MAKİNE ÖĞRENME ALGORİTMALARINI KULLANARAK AĞ TRAFİĞİNİN SINIFLANDIRILMASI

HÜSEYİN AHMET YİĞİDİM

YÜKSEK LİSANS TEZİ

BİLGİSAYAR MÜHENDİSLİĞİ

TOBB EKONOMİ VE TEKNOLOJİ ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

EYLÜL 2012

ANKARA

Fen Bilimleri Enstitü onayı

_____

Prof. Dr. Ünver KAYNAK

Müdür

Bu tezin Yüksek Lisans derecesinin tüm gereksinimlerini sağladığını onaylarım.

_____

Doç. Dr. Erdoğan DOĞDU

Anabilim Dalı Başkanı

Hüseyin Ahmet YİĞİDİM tarafından hazırlanan MAKİNE ÖĞRENME ALGORİTMALARINI KULLANARAK AĞ TRAFİĞİNİN SINIFLANDIRILMASI adlı bu tezin Yüksek Lisans tezi olarak uygun olduğunu onaylarım.

_____

Yrd. Doç. Dr. Tansel ÖZYER

Tez Danışmanı

Tez Jüri Üyeleri

Başkan : Doç. Dr. Erdoğan DOĞDU _____

Üye : Doç. Dr. Bülent TAVLI _____

Üye : Yrd. Doç. Dr. Tansel ÖZYER _____

TEZ BİLDİRİMİ

Tez içindeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edilerek sunulduğunu, ayrıca tez yazım kurallarına uygun olarak hazırlanan bu çalışmada orijinal olmayan her türlü kaynağa eksiksiz atıf yapıldığını bildiririm.

……

Hüseyin Ahmet YİĞİDİM

Üniversitesi : **TOBB Ekonomi ve Teknoloji Üniversitesi**
Enstitüsü : **Fen Bilimleri**
Anabilim Dalı : **Bilgisayar Mühendisliği**
Tez Danışmanı : **Yrd. Doç. Dr. Tansel ÖZYER**
Tez Türü ve Tarihi : **Yüksek Lisans – Eylül 2012**

Hüseyin Ahmet YİĞİDİM

MAKİNE ÖĞRENME ALGORİTMALARINI KULLANARAK AĞ TRAFİĞİNİN SINIFLANDIRILMASI

ÖZET

İnternetin hayatımıza girmesi ile birlikte ağ kaynaklarının verimli bir şekilde kullanılabilmesi için ağ trafiğinin yönetilmesi ve ağ akışlarının kontrol edilmesi kaçınılmaz olmuştur. Geleneksel sorgu ve raporlama araçlarının günümüzde yetersiz kalması nedeniyle toplanan bu verileri nasıl kullanılacağı üzerine yapılan araştırmalar makine öğrenme ve veri madenciliğinin kavramlarının hayatımıza daha çok girmesini sağlamıştır.

Ağ analizi için kullanılan geleneksel tabanlı yaklaşımlar (port-tabanlı, yük-tabanlı) internet kullanım davranışlarının ve teknolojilerinin değişmesiyle günümüzde yetersiz kalmaya başlamıştır. Bu nedenle; Makine öğrenme tabanlı ağ trafiği sınıflandırma yöntemi geliştirilmiştir.

Tez çalışmasında, ağ trafiği veri setini kullanarak, akışlar üzerinden ağ akışlarına ait nitelik kümeleri çıkartılarak, sınıflandırma ve kümeleme analizinde sıkça kullanılan C4.5, Naive Bayes, EM ve K-means algoritmaları, karşılaştırma metrikleri kullanılarak test edilecektir. Ayrıca algoritmaların performanslarını yükseltmek için kullanılan Adaboost algoritmasının sınıflandırma algoritmaları üzerindeki etkisi incelenecektir. Oluşturulan nitelik kümesindeki kullanılmayan veya gereksiz niteliklerin çıkarılmasını sağlayan nitelik seçme algoritmaları kullanılarak, en uygun nitelik sınıfı aranacak ve algoritmaların performansları üzerindeki etkisi araştırılacaktır.

Anahtar Kelimeler: Makine öğrenme, Veri madenciliği, Ağ trafiği sınıflandırma, Nitelik Seçimi, C4.5, Naive Bayes, EM, K-means, Adaboost

University : **TOBB Economics and Technology University**
Institute : **Institute of Natural and Applied Sciences**
Science Programme : **Computer Engineering**
Supervisor : **Associate Professor Tansel ÖZYER**
Degree Awarded and Date : **M.Sc. – September 2012**

Hüseyin Ahmet YİĞİDİM

NETWORK TRAFFIC CLASSIFICATION USİNG MACHİNE LEARNING ALGORITHMS

ABSTRACT

With beginning of extensive usage of internet on our daily lives, to control network flows and manage network traffic for efficient usage of network sources has become more and more important. Traditional query and reporting tools started to become insufficient, so, researches done to decide how these data will be used has made machine learning and data mining concepts to have much more value in our lives.

Traditional approaches performed for network analysis (port-based, payload-based) has become insufficient with the changes on network usage behaviors and technology. Therefore, Machine Learning based network traffic classification technique has been developed.

On this thesis study, by the usage of network traffic data set, attribute set for network flows will be calculated, and C4.5, Naïve Bayes, EM and K-means algorithms which are used frequently on classification and clustering analysis will be tested using comparison metrics. Also, effects on classification algorithms of Adaboost, used for boosting algorithms performances will be tested. Most convenient attribute set will be determined by using attribute selection algorithms which helps to be removed unused and unnecessary attributes on generated attribute set. Effects of this attribute set on algorithms' performance will be researched.

Keywords: Machine Learning, Data mining, Network Traffic Classification, Feature Selection, C4.5, Naive Bayes, EM, K-means, Adaboost

TEŞEKKÜR

Çalışmalarımda yardımcı olan ve katkılarıyla yönlendiren tez hocam Yrd. Doç. Dr. Tansel ÖZYER'e ve yüksek lisans eğitimim sırasında tecrübelerinden yararlandığım saygı değer TOBB Ekonomi ve Teknoloji Üniversitesi öğretim üyelerine,

Bana hayatım boyunca her zaman her konuda destek olan aileme teşekkürlerimi sunarım.

İÇİNDEKİLER

ÇİZELGELERİN LİSTESİ

Çizelge **Sayfa**

ŞEKİLLLERİ LİSTESİ

Şekil **Sayfa**

KISALTMALAR

Kısaltmalar	Açıklama
IEEE	Elektrik Elektronik Mühendisleri Enstitüsü
ICDM	Uluslararası Veri madenciliği Konferansı
DARPA	Savunma İleri Araştırma Projeleri Dairesi
IANA	İnternet Atanmış Numaralar Otoritesi
CAIDA	İnternet Veri Analizi Kooperatif Birliği
ITU	Uluslararası Telekomünikasyon Birliği
ICT	Bilgi ve İletişim Teknolojisi
NLANR	Gelişmiş Ağ İnternet Araştırma Laboratuarı
WAND	Waikato Üniversitesi Araştırma Grubu
EM	Beklenti Maksimizasyonu Algoritması
CFS	Korelâsyon-tabanlı Özellik Arama Algoritması
CON	Tutarlılık-tabanlı Alt Küme Arama Algoritması
TCP	Transmisyon Kontrol Protokolü
UDP	Kullanıcı Veri Bloğu İletişim Protokolü
ICMP	İnternet Kontrol Mesaj İletişim Protokolü
LAN	Yerel Ağ
FTP	Dosya Transfer Protokolü
SSH	Uzak Makine Bağlantı Sağlama Protokolü
HTTP	Hiper Metin Transfer Protokolü
HTTPS	Güvenli Hiper Metin Transfer Protokolü
DNS	Alan Adı Sistemi
SMTP	Basit E-posta Gönderme Protokolü
POP3	Postane Protokolü
IMAP	İnternet Mesaj Erişim Protokolü
SOCK	Network Paketlerini Rotalama Protokolü
NNTP	Ağ Haber Aktarım İletişim Kuralı
IRC	İnternet Chat Protokolü
NTP	Ağ Zaman Protokolü
SNMP	Basit Ağ Yönetim Protokolü
P2P	Veri Paylaşma Protokolü
NetAI	Ağ Trafiği Tabanlı Uygulama Tanımlama Paketi
NeTraMet	Network Trafik Ölçüm Yazılımı
URG	Acil İşaret Bayrağı
PSH	İtme Fonksiyon Bayrağı

1. GİRİŞ

Bilişim Teknolojilerinde yaşanan olumlu gelişmeler; Artan ve gelişen işlemci, bellek, sabit disk, … hız ve kapasiteleri, azalan güç tüketimi, ağırlık, maliyetler, … ile birlikte, insanlar ile bilgisayarlar arasındaki bağ her geçen gün biraz daha artmaya başlamıştır. Sadece basit hesaplamalarda kullanılan ve devasa büyüklükte olan bilgisayarların yerini alan yeni bilgisayar veya bilgisayar destekli teknolojiler ile birlikte, neredeyse günlük yaşamdaki tüm işlem ve problemler bilgisayar desteği ile çözümlenmektedir. İnsanlar, daha büyük ve daha çok miktarda veriyi dijital ortamda saklayabilmekte ve bu veriler üzerinde daha hızlı işlem yapabilmektedirler.

Veri; Temel bir tanım olarak, işlenmemiş ham kayıtlar şekilde ifade edilebilir. Saklanan verilerin kaynağı; insanlar, bitkiler, coğrafi değerler, yazılı metinler, … kısaca canlı veya cansız varlıklara ait her türlü nümerik veya nominal kayıtlar olabilir. Her geçen zaman diliminde de ortaya çıkan bu veri miktarı giderek artmaktadır.

Veriler, tek başlarına anlam ifade etmeye bilirler. Veriyi belli bir amaca göre işleyerek anlam kazandırma işlemine *veri analizi*, ortaya çıkan işlenmiş (anlaşılabilir) veriye de *bilgi* denir.

Zaman içerisinde teknolojinin gelişmesi, yapay zekâ, makine öğrenme gibi kavramların yaygınlık kazanması ile oluşan akıllı algoritmaları ve istatistiksel yöntemleri kullanarak, veri yığınları arasından keşfedilecek bilgilerin olduğunun düşüncesi üzerine yapılan çalışmalar sayesinde *veri madenciliği ve bilgi keşfi* kavramları ortaya çıkmaya başlamıştır.

Bilgisayar ağlarındaki ilerlemeler ile birlikte, yeni ve mevcut verilere başka bilgisayarlar aracılığı ile daha hızlı ulaşabilme, verileri diğer bilgisayarlar ile dijital ortamlarda paylaşabilme imkânı doğmuştur.

1.1. Problemin Tanımı

Artan veri miktarının kontrolü ve yönetilmesini sağlamak için geliştirilen veritabanı sistemlerinin kullanımı giderek yaygınlaşmakta ve hacimlerindeki olağanüstü artış, raporlama ve sorgu araçlarının yetersiz kalması nedeniyle, toplanan bu verilerden nasıl faydalanılabileceği sorusunu ortaya çıkarmaktadır.

İnternet gibi, dünya üzerindeki birçok bilgisayar ağının birbirleri ile bağlanması sonucu ortaya çıkan, gün geçtikte yaygınlaşan ve sürekli büyüyen iletişim ağının hayatımıza girmesi ile birlikte veriler, daha da çok artan, ulaşılması kolay, kontrol edilmesi ve analizi zor hâle dönüşmüştür.

Ağ analizi için kullanılan geleneksel tabanlı yaklaşımlar (port-tabanlı, yük-tabanlı) internet kullanım davranışlarının ve teknolojilerinin değişmesiyle günümüzde yetersiz kalmaya başlamıştır.

Ağ üzerinde dolaşan bu veri trafiğinin verimli bir şekilde gerçekleşmesini sağlamak, ağ verileri üzerinden kullanıcı analizlerini yapabilmek, ağ kaynaklarının yönetilmesi ve planlanması sağlamak veya ağ üzerindeki anormalliklerin ve saldırıların tespitini gerçekleştirmek için, yeni yaklaşımların ve yeni araçların geliştirilmesi ihtiyacı kaçınılmaz oluşmuştur.

1.2. Amaç

Ağ üzerindeki ağ akışlarını ve ağ protokollerini kullanarak, makine öğrenme algoritmalarının başarım karşılaştırılmalarının yapılması ve ağ trafiğinin sınıflandırılması tezin genel amacıdır.

Farklı veri setleri üzerinden sınıflandırma ve kümeleme algoritmaları performansları incelenerek, veri seti içerisinde kullanılan ağ protokolleri ve ağ protokol grupları hakkında sonuç çıkarımları yapılacaktır.

Nitelik seçme algoritmaları ve seçme yöntemleri anlatılarak, veri setlerinde kullanılan nitelik kümeleri üzerinde incelemeler yapılacaktır.

Sıklıkla bir arada kullanılan ve birçok noktada örtüşen, dolayısıyla karıştırılan, makine öğrenmesi ve veri madenciliği kavramları hakkında bilgi verilerek kavram kargaşalığı giderilmeye çalışılacak ve internet ağları hakkında temel bilgilerin verilmesi sağlanacaktır.

Tezin geri kalan anlatımlarında aşağıdaki şekilde bir organizasyon planı izlenmektedir.

