987). Principal component analysis, *Chemometrics and intelligent laboratory systems*, *2*(1-3), 37–52.

## APPENDICES

## APPENDIX A.1



10.0.30.32 3256

testuser8 ••••••

Last One ● Minute ○ Hour ○ Day

| Destination IP | Protocol | Length |
|---|---|---|
| 13.107.136.9 | TCP | 204768... |
| 157.240.1.18 | TCP | 104 |
| 45.124.254.163 | TCP | 2015 |
| 104.36.224.131 | TCP | 2716 |
| 52.16.164.248 | TCP | 1479 |
| 108.177.119.188 | TCP | 662 |
| 172.217.17.42 | TCP | 8966 |
| 52.114.158.50 | TCP | 35784 |
| 8.8.8.8 | UDP | 5210 |
| 255.255.255.255 | UDP | 680 |

**Figure A.1 :** Android VPN.

## APPENDIX A.2

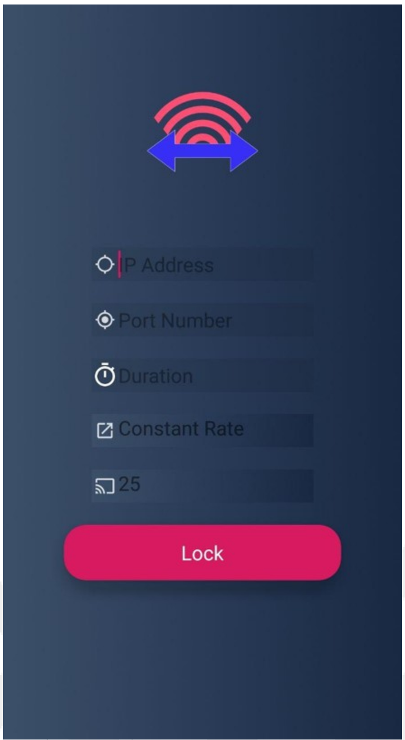
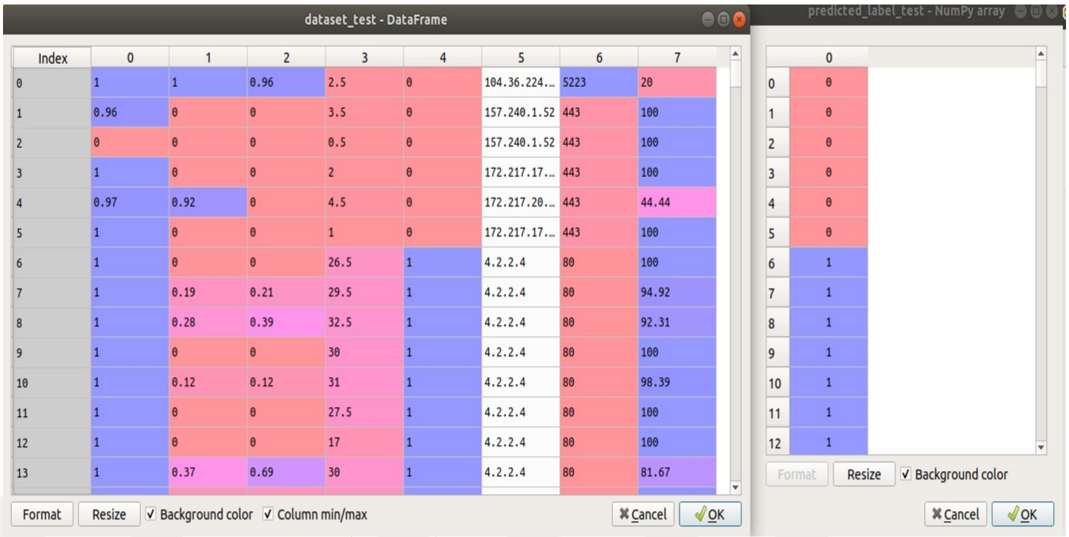


**Figure A.2 :** Packet Sender.

## APPENDIX A.3



| Index | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 1 | 0.96 | 2.5 | 0 | 104.36.224… | 5223 | 20 |
| 1 | 0.96 | 0 | 0 | 3.5 | 0 | 157.240.1.52 | 443 | 100 |
| 2 | 0 | 0 | 0 | 0.5 | 0 | 157.240.1.52 | 443 | 100 |
| 3 | 1 | 0 | 0 | 2 | 0 | 172.217.17.… | 443 | 100 |
| 4 | 0.97 | 0.92 | 0 | 4.5 | 0 | 172.217.20.… | 443 | 44.44 |
| 5 | 1 | 0 | 0 | 1 | 0 | 172.217.17.… | 443 | 100 |
| 6 | 1 | 0 | 0 | 26.5 | 1 | 4.2.2.4 | 80 | 100 |
| 7 | 1 | 0.19 | 0.21 | 29.5 | 1 | 4.2.2.4 | 80 | 94.92 |
| 8 | 1 | 0.28 | 0.39 | 32.5 | 1 | 4.2.2.4 | 80 | 92.31 |
| 9 | 1 | 0 | 0 | 30 | 1 | 4.2.2.4 | 80 | 100 |
| 10 | 1 | 0.12 | 0.12 | 31 | 1 | 4.2.2.4 | 80 | 98.39 |
| 11 | 1 | 0 | 0 | 27.5 | 1 | 4.2.2.4 | 80 | 100 |
| 12 | 1 | 0 | 0 | 17 | 1 | 4.2.2.4 | 80 | 100 |
| 13 | 1 | 0.37 | 0.69 | 30 | 1 | 4.2.2.4 | 80 | 81.67 |

|  | 0 |
|---|---|
| 0 | 0 |
| 1 | 0 |
| 2 | 0 |
| 3 | 0 |
| 4 | 0 |
| 5 | 0 |
| 6 | 1 |
| 7 | 1 |
| 8 | 1 |
| 9 | 1 |
| 10 | 1 |
| 11 | 1 |
| 12 | 1 |

**Figure A.3** : Machine Learning.

## CURRICULUM VITAE

**Name Surname** **:** Yalda MOTEVAKEL KHOSROSHAHI

**Place and Date of Birth:** 20/12/1991

**E-Mail** **:** khosroshahi15@itu.edu.tr

## EDUCATION:

- **B.Sc.:** 2015, Iran University of Science and Technology, Computer Engineering, Software Engineering.

- **M.Sc.:** 2019, Istanbul Technical University, Informatics Institute, Cyber Security Engineering and Cryptography.

## PUBLICATIONS, PRESENTATIONS AND PATENTS ON THE THESIS:

- Khosroshahi Y., Ozdemir E., 2019. "Detection of Sources Being Used in DDoS Attacks", 6th *International Conference IEEE CSCloud 2019*, June 21-23, 2019 Paris, France.