Derin Öğrenme Teknolojileri Kullanarak Dağıtık Hizmet Dışı Bırakma Saldırılarının Tespit Edilmesi

Ferhat Özgür Çatak
TÜBİTAK BİLGEM, Siber Güvenlik Enstitüsü
Kocaeli, Türkiye
ozgur.catak@tubitak.gov.tr

Ahmet Fatih Mustaçoğlu
TÜBİTAK BİLGEM
Kocaeli, Türkiye
afatih.mustacoglu@tubitak.gov.tr

Öz

Günümüzde kurumsal sistemlerde kötü amaçlı ağ trafiğinin miktarı, botnet, fuzzer, shellcode veya istismarların yayılması nedeniyle oldukça artmıştır. Bu kötü niyetli trafik günlük operasyonları tehdit etmesi sebebiyle oldukça önemli bir konu haline gelmiştir. Sınıflandırma modelleri kullanılarak saldırıların tahmin edilmesi ve ayırt edilmesi sağlanabilir. Erişilebilirliğe saldırmayı hedefleyen dağıtık hizmet dışı bırakma saldırıları, hizmete bağlanması gereken meşru kullanıcılar için hizmetlere erişimi engellemeyi amaçlamaktadırlar. Bu bildiride, ağ akış modellerine dayalı derin öğrenme yöntem ve teknolojileri tabanlı ağ trafiği sınıflandırma modeli önerilmektedir. Sınıflandırma performansını artırmaya yönelik olarak derin yapay sinir ağlarına dayalı model kullanılmıştır. Önerilen yöntemin sınıflandırma performansı grafik ve tablolarla gösterilmiştir. Simülasyon sonuçları, önerilen sınıflandırma yönteminin etkinliğini doğrulamaktadır.

1. Giriş

İnternet üzerinden hizmet veren servis sağlayıcılar ve işletmeler, düzenli olarak, botnet trafiğinin yayılması, dağıtılmış hizmet reddi (DDoS), arka kapı, exploit'ler, kabuk kodu ve fuzzerler gibi farklı siber saldırılara hedef olmaktadırlar [1], [2]. Servis sağlayıcı üzerinde oluşan zararlı trafik, ağ üzerinde olumsuz etkilere sebep olmakta ve meşru kullanıcıların hizmet almasını engellemektedir [3], [4]. Güvenlik duvarları veya saldırı tespit sistemleri gibi çeşitli ağ güvenliği çevre birimleri, servis sağlayıcılar üzerinde gerçekleşmekte olan bir takım saldırıları engellemektedir. Bu bileşenlerin güvenlik mekanizmaları genellikle imza tabanlıdır. Bu nedenle, kötü niyetli trafik değişiklikleri veya yeni türdeki ağ paket modelleri ile uyumlu olmamaktadırlar.

Kötü niyetli ağ trafiğini kısa sürede, tam ve gerçek zamanlı olarak algılayabilmemiz kritik önem taşmaktadır [5]. DDoS saldırılarında, daha önceden ele geçirilmiş olan bilgisayar kümeleri kullanılarak hedef sunucuya aşırı miktarlarda ağ trafiği gönderilmektedir. Fuzzing adı verilen, bir yazılıma geçersiz, rasgele veya beklenmedik veri enjeksiyonunu kullanarak uygulama hatalarını bulmayı içeren, otomatikleştirilmiş kara kutu yazılım test tekniğide ayrıca saldırganlar tarafından kullanılabilmektedir.

Ağ trafiğinde meydana gelebilecek bu gibi anormal değişiklikler tanımlayıcı istatistik yöntemleri kullanılarak tespit edilebilmektedir. Feinstein ve diğerleri [6] ağ trafiğinde yer alan anormallikleri tanımlamak için Ki-Kare istatistiklerini kullanan bir yöntem önermişlerdir. Kötü amaçlı trafiği algılamak için ağ akışında zaman penceresine dayalı entropi değişikliklerini kullanmayı önermişlerdir.