İkinci bölümde; Makine öğrenme ve veri madenciliği kavramları hakkında temel bilgiler, bu kavramların alt başlıklar altında incelenmesi, tezde kullanılacak algoritmaların anlatımları yapılmaktadır.

Üçüncü bölümde; İnternet ağlarında iletişimin nasıl gerçekleştiği, TCP/IP internet mimarisi ve ulaşım katmanı protokolleri, bağlantı türleri ve IP trafik akışının nasıl gerçekleştiği üzerine tanımlamalar yapılmaktadır.

Dördüncü bölümde; Literatürde anlatılan ve güncel olarak kullanılan ağ trafiği sınıflandırma yöntemlerinin artı ve eksi yönleri anlatılmakta, tez çalışması ile benzer özelliklere sahip literatür çalışmalarına değinilmektedir.

Beşinci bölümde; Makine öğrenme algoritmaları başarımı değerlendirme metrikleri, tez çalışmasında uygun olacak şekilde anlatılmaktadır.

Altıncı bölümde; Veri seti içerisindeki gereksiz niteliklerin belirlenmesi ve çıkarılmasını sağlayan nitelik seçimi algoritmaları ve nitelik seçim yöntemleri anlatılmaktadır.

Yedinci bölümde; Tez çalışmasında kullanılan ham veri seti ve yazılımların incelenmesi, veri setinden akış çıkarım yöntemi ve veri setine ait nitelik kümesi

elemanlarının tanımlamaları, tezde kullanılan yazılım içerisindeki makine öğreneme algoritmaları ve nitelik seçimi yöntemleri anlatılmaktadır.

Sekizinci bölümde; Tez çalışmasında kullanılan özel veri setleri ve test metodolojisi üzerine anlatımlar, deneysel gözlem ve analizler bulunmaktadır.

Dokuzuncu bölümde ise; Tez çalışması ait genel sonuçlar ve tez üzerinden gelecekte yapılabilecek çalışmalar anlatılmaktadır.

2. MAKİNE ÖĞRENMESİ VE VERİ MADENCİLİĞİ

Tarihsel süreçte farklı coğrafyalarda yaşayan farklı kültürdeki insanlar bile nasıl birbirleri ile birçok ortak noktada buluşup, etkileşimde bulunarak kendileri geliştirebiliyorlarsa, bilgisayar bilimleri, mühendislik ve istatistik gibi bilim dallarının araştırmalarıyla ortaya makine öğrenmesi ve veri madenciliği alanları da birçok ortak buluşup zamanla geliştirilmiş kavramlardır.

Veri analizi kavramının ve daha akıllı bilgisayarlar üretilebileceği düşüncesinin önem kazanması ile sürekli geliştirilen bu iki kavram, birbirlerinden farklı olmasına rağmen, birbirleri ile birçok açıdan örtüşmesi nedeniyle aynı kavramlarmış gibi algılanabilmektedir. [1][4][5]

2.1. Makine Öğrenmesi

Bilgisayar ve bilgisayar sistemlerinin ile aramızdaki bağın her geçen gün daha da artması ile bilgisayarın da insanlar gibi öğrenme işlemini gerçekleştirip gerçekleştiremeyeceği merak konusu olmuştur.

Yapay zekâ ve istatistik alanında yapılan önemli çalışmalar neticesinde öğrenme işlemini gerçekleştirebilecek algoritmalar geliştirilmiştir ve makine öğrenmesi kavramı hayatımıza girmeye başlamıştır.

Öğrenme, kelime anlamı olarak; Frank ve Witten'a göre *"Davranışlarını gelecekte daha iyi olacak şekilde değiştirme"* [1] veya Simon'un 1983 yılında dediği gibi "*Bir sistem içerisindeki aynı soruna(göreve) bir sonraki seferde daha etkili ve verimli bir şekilde adapte olabilme davranışı içersinde bulunmadır.*" [2] şeklinde tanımlanabilir. Yukarıdaki tanımdan yola çıkarak; Makine öğrenmesi kavramı, makinelerin karşılaştıkları durumlar karşısında kendini eğiterek daha iyi kararlar verebilmesini sağlayan algoritmaların geliştirilmesi olgusudur.

2.1.1. Makine Öğrenme Algoritmaları

Makine öğrenmesi algoritmaları farklı kaynaklarda farklı alt gruplara ayrılarak incelenebilmektedir.

Algoritmanın istenen sonucuna dayalı olarak sınıflandırılmasına göre, makine öğrenmesi algoritmaları aşağıdaki başlıklar altında incelenebilir. [3]

- **Denetimli Öğrenme** (*Supervised Learning*) **:** Algoritma girdileri istenen çıktılara göre etiketleyen bir fonksiyon oluşturarak öğrenme işlevini gerçekleştirir.
- **Denetimsiz Öğrenme** (*Unsupervised Learning*) **:** Girdilere bağlı olmadan çıkarım yapılması işlevidir.
- **Yarı-denetimli Öğrenme** (*Semi-supervised Learning*) : Hem etiketlenip hem de etiketlenmeyen örnekler beraber incelenerek uygun fonksiyon ve sınıflandırıcı oluşturulur.
- **Takviyeli (Pekiştirici) Öğrenme** (*Reinforcement Learning*) **:** Algoritma, dünya algısına dayalı öğrenme biçiminde olduğu gibi işlemini gerçekleştirir. Her eylem ortamda bir etki oluşturmakta ve ortam, öğrenme algoritmasına yol gösteren geri beslemeler vermektedir.
- **Transdüksiyon** (*Transduction*) **:** Denetimli öğrenmeye benzerdir. Fakat açıkça bir fonksiyon inşa etmez. Bunun yerine; Denenen girdilere, denenen çıktılara ve yeni girdilere bakarak yeni çıktılar hakkında tahmin yürütür.

- **Öğrenmeyi Öğrenme** (*Learning to Learn*) **:** Algoritma daha önceki deneyimlerine dayanarak tümevarımsal bir öğrenme gerçekleştirir.

Özellikle; Veri madenciliğinin gelişmesinde önemli rol oynayan ve sıklıkla kullanılan, denetimli öğrenme (*supervised learning*), denetimsiz öğrenme (*unsupervised learning*) ve yarı-denetimli öğrenme (*semi-supervised learning*) algoritmalarının anlatımları ve öğrenme işlevlerinin nasıl gerçekleştiği hakkındaki açıklamalar ilerleyen bölümlerde incelenen algoritmalar üzerinden daha detaylı olarak yapılacaktır.

Çeşitli gruplama şekilleri altında makine öğrenme algoritmaları [1][3-7] kaynaklarından daha detaylı olarak incelenebilir.

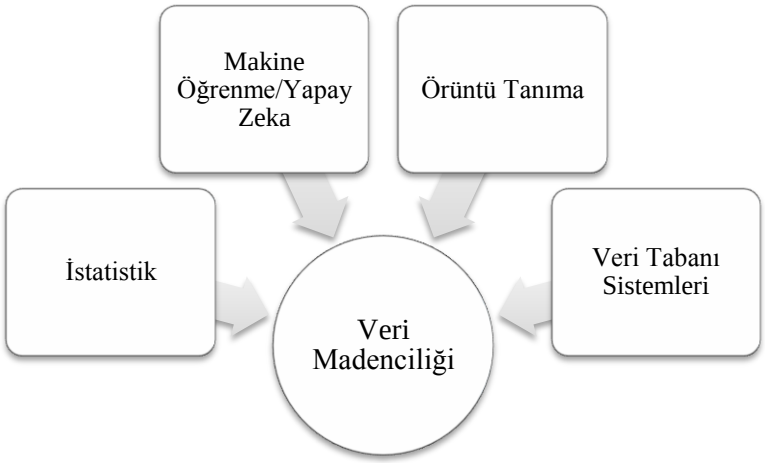
2.2. Veri Madenciliği

Hızla gelişen teknolojiler ve ucuzlayan maliyet nedeniyle günümüzde veriler anormal bir hızla artmakta (saatte gigabyte'lar seviyesinde), ortaya çıkan bu veriler toplanmakta ve depolanmaktadır.

İnternet, uydu ve gökyüzü teleskop verileri, banka ve kredi kartı verileri, genleri açıklamak için üretilen mikro diziler, bilimsel simülasyon verileri terabyte'lar seviyesini aşmıştır. [6]

Ortaya çıkan bu veri yığınları içerisinden yeni ilişkilerin çıkarılması düşüncesi ve geleneksel kullanılan veritabanı sorgu metotlarının ve raporlama araçlarının bu tarz verilerin detaylı incelenmesinde yetersiz kalmaları nedeniyle yeni analiz yöntemleri ihtiyacı oluşmuştur.

İstatistik, makine öğrenme, yapay zekâ, örüntü tanıma gibi çalışma alanlarının etkisiyle zamanla ortaya veri madenciliği kavramı çıkmıştır. [6][7]



Şekil 2.1 Veri madenciliğinin diğer kavramlarla ilişkisi

Yukarıdaki anlatımlardan yola çıkarak veri madenciliği; Büyük veri yığınları içersinden, geleneksel yöntemler kullanılarak çıkarılamayacak gizli kalmış bilgilerin, istatistiksel yöntem analizleri ve makine öğrenme algoritmalarını kullanarak ortaya çıkarılmasını sağlayan kavramdır. [7]

Daha basit bir ifade ile veri madenciliği; Eldeki, üstü kapalı çok net olmayan verilerden, kullanışlı bilgilerin çıkarılmasıdır.

Örneğin; Bir telefon rehberinde bazı isimlerin bazı yörelerde daha çok kullanılması bilgisini ortaya çıkaran kavrama veri madenciliği denir. Telefon rehberinden bir numarayı aramak veri madenciliği değildir. [6]

2.2.1. Veri Madenciliği Algoritmaları

Veri madenciliği algoritmaları da makine öğrenme algoritmaları gibi kendi aralarında farklı gruplara ayrılmasına rağmen, analiz yöntemlerine ve gördükleri işlevlere göre birbirinden aşağıdaki şekilde ayrılabilirler. [5-7]

- **Sınıflandırma** (*Classification*)**:** Veriyi önceden belirlenmiş sınıflara atarak tüm kayıtları doğru etiketlemeyi amaçlar. Verinin dağılımına göre bir değer bularak, niteliğin bilinmeyen veya gelecekteki değerinin tahmin edilmesinde kullanılır.

- **Kümeleme** (*Clustering*)**:** Hangi verinin hangi sınıfta olduğu ve sınıf sayısı bilinmez. Veriler gruplara ayrılarak incelenir. Amaç, aynı grup içerisinde nesnelerin birbirine benzer veya ilişkili olması, farklı gruptaki nesnelerin ise birbirinden farklı olması veya ilişkilerinin bulunmamasıdır.

- **Birliktelik Kuralı** (*Association Rule*)**:** Veri kümesi içerisindeki örüntülerin, nitelikler arasındaki gizli kalmış ilişkilerin çıkarılmasını sağlar. Diğer bir ifade ile bir niteliğin ortaya çıkma olasılığının, diğer niteliğin ortaya çıkma olasılığından tahmin edilmesidir.

- **Regresyon** (*Regression*)**:** Hedef değişken ile giriş değişkenleri arasında doğrusal veya doğrusal olmayan bir ilişki vardır. Diğer değişkene dayalı olarak sürekli değerli hedef değişken tahmin edilmek istenmektedir.

Yukarıdaki gruplama şekillerine bağlı olarak bu gruplara ait birçok alt grup ve bu alt gruba ait birçok veri madenciliği algoritması mevcuttur.

Örneğin; Kümeleme analizi kavramı; Bölümlemeli, hiyerarşik, yoğunluk tabanlı ve model tabanlı olmak üzere dört alt gruba ayrılabilir. Bölümlemeli kümelere algoritmalarına örnek olarak da K-means, CLARA, CLARANS ve PAM algoritmaları verilebilir.

2.3. Tezde Kullanılacak Algoritmalar ve Kavramlar

Tezin genel kapsamında bütün veri madenciliği ve makine öğrenmesi kavramlarının alt grupların açıklamalarına ve bu alt gruplara ait algoritmaların incelenmesine algoritma sayıların çok fazla olması nedeniyle girilmemiştir. Algoritmalar ve kavramlar hakkında detaylı bilgi için [1] [3-7] kaynakları incelenebilir.

Tez içerisinde kullanılacak algoritmalar, Aralık 2006 yılında Hong Kong'da yapılan IEEE ICDM'06 konferansında açıklanan en etkili 10 veri madenciliği algoritması arasından seçilmiştir. [8][9]

Tez kapsamında; Bu en iyi bilinen ve en çok kullanılan algoritmalar arasından seçilerek incelenecek algoritmaların, makine öğrenmesi ve veri madenciliğinde, farklı gruplara ait özellikler taşımalarına, algoritmalarının birbirleri ile karşılaştırmalarının yapılabilip yapılamayacağına dikkat edilerek seçilmiştir.

Tezde kullanılacak algoritmalar hakkında temel bilgiler ve algoritmaların çalışma prensipleri sırasıyla alt bölümlerde anlatılmaktadır. Daha detaylı bilgi almak için; IEEE ICDM'06 konferansı inceleme makalesi [8] ve [1][6-7] kaynakları incelenebilir.

