

T.C.
GEBZE YÜKSEK TEKNOLOJİ ENSTİTÜSÜ
MÜHENDİSLİK VE FEN BİLİMLERİ ENSTİTÜSÜ

İNTERNET TRAFİĞİNİN GERÇEK ZAMANLI
SINIFLANDIRILMASI

CİHANGİR BEŞİKTAŞ
YÜKSEK LİSANS TEZİ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

GEBZE

2012

T.C.
GEBZE YÜKSEK TEKNOLOJİ ENSTİTÜSÜ
MÜHENDİSLİK VE FEN BİLİMLERİ ENSTİTÜSÜ

İNTERNET TRAFİĞİNİN GERÇEK ZAMANLI
SINIFLANDIRILMASI

CİHANGİR BEŞİKTAŞ
YÜKSEK LİSANS TEZİ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

TEZ DANIŞMANI
DOÇ. DR. HACI ALİ MANTAR

GEBZE
2012



YÜKSEK LİSANS TEZİ JÜRİ ONAY SAYFASI

G.Y.T.E. Mühendislik ve Fen Bilimleri Enstitüsü Yönetim Kurulu'nun 02.03.2012 tarih ve 2012/12... sayılı kararıyla oluşturulan jüri tarafından 04.06.2012 tarihinde tez savunma sınavı yapılan ...Cihangir BEŞİKTAŞ...'ın tez çalışması ...Bilgisayar Mühendisliği.. Anabilim Dalında YÜKSEK LİSANS tezi olarak kabul edilmiştir.

JÜRİ

ÜYE

(TEZ DANIŞMANI) : Doç. Dr. Hacı Ali MANTAR

ÜYE

: Prof. Dr. İbrahim SOĞUKPINAR

ÜYE

: Yard. Doç. Dr. S. Süer ERDEM

ONAY

G.Y.T.E. Mühendislik ve Fen Bilimleri Enstitüsü Yönetim Kurulu'nun 15/06/2012 tarih ve 2012/22... sayılı kararı.



ÖZET

TEZİN BAŞLIĞI: İNTERNET TRAFİĞİNİN GERÇEK ZAMANLI SINIFLANDIRILMASI

YAZAR ADI : CİHANGİR BEŞİKTAŞ

İnternet trafik akışlarının uygulama veya protokol bazında sınıflandırılması ağ izleme, hizmet kalitesi, nüfuz tespiti, ağ güvenliği, trend analizi vb. alanlarda etkin bir şekilde kullanılmaktadır. Son zamanlarda bazı P2P uygulamalarının dinamik port numaraları ve maskeleyme teknikleri kullanmaları port bazlı tespit sistemlerinin yetersiz kalmasına neden olmuştur. Bu engellere alternatif olarak geliştirilen yük bazlı tespit sistemleri ise uygulamaların güvenlik ve takip edilmeme amacıyla şifreleme mekanizmalarına gitmesi ve uygulama katmanında imza arama işlemlerinin çok maliyetli olması nedeniyle yerini makine öğrenmesi ve istatistikî verilere dayanan yaklaşımlara bırakmıştır.

Bu tez çalışmasında gerçek zamanlı internet trafik sınıflandırması için DBSCAN kümeleme algoritmasına ve ağırlıklı kosinüs benzerliği hesabına dayanan yeni bir istatistiksel metot önerilmiştir. Önerilen metot şifreli internet trafiğinin de sınıflandırılmasına olanak sağlamaktadır. Metodun öğrenme aşamasında port ve yük tabanlı sınıflandırıcımız ağ akışlarının protokollerini tespit ederek her bir protokolün öğrenme veri kümesini oluşturur. Daha sonra her bir protokolün alt uygulama vektörleri DBSCAN kümeleme algoritması ile çıkartılır. Alt uygulama vektörleri ağ akışlarının ilk birkaç paketinden elde edilen işaretli paket uzunlukları ve normalize edilmiş standart sapma vektörlerinden oluşmaktadır. Sınıflandırma aşamasında ise yeni bir ağ akışının protokolü ağırlıklı kosinüs benzerliği ile hesaplanır. Test sonuçları önerilen yöntemin yüksek bir başarımla sağladığını göstermektedir. Sonuçlar alt uygulama vektörleri ve standart sapmanın kosinüs benzerliğinde ağırlık olarak kullanılması katkılarımızın doğruluk performansını efektif olarak arttırdığını da göstermektedir.

SUMMARY

THESIS TITLE : REAL TIME INTERNET TRAFFIC CLASSIFICATION

AUTHOR : CİHANGİR BEŞİKTAŞ

Internet traffic classification has a critical role on network monitoring, quality of service, intrusion detection, network security and trend analysis. The conventional port-based method is ineffective due to dynamic port usage and masquerading techniques. Besides, payload-based method suffers from heavy load and encryption. Due to these facts, machine learning based statistical approaches have become the new trend for the network measurement community.

In this thesis, we propose a new statistical approach based on DBSCAN clustering and weighted cosine similarity for real time internet traffic classification on encrypted/non-encrypted traffic. In the learning phase, our port and payload-based pre-classifier tags network flows to applications. Then, our sub-application vectors, which consist of packet size and normalized standard deviation vectors of the first few packets of flows, are extracted for each application using DBSCAN clustering algorithm. In the classification phase, weighted cosine similarity is used to classify an unknown flow. Our experimental test results show that the proposed approach achieves very high accuracy. The results also show that our contributions which are sub-application vectors and the usage of standard deviation as weight in cosine similarity measurement increase the accuracy in an efficient way.

TEŞEKKÜR

Bu çalışmanın ilk aşamasından itibaren tüm aşamalarındaengin bilgi birikimi ve deneyimiyle bana katkıda bulunan danışman hocam Doç. Dr. Hacı Ali Mantar’a, proje kaynak kodunun yazılmasında ve test edilmesinde verdikleri destekten dolayı Inforcept Networks firmasının yöneticisi Murat Balaban’a ve son olarak beni bu günlere getiren aileme sonsuz teşekkürlerimi sunarım.

İÇİNDEKİLER DİZİNİ

	<u>Sayfa</u>
ÖZET	iii
SUMMARY	iv
TEŞEKKÜR	v
SİMGELER VE KISALTMALAR DİZİNİ	viii
ŞEKİLLER DİZİNİ	x
ÇİZELGELER DİZİNİ	xi
1 GİRİŞ	1
2 KONU İLE İLGİLİ ÇALIŞMALAR	3
2.1 Güdümlü Sınıflandırma	3
2.2 Güdümsüz Sınıflandırma	4
2.3 Karma Sınıflandırma	6
3 MAKİNE ÖĞRENMESİ TABANLI GELİŞTİRİLEN METODOLOJİLER	7
3.1 Sistem Mimarisi	8
3.2 Çevrimdışı Öğrenme	10
3.3 Çevrimiçi Sınıflandırma	14
3.3.1 Vektör Uzay Modeli	17
3.3.2 Kosinüs Benzerliği	17
3.4 İncelenen Protokoller	18
3.4.1 HTTP	18
3.4.2 HTTPS	19
3.4.3 SMTP	20
3.4.4 ESMTP	21
3.4.5 POP3	21
3.4.6 Bittorrent	22
3.4.7 Edonkey	23
3.4.8 MSN	24

3.4.9	XMPP	25
3.4.10	FTP	26
3.4.11	DNS	26
4	METODOLOJİLERİN DENEYSEL ANALİZİ	28
4.1	Test Ortamı ve Veri Kümesi	28
4.2	Performans Metrikleri	30
4.3	Test Sonuçları ve Analiz	31
4.3.1	Yöntemlerin Çalışma Zamanı ve Ölçeklenebilirlik Açısından İncelenmesi	35
4.3.2	Farklı Paket Sayıları ve Epsilon Değerleri için Derinlemesine İncelemeler	36
5	SONUÇLAR VE ÖNERİLER	42
	KAYNAKLAR DİZİNİ	44
	ÖZGEÇMİŞ	46

SİMGELER VE KISALTMALAR DİZİNİ

<u>Kısaltma</u>	<u>Açıklama</u>
IP	İnternet Protokolü
TCP	İletişim Kontrol Protokolü (Transmission Control Protocol)
UDP	Kullanıcı Veri Bloğu Protokolü (User Datagram Protocol)
NN	En yakın Komşular (Nearest Neighbours)
LDA	Lineer Diskriminant Analizi (Linear Discriminant Analysis)
QDA	Karesel Diskriminant Analizi (Quadratic Discriminant Analysis)
P2P	Uçtan Uca (Peer to Peer)
RTT	Gidiş Dönüş Süresi (Round Trip Time)
MTU	Maksimum İletim Birimi (Maximum Transmission Unit)
HTTP	Hipermetin Aktarma İletişim Kuralı (Hypertext Transfer Protocol)
HTTPS	Güvenli Hiper Metin Aktarım İletişim Kuralı (Secure Hypertext Transfer Protocol)
SMTP	Elektronik Posta Gönderme Protokolü (Simple Mail Transfer Protocol)
POP3	Postane Protokolü 3 (Post Office Protocol 3)
IMAP	İnternet Mesaj Erişim Protokolü (Internet Message Access Protocol)
MSN	MicroSoft Network

XMPP	Geniřletilebilir Mesajlařma ve Varlık Protokolü (Extensible Messaging and Presence Protocol)
FTP	Dosya Aktarım Protokolü (File Transfer Protocol)
DNS	Alan Adı Sistemi (Domain Name System)
NTP	Ağ Zaman Protokolü (Network Time Protocol)
TLS	Tařıma Katmanı Güvenlięi (Transport Layer Security)
DBSCAN	Gürültü ile Uygulamaların Yoęunluk Tabanlı Mekansal Kümelenendirilmesi (Density-Based Spatial Clustering of Applications with Noise)
eps	Epsilon
minpts	Minimum Noktalar (Minimum Points)
PktUzVek	Paket Uzunlukları Vektörü
StdSapVek	Standart Sapma Vektörü
NStdSapVek	Normalize Edilmiş Standart Sapma Vektörü

ŞEKİLLER DİZİNİ

<u>Şekil</u>	<u>Sayfa</u>
3.1. Öğrenme Akış Şeması	9
3.2. Sınıflandırma Akış Şeması	10
3.3. Çevrimdışı Öğrenme Akış Şeması	12
3.4. DBSCAN Algoritması	13
3.5. Yöntem 1 için Geliştirilen Algoritma	15
3.6. Yöntem 2 için geliştirilen algoritma	16
3.7. Yöntem 3 için Geliştirilen Algoritma	16
3.8. Yöntem 4 için Geliştirilen Algoritma	16
4.1. Paket İzi 1 Protokollerinin Paket Sayıları Dağılımları	29
4.2. Paket İzi 2 Protokollerinin Paket Sayıları Dağılımları	30
4.3. Paket İzi 1 için Genel Sonuçlar	33
4.4. Paket İzi 2 için Genel Sonuçlar	34
4.5. Yöntem 3 için Epsilon 0.03 iken Farklı Paket Sayılarına Göre Sonuçlar	37
4.6. Yöntem 3 için Paket Sayısı 4 iken Farklı Epsilon Değerleri Sonuçları	38
4.7. Epsilon 0.02 ve Paket Sayısı 5 iken Yöntem 1 ve Yöntem 2 için Sonuçlar	39

TABLOLAR DİZİNİ

<u>Tablo</u>	<u>Sayfa</u>
3.1. Yöntemler ve Özellikleri	7
3.2. HTTP Trafiği Kuralları	19
3.3. HTTPS Trafiği Kuralları	20
3.4. SMTP Trafiği Kuralları	20
3.5. ESMTP Trafiği Kuralları	21
3.6. POP3 Trafiği Kuralları	22
3.7. Bittorrent Trafiği Kuralları	23
3.8. Edonkey Trafiği Kuralları	24
3.9. MSN Trafiği Kuralları	25
3.10. XMPP Trafiği Kuralları	25
3.11. FTP Trafiği Kuralları	26
3.12. DNS Trafiği Kuralları	27
4.1. Paket İzleri	28
4.2. Paket İzlerindeki Protokollere Ait Ağ Akışları Sayıları	29
4.3. Paket İzi 1 ve Paket İzi 2 için genel sonuçlar	31
4.4. Paket İzi 1 için Alt Uygulama Vektörleri Sayıları	35
4.5. Paket İzi 2 için Alt Uygulama Vektörleri Sayıları	35
4.6. Yöntem 3 için DP sonuçları	36
4.7. Yöntem 4 için DP sonuçları	39
4.8. Yöntemler için YP sonuçları	41

1 GİRİŞ

Gerçek zamanlı internet trafik sınıflandırması günümüzde bir çok ağ operasyonunda rol almaktadır. Bu operasyonlara örnek olarak nüfuz tespiti, hizmet kalitesi, trend analizi, filtreleme, yasal izleme vb. verilebilir. Bu operasyonlar sonucu sağlanan faydalar ise internet servis sağlayıcıları için nüfuz tespiti ile güvenlik, hizmet kalitesi ile ağın performanslı kullanılması, trend analizleri için kategorilerin oluşturulması, filtreleme ile farklı kategorilerde internet servis paketlerinin oluşturulmasıdır. Ülkemizde son zamanlarda üzerinde çalışılan güvenli internet için filtreleme hizmetlerinin kaliteli bir şekilde sağlanmasında da yüksek performanslı trafik sınıflandırması büyük bir rol oynayacaktır. Ayrıca hükümetlerin koyduğu yükümlülükler çerçevesinde (ülkemizde 5651 yasası) IP trafiğinin yasal olarak izlenmesinde de önemli bir yere sahiptir. İnternet trafik sınıflandırmasının sağlamış olduğu önemli hizmetler bu alanda bir çok akademik çalışmanın yapılmasına sebep olmuştur.