Literatürde çeşitli makine öğrenme yöntemleri kullanılarak dağıtık hizmet dışı bırakma saldırılarının algılanması ile ilgili çalışmalar mevcuttur.

Bhatia [7] yaptığı çalışmada hizmet dışı bırakma saldırılarını yapılan ağ katmanına göre farklı değerlendirilmesi gerektiğini belirtmiştir. SYN seli [8] saldırısı gibi ağ üzerinde paket trafiğini artırarak hizmetin bulunduğu sunucunun ağ bileşen kaynaklarını tüketmeye yönelik saldırılar olabileceği gibi, uygulama seviyesinde yapılan bir saldırı ile hem ağ bant genişliği hem de CPU kaynak tüketimi hedeflenebilmektedir. Bu çalışma kapsamında ağ seviyesinde gerçekleşen bir saldırının algılanması için gerekli kriterler uygulama seviyesinde yapılan saldırılar ile aynı olmayacağı ifade edilmiştir. Bu nedenle farklı nitelik ve yöntemler kullanılarak her iki tip saldırının algılanması ile modeller oluşturulmuştur.

Diğer bir çalışmada [9], ağ üzerinden yapılan smurf ve teardrop gibi çeşitli hizmet dışı bırakma saldırılarını algılamak için farklı makine öğrenme yöntemleri kullanmışlardır. Kullanılan algoritmalar sırasıyla naive bayes, bayes ağları, karar ağacı ve azalan hata budama (REPTree) şeklindedir. Çalışma kapsamında kullanılan veri kümesi, kddcup99 olarak adlandırılan ağ sızma algılama veri kümesidir. Çalışmada iki ve daha fazla algoritmanın beraber kullanılması ile elde edilen sınıflandırma modeline topluluk yöntemleri adını vermişlerdir. Yazarlar kddcup99 veri kümesi kullanılarak oluşturulan modeller ile %99.94117 oranında doğruluk elde etmişlerdir.

Osanaiye ve diğerleri [10] nitelik seçimlerinde topluluk yöntemleri kullanmışlardır. Bilgi kazanımı, kazanım oranı, ki-kare ve ReliefF yöntemleri kullanılarak kddcup99 veri kümesinin geliştirilmiş bir hali olan NSL-KDD veri kümesi üzerinde test etmişlerdir. Önerdikleri modeli kullanarak 13 nitelik seçmişler, seçilen nitelikler karar ağacı algoritması kullanılarak eğitilmiş ve ortaya çıkan sınıflandırma modelinin doğruluk oranını %99.67 olarak ölçümlemişlerdir.

Livadas ve diğerleri, IRC tabanlı botnet'lerin komuta kontrol trafiğini tespit etmek için ağ akış temelli bir yaklaşım önermişlerdir [11]. Yaptıkları çalışmada, ilk aşamada,

sınıflandırma algoritmaları ile trafik akışlarını IRC sohbet veya IRC sohbet-dışı akışlara sınıflandırırken, ikinci aşamada, IRC akışlarını kötü amaçlı veya kötü amaçlı olmayan olarak sınıflandırmaktadırlar. Modelin performans ölçüm değerlerine bakıldığında nispeten yüksek kabul edilebilecek %10-20 oranında hatalı negatif oranı ve %30-40 dolaylarında hatalı pozitif oranı elde etmişlerdir.

Zhao ve diğerleri trafik davranış analizi ve akış aralıklarına dayalı bir botnet algılama sistemi önermişlerdir [12]. Bu çalışma kapsamında kötü niyetli ve kötü amaçlı olmayan ağ trafiğini sınıflandırmak için azalan hata budama algoritmasını (REPTree) kullanmışlardır.

Saad ve diğerleri, kötü niyetli e-postalar, web siteleri, dosya paylaşım ağları ve geçici kablosuz ağlar aracılığıyla P2P botnet komutu ve kontrol aşamasını radar altında tespit etmek için ağ trafiği davranışının özelliklerini incelemişlerdir [13]. ISOT botnet veri kümesini kullanarak botnet trafiğini tanımlamak için beş farklı makine öğrenme algoritması kullanmışlar ve %89'luk en yüksek doğruluk oranı elde etmişlerdir.