2.3.1. K-Means Algoritması

En popüler kümeleme algoritmalardan biridir. Veri madenciliğindeki kümeleme analizi işlemlerinde kullanılır. Algoritma, bölümleme tabanlı kümeleme yapmaktadır ve analiz sırasında kendisini denetimsiz öğrenme yöntemini kullanarak geliştirir. [7]

K-means algoritması sayısal veriler kullanarak çalışan bir algoritmadır. Kullanılan veri setinin belirlenen k adet kümeye bölünmesi prensibine dayanır.

Algoritma adımları aşağıdaki gibidir.

1. Küme merkezleri belirlenir. Bu belirleme işlemi veriler arasından küme sayısı olan k adet noktanın, rastgele seçilmesi veya tüm verilerin ortalaması alınması yöntemiyle yapılabilir.

2. Her verinin seçilen merkez noktalara olan uzaklıkları hesaplanır. Elde edilen sonuçlara göre tüm veriler k adet kümeden kendisine en yakın olana yerleştirilir.

3. Ortaya çıkan kümelerin yeni merkez noktaları o küme ait tüm verilerin ortalama değerleri ile değiştirilir.

4. Merkez noktalar değişmeyene kadar 2. ve 3. adımlar tekrarlanır.

İkinci adımdaki, tüm verilerin kendisine en yakın olan kümeye yerleştirmesi işlemi sırasında farklı benzerlik ölçütleri kullanabilir. Tez kapsamında Öklid uzaklığı kullanılmıştır. [19]

K-means algoritması katı (*hard*) kümeleme yöntemini kullanmaktadır. Bir veri yalnızca bir kümeye aittir ve kümeler birbirlerinden kesin çizgiler ile ayrılmaktadır.

K-means algoritması denetimsiz öğrenme yöntemini kullanması nedeniyle algoritmaların girdi verileri hakkında bilgi sahibi olmadan kümeleme işlemini gerçekleştirir. Dolayısıyla; Küme sayısı hakkında önbilgi olması veya araştırmacının anlamlı olacak küme sayısına karar vermiş olması durumunda tercih edilmesi daha doğrudur. Ayrıca; K-means algoritması, gürültülü veya aşırı uç verilerde zayıf bir performans göstermektedir.

2.3.2. Beklenti Maksimasyonu (EM) Algoritması

EM algoritması genellikle veri madenciliğindeki kümeleme analizi kavramı altında incelenen algoritmalardandır ve model tabanlı kümeleme yapmaktadır. Algoritma yarı-denetimli öğrenme metodunu kullanarak öğrenme işlemini gerçekleştirmesi nedeniyle hem işaretlemiş hem de işaretlenmemiş verileri kullanarak işlem yapmaktadır. Dolayısıyla, hem sınıflandırma hem de kümeleme işlemlerinde kullanılabilir bir yapıya sahiptir. [20]

EM algoritması K-means algoritmasından farklı olarak, olasılıksal model oluştururken, küme sayısını ve her kümenin farklı olasılık dağılımlarını yöneten parametreleri tahminsel olarak kendisi belirler. [21]

EM algoritması ve K-means algoritması arasındaki diğer bir fark, verinin hangi kümeye dâhil edildiğinin tespitidir. EM algoritması yumuşak (*soft*) kümeleme yöntemini kullanmaktadır. Oluşan kümeler genellikle birbirlerinden kopuk bir yapıda değildirler.

Bir veri noktası, kümeler arasında geçiş noktası içerisinde olması veya veri uzayında boyut sıkıntısı yaşanması nedeniyle başka bir veri ile üst üste binmiş gibi algılanabilir. Bu nedenle; Bu veri noktası tahminsel olarak birçok kümenin elemanı olabilir. Dolayısıyla; Her veri noktası ve küme birleşimi için bir olasılık hesaplaması yapılır.

EM algoritması temelde iki adımdan oluşmaktadır. Bunlar; Beklenti (*expectation*) ve Maksimizasyon (*maximization*) adımlarıdır.

İlk beklenti adımında sözde rastlantısal sayıları, daha sonraki beklenti adımlarında da gözlenen verileri kullanılarak, parametre değerleri tahmin edilir ve beklenen logaritmik olasılık fonksiyonu oluşturulur. Maksimizasyon adımında, ortalama ve varyans kullanılarak beklenti adımında oluşturulan fonksiyonun parametrelerinin maksimizasyonu yapılır. Bulunan beklenen fonksiyon ile önceki parametrelerin fonksiyonları karşılaştırılır. Eğer; Fark varsa, hesaplanan parametre ile önceki parametreler yer değiştirilerek yeniden hesaplanır. Fark yoksa algoritma durdurulur.

2.3.3. C4.5 Algoritması

Veri madenciliğindeki sınıflandırma analizi kavramı altında incelenen, denetimli öğrenme metodunu kullanan tek değişkenli bir karar ağacı algoritmasıdır. [22] Basit karar ağacı oluşturmada kullanılan ID3 algoritmasının geliştirilmiş versiyonu olan

C4.5 algoritması, sınıflandırma yeteneği nedeniyle İstatistiksel Sınıflandırıcı olarak da bilinmektedir. [23]

ID3 algoritmasından farklı olarak C4.5 algoritması, nitelikler sayısal değere sahipse veya eksik değer varsa da kullanılabilir. Algoritma, oluşturulan karar ağacının gereksiz dallarını da tespit edebilme özelliğine sahiptir.

Karar ağacını oluşturma işlemini, niteliklerin entropi değerlerini hesaplayıp bilgi kazançlarını ölçerek yapmaktadır.

Verilen bir t düğümü için $P(j|t)$ t düğümdeki j sınıfına ait bağıl olasılık olmak üzere;

$$Entropy = -\sum_{j} P(j|t)\log P(j|t) \qquad (2.3.3.1)$$

P, k sayıya ayrılmış düğüm noktası ve n_i i. kısım içerisindeki kayıt sayısı olmak üzere bilgi kazancı;

$$GAIN_{split} = Entropy(P) - \left(\sum_{i=1}^{k} \frac{n_i}{n} Entropy(i) \right) \qquad (2.3.3.2)$$

formülünü kullanılarak hesaplanabilir. [6]

Bilgi kazancı en yüksek olan nitelik ağacın köküne yerleştirilir. Daha sonra ağacın düğümlerine ait alt kümelerinin bilgi kazançları hesaplanarak alt düğümler ve yapraklar oluşturulur. Bu durum; Örneklerin hepsi aynı sınıfa ait oluncaya veya örnekleri bölecek özellik kalmayıncaya kadar devam eder.

Eğer; C4.5 algoritması sayısal değerler ile çalışıyorsa, niteliğin değerleri sıralanarak verilerin orta noktası bulunup bir eşik değer tespit edilir. Nitelik değerleri, bu eşik değer ile karşılaştırılarak büyük veya küçük eşit olmak üzere ikiye ayrılır.

2.3.4. Naive Bayes Algoritması

Naive Bayes algoritması, veri madenciliğindeki sınıflandırma analizi işlemlerinde kullanılan bir algoritmadır. Denetimli öğrenme yöntemini kullanarak öğrenme işlemini gerçekleştirir.

Bayes teoremi tabanlı bir algoritmadır. [1] Basit Bayes, Bağımsız Bayes olarak da adlandırılmaktadır. [8] Daha önce gerçekleşen olayların gerçekleşme olasılıklarına bakarak, incelenen olayın gerçekleşme olasılığı hakkında tahminde bulunulmasını sağlar.

$X = (x_1, x_2, \ldots, x_n)$ nitelik vektörü, $C = (c_1, c_2, \ldots, c_m)$ veri kümesinde sınıflar olmak üzere, Bayes teoreminden;

$$P(c_i|X) = \frac{P(X \mid c_i)P(c_i)}{P(X)} \qquad (2.3.4.1)$$

elde edilir. Bayes teoremi sınıflandırma amaçlı kullanılırken olasılığı en yüksek olan durum hedef sınıf olarak seçilir.

$$\arg\max_{c_i \in C} P(c_i \mid X) \qquad (2.3.4.2)$$

(2.3.4.1) formülünde $P(X)$ olasılığı bütün sınıflar için sabit olduğundan göz ardı edilerek;

$$P(c_i|X) = P(X \mid c_i)P(c_i) \qquad (2.3.4.3)$$

olarak türetilebilir.

Naive Bayes algoritması, niteliklerin birbirlerinden bağımsız ve aynı derecede önemli olduğu kabulüne dayanır. Bu nedenle; (2.3.4.3) denkleminde,

$$P(X|\mathrm{c}_i) = \prod_{k=1}^{n} P\left(x_k|\mathrm{c}_i\right) \qquad (2.3.4.4)$$

denklemi yerine yazılarak düzenlendiğinde, hedef sınıf formülü (2.3.4.2) den;

$$\arg\max_{c_i \in C} \prod_{k=1}^{n} P\left(x_k|\mathrm{c}_i\right) P(c_i) \qquad (2.3.4.5)$$

olarak bulunur.

Veri setindeki niteliklerin nümerik olması durumunda nitelik uzayındaki değerlerin Gauss dağılımına (2.3.4.6) sahip olduğu varsayılır. Aranan olasılık değeri, nitelik değeri (x), ortalama (μ) ve standart sapma (σ) değerlerine bağlı olarak hesaplanır.

$$f(x) = 1/(\sqrt{2\pi}\sigma)\, e^{-(x-\mu)^2/(2\sigma^2)} \qquad (2.3.4.6)$$

2.3.5. Adaboost Algoritması

Yükseltme (*boosting*) tekniğini kullanarak zayıf öğrenme algoritmalarının performanslarını arttırılmasını sağlayan bir sınıflandırma algoritmasıdır. Nitelik uzayındaki her bir nitelik üzerinde bir zayıf sınıflandırıcıyı eğitilir ve zayıf sınıflama fonksiyonlarını bir araya getirip, doğrusal olarak birleştirerek güçlü sınıflandırıcıların oluşturulmasını sağlar.

Öğrenme kümesi üzerinde her örneğin bir ağırlığı vardır. Her bir nitelik için zayıf sınıflandırıcı eğitildikten sonra yanlış sınıflandırılan örneklerin ağırlıkları arttırılırken doğru sınıflanan örneklerin ağırlıkları azaltılır. Daha sonra bu ağırlıklar yardımı ile genel hata hesaplanır ve bir sonraki nitelik üzerinde yeni zayıf sınıflandırıcı bu hata değeri göz önünde bulundurularak eğitilir.

Adaboost algoritması güçlü algoritmalar kullanılarak da eğitilebilir. Fakat bu durumda, gelişim daha az olduğu gözlenmiştir.

Bu algoritma tez kapsamında, sınıflandırma algoritmaları ile çalıştırılarak algoritmaların performanslarındaki gelişmeler gözlemlenecektir.

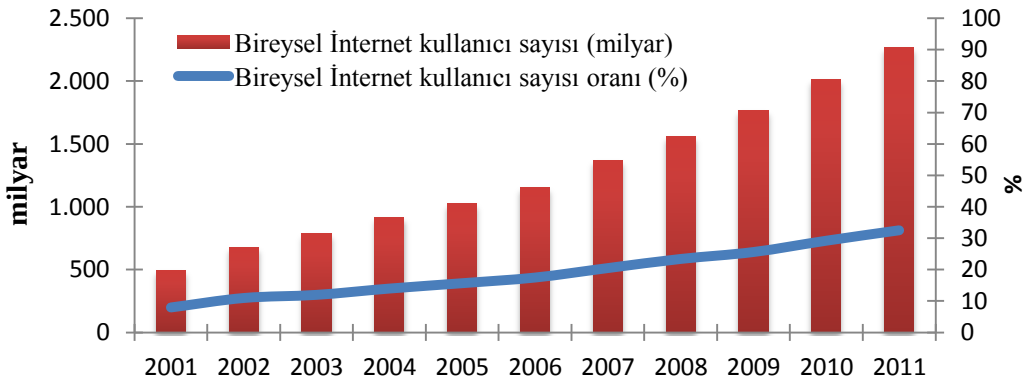
3. İNTERNET AĞLARINDA İLETİŞİM

DARPA'nın askeri amaçlar için kullanılması planlanan 1960'lı yıllarının sonralarına doğru geliştirdiği ARPANET ağının oluşturulması ile günümüz internet ağlarının temelleri atılmıştır. [10]

İnternet; Yerel ağların birleşmesi ile ortaya çıkan bir ağ topluluğudur ve herhangi bir ağ grubu interneti oluşturabilir. Günümüzde kullandığımız tanımla; Dünya üzerindeki birçok yerel ağın bağlanması ile oluşan haberleşme ağı olarak ifade edilir.