Günümüz internet trafik sınıflandırılması çalışmaları aynı kaynak IP, kaynak port, hedef IP, hedef port ve iletim protokolüne (TCP, UDP) sahip olan paketlerin oluşturduğu ağ akışlarının hangi uygulamaya veya protokole ait olduğunu tespit etmeye çalışmaktadır. En basit anlamda bu işlem ağ akışlarının IANA tarafından belirlenen protokollerin klasik TCP veya UDP port numaralarına [1] bakılarak gerçekleştirilir. Fakat bazı uygulamaların dinamik port numaraları kullanmaları, bazı uygulamaların da güvenlik duvarlarını aşmak için http, ftp gibi servislerin genelde internete açık olan klasik port numaralarını kullanması port bazlı yaklaşımları yetersiz kılmıştır. Moore ve Papagiannaki [2] port bazlı yaklaşımların %70 başarımlı olduğunu tespit etmişlerdir.

Port bazlı yaklaşımların yetersiz oluşu araştırma dünyasını yük bazlı yaklaşımlara yöneltmiştir [3, 4]. Yük bazlı yaklaşımlarda ağ akışlarındaki paketlerin uygulama katmanlarına bakılarak derinlemesine paket incelemesi yapılır. Paketlerin yüklerinde belirli imzalar aranarak ağ akışının hangi uygulamaya veya protokole ait olduğu tespit edilmeye çalışılır. Yük bazlı yaklaşımların ise aşağıda belirtilen sıkıntıları vardır:

- Kendine özgü yapıya sahip olan veya kaliteli dokümantasyona sahip olmayan protokoller için ileri seviyede tersine mühendislik çalışması gerektirmektedir.
- Şifreleme kullanan protokollerin tespit edilmesi mümkün değildir.
- Periyodik olarak protokollerin imzalarının güncellenmesi gerekmektedir.
- Paketler üzerinde yapılan imza arama işlemleri CPU ve bellek kaynaklarını yoğun bir şekilde tüketmektedir.

Yük bazlı yaklaşımların yukarıda listelenen problemlerinden dolayı CPU ve bellek kaynaklarını daha az tüketen ve şifreli trafiğin tespit edilmesine de olanak sağlayan makine öğrenmesi tabanlı istatistikî yaklaşımlar üzerine son zamanlarda birçok çalışma yapılmaktadır. Makine öğrenmesi tabanlı yaklaşımların uygulamaları bir kaç adımdan oluşmaktadır. Önce ağ akışlarının ayırt edilmesini sağlayan özellikler çıkartılır. Bu özellikler maksimum, minimum paket uzunlukları, paket uzunluklarının ortası, ortalaması, standart sapması, paketler arası süreler gibi ağ akışlarına ait davranışlardır. Daha sonra ise öğrenme aşamasına geçilir. Bu aşamada her bir uygulama veya protokol için seçilen özellikler cinsinden bir karakteristik belirlenir. Son olarak ise yeni gelen bir ağ akışının hangi uygulamaya veya protokole ait olduğunu öğrenilen karakteristiklerle karşılaştırma yaparak tespit eden sınıflandırma algoritması uygulanır.

Bu çalışma makine öğrenmesi metodolojisini kullanarak IP trafiğinin gerçek zamanlı olarak sınıflandırılması üzerine yoğunlaşmıştır.

2 KONU İLE İLGİLİ ÇALIŞMALAR

Makine öğrenmesi tabanlı IP trafik sınıflandırması çalışmaları metodolojileri açısından üçe ayrılmaktadırlar:

- GÜDÜMLÜ sınıflandırma
- GÜDÜMSÜZ sınıflandırma
- Karma sınıflandırma

2.1 GÜDÜMLÜ Sınıflandırma

Güdümlü sınıflandırmada öğrenme işleminde sınıfı belli olan veriler kullanılır. Veri kümesindeki sınıfı belli veriler ile her bir sınıfın seçilen özellikler cinsinden karakteristiği çıkartılır. Sınıflandırma aşamasında ise yeni bir verinin sınıfı öğrenme aşamasında çıkartılan karakteristikleri kullanan sınıflandırma algoritması ile tespit edilir.

IP trafiği için yapılan güdümlü sınıflandırma çalışmalarında ise öğrenme aşamasında kullanılan veri kümesindeki her bir verinin hangi uygulamaya ait olduğu önceden etiketlenir ve bu etiketlere göre karakteristikler çıkartılır. Dolayısıyla öğrenme aşamasındaki veri kümesinin port ve/veya yük bazlı trafik tespiti yapan yazılımlar aracılığı ile veya manuel olarak önceden etiketlenmesi gerekmektedir. Sınıflandırma aşamasında ise yeni bir ağ akışının hangi uygulamaya ait olduğu kullanılan sınıflandırma algoritması ile tespit edilir.

Güdümlü sınıflandırma çalışmalarında Saf Bayes, Saf Bayes Çekirdek Tahmini, Bayesian Ağlar, C4.5 Karar Ağaçları, k-En Yakın Komşular, Sinir Ağları ve Destek Vektör Makineleri genelde tercih edilen algoritmalar olmuştur.

Roughan ve diğerleri [5] NN, LDA ve QDA algoritmaları üzerine yaptıkları çalışmada paket uzunluklarının ortası, varyansı, kök orta karesi, ağ akışlarının süresi,

toplam boyutu, toplam paketleri, bağlantıların TCP pencere boyutu, bant genişliği dağılımı gibi özellikleri kullanmışlardır. Yapılan çalışma internet trafiğini toplu veri (FTP-data), interaktif (Telnet), streaming (RealMedia) ve işlemsel (DNS) olmak üzere 4 sınıfta incelemiştir.

Moore ve Zuev [6] ise ağ akışlarına ait 248 özelliği Saf Bayes algoritmasını kullanan sınıflandırıcılarını eğitmek için kullanmışlardır. Bu algoritma ile %65 akış keskinliği yakalamışlardır. Yaptıkları çalışmada toplu veri transferi, veritabanı, interaktif, mail, servisler, www, p2p, atak, oyun ve multimedya trafiklerini tespit etmeye çalışmışlardır. Daha sonra [7] de çalışmaları genişletilerek %90 üzerinde akış keskinliği yakalanan Bayesian Sinir Ağı yaklaşımı önerilmiştir.

[5], [6], [7] deki çalışmalar ağ akışlarına ait ağ akışlarının başlangıçlarında elde edilemeyen birçok özelliği kullandıkları için gerçek zamanda sınıflandırma özellikleri yoktur. Gerçek zamanda sınıflandırma çalışmalarından biri Crotti ve diğerlerinin [8] ağ akışlarının ilk paket uzunlukları, paketler arası geçen süreler ve paket erişim sıralarını kullanarak %91 ağ akış keskinliği yakaladıkları Basit İstatistiksel Protokol Parmakizi metodu olmuştur. Çalışmalarında değişkenlik gösteren RTT ve MTU değerlerinin neden oldukları gürültüleri azaltmak için protokol maskesi ve normalizasyon teknikleri kullanmışlardır.

Gerçek zamanda sınıflandırma çalışmalarına bir diğer örnek ise vektör uzay modelini kullanarak Chung ve diğerleri [9] tarafından yapılmıştır. Çalışmalarında özellik olarak paketlerin yüklerinde geçen kelimelerin (16 bitlik sayı) sıklıklarından oluşan yük vektörlerini kullanmışlardır. Paketlerin yüklerini incelediklerinden çalışmaları şifreli trafiğin tespitini sağlayamamaktadır.

2.2 GÜDÜMSÜZ SINIFLANDIRMA

Güdümsüz sınıflandırma sınıfı belli olmayan öğrenme veri kümesindeki elemanların gruplandırılmasına dayanır. Gruplar seçilen özellikler cinsinden benzer olan

elemanlardan oluşturulur. Gruplama işleminden sonra her bir grubun ismi/etiketi manüel olarak verilir. Ayrıca grup elemanları kullanılarak seçilen özelliklere göre grup karakteristiği çıkartılır. Sınıflandırma işleminde ise yeni bir verinin sınıfı grupların karakteristikleri ile karşılaştırma yapan sınıflandırma algoritması ile tespit edilir.

IP trafiği için yapılan güdümsüz sınıflandırma çalışmalarında internet trafik akışları seçilen özellikler cinsinden gruplandırılır ve manüel olarak derinlemesine paket incelemesi ile etiketlenir. Yeni bir ağ akışının sınıfı ise bir sınıflandırma algoritması ile belirlenir.

IP trafiği için güdümsüz sınıflandırma çalışmalarının ilk örneklerinden biri McGregor ve diğerlerinin [10] 2004 yılında Beklenti Maksimizasyonu algoritmasını kullanarak gerçekleştirdikleri yaklaşımdır. Bu yaklaşım güdümsüz sınıflandırma yaparak HTTP, FTP, SMTP, IMAP, NTP ve DNS protokollerini tespit etmeye çalışmıştır.

Güdümsüz sınıflandırma çalışmalarına örnek olarak Zander ve diğerlerinin [11] Beklenti Maksimizasyonu algoritmasını kullanan Bayesian sınıflandırıcısı, Bernaille ve diğerlerinin [12] basit K-Means algoritması üzerine yaptıkları çalışma, Erman ve diğerlerinin [13] öklit uzaklığı ve K-Means algoritmasını kullanarak yaptıkları çalışma gösterilebilir.

Bir diğer güdümsüz sınıflandırma çalışması ise Erman ve diğerlerinin [14] k-Means, Autoclass ve DBSCAN algoritmalarını karşılaştırdıkları çalışma olmuştur. Yaptıkları çalışmanın sonucunda DBSCAN algoritmasının daha iyi gruplar oluşturduğunu tespit etmişlerdir. Yapılan tespit bu tez çalışmasında daha iyi gruplar oluşturmak için DBSCAN kümeleme algoritmasının kullanılmasında önemli bir etken olmuştur.

2.3 Karma Sınıflandırma

Karma sınıflandırmada güdümlü ve güdümsüz sınıflandırma tekniklerini beraber kullanılır.

Erman ve diğerleri [15] güdümlü ve güdümsüz sınıflandırmanın bir kombinasyonu olan yarı güdümlü karma sınıflandırma çalışması yapmışlardır. Yaptıkları çalışmada öğrenme aşamasında sınıfı belli olan ve sınıfı belli olmayan veriler üzerinde gruplama yapmışlardır. Daha sonra oluşan grupların sınıfını belirlemek için grupta bulunan sınıfı belli verileri kullanmışlardır. Bunun için maksimum benzerlik üzerine kurulu bir olasılık hesabı oluşturmuşlardır.

3 MAKİNE ÖĞRENMESİ TABANLI GELİŞTİRİLEN METODOLOJİLER

Makine öğrenmesi tabanlı yaklaşımımız ağ akışlarının ilk bir kaç paketinin işaretli (paketin yönüne bağlı olarak) paket uzunlukları vektörünü ve bu vektörlerden elde edilen normalize edilmiş standart sapma vektörünü öznitelik olarak kullanmaktadır. Bu öznitelikler ve nasıl elde edildikleri “Çevrimdışı Öğrenme” bölümünde detaylı olarak anlatılacaktır. Burada standart sapmanın ağırlıklı kosinüs benzerliğinde ağırlık olarak kullanılmasının sınıflandırma keskinliğini arttıracak önerilmektedir.

Diğer bir önerimiz ise alt uygulama vektörlerinin çıkartılmasıdır. Bir uygulamanın alt uygulama vektörleri o uygulamanın alt karakteristiklerini oluşturmaktadır. Örneğin ftp protokolünün iki farklı trafik çeşidi vardır. Bunlardan ftp-command komut transferini sağlar. Ftp-data ise veri transferini sağlar. Ftp-command ve ftp-data birbirinden bağımsız trafik karakteristiklerine sahiptir. Dolayısıyla ftp protokolünün karakteristiğinin aslında iki alt karakteristiği vardır. Bu mantık ile uygulamaların farklı karakteristiklerini ifade edebilmek için alt uygulama vektörlerinin çıkartılması önerilmiştir.

Bu tez çalışmasında standart sapma vektörünün ve alt uygulama vektörlerinin internet trafik sınıflandırmasına etkisini hesaplayabilmek için dört farklı yöntem kullanılarak performans testleri yapılmıştır. Bu yöntemler Tablo 3.1’ de verilmiştir.