Lu ve diğerleri, kötü amaçlı IRC bot trafiğini normal aralıktan ayırmak için önceden tanımlanmış bir zaman aralığı boyunca yük üzerindeki 256 ASCII baytın zamansal sık özelliklerini analiz eden bir algılama sistemi önermişlerdir [14].

Masud ve diğerleri ana makine üzerinde kurulu birden çok günlük dosyası arasındaki korelasyonu göz önüne alarak, akış temelli bir algılama yöntemi kullanan botnet algılama sistemi önermişlerdir [15].

Tanımlayıcı istatistik tabanlı algılama yöntemleri, önceden bilenen verilere dayanmaktadır. Dikkat çeken en önemli zorluk, ağ akış anormalliklerinin zaman içerisinde değişen bir hedef olmasıdır [16]. Fakat zararlı ağ trafiği kümesini doğru olarak karakterize oldukça zor bir konudur. Yeni tip zararlı ağ trafiği-zaman içerisinde farklılıklar göstererek ilerleyecektir. Fakat, kullanılacak olan sınıflandırma modelinin önceden tanımlanmış zararlı ağ trafiği sınıfına uygun olarak aşırı öğrenmesi engellenmelidir [17].

Bu çalışmanın temel katkıları şu şekildedir:

- Derin öğrenme yöntemlerine dayanan zararlı ağ akışının algılanması için yeni bir sınıflandırma modeli önerilmektedir.
- Keras, tensorflow ve theano teknolojilerinin kullanımı
- Siber güvenlik veri kümeleri üzerinde derin öğrenme yöntem ve teknolojilerinin kullanılması

2. MATERYAL VE METOT

2.1 *Veri Kümesi*

Bu çalışma kapsamında kullanılan veri kümesi, Avustralya Siber Güvenlik Merkezi'nde bulunan Siber Güvenlik Laboratuvarından alınmıştır [18]. Veri kümesi oluşturulurken IXIA PerfectStorm aracı kullanılmış, bu şekilde normal aktiviteler ve saldırı davranışlarını içeren ağ trafiği PCAP dosya formatında kayıt edilmiştir. Veri kümesi *Fuzzers*, *Analysis*, *Backdoors*, *DoS*, *Exploits*, *Normal*, *Reconnaissance*, *Shellcode* ve *Worms* şeklinde 9 farklı etikete sahip veriler içermektedir. PCAP dosyasından 49 farklı niteliğin çıkarılması işlemi için Argus aracı kullanılmıştır. Bahsedilen 49 nitelik arasında *kaynak IP adresi*, *protokol* gibi sayısal olmayan niteliklerde mevcuttur. Bu çalışmada kullandığımız sınıflandırma algoritması sebebiyle seçilen niteliklerin sayısal olması gerekmektedir. Bu nedenle, bu çalışma kapsamında bahsedilen 49 nitelik arasından sınıflandırma algoritmalarında kullanılmak üzere 25 adet sayısal değer içeren nitelik seçilmiştir. Sayısal değer içermeyen diğer nitelikler çalışmaya dahil edilmemiştir. Seçilen nitelikler ve açıklamaları Tablo 1'de bulunmaktadır.

Tablo 2'de veri kümesinde yer alan kayıtların *normal* ve *saldırı* dağılımları gösterilmektedir.

2.2 *Sınıflandırma Model Ölçümü*

Bu çalışma kapsamında önerilen modelin sınıflandırma performansının ölçülmesi için hassaslık (precision), geri çağırma (recall), F_1-ölçüsü ve doğruluk olmak üzere dört farklı sınıflandırma model değerlendirme metriği kullanılmıştır. Sınıflandırma algoritmalarının öğrenme aşamasında kullanılan eğitim veri kümesinin örnek boyutunu dikkatle seçilerek çok yüksek doğruluğu elde etmek kolaydır. Yalnızca doğruluk metriğinin kullanılması, önerilen modelin sınıflandırma performansını test edilmesinde hatalı yorumlamalara neden olabilmektedir. Bu sorunun üstesinden gelmek için, eğitimin veri kümesinin büyüklüğüne ve test örneklerine bağlı olmayan hassaslık (pozitif öngörme değeri) ve geri çağırma değerleri dikkate alınarak model değerlendirilmesi yapılması gerekmektedir.