ITU'nun ICT verilerine göre, küresel çapta yapılan araştırmada 2001-2011 yılları arasında dünyadaki bireysel internet kullanıcı sayısı ve bireysel internet kullanıcılarının tüm dünya nüfusuna oranı giderek artmıştır. [11]

Tablo 3.1 Bireysel İnternet kullanıcı sayısı ve dünya nüfusuna oranı

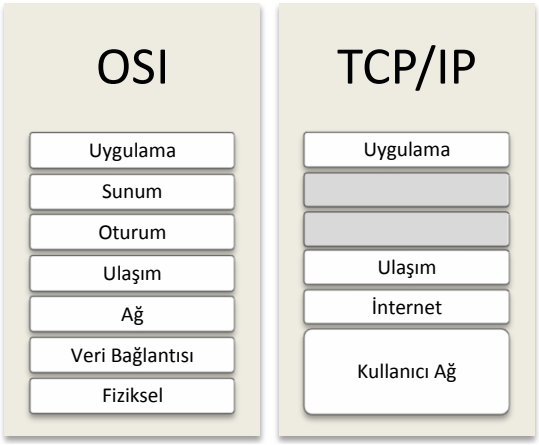


Kaynak: ITU World Telecommunication /ICT Indicators veritabanı

İnternet kullanımıyla ortaya çıkan olağanüstü büyüklükteki veri yığınları, tez için istenilen veri setinin kaynağının oluşturulmasını sağlamaktadır.

İnternet iletişimi belli katmanlardan ve protokol takımları üzerinden gerçekleşmektedir. Bu internet protokolleri takımı, TCP/IP protokol takımı olarak da adlandırılmaktadır.

Haberleşme ağlarında iki tür model kullanmaktadır. Bunlar; OSI ve TCP/IP modelleridir.



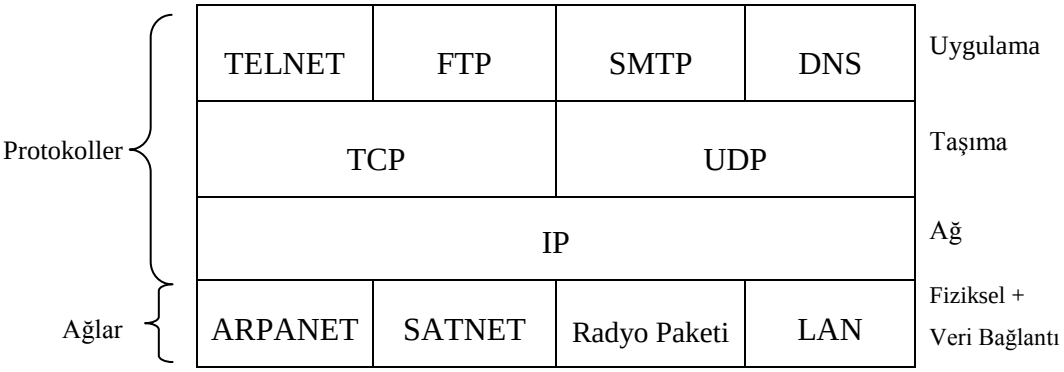
Şekil 3.1 OSI ve TCP/IP modelleri

OSI modeli ile ilişkilendirilen protokoller günümüzde kullanılmamasına rağmen modelin kendisi halen geçerli bir yapıya sahiptir.

TCP/IP modelinde ise; Model çok fazla kullanım bulmazken, protokoller geniş bir kullanıcı kitlesi tarafından kullanılmaktadır.

Günümüzde kullandığımız TCP/IP mimarisi bu iki yaklaşımdan hareket ile dört katmanda incelenebilir. Her katmanın değişik görevleri vardır ve katmanlar arası gerekli bilgi alış verişi sağlanabilir. Yollanan veriler katmanlara göre paketlenerek yollanır. Yollanan verinin gönderilme şekli ve yolu birbirinden farklıdır. Böylece;

Yeni teknolojilerin sistemlere entegrasyonu sırasında yaşanabilecek sıkıntılar önlenmeye çalışılmıştır. [12]



Şekil 3.2 TCP/IP mimarisi protokolleri ve ağları

İnternet üzerinden yönlendirmeler IP adresleri kullanılarak yapılmaktadır. Ağ katmanındaki verilere IP adresleri eklenerek taşıma katmanına ulaştırılır. Burada servisin kalitesine göre oluşturulacak bağlantı protokolü türü belirlenir.

Tez kapsamında iki tür protokol türü incelenecektir. Bunlar TCP ve UDP protokolleridir.

Veri akış kontrolü ve güvenli veri aktarımı TCP protokolü kullanılarak gerçekleştirilir. Bir bağlantının güvenilir olması için; İki bilgisayarın veri alışverişine geçmeden önce birbirleri ile anlaşmaları gerekmektedir.

TCP protokolü üç yollu el sıkışması (*3-way handshake*) denilen yöntem ile bu bağlantıyı gerçekleştirir.

- A bilgisayarı B bilgisayarına TCP SYN mesajı yollar.
- B bilgisayarı SYN+ACK mesajı yolarak, A bilgisayarının isteğinin alındığını belirtir.
- A bilgisayarı B bilgisayarına TCP ACK mesajı yollar.
- B bilgisayarı bir ACK “TCP bağlantısı kuruldu” mesajı alır.

TCP bağlantı sonlandırılması aşağıdaki şekilde yapılmaktadır.

- A bilgisayarı TCP FIN mesajı yollayarak B bilgisayarına bağlantıyı sonlandırmak istediğini belirtir.
- B bilgisayarı TCP ACK mesajı yollayarak A bilgisayarına bağlantıyı sonlandırma isteği aldığını belirtir.
- B bilgisayarı TCP FIN mesajı yollayarak A bilgisayarına bağlantıyı sonlandırmak isteğini belirtir.
- A bilgisayarı TCP ACK mesajı yollayarak B bilgisayarına bağlantıyı sonlandırma isteği aldığı belirtir.

UDP protokolü gönderilen paketin doğru şekilde iletilip iletilmediğini takip etmez. Bu nedenle, güvenilir olmayan ve bağlantı gerektirmeyen bir protokol türüdür.

TCP ve UDP protokolleri, IP adresleri üzerinden gerekli veri alışverişini sağlamak için hedef ve kaynak port numaralarını kullanır. Bu şekilde; Aynı anda, aynı IP adresini kullanarak, farklı programlar ile veri alış verişi gerçekleştirilebilir.

Bir IP trafik akışı, temel olarak beş öğeye (*Kaynak IP adresi, Kaynak Port, Hedef IP adresi, Hedef Port, Protokol*) bakılarak tanımlanır.

İki nokta arasındaki iletişim, çift yönlü veya tek yönlü olabilir. Tek yönlü iletişim, ileri yönlü (*forward direction*) ve geriye yönlü (*backward direction*) olmak üzere iki şekilde gerçekleşebilir. İletişim yönü, Kaynak IP adresi üzerindeki Kaynak Port'undan (genellikle kullanıcı (*client*) olmaktadır.), Hedef IP adresi üzerindeki Hedef Port'una (genellikle sunucu (*server*) olmaktadır.) doğru ise ileri yönlü, tersi durumda ise geriye yönlü olarak ifade edilir. İletişim, iki sunucu veya iki kullanıcı arasında da gerçekleşebilir.

İnternet ağları üzerinde güvenli bağlantı yöntemi, güvensiz bağlantı yöntemine göre çok daha fazla tercih edilmektedir. Bu nedenle; TCP bağlantı yöntemi kullanılarak

gerçekleştirilen çalışma sayısı UDP bağlantı kullanarak geliştirilen çalışma sayısından çok daha fazladır. Fakat; CAIDA 'nın [35] 2002-2009 tarihleri arasında farklı zamanlarda, farklı coğrafya ve ağlar üzerinde yaptığı araştırmaya göre, bağlantı kalitesindeki artışa bağlı olarak, özellikle internet üzerinden video, ses, çevrimiçi oyun oynama ve P2P kullanım oranının git gide artması nedeniyle, TCP ve UDP bağlantı kullanımı oranı arasındaki fark giderek azalmıştır. [36] Bu nedenle, tez veri setinde hem TCP hem de UDP protokol verileri kullanılacaktır.

Tez kapsamında; Ağ trafiği denildiğinde anlatılmak istenen, dünya üzerindeki en büyük küresel ağ olan İnternet dir. Veri, kelimesi ile kullanılan tanımlama ile de, internet ağı üzerindeki her bir akış kastedilmektedir.

4. AĞ TRAFİĞİ SINIFLANDIRMA YÖNTEMLERİ VE LİTERATÜR ÇALIŞMALARI

İnternet kullanımı ile ortaya çıkan olağanüstü veri yığınlarını incelemek için geliştirilen yöntemler, *"Günümüzde en iyi trafik sınıflandırma metodu hangisidir?, Hangi şartlar altında?, Neden?"* sorularının ortaya çıkmasına neden olmuştur. [30]

Bu soruların cevaplarını araştırmak için günümüze kadar birçok çalışma gerçekleştirilmesine rağmen, hâlâ, araştırmacılar kesin bir sonuca ve fikir birliğine ulaşamamıştır. [24][29]

H. Kim ve ark. [30] göre, sınıflandırma metotlarının titiz bir şekilde karşılaştırılamamaları üç nedene bağlıdır. İlk olarak, herkesin erişebileceği açık bir yüklü kayıtlı (*payload trace*) veri seti yoktur. Bu nedenle, geliştirilen her model yerel olarak toplanılan veri setleri incelenerek değerlendirilmeye çalışılmaktadır. İkinci olarak; Literatürdeki çalışmalar; Farklı nitelikleri, farklı parametreleri, farklı tanımlamaları ve farklı uygulamaları kullanarak değerlendirme yapmaktadır. Üçüncü olarak, araştırmacılar çalışmalarının sonuçlarını açıklamalarına rağmen, geliştirdikleri uygulama kodlarını paylaşmamaktadırlar.

Ağ kaynakları üzerinde dolaşan veri trafiğinin verimli bir şekilde gerçekleşmesini sağlamak, ağ verileri üzerinden kullanıcı analizlerini yapabilmek [31], ağ kaynaklarının yönetilmesi ve planlanması sağlamak [32] veya ağ üzerindeki anormalliklerin ve saldırıların [33-34] tespitini gerçekleştirmek için trafik sınıflandırma yöntemleri kullanılabilir.

İnternet trafiği analizi, hem çevrimiçi (gerçek zamanlı) hem de çevrimdışı (pasif olarak) yapılabilir**.** Çevrimiçi trafik analizi yapılırken ağ üzerinden akan veri paketleri anlık olarak yakalanarak analiz edilir. Çevrimdışı trafik analizi yapılırken ise, paketler ilk önce yakalanıp depolanır, daha sonra analiz edilerek sınıflandırılır.[29]

Literatürde sıklıkla kullanılan üç tür sınıflandırma tekniği vardır. Bunlar; Port-tabanlı sınıflandırma, yük-tabanlı sınıflandırma ve makine öğrenme-tabanlı sınıflandırmadır. Bu tekniklerin, avantajları ve dezavantajları detaylı olarak alt bölümlerde bahsedilmektedir.

4.1. Port-tabanlı (*Port-based*) Sınıflandırma

İnternet trafiğini sınıflandırmak için port numaralarının kullanılması prensibine dayanır.

Önceki zamanlar için, port tabanlı sınıflandırma metodu oldukça başarılı bir yöntemdir. En çok bilinen ve kullanılan uygulamalar, IANA [13] tarafından belirlenen sabit port numaralarını kullanmaktadır. Örneğin; FTP trafiği 21 portunu, DNS trafiği 53 portunu kullanır.

Günümüzde ise; T. Karagiannis ve ark. [14] çalışmalarından da anlaşılacağı gibi, özellikle P2P uygulamalarının yaygınlık kazanması ile birlikte, bazı uygulamaların firewall ve ağ güvenlik araçlarından sızmak için standart olmayan port numaralarını

kullanması, port gizleme ve dinamik port numarası yöntemlerini kullanması nedeniyle bu yöntem etkisini yitirmeye başlamıştır. [15][17-18]

4.2. Yük-tabanlı (*Payload-based*) Sınıflandırma

İnternet trafiğini TCP/UDP paket yüklerinin analizlerini yaparak sınıflandırılması prensibine dayanır. Yüklerin analizi, bilinen uygulamaların karakteristik imza içerip içermediğini belirleyerek gerçekleştirilir. [14][16]

Bu yöntem; Paketler şifreli olmadığı zaman ve P2P trafiğini içeren internet trafiği incelendiğinde oldukça başarılı çalışmaktadır. Fakat; Mahremiyet ve güvenlik kaygıları yaratması, bazı uygulamaların şifreli paketler kullanarak haberleşmesi, sadece daha önceki trafik sınıflandırma yöntemlerince tecrübe edilen imzalara göre değerlendirme yapabilen bir yöntem olması, yüksek işlemci ve depolama kapasitesi gerektirdiği için gerçek zamanlı sınıflandırmaya uygun olmaması nedeniyle günümüzde gerçekçi olmayan bir yaklaşımdır. [15][17-18]

4.3. Makine Öğrenme-tabanlı (*Machine Learning-based*) Sınıflandırma

Makine öğrenme algoritmalarını kullanarak ağ trafiğini sınıflandırma yöntemi, günümüzde en popüler trafik sınıflandırma yöntemidir. Yapılan çalışmalarda genellikle denetimli ve denetimsiz öğrenme algoritmaları kullanılarak sınıflandırma yapılmaktadır.

Denetimli öğrenme algoritmaları veri madenciliğindeki sınıflandırma analizi yöntemlerini, denetimsiz öğrenme algoritmaları ise kümeleme analizi yöntemlerini kullanarak sınıflandırma işlemini gerçekleştirirler.