Tablo 3.1. Yöntemler ve Özellikleri

Metod	Alt Uygulama Vektörleri	Standart Sapma	Kosinüs Benzerliği	Ağırlıklı Kosinüs Benzerliği
Sadece Paket Uzunlukları Yöntemi	-	-	+	-
Paket Uzunlukları ve Standart Sapma Yöntemi	-	+	-	+
Sadece Paket Uzunlukları ve Alt Uygulama Vektörleri Yöntemi	+	-	+	-
Standart Sapma ve Alt Uygulama Vektörleri Yöntemi	+	+	-	+

Birinci yöntem performans testlerinde referans olarak kullanılan ve en yalın olan metottur. Bu yöntemde öznitelik olarak sadece paket uzunlukları vektörü kullanılır. İkinci yöntem ise normalize edilmiş standart sapma vektörünün internet trafik sınıflandırmasına etkisini ölçmek için önerilen bir metottur. Bu yöntemde normalize edilmiş standart sapma vektörü ağırlıklı kosinüs benzerliğinde ağırlık olarak kullanılmaktadır. Üçüncü yöntem ise alt uygulama vektörlerinin performansının ölçülmesini sağlayan metottur. Son yöntemde ise normalize edilmiş standart sapma vektörü ve alt uygulama vektörlerinin beraber kullanılması sonucu nasıl bir performans elde edileceği hedeflenmiştir.

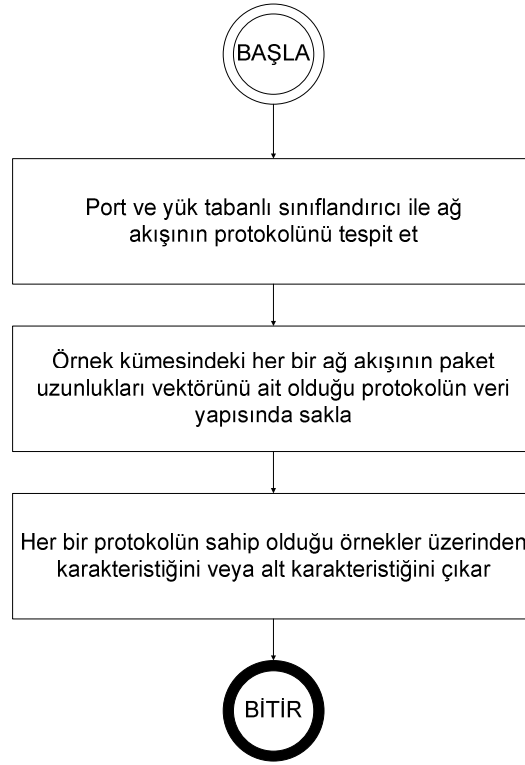
Tezin ilerleyen bölümlerinde birinci yöntem için “Yöntem 1”, ikinci yöntem için “Yöntem 2”, üçüncü yöntem için “Yöntem 3” ve dördüncü yöntem için “Yöntem 4” söz grupları kullanılacaktır.

3.1 Sistem Mimarisi

Makine öğrenmesi tabanlı önerilen yaklaşımın iki temel bileşeni vardır:

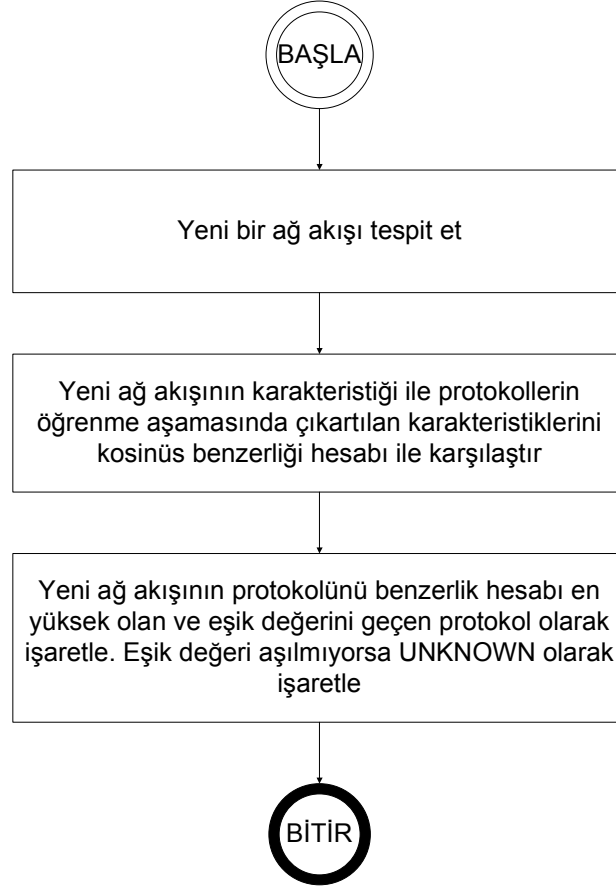
- Çevrimdışı Öğrenme
- Çevrimiçi Sınıflandırma

Çevrimdışı öğrenmede port ve yük tabanlı ön sınıflandırıcı her bir ağ akışının protokolünü tespit eder ve ağ akışının bilgilerini tespit edilen protokolün veri yapısında saklar. Daha sonra protokollerin veri yapılarında bulunan ağ akışlarının bilgilerinden her bir protokolün karakteristiklerini oluşturan paket uzunlukları vektörü ve normalize edilmiş standart sapma vektörü çıkartılır. Alt uygulama vektörünü kullanan yöntemlerde ise her bir protokolün veri yapısındaki ağ akışı bilgileri ve DBSCAN kümeleme algoritması kullanılarak her bir protokolün alt uygulama vektörleri çıkartılır. Daha sonra ise her bir alt uygulama vektörünün paket uzunlukları ve normalize edilmiş standart sapma vektörü çıkartılır.



Şekil 3.1. Öğrenme Akış Şeması

Çevrimiçi sınıflandırma ise gerçek zamanlı olarak gerçekleştirilir. Gerçek zamanda ağ kartı üzerinde tespit edilen bir ağ akışının hangi protokole ait olduğu tespit edilir. Bunun için tespit edilen ağ akışının paket uzunlukları vektörü ile protokollerin öğrenme aşamasında hesaplanan paket uzunlukları vektörleri kosinüs benzerliği hesabı ile karşılaştırılır. Ağ akışı benzerlik hesabı en yüksek olan protokol olarak işaretlenir. Standart sapma vektörünü kullanan yöntemlerde ise ağırlıklı kosinüs benzerliği hesabı yapılır.



Şekil 3.2. Sınıflandırma Akış Şeması

3.2 Çevrimdışı Öğrenme

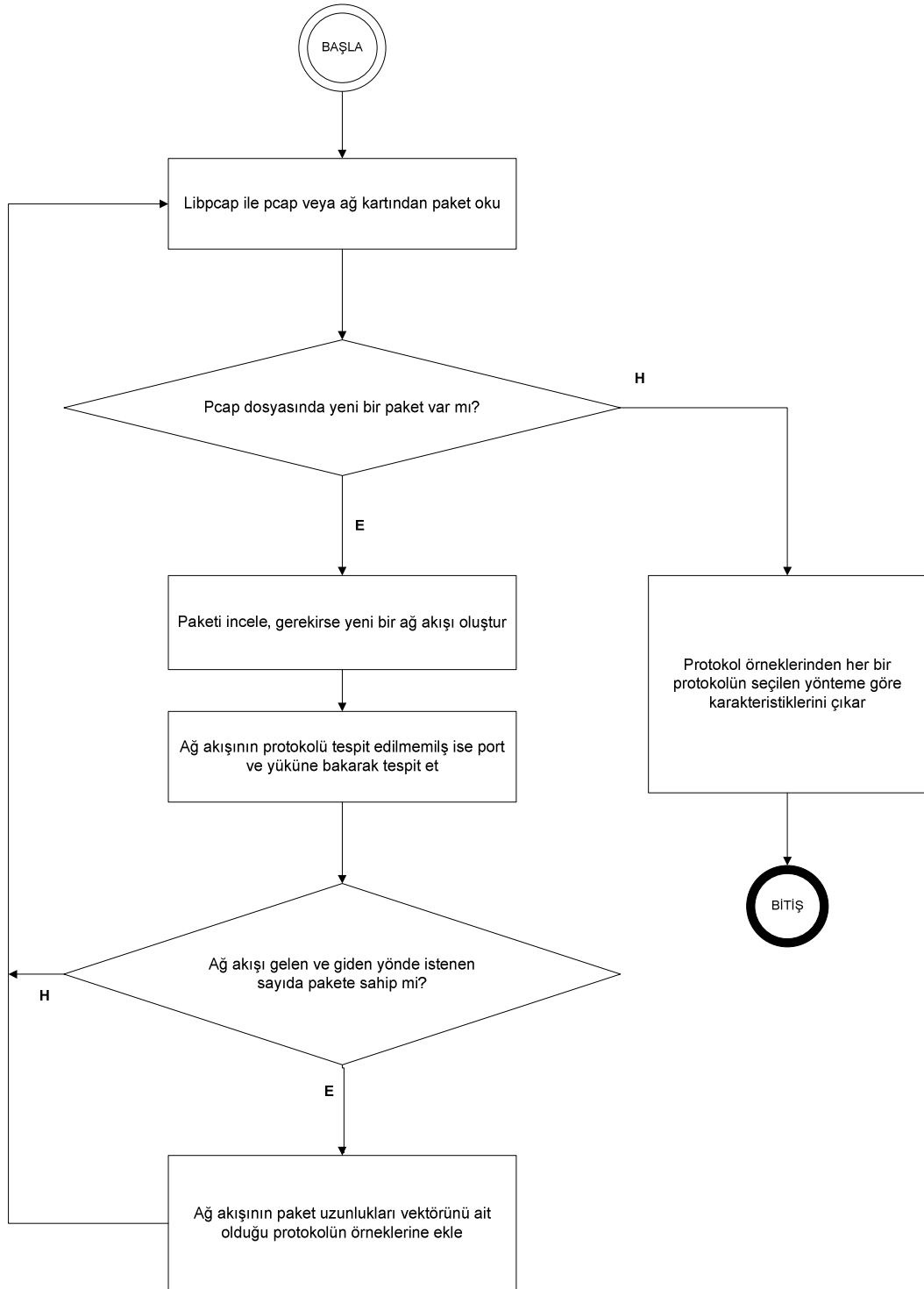
IP trafiğinden geçen aynı uygulamaya ait ağ akışlarının vektör uzay modeli kullanılarak karakteristiklerinin çıkartılması makine öğrenmesi tabanlı yaklaşımımızın öğrenme kısmını oluşturmaktadır. Öğrenme aşaması çevrimdışı olarak önceden toplanmış öğrenme veri kümesi ile yapılmaktadır. Bu işlem aşağıdaki aşamalardan oluşur:

1. Veri kümesi üzerindeki her bir ağ akışı port ve yük bazlı sınıflandırıcı ile incelenir ve hangi uygulamaya ait olduğu tespit edilir. Her bir ağ akışının gelen ve giden yönde ilk bir kaç paketinin paket uzunlukları kayıt altına alınır. Bu adımın sonunda her bir uygulama için o uygulamaya ait örneklerin gelen ve

giden yönde ilk bir kaç paketine ait paket uzunlukları elde edilmiş olunur. Paket uzunlukları vektöründe giden yöndeki paket uzunluğu negatif, gelen yöndeki paket uzunluğu pozitif olarak işaretlenir. Giden yön istemciden sunucuya doğrudur.

2. Eğer uygulanacak yöntem ile alt uygulama vektörleri oluşturacak ise her bir uygulamanın verileri DBSCAN algoritması kullanılarak gruplara ayrılır. Her bir grup minpts sayısında elemana ve gruptaki elemanlar birbirlerine en fazla epsilon değerinde uzaklığa sahiptirler. Minpts ve epsilon DBSCAN algoritmasının parametreleridir. İki ağ akışı arasındaki epsilon uzaklığı ise kosinüs benzerliği hesabı ile tespit edilir.
3. Daha sonra ise kullanılan yöntemle göre her bir uygulama veya alt uygulama için sahip olduğu örneklerin gelen ve giden yönde ilk bir kaç paketine ait paket uzunluklarının ortalaması alınarak gelen ve giden paket uzunlukları vektörü, paket uzunluklarının standart sapması hesaplanarak da gelen ve giden standart sapma vektörü hesaplanır. Standart sapma vektörü de normalizasyondan geçirilerek normalize edilmiş standart sapma vektörü elde edilir.

Öğrenme aşamasının akış şeması Şekil 3.3' de verilmiştir.



Şekil 3.3. Çevrimdışı Öğrenme Akış Şeması

Alt uygulama vektörlerinin çıkartılmasında DBSCAN algoritmasının tercih edilmesinin sebebi [16] da yapılan ölçümlerde DBSCAN algoritmasının daha iyi gruplar oluşturduğunun saptanmasıdır. Ayrıca DBSCAN değişken sayıda birbirine sıkıca bağlı elemanlardan oluşan gruplar oluşturmaktadır. Bu da uygulamaların kendilerini daha keskin karakterize eden alt uygulama vektörlerine sahip olmasını sağlamaktadır.

DBSCAN algoritması [17] Şekil 3.4’de verilmiştir.

```

DBSCAN(D, eps, MinPts)
  C = 0
  for each unvisited point P in dataset D
    mark P as visited
    N = getNeighbours(P, eps)
    if sizeof(N) < MinPts
      mark P as NOISE
    else
      C = next cluster
      expandCluster(P, N, C, eps, MinPts)
  expandCluster(P, N, C, eps, MinPts)
    add P to cluster C
    for each point P' in N
      if P' is not visited
        mark P' as visited
        N' = getNeighbours(P', eps)
        if sizeof(N') >= MinPts
          N = N joined with N'
    if P' is not yet member of any cluster
      add P' to cluster C

```

Şekil 3.4. DBSCAN Algoritması

Şekil 3.4’de verilen DBSCAN algoritması verilen bir D kümesini alt kümelere ayırmak için D de bulunan her bir elemanın getNeighbours fonksiyonu ile epsilon komşuları, expandCluster ile de epsilon komşularının komşuları bulunur. Bu şekilde birbirlerine epsilon uzaklıkta elemanlardan oluşan kümeler oluşturulmuş olunur. Burada getNeighbours fonksiyonunda epsilon uzaklığı kosinüs benzerliğinin 1’den çıkartılmasıyla tespit edilmektedir.