Hassaslık elde edilen pozitif örneklerden (bu çalışma için zararlı ağ trafiği) gerçek pozitif olanların elde edilen değerlere olan oranıdır.

$$Hassaslik = \frac{Gercek\,Pozitif}{Toplam\,Pozitif\,Etiketlenen} \tag{1}$$

Geri çekilme oranı ise elde edilen pozitif örneklerden gerçek pozitif olanların örneklem kümesi içerisinde bulunan toplam pozitif örnek sayısına olan oranıdır.

$$Geri\,Cekilme = \frac{Gercek\,Pozitif}{Toplam\,Pozitif} \tag{2}$$

Hassaslık ve geri çağırma değerleri birbirlerine ters orantılı olarak davranmaktadır ve normalde bu iki değer arasında bir dengelenme mevcuttur. Bu nedenle hassaslık ve geri çekilmenin harmonik ortalaması olan F_1-ölçüsü denilen bir başka değerde oluşturulan modelin değerlendirmesinde dikkate alınmıştır.

Tablo 1: Veri kümesinde bulunan nitelikler.

Nitelik Adı	Açıklama
ackdat	TCP bağlantı kurulum zamanı SYN_ACK ve ACK paketleri arasındaki süre.
ct_flw_http_mthd	Http hizmetinde Get ve Post gibi yöntemler olan akış sayısı.
ct_ftp_cmd	Ftp oturumunda bir komut olan akışların sayısı.
dbytes	Hedef işlem boyutu
dinpkt	Hedef katmanlararası varış süresi (ms)
djit	Hedef jitter (ms)
dload	Hedef bit / saniye
dloss	Yeniden aktarılan veya silinen hedef paket sayısı
dmean	Hedef tarafından iletilen Ham paket boyutunun ortalaması.
dpkts	Hedef paket sayısı
dur	Toplam süre
dwin	Hedef TCP pencere değeri
is_ftp_login	Ftp oturumu
response_body_len	Sunucu http servisi tarafından cevap boyutu
sbytes	Kaynak işlem boyutu
sinpkt	Kaynak katmanlararası varış süresi (ms)
sjit	Kaynak jitter (ms)
sload	kaynak bit / saniye
sloss	Yeniden aktarılan veya silinen kaynak paket sayısı
smean	Kaynak tarafından iletilen Ham paket boyutunun ortalaması.
spkts	Kaynak paket sayısı
swin	Kaynak TCP pencere değeri
synack	TCP bağlantısı kurulum zamanı SYN ve SYN_ACK paketleri arasındaki süre.
tcprtt	TCP bağlantısı kurulumu gidiş-dönüş süresi 'SYN_ACK' ve 'ACKDAT' toplamı.
trans_depth	Http request / response transaction bağlantısındaki derinlik.

Tablo 2: Veri kümesinde bulunan nitelikler.

Trafik	Toplam kayıt
Normal	37000
Saldırı	45332

$$F_1 = 2 \times \frac{(Hassaslik) \times (Geri\,Cagirma)}{(Hassaslik) + (Geri\,Cagirma)} \quad (3)$$

Bu çalışma kapsamında ortaya konulan modellerin doğruluğunu göstermek amacıyla bahsedilen dört farklı metrik kullanılacaktır.

3. BULGULAR

3.1 *Sınıflandırma Modeli*

Sınıflandırma modeli oluşturulurken girdi katmanında 800, gizli katmanda 500 olacak şekilde oluşturulmuştur. Dropout, derin yapay sinir ağlarında popüler bir düzenleyici tekniktir. Temel olarak yaptığı işlem, ağ üzerinde yer alan birimlerin rasgele maskelenmesidir. Bu model oluşturulurken Dropout değeri 0.2 seçilmiş ve her katmana uygulanmıştır. Eğitim için oluşturulan modelin detayları Şekil 1'de gösterilmektedir.