Makine öğrenme algoritmaları ağ trafiğini sınıflandırma işlemini iki adımda gerçekleştirir. İlk adımda sınıflandırma modeli oluşturulur, ikinci adımda ise

sınıflandırma işlemi yapılır. Sınıflandırma işlemini gerçekleştirirken genellikle istatistiksel yöntem ve hesaplamalardan yararlanılmaktadır.

Makine öğrenme tabanlı sınıflandırma yöntemi akış tabanlı sınıflandırma yaparken, akışa ait; Paket büyüklük istatistikleri (minimum, maksimum, ortalama), Toplam Paket sayısı (ileri yönlü, ters yönlü), Toplam byte miktarı (ileri yönlü, ters yönlü), Paketler arası geliş gidiş zamanı (minimum, maksimum, ortalama), Akış süresi, … vb. TCP ve UDP istatistiksel niteliklerini kullanmaktadır.

A. Moore ve ark. [37] çalışmasından, TCP protokolüne ait daha fazla istatistiksel nitelik bilgilerine ulaşılabilir. Tez içerisinde kullanılan nitelikler, Bölüm 6 dan incelenebilir.

L. Yingqiu ve ark. [18] çalışmalarında, K-means algoritmasını kullanarak orijinal ve logaritmik dönüşümlü veri seti üzerinden ağ trafiğini sınıflandırmaya çalışmışlardır. J. Erman ve ark. [21] çalışmasında, kümeleme algoritmaları olan K-means, DBSCAN ve Autoclass algoritmalarını, [17] çalışmasında, Naive Bayes ve Autoclass algoritmalarını kullanarak AucklandIV veri seti üzerinde incelemeler ve karşılaştırmalar yapmışlardır. S. Zander [54] WAND Araştırma Grubu'nun [46] web sayfasından indirilebilen açık veri setlerini kullanarak farklı veri setleri üzerinde EM algoritması ve nitelik seçimi yöntemini kullanarak ağ trafiğini üzerinde sınıflandırmalar yapmışlardır. Ayrıca; S. Zander ve ark. [42-43] ve S. Agrawal ve ark. [23][55-56] C4.5, Bayes Net, Navie Bayes, … gibi sınıflandırma algoritmalarını ve nitelik seçimi yöntemlerini kullanarak, sınıflandırma algoritmaları üzerine kapsamlı karşılaştırmalar yapmışlardır.

Literatürde, kullanılan bu üç teknik dışında, T. Karagiannis ve ark. nın, TCP veya UDP protokollerini yerine host servis sağlayıcılarını kullanarak geliştirdikleri sınıflandırma tekniği [38] ve L. Bernaille ve ark. nın sadece ilk gelen birkaç TCP paketine bakarak ve denetimsiz öğrenme yöntemini kullanarak gerçekleştirdikleri sınıflandırma tekniği [39], literatürdeki önemli çalışmalar arasındadır.

5. ALGORİTMA BAŞARIMI DEĞERLENDİRME METRİKLERİ

Makine öğrenme algoritmalarının karşılaştırmaları yapılırken, algoritmaların implementasyonlarına bakılarak yapılan (zaman karmaşıklığı, alan karmaşıklığı, …) değerlendirmelerin yanı sıra, sınıflandırma başarımlarını değerlendirmek için kullanılan; Model başarımı ve sınıflandırma yapmak için gereken süre ölçümleri, gürültülü veya eksik verilerde algoritmaların doğru çalışabilme kabiliyetlerinin ölçümleri, verilerin kullanıcılar tarafından yorumlanabilir olması ve verileri doğru sınıflandırma başarımlarının ölçümleri, öğrenme algoritmalarının karşılaştırmasında önemli bir rol oynamaktadır.

Makine öğrenme algoritmalarının, sınıflandırma başarımlarının ölçümleri, Şekil 5.1 [24] de gösterilen, sınıflandırma algoritmaları için *karışıklık matrisi*, kümeleme algoritmaları için *eşleşme matrisi* olarak ifade edilen değerlendirme tablosu ve metrikleri kullanılarak incelenebilir.

Tablo 5.1 Değerlendirme tablosu ve metrikleri

Sınıflandırılmış → ***Kümeye atanmış***	X	$\bar{X}$
X	TP	FN
$\bar{X}$	FP	TN

Tablonun satırları örneğe ait gerçek değerleri, matrisin sütunları ise örneğe ait sınıflandırılmış veya kümeye atanmış tahmini değerleri vermektedir. Bu ifadeye göre, yukarıda adı geçen metrikler şu şekilde tanımlanabilir.

- **Doğru Pozitif (TP) :** Gerçekte X sınıfına ait ve X sınıfına ait olduğu doğru olarak tahmin edilmiş örnek sayısı

- **Doğru Negatif (TN) :** Gerçekte X sınıfına ait olmayan ve X sınıfına ait olmadığı doğru olarak tahmin edilmiş örnek sayısı

- **Yanlış Pozitif (FP) :** Gerçekte X sınıfına ait olmayan fakat X sınıfına ait olduğu yanlış olarak tahmin edilmiş örnek sayısı

- **Yanlış Negatif (FN) :** Gerçekte X sınıfına ait olan fakat X sınıfına ait olmadığı yanlış olarak tahmin edilmiş örnek sayısı

Yukarıdaki metrik tanımlarından hareket ederek; İyi bir trafik sınıflandırmasındaki amaç, FN ve FP değerlerini minimize etmektir ve bu metrikleri kullanarak algoritma başarımlarını değerlendirirken literatürde trafik sınıflandırmasında sıklıkla kullanılan aşağıdaki kavramlara ulaşılabilir. [24]

Tez kapsamında akış tabanlı ağ trafiği sınıflandırması incelendiğinden tanımlamalar aksi belirtilmedikçe aşağıdaki şekilde kullanılacaktır.

- **Doğruluk (*Accuracy*) :** En sık kullanılan karşılaştırma kavramlarından biridir. Doğru sınıflandırılmış akış sayısına bağlı olarak çıkarım yapılmasında kullanılır. Bütün veri seti içersindeki doğru olarak sınıflandırılmış akış sayısının, veri setindeki toplam akış sayısına oranı alınarak hesaplanır.

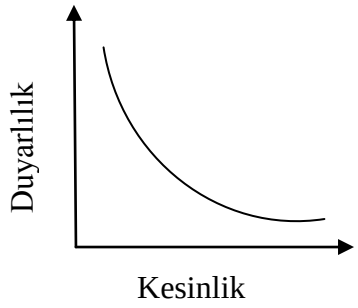
$$Doğruluk = \frac{TP + TN}{TP + TN + FP + FN} \tag{5.1}$$

- **Kesinlik (*Precision*) :** Doğru sınıfa ait olduğu tespit edilmiş akış sayısının, veri setindeki doğru akış sınıfına ait olduğu belirlenmiş akışların toplam sayısına oranıdır.

$$Kesinlik = \frac{TP}{TP + FP} \tag{5.2}$$

- **Duyarlılık (*Recall*) :** Doğru sınıfına ait olduğu tespit edilmiş akış sayısının, gerçekte o sınıfa ait olan toplam akış sayısına oranıdır.

$$Duyarlılık = \frac{TP}{TP + FN} \tag{5.3}$$



Şekil 5.1 Duyarlılık ve Kesinlik ilişki grafiği

Duyarlılık ve kesinlik arasında Şekil 5.1 de olduğu gibi ters orantı vardır.

X algoritması, Y algoritmasından daha iyi duyarlılık ve kesinlik değerine sahip ise X algoritması daha iyi bir sınıflandırıcı olduğu söylenebilir.

Bazı durumlarda, X algoritması Y algoritmasından daha iyi duyarlılık değerine sahip olmasına rağmen, Y algoritması X algoritmasından daha iyi kesinlik değerine sahip olabilir. Bu durumda, F-ölçütüne bakarak karar vermek daha doğru olur.

- **F-ölçütü (*F-measure*) :** Duyarlılık ve kesinlik metriklerinin beraber incelenerek ölçüm yapılmasıdır. Bu ölçüm, iki metriğin harmonik ortalaması alınarak hesaplanır.

$$F - Ölçütü = \frac{2 \times Duyarlılık \times Kesinlik}{Duyarlılık + Kesinlik} \tag{5.4}$$

Yukarıda yapılan tanımlamaların dışında sınıflandırma ve kümelere algoritmalarının sonuçlarının doğrulamasında kullanılan farklı yöntemler de mevcuttur. Bu doğrulama yöntemleri [25-28] makalelerinden incelenebilir. Tez kapsamında bu hesaplamalar kullanılmayacaktır.

6. NİTELİK SEÇİMİ

Makine Öğrenme algoritmalarının, sınıflandırma ve kümeleme işlemleri sırasında kullanacağı niteliklerin belirlenmesi algoritmaların başarımları üzerinde önemli bir etkisi vardır. Bu nedenle; Algoritmaların kullanacağı nitelik sayısına ve niteliğin türüne dikkat edilmelidir.

Örneğin; Kümeleme analizi sırasında veriler aralarındaki uzaklıklara bakılarak gruplandırma yapılsın. Eğer veriler arasındaki ilişkiler düzgün seçilmiş ise, benzer özelliklere sahip veriler arasındaki uzaklık minimum, farklı özelliklere sahip veriler arasındaki uzaklık maksimum olarak ortaya çıkar ve veriler bu şekilde birbirlerinden ayrılarak belirli gruplar oluştururlar. Böyle bir durumda, nitelik sayısının fazla veya az olmasında da problem oluşacaktır.

İlk olarak, nitelik sayısının girerek arttırıldığı, her nitelik arttırılışında noktalar kümesinin yarısının aynı grup içerisinde düştüğü varsayılsın. Bu durumda, İlk etapta noktalar kümesinin yarısı aynı grup içerisine, İkinci etapta 1/4'ü aynı grup içerisine, üçüncü etapta 1/8'i aynı grup içerisine düşecektir. Boyut sayısı arttırılmaya devam ettiğinde noktalar arasındaki uzaklık nedeyse eşit olacaktır ve uzaklık kavramı önemsizleşecektir. Dolayısıyla da anlamlı küme oluşturma işlemi gerçekleştirilemeyecektir. Bu duruma literatürde, boyutsallığın laneti (*the curse of dimentionality*) denir. [40]

İkinci olarak, nitelik sayısının giderek azaltıldığı varsayılsın. Bu durumda; Zamanla, aslında farklı gruplara ait olan veriler aynı gruplara düşmeye başlayacak ve kümelerin doğruluğu giderek azalacaktır.

Ayrıca; Kullanılan nitelik sayısının gereğinden fazla olması durumunda, algoritmadaki gereksiz bilgi sayısının artması nedeniyle, algoritmanın öğrenme süresi ve sınıflandırma zamanı artacak, algoritmanın sınıflandırma başarısı düşecektir.

Kullanılmayan niteliklerin çıkartılması işlemi sırasında bütün nitelik kümesinin alt kümelerine tek tek bakarak, sınıflandırma başarısına göre nitelik seçimlerini gerçekleştirilmek istenebilir. Fakat bu durumda, n adet nitelik için 2^n sayıda alt kümede inceleme yapmak gerekecektir. Dolayısıyla bu yöntem, nitelik sayısının fazla olması durumda oldukça verimsiz olacaktır.

Yukarıda anlatılan nedenlerden dolayı ortaya çıkan nitelik seçme algoritmaları, makine öğrenme algoritmalarının başarımları üzerinde önemli rol oynamaktadır.

Nitelik seçme algoritmaları filtre model (*filter model*) algoritmaları ve sarmal model (*wrapper model*) algoritmalar olmak üzere iki kategoriye ayrılabilir. [41]

Sarmal model algoritmalar, veri kümelerini alt kümelere ayırarak sınıflandırma algoritmalarının alt kümeler üzerindeki etkisini değerlendirir. Bu nedenle, kümeleme algoritmalarını değerlendirmek için uygun değildir. [18]

Sarmal model algoritmalar, algoritmaların öğrenme zamanı önemli değilse, en iyi algoritmalardan biridir. Fakat; Filtreleme algoritmalarına göre oldukça yavaştırlar ve büyük veri setleri için uygun değildirler. [41]

Tez kapsamında, büyük ölçekli veri seti kullanılması ve sınıflandırma algoritmalarının yanında kümeleme algoritmalarının da değerlendirilmesi nedeniyle sarmal model algoritmalar kullanılmayacaktır.

6.1. Filtre Modeli

Filtre modeli algoritmaları, sarmal modelinden farklı olarak nitelikleri belirlemek için özelleştirilebilir bazı metrikler kullanır. Bu nedenle; bütün makine öğrenme algoritmalarında kullanılacak şekilde nitelikleri seçebilir.

Filtre model algoritmaları sıralayıcı (*ranker*) ve alt küme arama (*subset search*) olmak üzere iki sınıfta incelenebilir. [42] Sıralayıcı algoritmalar, bütün nitelikleri belli bir başarı sırasına göre sıralar, kullanılmaması gereken niteliklerin çıkarımını kullanıcıya bırakır. Alt küme arama algoritmaları ise niteliklerin uygun altkümelerini çıkararak kullanılmaması gereken nitelikleri belirleyebilir.

Tez kapsamında korelâsyon tabanlı ve tutarlılık tabanlı olmak üzere iki çeşit alt küme arama algoritması incelenecektir.