Öğrenme aşamasında her bir uygulama için çıkartılan paket uzunlukları ve normize edilmiş standart sapma vektörlerinin matematiksel tanımlamaları aşağıdaki gibidir.

C uygulamasına veya alt uygulama grubuna ait örneklerin S kümesinde olduğunu varsayalım. N adet örnek içeren S kümesindeki her bir örnek ise n uzunluğunda paket uzunlukları vektörüne sahip olsun. Eğer S kümesindeki j. örneğin paket uzunlukları vektörünün i. elemanı PU_{ji} , olarak gösterilirse, C uygulamasının veya alt uygulama grubunun paket uzunlukları vektörünün i. elemanı $PktUzVek(i)$ 1. denklem ile hesaplanır.

$$PktUzVek(i) = \left(\frac{\sum_{j=1}^N PU_{ji}}{N} \right) \quad (1)$$

Daha sonra normalize edilmiş standart sapma vektörünün i. elemanı 2. ve 3. denklemler ile hesaplanır.

$$StdSapVec(i) = \left(\sqrt{\frac{\sum_{j=1}^N (PU_{ji} - PktUzVek(i))^2}{N}} \right) \quad (2)$$

$$NStdSapVec(i) = \left(\frac{1}{1 + StdSapVec(i)} \right) \quad (3)$$

3.3 Çevrimiçi Sınıflandırma

Her bir uygulamaya ait karakteristikler çıkartıldıktan sonra yeni bir ağ akışının hangi uygulamaya ait olduğunu tespit etme makine öğrenmesi tabanlı yaklaşımımızın

sınıflandırma aşamasını oluşturmaktadır. Bu aşama gerçek zamanda çevrimiçi olarak yapılmaktadır ve aşağıdaki adımlar ile gerçekleştirilmektedir:

1. Ağ üzerinde yeni bir akış tespit edilir.
2. Bu akışa ait ilk bir kaç paketin paket uzunlukları ile öğrenme aşamasında çıkartılan diğer uygulamaların paket uzunlukları vektörü ve normalize edilmiş standart sapma vektörü arasında ağırlıklı kosinüs benzerliği hesabı yapılır. Standart sapmanın kullanılmadığı yöntemlerde normal kosinüs benzerliği hesabı yapılır. Normalize edilmiş standart sapma vektörü kosinüs benzerliği hesaplamasında ağırlık olarak kullanılarak, öğrenme paket uzunluğu değişkenlik gösteren vektör elemanlarının benzerlik hesabında daha az etki yapması hedeflenmiştir.
3. Yapılan hesaplama sonucu kosinüs benzerliği en yüksek olan uygulama tespit edilen ağ akışına atanır. Böylece sınıflandırma aşaması tamamlanmış olur.

Yöntem 1, Yöntem 2, Yöntem 3 ve Yöntem 4 için geliştirilen algoritmalar Şekil 3.5, Şekil 3.6, Şekil 3.7 ve Şekil 3.8’de verilmiştir.

```

CLASSIFY_Just_PktUzVek()
  max = 0, uyg = UNKNOWN
  for each UygVek(k) do
    sonuc = Kosinus_Benzerligi(PktUzVek(UNKNOWN), PktUzVek(k))
    if sonuc > max and sonuc > esik_deg then
      max = sonuc
      uyg = UYGULAMA(k)
    end if
  end for
  return uyg

```

Şekil 3.5. Yöntem 1 için Geliştirilen Algoritma

```

CLASSIFY_With_StdSapVek()
  max = 0, uyg = UNKNOWN
  for each UygVek(k) do
    sonuc = Agirlikli_Kosinus_Benzerligi(PktUzVek(UNKNOWN),
                                         PktUzVek(k), StdSapVek(k))
    if sonuc > max and sonuc > esik_deg then
      max = sonuc
      uyg = UYGULAMA(k)
    end if
  end for
  return uyg

```

Şekil 3.6. Yöntem 2 için geliştirilen algoritma

```

CLASSIFY_AltUygVek_Just_PktUzVek()
  max = 0, uyg = UNKNOWN
  for each Uyg(i) do
    for each AltUygVek(k) do
      sonuc = Kosinus_Benzerligi(PktUzVek(UNKNOWN), PktUzVek(i,k))
      if sonuc > max and sonuc > esik_deg then
        max = sonuc
        uyg = UYGULAMA(k)
      end if
    end for
  end for
  return uyg

```

Şekil 3.7. Yöntem 3 için Geliştirilen Algoritma

```

CLASSIFY_AltUygVek_Just_PktUzVek()
  max = 0, uyg = UNKNOWN
  for each Uyg(i) do
    for each AltUygVek(k) do
      sonuc = Agirlikli_Kosinus_Benzerligi(PktUzVek(UNKNOWN),
                                         PktUzVek(i,k), StdSapVek(i,k))
      if sonuc > max and sonuc > esik_deg then
        max = sonuc
        uyg = UYGULAMA(k)
      end if
    end for
  end for
  return uyg

```

Şekil 3.8. Yöntem 4 için Geliştirilen Algoritma

3.3.1 Vektör Uzay Modeli

Vektör uzay modeli doküman sınıflandırmada kullanılan ve metin dokümanlarını cebirsel olarak temsil eden vektörlerden oluşan bir modeldir. Bu modellemede her bir doküman, barındırdığı metinlerin sıklıklarına göre bir vektöre sahip olur. Dokümanlar sahip oldukları vektörler ile yapılan benzerlik hesapları ile sınıflandırılırlar. Bu modelleme IP trafik sınıflandırmasına da uygulanabilir. Bunun için ağ üzerinden geçen aynı uygulamaya ait akışlar bir doküman gibi ele alınıp ona ait bir vektör oluşturulur. Dokümanları temsil eden metinlerin sıklıkları yerine ise akışların paket uzunlukları ve bu paket uzunluklarının normalize edilmiş standart sapmaları kullanılabilir.

3.3.2 Kosinüs Benzerliği

Kosinüs benzerliği iki vektör arasındaki açının kosinüsünün hesaplanmasına dayanan benzerlik hesabıdır. Kosinüs sıfır bir, kosinüs doksan sıfır olduğundan vektörler arası açı ne kadar az ise kosinüs değeri de 1' e o kadar yakındır. Dolayısıyla iki vektör birbirine ne kadar benzer ise kosinüs değeri de 1'e o kadar yakındır. Kosinüs benzerliğinin matematiksel tanımlaması aşağıdaki gibidir.

A ve B K uzunluğunda iki vektör olsun. $A(i)$ ve $B(i)$ vektörlerin i . elemanı olsun. Bu durumda A ve B vektörleri arasındaki kosinüs benzerliği 4. denklem ile hesaplanır.

$$KB(A, B) = \frac{\sum_{i=1}^K A(i)B(i)}{\sqrt{\sum_{i=1}^K A(i)^2 * \sum_{i=1}^K B(i)^2}} \quad (4)$$

Eğer W ağırlık vektörü ve $W(i)$ bu vektörün i . elemanı ise, A ve B arasındaki ağırlıklı kosinüs benzerliği 5. denklem ile hesaplanır.

$$AKB(A, B, W) = \frac{\sum_{i=1}^K A(i)B(i)W(i)}{\sqrt{\sum_{i=1}^K W(i)A(i)^2 * \sum_{i=1}^K W(i)B(i)^2}} \quad (5)$$

3.4 İncelenen Protokoller

Bu çalışmada deneysel olarak incelenen protokoller ve bu protokollerin port ve yük bazlı sınıflandırıcımız tarafından tespitini sağlayan port ve yük bilgileri bu bölümde verilmiştir.

3.4.1 HTTP

RFC 2616 [18] tanımlanan Hypertext Transfer Protokol (HTTP) TCP tabanlı uygulama protokolüdür. İstemci-sunucu modeli ile çalışır. Dünya üzerinde çok yaygın bir protokoldür. Klasik olarak 80 portunu kullanmaktadır.

HTTP protokolü istemcileri büyük çoğunlukla web tarayıcılarıdır. Yaygın web tarayıcıları Firefox, Internet Explorer, Chrome, Safari ve Netscape'dir. Protokolün sunucuları ise web servisleridir. Apache yaygın olarak kullanılan bir web servisidir.

Dünya üzerindeki yaygın kullanımı ve web tarayıcılarda gömülü olması sebebiyle klasik 80 portu yaygın olarak HTTP için kullanılmaktadır ve güvenlik duvarları tarafından genelde engellenmemektedir. Fakat güvenlik duvarları tarafından 80 portunun engellenmemesi diğer bazı uygulamaların bu portu kullanmalarına sebep olmuştur. Dolayısıyla HTTP 80 portundan geçmektedir, fakat 80 portundan geçen her ağ akışı HTTP değildir. Bu sebeple port ve yük bazlı sınıflandırıcımız HTTP trafiğini Tablo 3.2'deki kurallara göre tespit eder.

Tablo 3.2. HTTP Trafiği Kuralları

Protokol	HTTP
Port	80
Yük	GET HTTP/1. POST OPTIONS HEAD PUT DELETE CONNECT

3.4.2 HTTPS

RFC 2818 [19] de tanımlanan HTTP over TLS (HTTPS) protokolü TCP tabanlıdır. HTTP trafiğinin güvenli bir şekilde şifreli olarak iletilmesi ihtiyacı üzerine oluşturulmuştur. HTTPS trafiği HTTP ile aynı yapıya sahiptir. Fakat ağ üzerindeki paketleri şifrelidir. Dolayısıyla yük tabanlı sınıflandırıcılar ile sınıflandırılması mümkün değildir. Klasik olarak 443 portunu kullanmaktadır.

Port ve yük tabanlı sınıflandırıcımız HTTPS protokolünü Tablo 3.3'deki kurallara göre sınıflandırmaktadır.

Tablo 3.3. HTTPS Trafiği Kuralları

Protokol	HTTPS
Port	443
Yük	-

3.4.3 SMTP

Simple Mail Transfer Protokol (SMTP) internet üzerinden elektronik posta iletimini sağlayan TCP tabanlı protokoldür. RFC 821 [20] 'de tanımlanmıştır. En son güncel hali 2008 yılında RFC 5321 [21]'de mevcuttur. Klasik olarak 25 portunu kullanmaktadır.

SMTP elektronik posta gönderimini ve alımını sağlamaktadır. SMTP protokolünün standart komutları bulunduğundan yük tabanlı sınıflandırıcılar ile sınıflandırılması mümkündür. Port ve yük tabanlı sınıflandırıcımız SMTP protokolünü Tablo 3.4'deki kurallara göre sınıflandırmaktadır.

Tablo 3.4. SMTP Trafiği Kuralları

Protokol	SMTP
Port	25
Yük	HELLO, MAIL, RCPT AUTH, STARTTLS DATA, NOOP, RSET 220, 250, 235, 334, 354

3.4.4 ESMTP

Extended SMTP (ESMTP) RFC1869 [22] da tanımlıdır. SMTP protokolünün uzatılmış versiyonudur. TCP ve UDP tabanlıdır. Klasik olarak 27 portunu kullanmaktadır.

Port ve yük tabanlı sınıflandırıcımız ESMTP protokolünü Tablo 3.5'deki kurallara göre sınıflandırmaktadır.

Tablo 3.5. ESMTP Trafiği Kuralları

Protokol	ESMTP
Port	27
Yük	-

3.4.5 POP3

Post Office Protokol (POP3) e-posta istemcilerinin e-posta sunucularında bulunan elektronik postalarını almalarını sağlayan TCP tabanlı protokoldür. RFC 1939 [23] da tanımlanmıştır. Klasik olarak 110 portunu kullanmaktadır.

Port ve yük tabanlı sınıflandırıcımız POP3 protokolünü Tablo 3.6'daki kurallara göre sınıflandırmaktadır.

Tablo 3.6. POP3 Trafiği Kuralları

Protokol	POP3
Port	110
Yük	+OK, -ERR, AUTH APOP, USER, PASS CAPA, LIST, STAT UIDL, RETR, DELE

3.4.6 Bittorrent

Bittorrent peer-to-peer yapıda dosya paylaşımı sağlayan protokoldür. Fakat uç birimlerin yönetimi tracker'lar tarafından yapılmaktadır. Tracker'lar ile olan bağlantı klasik olarak TCP 6969 portu üzerinden gerçekleştirilir. İki uç arası bağlantı ise dinamik portlar üzerinden sağlanmaktadır. Ayrıca güvenlik duvarlarını aşmak için HTTP, HTTPS gibi yaygın ve güvenlik duvarları tarafından genellikle engellenmeyen protokollerin port numaralarını da kullanmaktadır. Şifreli ve şifresiz versiyonları mevcuttur. Dolayısıyla sınıflandırılmasının port ve yük bazlı sınıflandırıcılar ile mümkün olmadığı durumlar vardır.

Bu tez çalışmasında şifresiz olan Bittorrent trafiği incelenmiştir. Port ve yük tabanlı sınıflandırıcımız Bittorrent protokolünü Tablo 3.7'deki kurallara göre sınıflandırmaktadır.