3.2 *Sınıflandırma Modeli Eğitim Aşamaları*

Model eğitilirken CPU ve GPU platformları kullanılmıştır. Modelin doğruluk değerlerinde herhangi bir değişiklik gözlemlenmemiştir. Fakat eğitim aşamasında her bir iterasyon ve toplam eğitim süresi incelendiğinde GPU'nun daha performanslı olduğu gözlemlenmiştir. Önerilen model hem CPU, hem de NVIDIA Quadro 1000M grafik işlemci ile test edilmiştir. İlgili altyapılar Tablo 3'de gösterilmiştir.

Tablo 3: Kullanılan altyapılar.

Platform	CPU	Hafıza
Quadro 1000M	96 CUDA cores @ 1 GHz	16 GB
Intel i7-600	4 Çekirdek @ 4 GHz	16 GB

Model eğitilirken epoch sayısı 200 olarak belirlenmiştir. Bir epoch ile beraber modelin doğruluk oranı ve kayıp değerleri Şekil 2 - 3'de gösterilmiştir.

Tablo 4'de modelin değerlendirme sonuçları yer almaktadır.

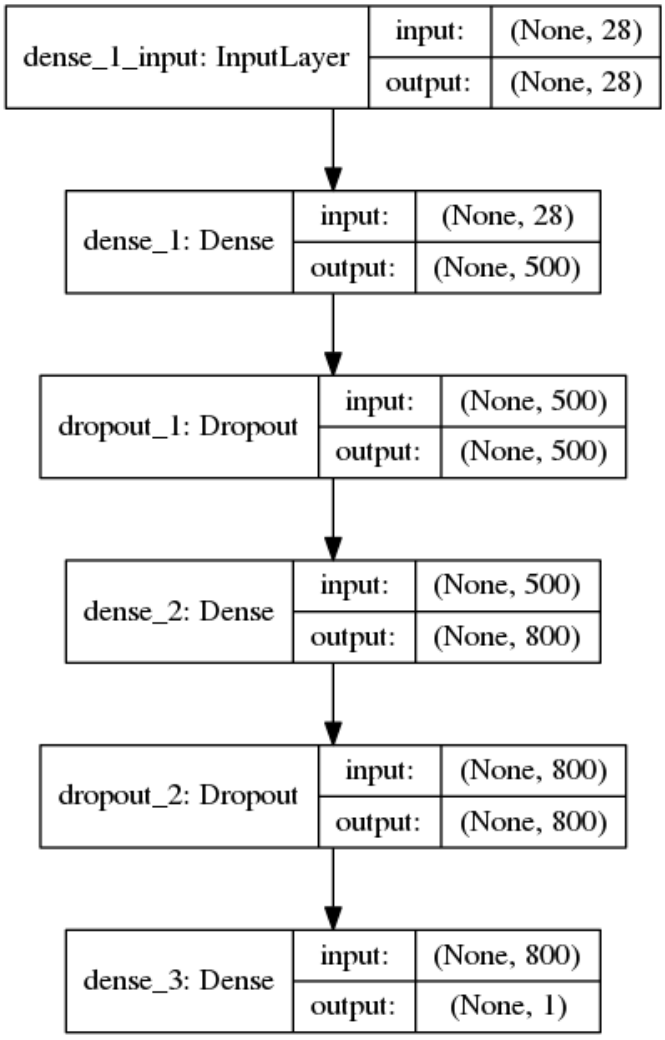
Tablo 4: Modelin değerlendirme sonuçları.