6.1.1. Korelâsyon-tabanlı nitelik seçimi (CFS) algoritması

CFS algoritması [44], nitelikler arasındaki korelâsyon seviyelerine bakarak sezgisel çıkarımda bulunulur. Niteliklerin sahip oldukları değerler birbirleri ile simetrik olarak değişiyor ise niteliklerin birbirleri ile ilişkili olduğu, aksi durumda niteliklerin birbirleri ile ilişkisiz olduğu kabul edilmektedir. Nitelikler birbirleri ile ilişkili ise korelâsyon değerleri yüksektir.

Nitelikler arası ve sınıf ile nitelik arasındaki korelâsyonun ölçülmesinde koşullu entropi kullanılmaktadır. $H(X)$, X niteliğinin entropisi ve $H(X|Y)$, Y niteliğinin gözlemine göre X niteliğinin entropisi olmak üzere X ve Y arasındaki korelâsyon,

$$C(X|Y) = \frac{H(X) - H(X|Y)}{H(Y)} \qquad (6.1.1.1)$$

simetrik belirsizlik formülünü kullanarak hesaplanabilir. [43] Ortaya çıkan, örneğin sınıfı bir nitelik olarak kabul edilir. Alt kümenin kalitesi,

$$G_{subset} = \frac{k\,\bar{r}_{ci}}{\sqrt{k + k(k-1)\,\bar{r}_{ii}}} \qquad (6.1.1.2)$$

formülü kullanarak hesaplanır. Burada, k alt kümedeki toplam nitelik sayısı, $\bar{r}_{ii}$ niteliklerin birbirleri arasındaki ortalama korelâsyonu, $\bar{r}_{ci}$ sınıf ile nitelikler arasındaki ortalama korelâsyonu göstermektedir. Nitelik ile sınıf arasındaki ve nitelikler arasındaki korelâsyonlar simetrik belirsizlik katsayılarıdır. [43]

6.1.2. Tutarlılık-tabanlı alt küme arama (CON) algoritması

CON algoritması [45], niteliklerin alt kümelerini eş zamanlı olarak değerlendirerek en uygun alt kümeyi bulmaya çalışır. En uygun alt küme, bütün nitelik kümesi kadar uygun bir şekilde sınıfın örneklerini belirleyebilen en küçük niteliklerin alt kümesidir.

Bir alt kümenin uygunluğuna karar vermek için niteliklerin değerlerinin kombinasyonunu gösteren bir örüntü etiketi verilir ve belirli bir örüntüdeki bütün örneklerin aynı sınıfta olması beklenir. Eğer aynı örüntüdeki iki örnek farklı sınıfları temsil ediyor ise, o örüntü tutarsız olarak kabul edilir. Bir p örüntüsündeki tüm tutarsızlıklar;

$$IC(p) = n_p - c_p \qquad (6.1.2.1)$$

ile bulunur. Burada n_p, örüntüdeki örneklerin sayısını, c_p, n_p örneklerinin çoğunluk sınıfındaki örneklerin sayısıdır. [42] S alt nitelik kümesindeki genel tutarsızlıklar, bütün örüntülerdeki tutarsızlıkların toplamının, bütün örüntü örneklerinin toplamına (n_s) oranıdır. [42]

$$IR(S) = \frac{\sum_{p} IC(p)}{n_s} \qquad (6.1.2.2)$$

Bütün nitelik kümesinin düşük tutarsızlık orasına sahip olduğu ve en benzer veya aynı altkümenin en uygun altküme olduğu kabul edilir. [42]

6.2. Arama Teknikleri

Nitelik seçme algoritmaları, nitelik uzayından aday alt kümenin oluşturulmasında arama tekniklerine ihtiyaç duymaktadır. En yaygın olarak kullanılan, en iyi ilk (*best first*) ve hırslı (*greedy*) arama teknikleridir. Bu teknikler iki türlü işlem gerçekleştirebilir. Eğer, algoritma nitelik kümesine ekleme yapıyorsa ileri yönlü (*forward direction*) arama, nitelik kümesinden çıkarma yapıyorsa geri yönlü (*backward direction*) arama yaparak işlemleri gerçekleştirmektedir.

- **Hırslı Arama (*Greedy Search*) :** Hırslı arama tekniği, nitelikleri ekleme veya kaldırma yoluyla mevcut alt yerel değişiklikleri değerlendirmektedir. Verilen belirli bir üst (*parent*) küme için hırslı arama nitelliklerin eklenmesi veya çıkarılması yoluyla tüm olası alt (*child*) kümeleri inceler. Alt küme, üst kümeden daha iyi ölçüm göstermesi durumunda alt küme ve üst küme yer değiştirilir. Daha fazla iyileştirme yapılamadığı zaman süreç sonlandırılır.[43]

- **En iyi ilk (*Best First*) :** Hırslı aramaya benzer olarak mevcut alt kümeye nitelik ekleyerek veya nitelik çıkararak yeni alt küme oluşturur. Ancak; Mevcut yol artık gelişme göstermediği zaman farklı olasılıkları keşfetmek için alt seçim yolunda geri izleme yeteneğine sahiptir. Nitelik uzayındaki tüm olasılıkların geri izlenmesini önlemek için gelişme göstermeyen alt kümeler üzerine bir sınır değeri konur. [43] Tez çalışmasında kullanılacak WEKA [51] yazılımında bu sınır değer 5 olarak belirlenmiştir.

7. KULLANILAN VERİ SETİNİN VE YAZILIMLARIN İNCELENMESİ

Bölüm 4'de de bahsedildiği gibi, literatürde ağ trafiği sınıflandırma karşılaştırmalarındaki temel sıkıntıların başında, ortak kullanılan büyük boyutlu veri setlerinin azlığı ve araştırmacıların kendi hazırladığı implementasyonları diğer araştırmacılar ile paylaşmaması gelmektedir. Bu nedenle, karşılaştırmaların titiz bir şekilde yapılabilmesi için tez çalışmasında, ağ sınıflandırmalarında kullanılan büyük boyutlu ortak veri seti ile veri madenciliğinde ve bilgi keşfinde kullanılan makine öğrenme algoritmalarının standart implementasyonlarının bulunduğu analiz araçları [46] tercih edilmiştir.

7.1. Kullanılan Veri Seti

Veri seti olarak daha önce NLANR [47] tarafından oluşturulan, günümüzde WAND Araştırma Grubu'nun [48] geliştirmelerine devam ettiği Auckland Üniversitesi kampüs internet altyapı ölçümlerine ait, 10 Haziran 2001 tarihli 24 saatlik Auckland-IV-20010610 kayıtları kullanılmıştır. Web sitesi içerisinde kayıtlar tek yönlü ve 6 saatlik dilimler halinde bulunmaktadır. Bu kayıtlar, libpcap [56], network trafiği yakalama kütüphanesi ile çift yönlü 24 saatlik olacak şekilde uygun olarak birleştirilmiştir.

Auckland-IV-20010610 veri seti içerisinde toplam 86 milyonun üzerinde paket bulunmaktadır. Bu paketler yük bilgisi içermemektedir. Veri seti içerisinde TCP, UDP ve ICMP protokollerinin trafik kayıtları gösterilmektedir ve IP trafiği olmayan bütün kayıtlar çıkarılmıştır.

7.2. Veri Seti İçerisinden Akış Çıkarımı

Veri seti içersindeki akışların belirlenebilmesi ve nitelik değerlerinin hesaplanabilmesi için NetMate [49] yazılımı kullanılmıştır ve yazılıma NetAI paketi eklenmiştir.

Tezde kullanılacak veri seti akışları iki yönlü olacak şekilde ayarlanmıştır ve sadece TCP ve UDP akışları dikkate alınmıştır. Akışın yönü olarak, ilk paketin gönderildiği nokta ileri yön olarak tayin edilmiştir. Başarısız (cevaplanamayan, zaman aşımına uğramış) veya veri iletişimi olmayan akışlar dikkate alınmamıştır.

Akışların belli bir zaman aşımı süreleri vardır. Akışın zaman aşımı süresi olarak NeTraMet [50] tarafından varsayılan olarak belirlenen ve benzer araştırmalarda da sıklıkla kullanılan 600 saniye belirlenmiştir. UDP akışları zaman aşım sürelerine bakılarak sonlandırılır. TCP akışları ise bağlantı sonlandırma işlemi (en son yollanan FIN paketine) veya zaman aşım süresine bakılarak sonlandırılır. (İlk gerçekleşen duruma bakılır.)

Başarılı akış kümeleri oluşturulduğu zaman, bu akışlara ait Tablo 7.1 de belirtilen nitelik değerlerine ulaşılabilir.

Tablo 7.1 Nitelik kümesi elemanları ve açıklamaları

Kısaltmalar	**Açıklamalar**
proto	Protokol türü
total_fpackets	İleri yönde toplam paket sayısı
total_fvolume	İleri yönde toplam bayt miktarı
total_bpackets	Geri yönde toplam paket sayısı
total_bvolume	Geri yönde toplam bayt miktarı
min_fpktl	İleri yönde gönderilen en küçük boyuttaki paket (bayt)
mean_fpktl	İleri yönde gönderilen paketlerin ortalama boyutu (bayt)
max_fpktl	İleri yönde gönderilen en büyük paket boyutu (bayt)
std_fpktl	İleri yönde gönderilen paketlerin ortalamadan standart sapması (bayt)
min_bpktl	Geri yönde gönderilen en küçük boyuttaki paket (bayt)

mean_bpktl	Geri yönde gönderilen paketlerin ortalama boyutu (bayt)
max_bpktl	Geri yönde gönderilen en büyük paket boyutu (bayt)
std_bpktl	Geri yönde gönderilen paketlerin ortalamadan standart sapması (bayt)
min_fiat	İleri yönde gönderilen iki paket arasındaki minimum süre (mikrosaniye)
mean_fiat	İleri yönde gönderilen iki paket arasındaki ortalama süre (mikrosaniye)
max_fiat	İleri yönde gönderilen iki paket arasındaki maksimum süre (mikrosaniye)
std_fiat	İleri yönde gönderilen iki paket arasındaki sürenin ortalamadan standart sapması (mikrosaniye)
min_biat	Geri yönde gönderilen iki paket arasındaki minimum süre (mikrosaniye)
mean_biat	Geri yönde gönderilen iki paket arasındaki ortalama süre (mikrosaniye)
max_biat	Geri yönde gönderilen iki paket arasındaki maksimum süre (mikrosaniye)
std_biat	Geri yönde gönderilen iki paket arasındaki sürenin ortalamadan standart sapması (mikrosaniye)
duration	Akış süresi (mikrosaniye)
min_active	Akışın hareket etmeden önce boşta aktif olarak beklediği minimum zaman (mikrosaniye)
mean_active	Akışın hareket etmeden önce boşta aktif olarak beklediği ortalama zaman (mikrosaniye)
max_active	Akışın hareket etmeden önce boşta aktif olarak beklediği maksimum zaman (mikrosaniye)
std_active	Akışın hareket etmeden önce boşta aktif olarak beklediği ortalama zaman miktarından (mikrosaniye) standart sapması
min_idle	Akışın aktif hale gelmeden önce boşta beklediği minimum zaman (mikrosaniye)
mean_idle	Akışın aktif hale gelmeden önce boşta beklediği ortalama zaman (mikrosaniye)
max_idle	Akışın aktif hale gelmeden önce boşta beklediği maksimum zaman (mikrosaniye)
std_idle	Akışın aktif hale gelmeden önce boşta beklediği ortalama zamandan standart sapması (mikrosaniye)
sflow_fpackets	İleri yönde bir alt akış içindeki paketlerin ortalama sayısı
sflow_fbytes	İleri yönde bir alt akış içindeki paketlerin ortalama bayt miktarı

sflow_bpackets	Geri yönde bir alt akış içindeki paketlerin ortalama sayısı
sflow_bbytes	Geri yönde bir alt akış içindeki paketlerin ortalama bayt miktarı
fpsh_cnt	İleri yönde seyahat içerisinde PSH bayrağı ayarlanma sayısı
bpsh_cnt	Geri yönde seyahat içerisinde PSH bayrağı ayarlanma sayısı
furg_cnt	İleri yönde seyahat içerisinde URG bayrağı ayarlanma sayısı
burg_cnt	Geri yönde seyahat içerisinde URG bayrağı ayarlanma sayısı
total_fhlen	İleri yöndeki başlıklar için kullanılan toplam bayt miktarı
total_bhlen	Geri yöndeki başlıklar için kullanılan toplam bayt miktarı

Yukarıdaki nitelik tanımlarında geçen alt akış, adından da anlaşıldığı gibi akış içerisindeki alt akışları belirtmektedir. Akışın boşta belli bir bekleme süresini aşan hareketsizlik zamanı ile ayrılır. Alt akış için zaman aşımı süresi 1 saniye olarak ayarlanmıştır. URG ve PSH bayrakları, TCP bağlantı türü için kullanılır ve paketler içerisindeki önceliklerin belirlenmesini sağlar. UDP bağlantı türü için bu değerler 0 dır.

7.3. Analiz Araçları

Ağ trafiği analizinin ve makine öğrenme algoritmalarının karşılaştırmalarının yapılabilmesi için, literatürde en çok kullanılan yazılımların başında olan, WEKA [51] programı tercih edilmiştir.