Tablo 3.7. Bittorrent Trafiği Kuralları

Protokol	Bittorrent
Port	Dinamik
Yük	BitTorrent protocol GET /webseed?info_hash= GET /data?fid=
	HTTP UserAgent Satırı: Azureus BitTorrent BTWebClient Shareaza & X-Queue:

3.4.7 Edonkey

Edonkey de Bittorrent gibi peer-to-peer olarak dosya paylaşımına olanak sağlayan bir sistemdir. Yaygın olarak kullanılmaktadır. Dinamik portları kullanmaktadır. Şifreli ve şifresiz versiyonları mevcuttur. Dolayısıyla Bittorrent de olduğu gibi port ve yük bazlı sınıflandırıcılar ile sınıflandırılması mümkün olmayabilir.

Bu tez çalışmasında şifresiz olan Edonkey trafiği incelenmiştir. Port ve yük tabanlı sınıflandırıcımız Edonkey protokolünü Tablo 3.8'deki kurallara göre sınıflandırmaktadır.

Tablo 3.8. Edonkey Trafiği Kuralları

Protokol	Edonkey
Port	Dinamik
Yük	Kendine özgü imza

3.4.8 MSN

MSN Microsoft tarafından geliştirilen anlık mesajlaşma uygulamasıdır. MSN istemci uygulamasına sahip kullanıcılar Microsoft tarafından yönetilen MSN sunucularına bağlanarak iletişim sağlarlar. Her bir kullanıcının kendine ait kişi listesi vardır ve bu kişi listesinde bulunan diğer kullanıcılar ile anlık mesajlaşma, dosya paylaşımı, sesli ve görüntülü görüşme yapabilmektedirler.

İstemciler ile sunucular arasındaki haberleşme ve yukarıda belirtilen farklı kategorideki işlemler Microsoft tarafından belirlenen MSN protokolüne göre yapılmaktadır. MSN protokolüne ait ağ paketleri klasik olarak 1863 portundan geçmektedir. Protokol port ve yük bazlı sınıflandırıcılar ile tespit edilebilmekte, fakat Microsoft tarafından yeni versiyonları çıkartılıp güncellendiğinden sınıflandırıcı tarafında güncellemeler istemektedir. Ayrıca MSN trafiğinden ağ koklama işlevi yapacak uygulamalar için ileri seviyede tersine mühendislik istemektedir. Çünkü örneğin MSN'in kendine özgü RTP haberleşme yapısı ve bu haberleşmede kullanılan kendine özgü ses/video codec'leri mevcuttur.

Port ve yük tabanlı sınıflandırıcımız MSN protokolünü Tablo 3.9'daki kurallara göre sınıflandırmaktadır.

Tablo 3.9. MSN Trafiği Kuralları

Protokol	MSN
Port	1863
Yük	VER, CVR, USR ADL, PRP, UUX

3.4.9 XMPP

RFC 3921 [24] de tanımlanan Extensible Messaging and Presence Protocol XML tabanlı anlık mesajlaşma protokolüdür. Protokol ilk olarak Jabber open source community tarafından geliştirilmiştir. Günümüzde yaygın olarak kullanılmakta ve protokolün standartlarına uygun birçok uygulama geliştirilmektedir.

Klasik olarak SSL ile şifreli versiyonu 5223 portunu, şifresiz olan versiyonu 5222 portunu kullanmaktadır. Port ve yük tabanlı sınıflandırıcımız XMPP protokolünü Tablo 3.10'daki kurallara göre sınıflandırmaktadır.

Tablo 3.10. XMPP Trafiği Kuralları

Protokol	XMPP
Port	5222, 5223
Yük	-

3.4.10 FTP

İlk olarak RFC 114 [25] de tanımlanan File Transfer Protocol istemci-sunucu mimarisine sahip TCP tabanlı iki uç arasında dosya transferi için geliştirilen bir protokoldür. Yaygın olarak kullanılmaktadır.

Klasik olarak TCP 21 portunu kullanmaktadır. Port ve yük bazlı sınıflandırıcılar ile sınıflandırılması mümkündür. Port ve yük bazlı sınıflandırıcımız FTP protokolünü Tablo 3.11'deki kurallara göre sınıflandırmaktadır.

Tablo 3.11. FTP Trafiği Kuralları

Protokol	FTP
Port	21
Yük	220 USER

3.4.11 DNS

RFC 882 [26] de tanımlanan DNS UDP tabanlı bir protokoldür. Alan adlarının IP adresleri eşleşmelerini öğrenmek için kullanılmaktadır. İnternet kullanıcılarının IP adreslerini akıllarında tutmalarının zorluğundan dolayı geliştirilmiştir. Bu sistem ile bağlanılmak istenen bir uç nokta IP adresi yerine alan adı ile erişilmektedir.

DNS protokolünde istemci DNS sunucusuna IP adresi gönderip bu IP adresine karşılık gelen alan adını alabilmekte ya da tam tersi şekilde alan adından IP adresini alabilmektedir. Klasik olarak 53 portunu kullanan DNS protokolü port ve yük bazlı sınıflandırıcımız ile Tablo 3.11'deki kurallara göre sınıflandırılmaktadır.

Tablo 3.12. DNS Trafiği Kuralları

Protokol	DNS
Port	53
Yük	Standart DNS başlığındaki sabit alanlar

4 METODOLOJİLERİN DENEYSEL ANALİZİ

Bu bölümde normalize edilmiş standart sapma vektörünün ve alt uygulama vektörlerinin internet trafik sınıflandırmasına etkilerini ölçen test ortamı, veri kümesi anlatılmış ve ölçüm sonuçları ve analizleri verilmiştir.

4.1 Test Ortamı ve Veri Kümesi

Test için geliştirilen internet trafik sınıflandırıcısı libpcap paket yakalama kütüphanesini kullanmaktadır. Sınıflandırıcı ağ akışlarının oturumlarını oluşturan bir TCP/IP yığına sahiptir. Bu TCP/IP yığını ağ kartından geçen her paketi yakalayarak bir callback fonksiyonunu ilgili paketin ağ akışı oturumu ile birlikte çağırır. Bir ağ akışının öğrenilmesi veya sınıflandırılması o ağ akışına ait istenen sayıda paket yakalandığında gerçekleştirilir. C programlama dili ile geliştirilen programın kodu FreeBSD 8.2 işletim sistemi üzerinde test edilmiştir.

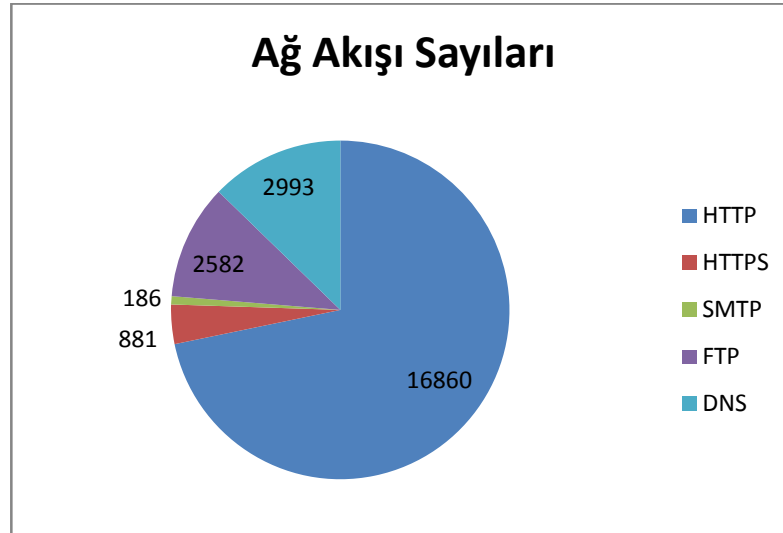
Veri kümesi olarak iki farklı tam yüklü paket izi toplanmıştır. Bu paket izlerinden ilki Naval Bilgisayar Bilimleri Bölümünün DEEP (Digital Evaluation and Exploitation) çalışma grubunun 2009 Kasım ve Aralık aylarında kayıt ettiği paket izleridir [18]. Diğeri ise 70 MB/s bant genişliğine sahip olan bir telekomünikasyon servis sağlayıcısının 3G omurgasında toplanan ve test edilen paket izleridir. Paket izlerine ait detaylı bilgiler Tablo 4.1 ve Tablo 4.2 de verilmiştir.

Tablo 4.1. Paket İzleri

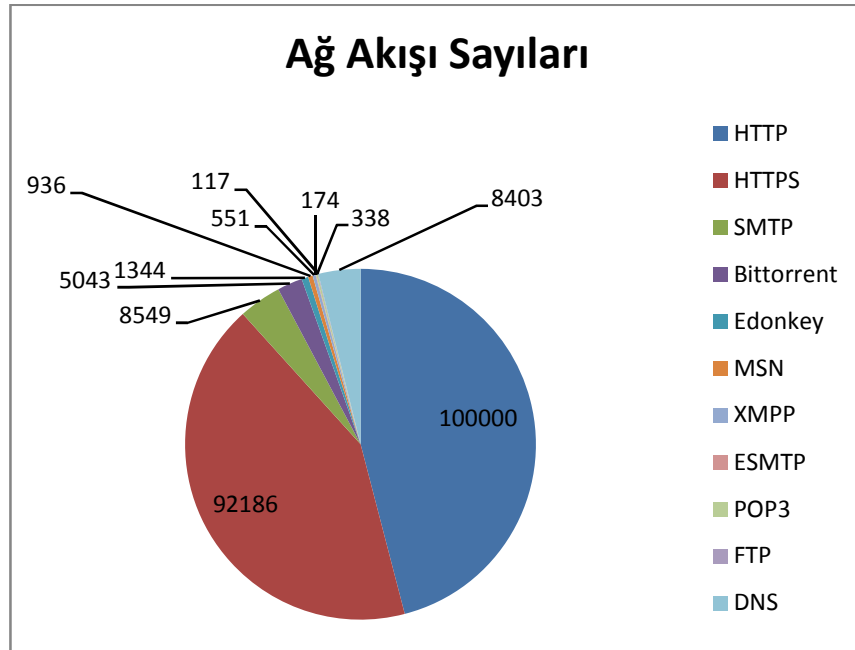
	Paket İzi 1	Paket İzi 2
Ağ Akışı Sayısı	20509	217689
Tarih	Kasım – Aralık 2009	07.10.2011

Tablo 4.2. Paket İzlerindeki Protokollere Ait Ağ Akışları Sayıları

	Ağ Akışı Sayısı	
Protokol	Paket İzi 1	Paket İzi 2
HTTP	16860	100000
HTTPS	881	92186
SMTP	186	8549
Bittorrent	-	5043
Edonkey	-	1344
MSN	-	936
XMPP	-	551
ESMTP	-	174
POP3	-	338
FTP	2582	117
DNS	2993	8403



Şekil 4.1. Paket İzi 1 Protokollerinin Paket Sayıları Dağılımları



Şekil 4.2. Paket İzi 2 Protokollerinin Paket Sayıları Dağılımları

4.2 Performans Metrikleri

İnternet trafik sınıflandırma yaklaşımlarının keskinliklerini karşılaştırmada kullanılan temel performans metrikleri şunlardır:

- **Yanlış Negatif (YN):** Üzerinde çalışılan uygulamaya ait olup yanlış tespit edilen ağ akışlarının yüzdesi.
- **Doğru Pozitif (DP):** Üzerinde çalışılan uygulamaya ait olup doğru tespit edilen ağ akışlarının yüzdesi ($100 - \text{YN}$).
- **Yanlış Pozitif (YP):** Üzerinde çalışılan uygulamaya ait olmayıp üzerinde çalışılan uygulama olarak tespit edilen ağ akışlarının yüzdesi.
- **Doğru Negatif (DN):** Üzerinde çalışılan uygulamaya ait olmayıp üzerinde çalışılan uygulama olarak tespit edilmeyen ağ akışlarının yüzdesi.

4.3 Test Sonuçları ve Analiz

Bu bölüm alt uygulama vektörleri ve normalleştirilmiş standart sapma vektörünün kosinüs benzerliğinde ağırlık olarak kullanılması önerilerimizin deneysel sonuçlarını içermektedir. Bu kapsamda Bölüm 3’de anlatılan dört yöntemin Paket İzi 1 ve Paket İzi 2 paket izlerinde test edilmesi sonucu elde edilen değerler paylaşılmıştır. Elde edilen değerler üzerinden önerilen yaklaşımların analizi yapılmıştır.

Tablo 4.3 Paket İzi 1 ve Paket İzi 2 üzerinde yapılan testlerin sonuçlarını içermektedir. Bu tabloda Yöntem 3 ve Yöntem 4 de alt uygulama vektörlerinin oluşturulmasında kullanılan DBSCAN algoritmasının epsilon ve minpts giriş parametreleri sırasıyla 0.02 ve 10 olarak verilmiştir. İncelenen paketlerin sayısı ise 5 olarak seçilmiştir. Bu şekilde yapılan seçim 4 yöntem için elde edilen en iyi sonuçların paylaşılması içindir.