Sınıf	Hassaslık	Geri Çekilme	F_1	Doğruluk
Normal	0.99	0.99	0.99	0.9889
Saldırı	0.95	0.96	0.96	0.9889

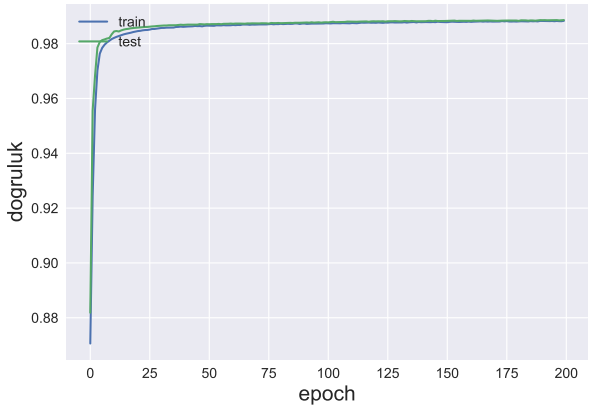
Tablo 5'de modelin eğtim süreleri yer almaktadır.

Tablo 5: İşlem Süresi.

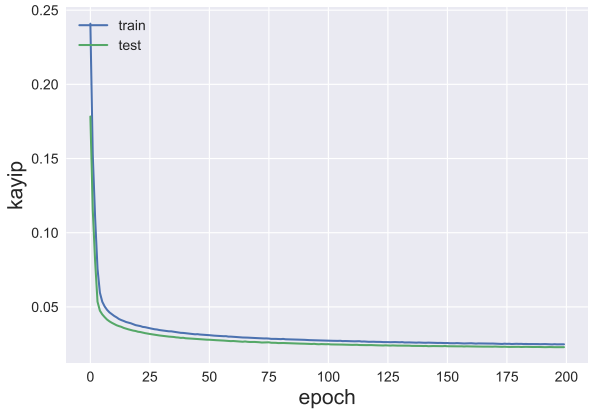
Platform	Eğitim Zamanı (sn)
Quadro 1000M	3516,21
Intel i7-600	10601,42



Şekil 1: Sınıflandırma modeli.



Şekil 2: Modelin iterasyon doğrulukları.



Şekil 3: Modelin iterasyon kayip değerleri.

4. SONUÇLAR

Bu çalışma kapsamında günümüzde en sık karşılaşılan siber saldırılardan biri olan zararlı ağ trafiğinin makine öğrenme yöntemleri ve derin öğrenme teknolojileri kullanılarak algılanabilmesi için bir model önerilmiştir. Önerilen modelin eğitimi GPU üzerinde olması sebebiyle eğitim süresinin azaldığı sonucuna varılmıştır.

Önerilen yöntemin yüksek boyutlu siber güvenlik alanında kullanılan veri kümelerine uygulanabilir olduğu gösterilmiştir. Bu yöntem kullanılarak zararlı ağ trafiğinin algılanmasında kaynak IP adresi gibi yanıltıcı alanlara bakılmadan, gelen paketler içerisinde yer alan büyüklük, varış süresi, yük boyutu gibi sunucuya gelen isteklerin niteliklerine bakılarak model oluşturmaktadır. Bu sebeple paketler üzerinde saldırganlar tarafında IP, MAC adres sahteciliği gibi yöntemlerden etkilenmemektedir.

Gelecek çalışma olarak nitelik çıkarımının danışmansız olarak yapılabildiği auto-encoder yöntemleri kullanılarak oluşturulacak olan veri kümesinin derin öğrenme yöntemleri ile eğitilmesi amaçlanmaktadır. Bu şekilde model sınıflandırma performansının artacağı değerlendirilmektedir.

KAYNAKÇA

[1] N. Ben-Asher and C. Gonzalez, "Effects of cyber security knowledge on attack detection," *Computers in Human Behavior*, vol. 48, pp. 51 – 61, 2015.

[2] B. Jasiul, M. Szpyrka, and J. Śliwa, "Detection and modeling of cyber attacks with petri nets," *Entropy*, vol. 16, no. 12, pp. 6602–6623, 2014.

[3] D. Jiang, Z. Xu, P. Zhang, and T. Zhu, "A transform domain-based anomaly detection approach to network-wide traffic," *Journal of Network and Computer Applications*, vol. 40, pp. 292 – 306, 2014.