WEKA yazılımı içerisindeki sınıflandırma ve kümeleme işlemlerinde kullanılan makine öğrenme algoritmalarının, öğrenme ve test kümelerinin oluşturulmasında farklı yöntemler kullanılabilir.

Sınıflandırma algoritmalarının öğrenme ve test kümesi farklı veri setleri olabileceği gibi, kullanılan veri setinin tamamı veya belli bir yüzdesi de test kümesi olarak

kullanılabilir veya k-kere çapraz doğrulama (*k-fold cross validation*) metodu kullanılabilir.

Özellik seçimi işlemlerinde de kullanılan k-kere çapraz doğrulama metodunda, veri seti belirtilen k sayıda kümeye bölünerek alt kümelerden birisi test kümesi, diğer kümeler öğrenme kümesi olarak seçilir ve sistem eğitilir. Her küme bir kere test kümesi olacak şekilde bu işlem k kere tekrarlanarak sistem iyileştirilmeye çalışılır.

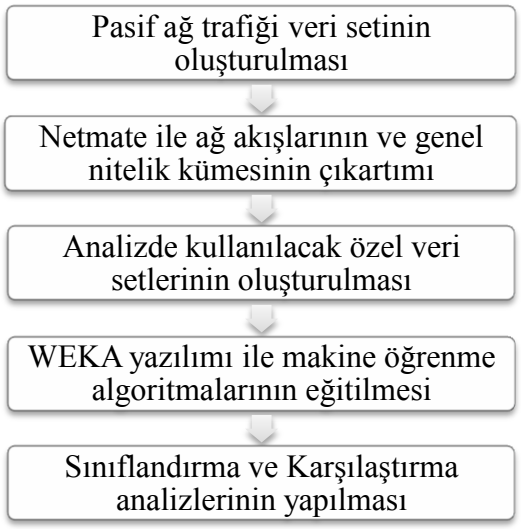
Kümeleme algoritmalarının öğrenme kümesi, sınıflandırma algoritmalarında da olduğu gibi farklı veri setleri, veri setinin tamamı veya veri setinin belli bir yüzdesi olabilir. Ayrıca; Kümeleme işlemi kümeleri değerlendirme sınıflarına göre ayırarak da incelenebilir. Bu durumda, sınıf niteliği veri setinden çıkarılarak veri setinin tamamı öğrenme amaçlı kullanılarak bu niteliğe göre kümeler oluşturulur.

AucklandIV ham veri setinin birleştirilmesi ve içerisinden ağ akışlarının çıkarılması sırasında; Intel Core2 Duo T8100 işlemci, 4gb ram ve 320gb disk kapasitesine sahip Linux işletim sistemli bilgisayar kullanılmıştır.

Oluşturulan işlenmiş veri setlerinin analizleri için Intel Core i7-2720 işlemci, 8GB ram ve 128GB SSD disk kapasitesine sahip Windows işletim sistemli sunucu kullanılmıştır.

8. DENEYSEL GÖZLEM VE İNCELEMELER

Çalışmada yapılmak istenen makine öğrenme algoritmalarını kullanarak çevrimdışı (pasif) ağ trafiğinin sınıflandırılması ve makine öğrenme algoritmalarının karşılaştırmalarının yapılması sırasında kullanılan işlem adımları Şekil 8.1 de gösterilen akış diyagramı ile ifade edilebilir.



Şekil 8.1 Tez akış diyagramı

Çalışmada kullanılan Auckland-IV-20010610 ham veri setinden, TCP ve UDP protokollerine ait ağ akışlarının ve nitelik kümesinin çıkarılmasının ardından, tezde kullanılacak olan Tablo 8.1 de gösterilen TCP ve UDP protokol türelerine ulaşılabilir. Auckland-IV-20010610 veri seti, internet üzerinde herkesin ulaşılabileceği şekilde açık olması nedeniyle, yük bilgisi içermemekte sadece TCP/IP başlık bilgilerini içermektedir. Bu nedenle; Sınıflandırmalarda kullanılacak özel veri setlerinin hazırlanmasında doğru sınıflandırma işlemlerinin gerçekleştirilebilmesi için akışlar hedef port bilgilerine bakılarak sınıflandırılmıştır. Daha önce tez çalışması içerisinde, port-tabanlı sınıflandırmanın günümüzde etkisini yitirmeye başladığı söylenmesine rağmen, J. Erman ve ark. [17][21] ve S. Zender ve ark. [42-43] çalışmalarında da belirtildiği gibi, P2P uygulamaları IANA tarafından belirlenmiş varsayılan port numaraları yerine, geçici olarak rastgele port numaralarını kullanmaktadır. [52] Yapılan tez çalışmasında ise, varsayılan ve sıklıkla kullanılan port numaraları üzerinden sınıflandırmalar ve karşılaştırmalar yapılacaktır. Ayrıca; P2P uygulamalarının dinamik port numaraları kullanımı, 2002 yılının sonlarına doğru ortaya çıkmaya başlamıştır. Kullanılan veri setindeki kayıtlar 2001 yılına aittir.[53]

Tablo 8.1 Ham veri seti protokol bilgileri ve dağılımları

Protokol (Port #)	Akış oranı (%)	Akış sayısı
FTP (21)	0,46	5418
SSH (22)	0,64	7495
SMTP (25)	1,47	17276
DNS (53)	6,57	76973
HTTP (80)	60,98	714796
HTTP (8080)	0,22	2586
HTTPS (443)	2,47	28953
POP3 (110)	0,34	4015
SOCK (1080)	4,71	55215
NNTP (119)	0,28	3251
IMAP (143)	0,07	836
IRC (113)	0,06	699
Half-Life (27015)	2,65	31089
NTP (123)	0,33	3813
SNMP (161)	0,09	998
Diğer Protokoller	18,66	218735
TOPLAM	100,00	1172148

Yukarındaki tablodan da anlaşılacağı gibi, veri setindeki kayıtların büyük bir çoğunluğunu HTTP(80) kayıtları oluşturmaktadır. Böyle bir durumda veri seti üzerinde yapılan incelemeler sağlıklı ve adil sonuçlar vermeyecektir. Bu nedenle, tez çalışmasında kullanılmak üzere, "protokol veri seti" ve "kategorik veri seti" olmak üzere iki veri seti hazırlanmıştır.

Veri setinin analizi için kullanılacak olan C4.5 algoritması, WEKA yazılımı içerisinde J48 olarak geçmektedir. Adaboost algoritmasında sınıflandırıcı olarak C4.5 ve Naive Bayes algoritmaları kullanılmıştır. Bu şekilde yükseltme (*boosting*) yönteminin algoritmalar üzerindeki etkisi incelenmeye çalışılmıştır.

Kümeleme algoritmalarında kümeler değerlendirme sınıflarına ayrılarak değerlendirilmişlerdir. Kümeleme algoritmalarının diğer algoritmalar ile karşılaştırmalarını yaparken, [36] çalışmasına benzer şekilde, algoritmaların doğruluk oranına (doğru kümelenmiş örnek sayısı oranı) ve veri setine ait bütün sınıflarının keşfedildiği optimal küme sayısına bakılarak karşılaştırma yapılmıştır.

Sınıflandırma algoritmaları analizinde test ve öğrenme işlemleri için, özellik seçimi algoritmalarında da olduğu gibi k-kare çapraz doğrulama yöntemi kullanılmıştır. Belirtilecek k sayısı olarak, sınıflandırma ve özellik seçiminde literatürde sıklıkla kullanılan on değeri alınmıştır.

Her iki veri setinin nitelik kümesi olarak, Tablo 7.1 de gösterilen kırk farklı nitelikten oluşan "tüm nitelik kümesi" kullanılmıştır. Nitelik kümesindeki bütün nitelikler nümerik değere sahiptir. Ayrıca; Nitelik seçme algoritmaları kullanılarak, analiz sırasında kullanılmayan veri seti içerisindeki gereksiz nitelikler çıkarılmış ve her iki veri seti içinde "seçilmiş nitelik kümesi" oluşturulmuştur. Böylece nitelik seçme işleminin, algoritmalar üzerindeki başarımlarına olan etkisi incelenmiştir.

8.1 Protokol Veri Seti Analizleri

Çalışmanın sağlıklı ve adil bir şekilde gerçekleştirilebilmesi için her veri türünde rastgele olarak beş yüz kayıt alınarak mümkün olduğuca farklı veri türünü içeren yeni homojen bir "protokol veri seti" oluşturulmuştur.

İncelenen protokol türü açısından tez çalışması, şu ana kadar yapılan çalışmalar arasında en fazla protokol türünü inceleyen çalışmaların başında olma özelliğine sahiptir.

Tablo 8.1 den de inceleneceği gibi on beş farklı uygulama katmanı protokolü seçilmiştir. Bu şekilde algoritmaların ağ trafiğini sınıflandırma hassasiyetleri daha yakından test edilmesi amaçlanmıştır.

Ağ trafiğini sınıflandırma işlemi sırasında protokol veri seti içerisinden kullanılmayan veya yeterince ayırt edici olmayan niteliklerin çıkarımı işleminde kullanılan algoritmalara ait nitelik seçimi işlemi sonuçları Tablo 8.2 gösterilmiştir.

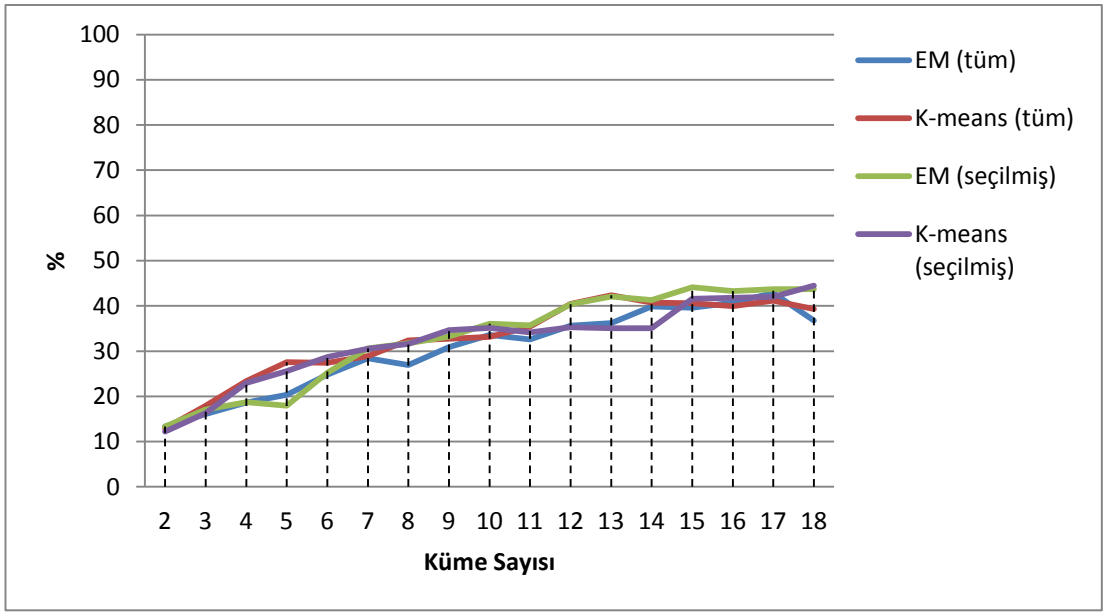
Tablo 8.2 Protokol veri seti nitelik seçimi işlemi sonuçları

Korelasyon-tabanlı (CFS)	**En iyi ilk ileri yönlü**	min_fpktl, mean_fpktl, max_fpktl, std_fpkl, max_bpktl, std_bpktl, min_biat, fpsh_cnt, bpsh_cnt
	En iyi ilk ileri yönlü	min_fpktl, mean_fpktl, max_fpktl, std_fpkl, max_bpktl, std_bpktl, min_biat, fpsh_cnt, bpsh_cnt
	Hızlı arama ileri yönlü	min_fpktl, mean_fpktl, max_fpktl, std_fpkl, max_bpktl, std_bpktl, min_biat, fpsh_cnt, bpsh_cnt
	Hızla arama geri yönlü	min_fpktl, mean_fpktl, max_fpktl, std_fpkl, max_bpktl, std_bpktl, min_biat, fpsh_cnt, bpsh_cnt
Tutarlılık-tabanlı (CON)	**En iyi ilk ileri yönlü**	total_fpackets, max_fpktl, max_bpktl, duration, fpsh_cnt
	En iyi ilk ileri yönlü	std_biat, max_active, std_idle, sflow_fbytes, sflow_bbytes, bpsh_cnt, total_fhlen, total_bhlen
	Hızlı arama ileri yönlü	Bütün nitelik kümesi
	Hızla arama geri yönlü	total_fpackets, total_bpackets, total_bvolume, mean_fpktl, mean_fpktl, max_fpktl, mean_fiat, duration

CFS algoritması bütün arama yöntemlerinde aynı baskın dokuz niteliği bularak oldukça başarılı bir performans göstermiştir. CON algoritması ise kullandığı her arama yöntemi altında farklı sonuçlar vermesi nedeniyle kullanılan veri setine ait yeterince tutarlı sonuçlar vermediği ortaya çıkmıştır. Bu nedenle; Seçilmiş nitelik kümesi kullanılarak protokol veri seti üzerinde yapılacak ağ trafiği analizde CFS algoritması kullanılmıştır.