Tablo 4.3. Paket İzi 1 ve Paket İzi 2 için genel sonuçlar

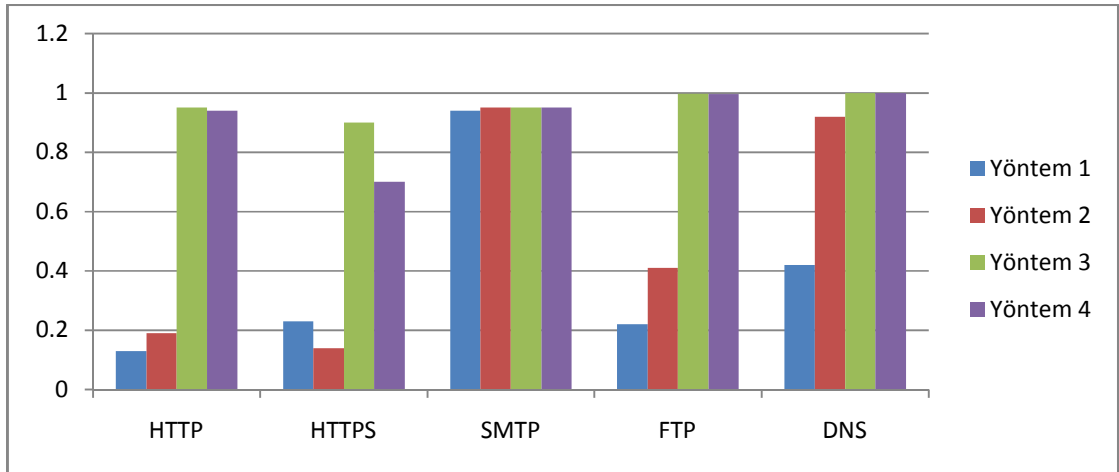
Paket İzi:	Paket İzi 1				Paket İzi 2			
Yöntem:	Yöntem 1	Yöntem 2	Yöntem 3	Yöntem 4	Yöntem 1	Yöntem 2	Yöntem 3	Yöntem 4
Süre:	0.28 sn	0.49 sn	4.57 sn	5.38 sn	3.01 sn	4.52 sn	45.55 sn	54.02 sn
HTTP	0.13	0.19	0.95	0.94	0.47	0.54	0.89	0.92
HTTPS	0.23	0.14	0.90	0.70	0.28	0.57	0.97	0.96
SMTP	0.94	0.95	0.95	0.95	0.40	0.45	0.71	0.81
Bittorrent	-	-	-	-	0.35	0.40	0.74	0.63
Edonkey	-	-	-	-	0.82	0.83	0.75	0.82
MSN	-	-	-	-	0.20	0.68	0.93	0.96
XMPP	-	-	-	-	0.68	0.68	0.94	0.91
ESMTP	-	-	-	-	0.81	0.83	0.93	0.93
POP3	-	-	-	-	0.28	0.30	0.87	0.91
FTP	0.22	0.41	0.997	0.996	0.00	0.0	0.84	0.84
DNS	0.42	0.92	1.00	1.00	0.80	0.82	0.98	0.94

Genel sonuçlar üzerinde değerlendirme yapacak olursak Paket İzi 1 üzerinde yapılan testlerde Yöntem 1 ve Yöntem 2 için SMTP hariç diğer protokollerin performans yüzdelerinin çok düşük kaldığı görülmektedir. Paket İzi 2 için de benzer sonuçların alındığı görülmektedir. Fakat Yöntem 2'nin Yöntem 1'e göre daha başarılı olduğu görülmektedir. Bu da normalleştirilmiş standart sapma vektörünün performansı arttırdığını göstermektedir.

Yöntem 3 ve Yöntem 4' ün sonuçlarına bakacak olursak büyük bir performans artışının gerçekleştiği görülmektedir. Bu da alt uygulama vektörleri önerimizin ne kadar isabetli bir yaklaşım olduğunu göstermektedir. Ayrıca Yöntem 4'ün genel olarak Yöntem 3' e göre daha başarılı olduğu görülmektedir. Bu da hem alt uygulama vektörlerinin hem de normalleştirilmiş standart sapma vektörünün beraber kullanılmasının en yüksek performans değerlerine ulaşılmasını sağladığını göstermektedir.

Yöntemler çalışma süreleri olarak karşılaştırıldığında alt uygulama vektörlerinin çalışma süresini yapılan test ortamında 10 kat arttırdığı görülmektedir. Bunun sebebi Yöntem 1 ve 2 nin incelediği protokol sayısına karşılık Yöntem 3 ve 4 için protokol sayısının yaklaşık 10 katı alt uygulama vektörünün olmasıdır. Ayrıca standart sapma hesabının da çalışma süresini az da olsa arttırdığı Yöntem 1 ile Yöntem 2 ve Yöntem 3 ile Yöntem 4 arasındaki çalışma süresi farklarından anlaşılmaktadır.

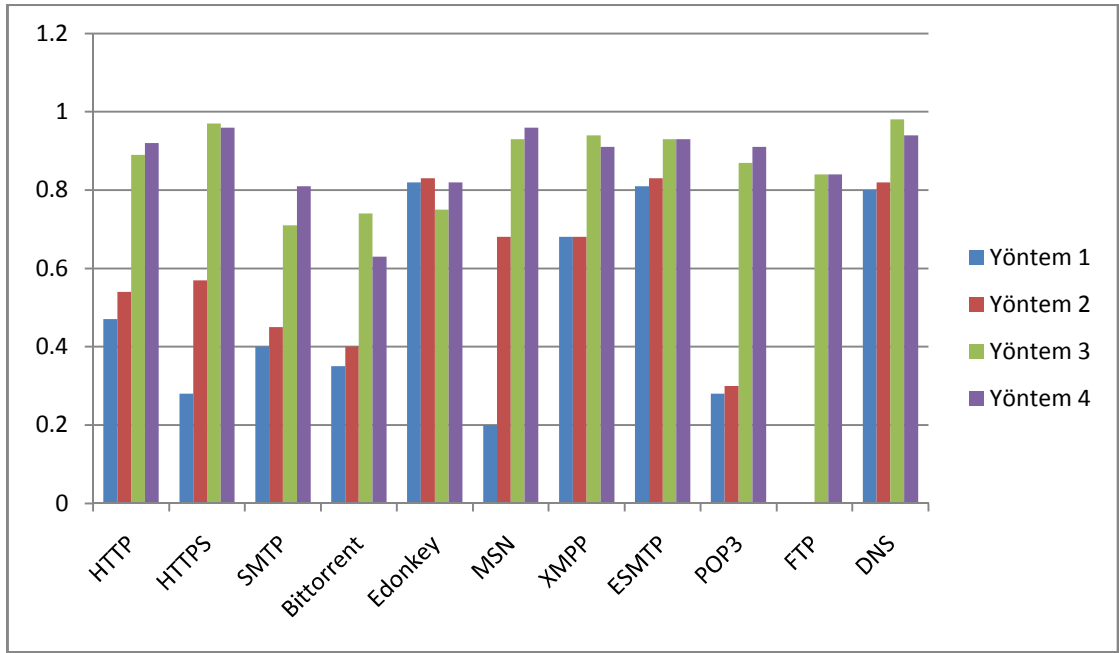
Sonuçların Paket İzi 1 ve Paket İzi 2 üzerinde benzer yapıda olması yaklaşımlarımızın hem uç ağlarda hem de ana omurgalarda yüksek performansta çalıştığını göstermektedir. Çünkü bölüm 4.1'de anlatıldığı üzere Paket İzi 1 bir uç ağdan, Paket İzi 2 de bir ana omurgadan toplanmıştır.



Şekil 4.3. Paket İzi 1 için Genel Sonuçlar

Şekil 4.3’de grafiksel olarak Paket İzi 1 için verilen sonuçlar incelenecek olunursa Yöntem 3 ve Yöntem 4’ün ortalama %95 civarı yüksek bir performans gösterdiği görülmektedir. Yöntem 1’in ortalama %40, Yöntem 2’nin ise ortalama %52 başarımlı gösterdiği görülmektedir.

Protokol bazında sonuçlar incelenirse SMTP ve DNS protokolleri üzerinde elde edilen sonuçların diğerlerine göre farklı karakteristik gösterdiği görülmektedir. Bunun sebebi Tablo 4.4’de de görüldüğü gibi bu protokollerin az sayıda alt uygulama vektörüne sahip olmasıdır. SMTP protokolü 1 adet karakteristiğe sahiptir. Bundan dolayı SMTP için tüm yöntemlerde yüksek bir başarımlı elde edilmiştir. DNS ise ortalama 6 adet alt karakteristiğe sahiptir. Yöntem 2’de kullanılan normalleştirilmiş standart sapma vektörü DNS için elde edilen başarımlı Yöntem 1’e göre arttırmıştır. Bunun sebebi ise DNS’in sahip olduğu 6 alt uygulama vektöründe sadece bazı paketler için farklılıklar olduğu ve bu farklılıkların normalleştirilmiş standart sapma vektörü ile etkisinin azaltılmasından dolayıdır.



Şekil 4.4. Paket İzi 2 için Genel Sonuçlar

Şekil 4.4’de grafiksel olarak Paket İzi 2 için verilen sonuçlar incelenecek olunursa Yöntem 3 için ortalama %88 ve Yöntem 4 için ortalama %90 başarımın elde edildiği görülmektedir. Yöntem 1’ in %46 ve Yöntem 2’nin ise %56 başarım gösterdiği görülmektedir.

Protokol bazında sonuçlar incelenirse MSN, XMPP, ESMTP ve DNS protokolleri Yöntem 2’ de yüksek başarımlar göstermiştir. Bunun sebebi ise Tablo 4.5’de görüldüğü üzere bu protokollerin az sayıda alt uygulama vektörüne sahip olması ve normalleştirilmiş standart sapma vektörünün kullanılması ile başarımın artırılmasıdır. FTP’ de Yöntem 1 ve Yöntem 2 için %0 sonucunun elde edilmesi bu protokolün Yöntem 1 ve Yöntem 2 de oluşturulan karakteristiğinin başka protokol ile yüksek benzerlik göstermesi ve bunun sonucu olarak hiç tespit edilememesidir. Ayrıca Tablo 4.2’den de görüleceği üzere Paket İzi 2’de en az sayıda pakete sahip protokol FTP’dir. Bu da Paket İzi 2 ile FTP’nin iyi öğrenilemediğini göstermektedir. Diğer protokoller için ise alt uygulama vektörlerinin başarımı büyük sıçramalar ile arttırdığı görülmektedir.

4.3.1 Yöntemlerin Çalışma Zamanı ve Ölçeklenebilirlik Açısından İncelenmesi

Yöntemler n toplam protokol sayısı ve m her bir protokolün alt uygulama vektör sayısı olmak üzere $O(nm)$ hızında çalışmaktadır. Yöntem 1 ve Yöntem 2 için m 1 olduğundan bu yöntemler $O(n)$ hızında çalışmaktadır. Yöntem 3 ve Yöntem 4 için ise m skalar bir seviyede tutulursa $O(n)$ hızında çalıştırılabilir. Bu sayı ise DBSCAN algoritmasında kullanılan epsilon değerine göre değişkenlik göstermektedir. Tablo 4.4 ve Tablo 4.5’de Paket İzi 1 ve Paket İzi 2 için protokollerin farklı epsilon değerlerine göre DBSCAN algoritması ile elde edilen alt uygulama vektörleri sayıları verilmiştir.

Tablo 4.4 ve Tablo 4.5’den görüldüğü üzere epsilon değeri arttıkça protokollerin alt uygulama vektörlerinin sayısının azaldığı görülmektedir. Fakat diğer taraftan epsilon değerinin artması protokollerin alt karakterlerinin temsil gücünü ve hassasiyetini azaltmaktadır. Bu da hassasiyet ve alt uygulama vektörleri sayısı arasında dengeli bir seçim yapılmasını gerektirmektedir.