[4] B. C. M. Cappers and J. J. van Wijk, "Snaps: Semantic network traffic analysis through projection and selection," in *2015 IEEE Symposium on Visualization for Cyber Security (VizSec)*, pp. 1–8, Oct 2015.

[5] F. Han, L. Xu, X. Yu, Z. Tari, Y. Feng, and J. Hu, "Sliding-mode observers for real-time ddos detection," in *2016 IEEE 11th Conference on Industrial Electronics and Applications (ICIEA)*, pp. 825–830, June 2016.

[6] L. Feinstein, D. Schnackenberg, R. Balupari, and D. Kindred, "Statistical approaches to ddos attack detection and response," in *Proceedings DARPA Information Survivability Conference and Exposition*, vol. 1, pp. 303–314 vol.1, April 2003.

[7] S. Bhatia, "Ensemble-based model for ddos attack detection and flash event separation," in *2016 Future Technologies Conference (FTC)*, pp. 958–967, Dec 2016.

[8] D. Kshirsagar, S. Sawant, A. Rathod, and S. Wathore, "Cpu load analysis and minimization for tcp syn flood detection," *Procedia Computer Science*, vol. 85, pp. 626 – 633, 2016. International Conference on Computational Modelling and Security (CMS 2016).

[9] V. D. Katkar and S. V. Kulkarni, "Experiments on detection of denial of service attacks using ensemble of classifiers," in *2013 International Conference on Green Computing, Communication and Conservation of Energy (ICGCE)*, pp. 837–842, Dec 2013.

[10] O. Osanaiye, H. Cai, K.-K. R. Choo, A. Dehghantanha, Z. Xu, and M. Dlodlo, "Ensemble-based multi-filter feature selection method for ddos detection in cloud computing," *EURASIP Journal on Wireless Communications and Networking*, vol. 2016, no. 1, p. 130, 2016.

[11] C. Livadas, R. Walsh, D. Lapsley, and W. T. Strayer, "Usilng machine learning technliques to identify botnet traffic," in *Proceedings. 2006 31st IEEE Conference on Local Computer Networks*, pp. 967–974, Nov 2006.

[12] D. Zhao, I. Traore, B. Sayed, W. Lu, S. Saad, A. Ghorbani, and D. Garant, "Botnet detection based on traffic behavior analysis and flow intervals," *Computers & Security*, vol. 39, pp. 2 – 16, 2013. 27th IFIP International Information Security Conference.

[13] S. Saad, I. Traore, A. Ghorbani, B. Sayed, D. Zhao, W. Lu, J. Felix, and P. Hakimian, "Detecting p2p botnets through network behavior analysis and machine learning," in *2011 Ninth Annual International Conference on Privacy, Security and Trust*, pp. 174–180, July 2011.

[14] W. Lu, G. Rammidi, and A. A. Ghorbani, "Clustering botnet communication traffic based on n-gram feature selection," *Computer Communications*, vol. 34, no. 3, pp. 502 – 514, 2011. Special Issue of Computer Communications on Information and Future Communication Security.

[15] M. M. Masud, T. Al-khateeb, L. Khan, B. Thuraisingham, and K. W. Hamlen, "Flow-based identification of botnet traffic by mining multiple log files," in *2008 First International Conference on Distributed Framework and Applications*, pp. 200–206, Oct 2008.

[16] M. H. Bhuyan, D. Bhattacharyya, and J. Kalita, "An empirical evaluation of information metrics for low-rate and high-rate {DDoS} attack detection," *Pattern Recognition Letters*, vol. 51, pp. 1 – 7, 2015.

[17] V. Srihari and R. Anitha, *DDoS Detection System Using Wavelet Features and Semi-supervised Learning*, pp. 291–303. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014.

[18] N. Moustafa and J. Slay, "The evaluation of network anomaly detection systems: Statistical analysis of the unsw-nb15 data set and the comparison with the kdd99 data set," *Inf. Sec. J.: A Global Perspective*, vol. 25, pp. 18–31, Apr. 2016.