Tablo 4.4. Paket İzi 1 için Alt Uygulama Vektörleri Sayıları

Uygulama:	HTTP	HTTPS	SMTP	FTP	DNS
Eps: 0.01	72	10	2	11	6
Eps: 0.02	53	10	1	9	6
Eps: 0.03	42	10	1	8	6
Eps: 0.04	35	10	1	8	5

Tablo 4.5. Paket İzi 2 için Alt Uygulama Vektörleri Sayıları

Application:	HTTP	HTTPS	SMTP	Bittorrent	Edonkey	MSN	XMPP	ESMTP	POP3	FTP	DNS
Eps: 0.01	193	87	27	27	15	6	6	3	12	2	5
Eps: 0.02	116	62	16	19	10	5	5	2	11	2	3
Eps: 0.03	75	47	9	16	10	3	4	2	9	2	3
Eps: 0.04	54	42	10	15	9	3	3	2	8	2	3

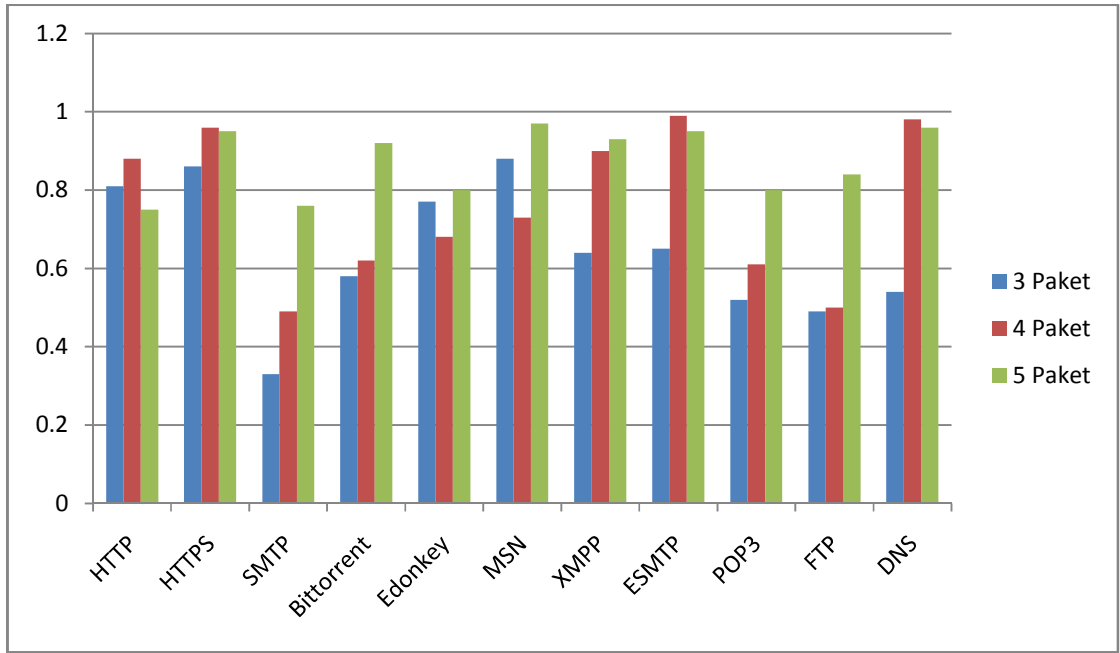
Ölçeklenebilirlik açısından ise Yöntem 3 ve Yöntem 4 için alt uygulama vektörleri sayıları kritik önem arz etmektedir. Bu kapsamda yüksek sayıda olabilecek alt uygulama vektörleri sayısının bu yöntemlerin gerçek zamanda çalışması açısından engel teşkil edebileceği unutulmamalıdır.

4.3.2 Farklı Paket Sayıları ve Epsilon Değerleri için Derinlemesine İncelemeler

Bu bölümde Yöntem 3 ve Yöntem 4 için Paket İzi 2 kullanılarak çıkartılan farklı epsilon ve farklı paket sayıları ile elde edilen sonuçlar paylaşılmıştır. Tablo 4.6 ve Tablo 4.7 bu sonuçları içermektedir.

Tablo 4.6. Yöntem 3 için DP sonuçları

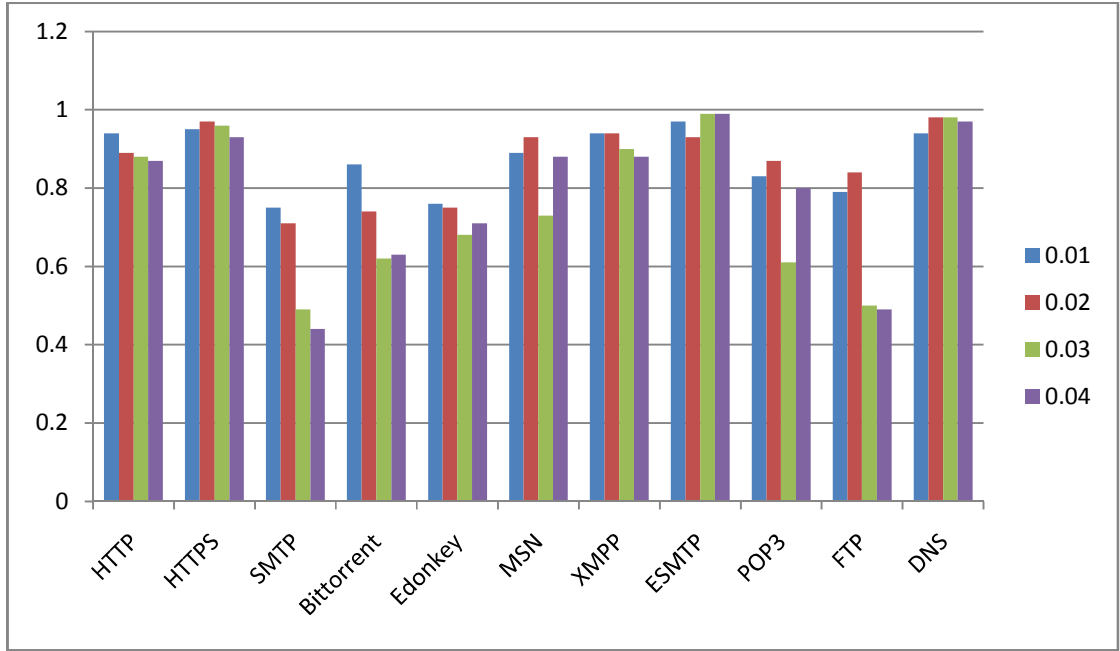
Paket Uz.	3				4				5			
Epsilon:	0.01	0.02	0.03	0.04	0.01	0.02	0.03	0.04	0.01	0.02	0.03	0.04
Süre:	32s	29s	28s	27s	43s	36s	33s	31s	57s	46s	41s	38s
HTTP	0.69	0.75	0.81	0.78	0.94	0.89	0.88	0.87	0.92	0.93	0.75	0.71
HTTPS	0.79	0.85	0.86	0.87	0.95	0.97	0.96	0.93	0.96	0.94	0.95	0.92
SMTP	0.55	0.47	0.33	0.53	0.75	0.71	0.49	0.44	0.93	0.89	0.76	0.76
Bittorrent	0.73	0.58	0.58	0.66	0.86	0.74	0.62	0.63	0.90	0.92	0.92	0.81
Edonkey	0.58	0.70	0.77	0.67	0.76	0.75	0.68	0.71	0.80	0.85	0.80	0.75
MSN	0.52	0.72	0.88	0.79	0.89	0.93	0.73	0.88	0.97	0.95	0.97	0.97
XMPP	0.91	0.92	0.64	0.64	0.94	0.94	0.90	0.88	0.93	0.93	0.93	0.97
ESMTP	0.38	0.38	0.65	0.65	0.97	0.93	0.99	0.99	0.93	0.95	0.95	0.99
POP3	0.84	0.55	0.52	0.44	0.83	0.87	0.61	0.80	0.80	0.79	0.80	0.77
FTP	0.77	0.49	0.49	0.50	0.79	0.84	0.50	0.49	0.83	0.84	0.84	0.84
DNS	0.54	0.54	0.54	0.94	0.94	0.98	0.98	0.97	0.96	0.96	0.96	0.96



Şekil 4.5. Yöntem 3 için Epsilon 0.03 iken Farklı Paket Sayılarına Göre Sonuçlar

Şekil 4.5’de görüldüğü üzere Yöntem 3 için paket uzunlukları vektörlerinde 4 paketin daha iyi sonuçlar verdiği durumlar gözükse de genel olarak 5 paketin en iyi sonucu verdiği görülmektedir. Paket uzunlukları vektörlerinde 3 paketin ise protokolleri yeterince temsil edemediği gösterdiği düşük performanslardan anlaşılmaktadır.

Şekil 4.6’da ise farklı epsilon değerleri için grafiksel sonuçlar verilmiştir. Bu sonuçlardan genel olarak epsilon’un 0.01 ve 0.02 olduğu durumlarda en iyi sonuçların elde edildiği görülmektedir. Diğer epsilon değerleri bazı protokoller için en iyi sonuçların alınmasını sağlasa da bazı protokoller için yetersiz kalmaktadır. Epsilon değerinin büyük olmasının ölçeklenebilirliği arttırdığı düşünülürse epsilonun 0.02 olarak seçilmesi diğerlerinin yanında daha mantıklı olacaktır.

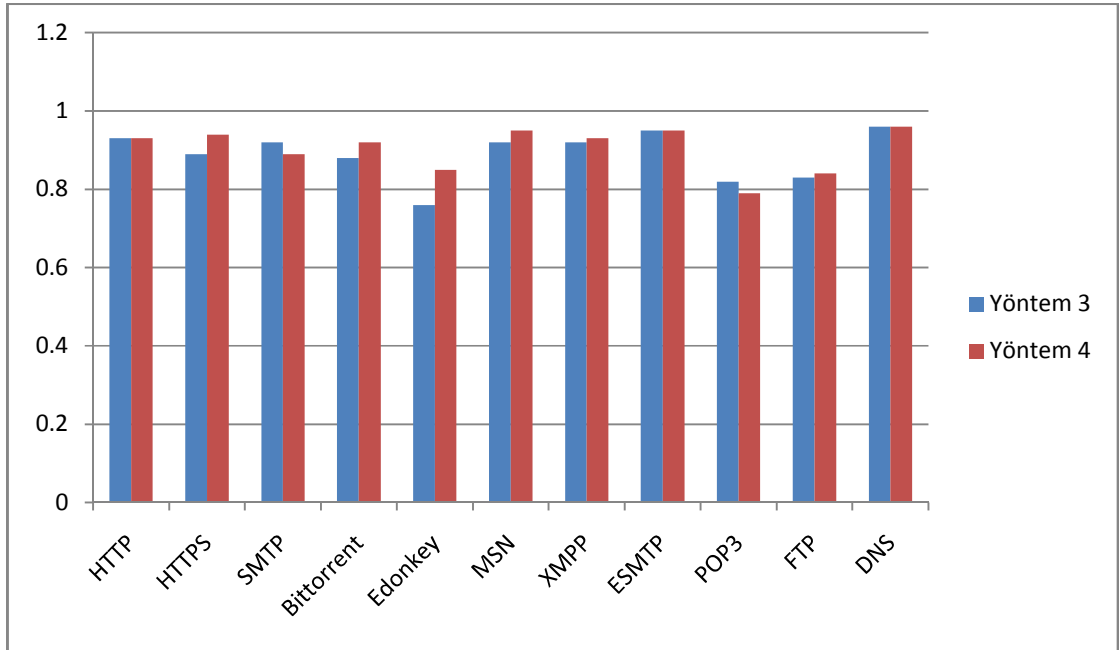


Şekil 4.6. Yöntem 3 için Paket Sayısı 4 iken Farklı Epsilon Değerleri Sonuçları

Tablo 4.7’de Yöntem 4 için farklı epsilon ve paket sayıları değerleri ile elde edilen sonuçlar verilmiştir. Tablodan anlaşılacağı üzere Yöntem 3 deki benzer çıkarımlar yapılabilir. Yani paket sayısı olarak 5 paketin, epsilon değeri olarak ise 0.02’nin en optimum başarıımı verdiği anlaşılmaktadır.

Tablo 4.7. Yöntem 4 için DP sonuçları

Paket Uz.	3				4				5			
Epsilon:	0.01	0.02	0.03	0.04	0.01	0.02	0.03	0.04	0.01	0.02	0.03	0.04
Süre:	35s	31s	29s	28s	50s	40s	36s	34s	69s	54s	47s	42s
HTTP	0.77	0.78	0.76	0.66	0.93	0.92	0.88	0.88	0.89	0.93	0.76	0.71
HTTPS	0.80	0.84	0.83	0.83	0.94	0.96	0.93	0.91	0.86	0.89	0.94	0.93
SMTP	0.43	0.45	0.20	0.23	0.75	0.81	0.60	0.60	0.91	0.92	0.83	0.81
Bittorrent	0.69	0.57	0.60	0.62	0.54	0.63	0.60	0.60	0.91	0.88	0.91	0.86
Edonkey	0.71	0.71	0.83	0.88	0.76	0.82	0.74	0.79	0.81	0.76	0.75	0.73
MSN	0.84	0.79	0.89	0.86	0.73	0.96	0.79	0.92	0.95	0.92	0.97	0.98
XMPP	0.91	0.92	0.64	0.63	0.94	0.91	0.86	0.84	0.92	0.92	0.92	0.96
ESMTP	0.81	0.16	0.38	0.65	0.97	0.93	0.98	0.98	0.97	0.95	0.95	0.98
POP3	0.79	0.40	0.45	0.37	0.88	0.91	0.69	0.82	0.78	0.82	0.84	0.84
FTP	0.84	0.47	0.48	0.49	0.85	0.84	0.84	0.84	0.83	0.83	0.83	0.83
DNS	0.24	0.24	0.19	0.22	0.89	0.94	0.94	0.94	0.96	0.96	0.96	0.96



Şekil 4.7. Epsilon 0.02 ve Paket Sayısı 5 iken Yöntem 1 ve Yöntem 2 için Sonuçlar

Epsilon ve paket sayısı değerlerinin optimum olduğu 0.02 ve 5 değerleri için Yöntem 3 ve Yöntem 4' e ait sonuçlar Şekil 4.7'de grafiksel olarak görülmektedir. Bu

grafikten Yöntem 3 ve Yöntem 4'ün birbirlerine yakın başarımlara sahip olduğu gözükse de Yöntem 4'ün daha iyi performansa sahip olduğu görülmektedir.

Yöntem 3 ve Yöntem 4'ün çalışma sürelerine bakıldığında standart sapmanın çalışma süresini belirgin bir şekilde arttırdığı görülmektedir. Dolayısıyla Yöntem 4, Yöntem 3'ün çalışma performansını fazla arttırmadığından Yöntem 3'ün seçilmesi çalışma süreleri açısından daha uygun olacaktır.

Çalışma sürelerini etkileyen diğer iki faktör ise epsilon ve paket sayısı değerleridir. Paket sayısının artması çalışma süresini arttırmakta, epsilon değerinin artması ise azaltmaktadır. Dolayısıyla doğruluk performansı açısından birbirine yakın olan 4 ve 5 paket sayısından çalışma süreleri göz önüne alındığında 4 paketin seçilmesi daha uygun olacaktır. Epsilon değeri açısından incelersek epsilon değerinin artması çalışma süresini azaltsa da doğruluk performansını arttıracaktır. Burada hem çalışma süresi hem de doğruluk performansı açısından en optimum değerin seçilmesi gerekmektedir.

4.3.2.1 Alt Uygulama Vektör Sayısı Seçimi

Tablo 4.6 ve Tablo 4.7 'deki sonuçlara bakıldığında epsilon değerinin artması çalışma süresini düşürmektedir. Bunun sebebi Tablo 4.5' de görüleceği üzere epsilon değerinin artması ile toplam alt uygulama vektör sayısının azalmasıdır. Alt uygulama vektör sayısının azalması da yeni bir ağ akışının protokolünün belirlenmesi hesabındaki karşılaştırma sayısını azaltacağından çalışma süresini de azaltacaktır. Fakat Tablo 4.3'e bakıldığında Paket İzi 2 için 12 protokol inceleyen Yöntem 1'in 1.32 sn, 412 alt uygulama vektörü inceleyen Yöntem 3'ün ise 45.55 sn sürdüğü görülmektedir. Bu da alt uygulama vektör sayısının protokol sayısına yakın olması gerektiğini göstermektedir. Bunun için alt uygulama vektör sayısı az olan protokollerin kullanılması daha uygun olacaktır. Örneğin alt uygulama vektör sayısı çok fazla olan ve paket yüküne bakılarak tespit edilebilen HTTP protokolünün benzerlik hesabından çıkartılması çalışma hızının düşürülmesine pozitif etki yapacaktır.

4.3.2.2 Yanlış Pozitif Analizi

Tablo 4.8. Yöntemler için YP sonuçları

	YP Yüzdesi			
Protokol	Yöntem 1	Yöntem 2	Yöntem 3	Yöntem 4
HTTP	0.024	0.009	0.093	0.043
HTTPS	0.051	0.044	0.054	0.058
SMTP	0.000	0.000	0.001	0.001
Bittorrent	0.016	0.100	0.001	0.003
Edonkey	0.023	0.008	0.001	0.001
MSN	0.007	0.000	0.000	0.000
XMPP	0.002	0.004	0.000	0.000
ESMTP	0.001	0.000	0.000	0.000
POP3	0.000	0.001	0.001	0.003
FTP	0.001	0.000	0.001	0.001
DNS	0.000	0.000	0.000	0.000

Epsilon ve paket sayısı değerlerinin optimum olduğu 0.02 ve 5 değerleri için Yöntem 1, Yöntem 2, Yöntem 3 ve Yöntem 4' e ait yanlış pozitif sonuçları Tablo 4.8'de verilmiştir. Bu tablodan genel olarak Yöntem 1 den Yöntem 4'e doğru YP oranının azaldığı görülmektedir. Bu da alt uygulama vektörleri ve normalize edilmiş standart sapma önerilerinin DP yanında YP açısından da başarıyı iyi bir oranda arttırdığını göstermektedir.

5 SONUÇLAR VE ÖNERİLER

Gerçek zamanlı internet trafik sınıflandırmasında port numarası ve L7 paket yükü protokollerin tespit edilmesinde etkin bir rol oynamaktadır. Fakat son zamanlarda özellikle P2P yapıdaki protokollerin dinamik port numaraları ve şifreleme yöntemleri kullanmaları bu yöntemlerin yetersiz kalmasına neden olmuştur. Bunun üzerine araştırma dünyasının makine öğrenmesi tabanlı istatistiki yöntemler üzerine yoğunlaştığı görülmüştür.

Bu tez çalışmasında araştırma dünyası tarafından yapılan makine öğrenmesi tabanlı istatistiki çalışmalar incelenmiş ve bunların üzerine bazı yeni yaklaşımlar öne sürülmüştür. Bu yaklaşımlar protokollerin alt uygulama vektörlerinin incelenmesi ve öğrenme kümesindeki verilerden her bir protokolün standart sapma vektörünün sınıflandırmada ağırlık olarak kullanılmasıdır. Bu kapsamda hem gerçek zamanlı sınıflandırmaya imkan vermesi hem de protokolden protokole farklılık göstererek bir anlamda ayırt edici özellikler barındıran ağ akışlarının ilk birkaç paketi incelenmiştir. Öğrenme aşamasında her bir protokolün paket yönlerini de içeren işaretli paket uzunlukları vektörü ve normalleştirilmiş standart sapma vektörü çıkarılmıştır. Ayrıca DBSCAN kümeleme algoritması kullanılarak her bir protokolün alt uygulama vektörleri çıkarılmıştır. Sınıflandırma aşamasında ise normalize edilmiş standart sapma vektörü kosinüs benzerliğinde ağırlık olarak kullanılmıştır.

Önerilen yaklaşımların performanslarını ölçmek için 4 yöntem geliştirilmiştir. Bu yöntemler ile yapılan testler sonucu DBSCAN ile oluşturulan alt uygulama vektörlerinin sınıflandırma doğruluğunu %50'lerden %90'ların üzerine çıkardığı görülmüştür. Ayrıca normalize edilmiş standart sapma vektörlerinin de bu doğruluğu az da olsa arttırdığı görülmüştür. Testlerin hem uç bir ağdan hem de ana bir omurgadan toplanan paket izleri ile yapılması ve bu paket izlerinde benzer sonuçların alınması yaklaşımların hem uç ağlarda hem de ana omurgalarda yüksek bir performansla çalıştığını göstermektedir.

Önerilen yaklaşımların tespit edilen kısıtları ise DBSCAN algoritması ile üretilen protokollerin alt uygulama vektörlerinin sayılarının fazla olabilmesidir. Bu da ölçeklenebilirlik sorunu çıkartarak gerçek zamanlı olarak sınıflandırma yapılmasını etkilemektedir. Bu amaçla alt uygulama vektör sayısı az olan protokollerin önerilen yöntemler ile tespit edilmesi daha mantıklı olacaktır. Ya da alt uygulama vektörleri ile bir ikili ağaç oluşturup bu ikili ağaç üzerinde dolaşma işlemi yapılmalıdır. Bu da sınıflandırma işleminin çalışma zamanını $O(n)$ den $O(\log(n))$ e düşürerek büyük bir hız artışı sağlayacaktır. Fakat ikili ağaç oluşturmak için gerekli vektörler arası ilişki kurmak oldukça zordur.

Bu çalışmada yapılan testler şifresiz trafik üzerinde yapılmıştır. Şifreli olan HTTPS trafiği sadece port numarasına bakılarak incelenmiştir. Fakat yaklaşım istatistiki tabanlı olduğu için şifreli trafiğin sınıflandırılmasında da iyi bir başarımlı sağlayacağı düşünülmektedir. Testlerin manüel olarak toplanmış şifreli paket izleri üzerinde de yapılması gelecek bir çalışma olarak düşünülebilir.

Son olarak makine öğrenmesi tabanlı istatistiki yaklaşımların protokolleri tespit etmesini engellemek amacıyla bir takım metotlar geliştirilebilir. Bu metotlara örnek olarak paket boyutlarını rastgele yapmak için paketlere ekleme yapılması, RTT sürelerinin rastgele hale getirilmesi verilebilir. Bu çalışmaları öngörmek ve bunlara karşı yeni yöntemler geliştirmek araştırma dünyası için fikir verecektir.

KAYNAKLAR DİZİNİ

- [1] IANA, Internet Assigned Numbers Authority, <http://www.iana.org/assignments/service-names-portnumbers/service-names-port-numbers.xml>
- [2] A. Moore ve K. Papagiannaki, “Toward the accurate identification of network applications”, Proc. Passive and Active Measurement Workshop (PAM2005), Boston, MA, USA, March/April 2005
- [3] P.-C. Lin, Y.-D. Lin, T.-H. Lee, ve Y.-C. Lai. “Using string matching for deep packet inspection”, IEEE Computer Practices, 41(4):23–28, April 2008.
- [4] S. Sen, O. Spatscheck, ve D. Wang, “Accurate, scalable in network identification of P2P traffic using application signatures”, WWW2004, New York, NY, USA, May 2004.
- [5] M. Roughan, S. Sen, O. Spatscheck, ve N. Duffield, “Class-of-service mapping for QoS: A statistical signature-based approach to IP traffic classification”, Proceedings of ACM/SIGCOMM Internet Measurement Conference (IMC) 2004, Taormina, Sicily, Italy, November 2004.
- [6] A. Moore ve D. Zuev, “Internet traffic classification using Bayesian analysis techniques”, ACM International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS) 2005, Banff, Alberta, Canada, June 2005.
- [7] T. Auld, A. W. Moore, ve S. F. Gull, “Bayesian neural networks for Internet traffic classification”, IEEE Transactions on Neural Networks, no. 1, pp. 223–239, January 2007.
- [8] M. Crotti, M. Dusi, F. Gringoli, ve L. Salgarelli, “Traffic classification through simple statistical fingerprinting,” SIGCOMM Comput. Commun. Rev., vol. 37, no. 1, pp. 5–16, 2007.
- [9] J.Y. Chung, B. Park, Y.J. Won, J. Strassner ve J.W. Hong, "Traffic Classification Based on Flow Similarity", IPOM 2009, LNCS 5843, pp. 65-77, 2009
- [10] A. McGregor, M. Hall, P. Lorier, ve J. Brunskill, “Flow clustering using machine learning techniques”, Proc. Passive and Active Measurement Workshop (PAM2004), Antibes Juan-les-Pins, France, April 2004
- [11] S. Zander, T. Nguyen, ve G. Armitage, “Automated traffic classification and application identification using machine learning”, IEEE 30th Conference on Local Computer Networks (LCN 2005), Australia, October 2005.

- [12] L. Bernaille, R. Teixeira, I. Akodkenou, A. Soule, ve K. Salamatian, “Traffic classification on the fly”, ACM Special Interest Group on Data Communication (SIGCOMM) Computer Communication Review, vol. 36, no. 2, 2006.
- [13] J. Erman, A. Mahanti, M. Arlitt, ve C. Williamson, “Identifying and discriminating between web and peer-to-peer traffic in the network core”, WWW ’07: Proceedings of the 16th international conference on World Wide Web. Banff, Alberta, Canada: ACM Press, May 2007, pp. 883–892.
- [14] J. Erman, M. Arlitt, A. Mahanti, “Traffic Classification Using Clustering Algorithms”, SIGCOMM06 Workshops, Pisa, Italy, (2006)
- [15] J. Erman, A. Mahanti, M. Arlitt, I. Cohen, and C. Williamson, “Semisupervised network traffic classification,” ACM International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS) Performance Evaluation Review, vol. 35, no. 1, pp. 369–370, 2007.
- [16] Erman, J., Arlitt, M., Mahanti, A.:Traffic Classification Using Clustering Algorithms. In: SIGCOMM06 Workshops, Pisa, Italy, (2006)
- [17] DBSCAN Wikipedia, <http://en.wikipedia.org/wiki/DBSCAN>
- [17] RFC 2616, Hypertext Transfer Protocol -- HTTP/1.1, <http://tools.ietf.org/html/rfc2616>
- [19] RFC 2818, HTTP over TLS, <http://tools.ietf.org/html/rfc2818>
- [20] RFC 821, Simple Mail Transfer Protocol, <http://tools.ietf.org/html/rfc821>
- [21] RFC 5321, Simple Mail Transfer Protocol (update), <http://tools.ietf.org/html/rfc5321>
- [22] RFC 1869, SMTP Service Extensions, <http://tools.ietf.org/html/rfc1869>
- [23] RFC 1939, Post Office Protocol – Version 3, <http://tools.ietf.org/html/rfc1939>
- [24] RFC 3921, Extensible Messaging and Presence Protocol, <http://tools.ietf.org/html/rfc3921>
- [25] RFC 114, A File Transfer Protocol, <http://tools.ietf.org/html/rfc114>
- [26] RFC 882, DOMAIN NAMES – CONCEPTS and FACILITIES, <http://tools.ietf.org/html/rfc882>
- [27] DEEP (Digital Evaluation and Exploitation) Paket İzleri, <https://domex.nps.edu/corp/scenarios/2009-m57/net/>, Department of Computer Science, Naval Postgraduate School, Monterey, CA, 2009

ÖZGEÇMİŞ

Cihangir Beşiktaş 1987 yılında İstanbul’da doğdu. 2001 yılında Cevizli İlköğretim Okulu’ndan mezun oldu. Lise öğrenimini 2005 yılında Maltepe Anadolu Lisesi’nde tamamlayan yazar, aynı yıl İstanbul Teknik Üniversitesi Bilgisayar Mühendisliği Bölümü’ne girmeye hak kazandı. Buradan ise 2009 yılında başarılı bir şekilde mezun olarak Gebze Yüksek Teknoloji Enstitüsü Bilgisayar Bilimleri Anabilim Dalı’nda yüksek lisans öğrenimine başladı. 2009 yılından beri TÜBİTAK Ulusal Elektronik Kriptoloji Araştırma Enstitüsü’nde araştırmacı olarak çalışmaktadır. Çalışma alanları sistem mühendisliği, ağ ve güvenlik teknolojileridir.