Kümeleme algoritmalarının analizlerinde kullanılan EM ve K-means algoritmalarının tüm nitelik kümesi ve seçilmiş nitelik kümesi analiz sonuçları Tablo 8.3 de gösterilmiştir.

Tablo 8.3 Protokol veri seti küme sayısı – doğruluk oranı ilişki grafiği



Tablo 8.3 den de anlaşılacağı gibi, K-means ve EM algoritmalarının doğruluk oranı protokol veri setindeki protokol sayısına yaklaştıkça arttığı gözlenmiştir. Fakat, her iki kümedeki doğruluk artışı logaritmik olup küme sayısı daha da arttırılıp en uygun küme sayısı aralığından (veri setindeki protokol sayısı veya yakın komşuluğu) uzaklaşmaya başlandığında doğruluk oranların düşmeye başladığı gözlenmiştir.

Tüm nitelik kümesini kullanarak yapılan incelemede, beklenen en uygun küme sayısı aralığında K-means algoritması maksimum doğruluk değerine (%42,36) küme sayısı on üç olduğunda ulaşmıştır. Bütün protokollerin baskın olarak oluşturduğu küme sayısı olan on yedi de ise doğruluk oranı (%41) daha düşüktür. En yüksek doğruluk oranında POP3 ve NTP protokollerinin baskın küme sınıfı oluşturamadıkları gözlemlenmiştir.

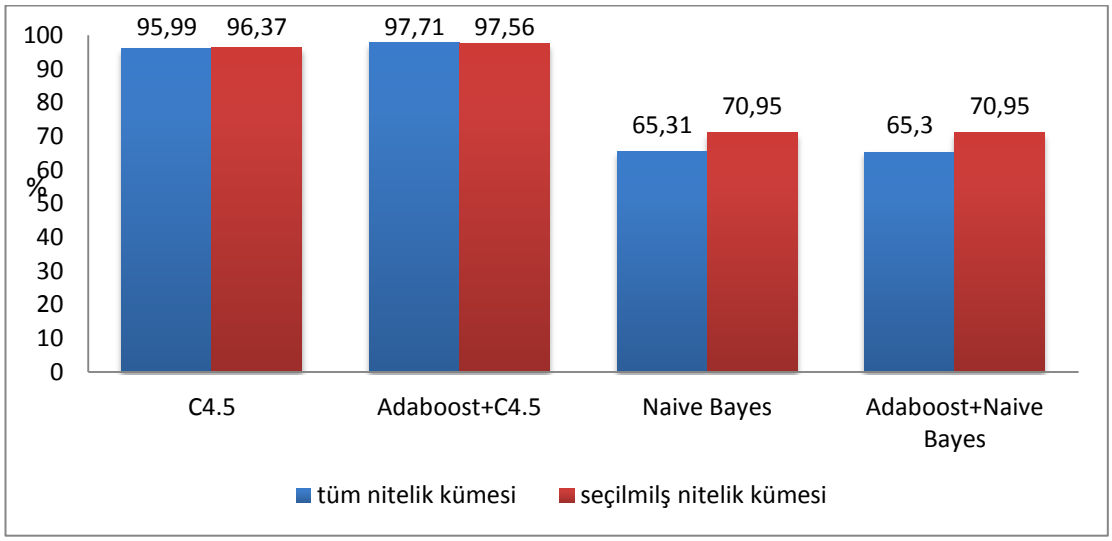
EM algoritması küme sayısı belirleyebilme yeteneğine sahip olmasına rağmen tüm nitelik kümesi kullanılarak yapılan incelemede sadece HTTP(8080), SSH, NNTP ve IMAP protokol küme sınıflarını oluşturabilmiş, doğruluk oranı çok küçük bir değer (%18,67) de kalmıştır. Beklenen küme aralığında ise, en yüksek doğruluk oranına

(%39,88) küme sayısı on dört olduğunda ulaşmıştır. Bu küme sayısında NNTP ve IRC küme sınıflarını oluşturamamıştır.

Nitelik seçimi kullanılarak incelenen protokol veri setinde her iki kümeleme algoritmasının da başarılarının arttığı gözlemlenmiştir. K-means algoritması en yüksek doğruluk oranına küme sayısı on sekize ulaştığında (%44,47), EM algoritması ise küme sayısı on beş olduğunda (%44,1) ulaşmıştır. Fakat; EM algoritması kendi küme sayısını belirleme sırasında HTTP(8080), NTP ve SMTP olmak üzere üç küme sınıfı oluşturmuş, doğruluk oranı tüm nitelik kümesindeki doğruluk oranına göre %1,7 azalmıştır.

Sınıflandırma algoritmalarının analizinde kullanılan C4.5, Naive Bayes ve bu algoritmaların sınıflandırma yeteneklerinin yükseltilebilmesini sağlayan Adaboost algoritmalarının, tüm nitelik kümesi ve seçilmiş nitelik kümesine göre doğruluk oranları Tablo 8.4 de gösterilmiştir.

Tablo 8.4 Protokol veri seti sınıflandırma algoritmaları doğruluk oranları grafiği



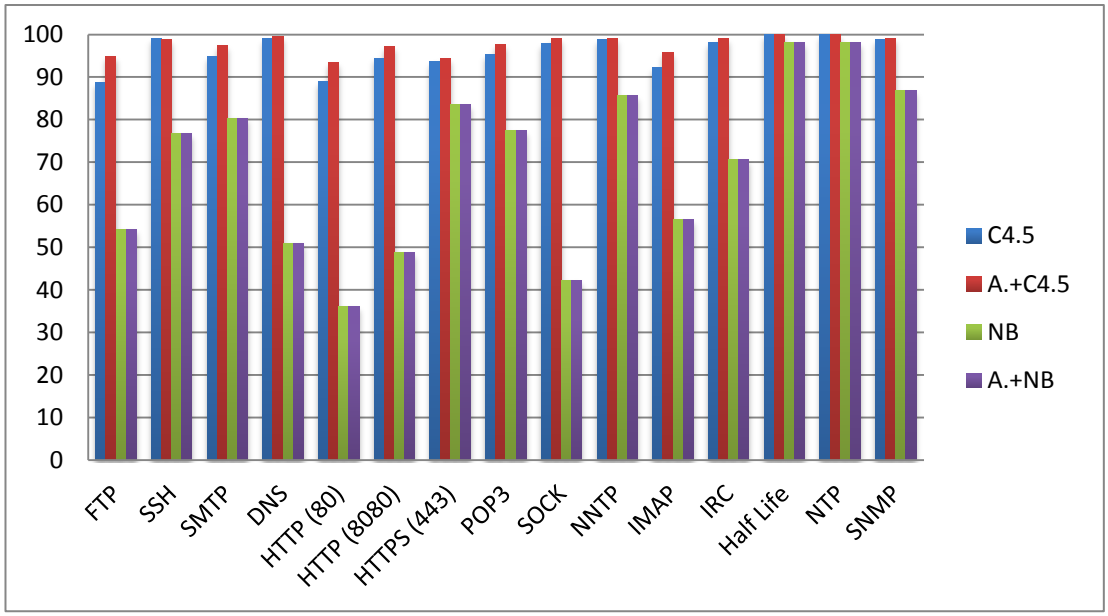
Protokol veri setine göre sınıflandırma algoritmalarının, hem tüm nitelik kümesine göre sınıflandırmada hem de seçilmiş nitelik kümesine göre sınıflandırmada kümeleme algoritmalarına göre çok daha başarılı oldukları gözlemlenmiştir.

Sınıflandırma algoritmalarının seçilmiş nitelik kümesi kullanarak elde ettikleri doğruluk değerleri, tüm nitelik kümesi kullanımlarına göre daha yüksek veya çok yakın olduğu gözlemlenmiştir.

Adaboost algoritması protokol veri seti üzerinde Naive Bayes algoritmasının kullanımında, algoritmanın sınıflandırma başarısını değiştirmemiştir. C4.5 algoritmasında ise, tüm nitelik kümesi üzerinden veri setini sınıflandırmada yaklaşık %1,7 lik, seçilmiş nitelik kümesi üzerinde sınıflandırmada da yaklaşık %1,2 lik geliştirme göstermiştir.

Tablo 8.5 den de anlaşılacağı gibi bütün protokollerde C4.5 algoritmasının kesinlik değerleri Naive Bayes algoritmasına göre daha yüksektir. Sadece SSH protokolünde Adaboost+C4.5 algoritmasının kesinlik değeri, C4.5'a göre düşüktür. C4.5 ve Naive Bayes algoritmaları en yüksek kesinlik değerlerine NNTP, Half Life, NTP ve SNMP protokollerini sınıflandırırken yakalamışlardır. C4.5 algoritması ayrıca, SSH, SOCK, IRC ve DNS protokollerini sınıflandırırken yine oldukça yüksek kesinlik başarımı yakalamıştır. Bu algoritmalar beraber incelendiklerinde düşük kesinlik yüzdelerini, FTP, HTTP(80), IMAP gibi protokolleri sınıflandırırken almışlardır. HTTP(8080), DNS ve SOCK protokolünü sınıflandırırken Naive Bayes algoritmasının kesinlik değeri ortalamanın altında bir başarı göstermesine rağmen, C4.5 algoritmasının kesinlik değeri ortalamanın üstünde veya ortalamaya yakın kalmıştır. HTTPS protokolünün sınıflandırılmasında ise tam ters bir durum söz konusudur. Naive Bayes algoritması ortalamanın üstünde başarım göstermesine rağmen, C4.5 algoritması kendi ortalamasının altında bir başarım göstermiştir.

Tablo 8.5 Tüm nitelik kümesi ile sınıflandırma algoritmaları kesinlik grafiği



Sınıflandırma algoritmalarının protokol veri setini kullanarak tüm nitelik kümesi üzerindeki ortalama, minimum ve maksimum kesinlik başarım değerleri Tablo 8.6 da gösterilmektedir.

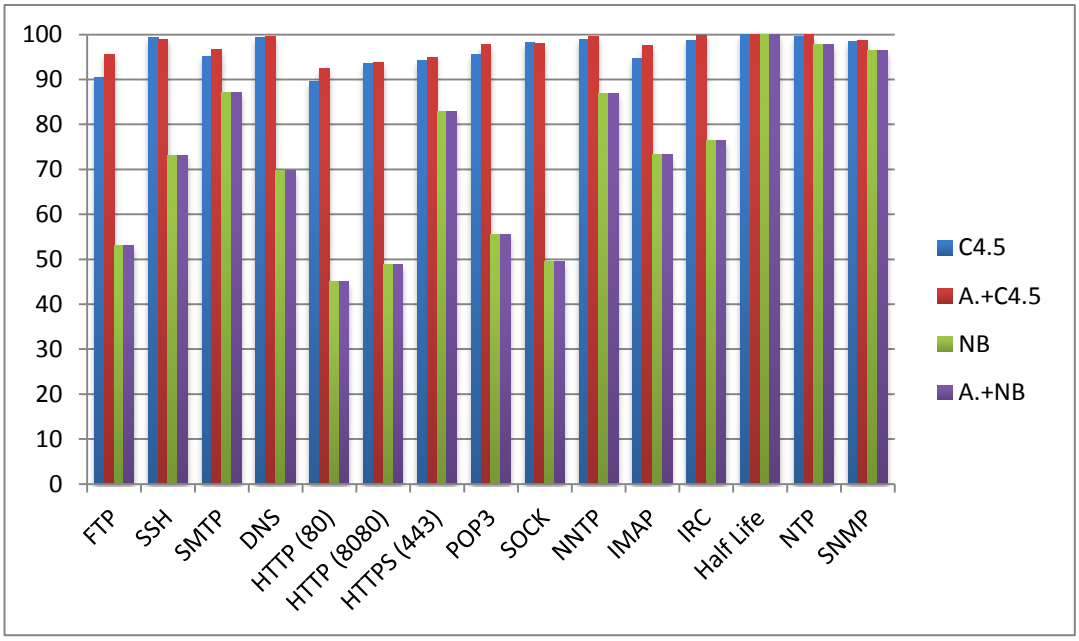
Tablo 8.6 Tüm nitelik kümesi ile sınıflandırma algoritmaları kesinlik değerleri

	C4.5	**A.+C4.5**	**NB**	**A.+NB**
Maksimum	100	100	98,2	98,2
Minimum	88,7	93,4	36	36
Ağırlıklı Ortalama	96	97,71	69,76	69,76

Tablo 8.7 ve Tablo 8.8 de, Sınıflandırma algoritmaların seçilmiş nitelik kümesini kullanarak göstermiş oldukları kesinlik başarımları bulunmaktadır.

Her iki algoritmanın kesinlik değerleri, tüm nitelik kümesi ile yapılan kesinlik değerlerine göre daha yüksektir. Seçilmiş nitelik kümesi sonuçları da tüm nitelik kümesi sonuçları ile genel olarak örtüşmektedir. Naive Bayes algoritmasının kesinlik değerleri tüm nitelik kümesinde olduğu gibi Adaboost+Naive Bayes algoritması kesinlik değerleri ile aynıdır. Adaboost+C4.5 algoritması ise SSH protokolü hariç C4.5 algoritmasına göre daha yüksek kesinlik değerleri vermektedir.

Tablo 8.7 Seçilmiş nitelik kümesi ile sınıflandırma algoritmaları kesinlik grafiği



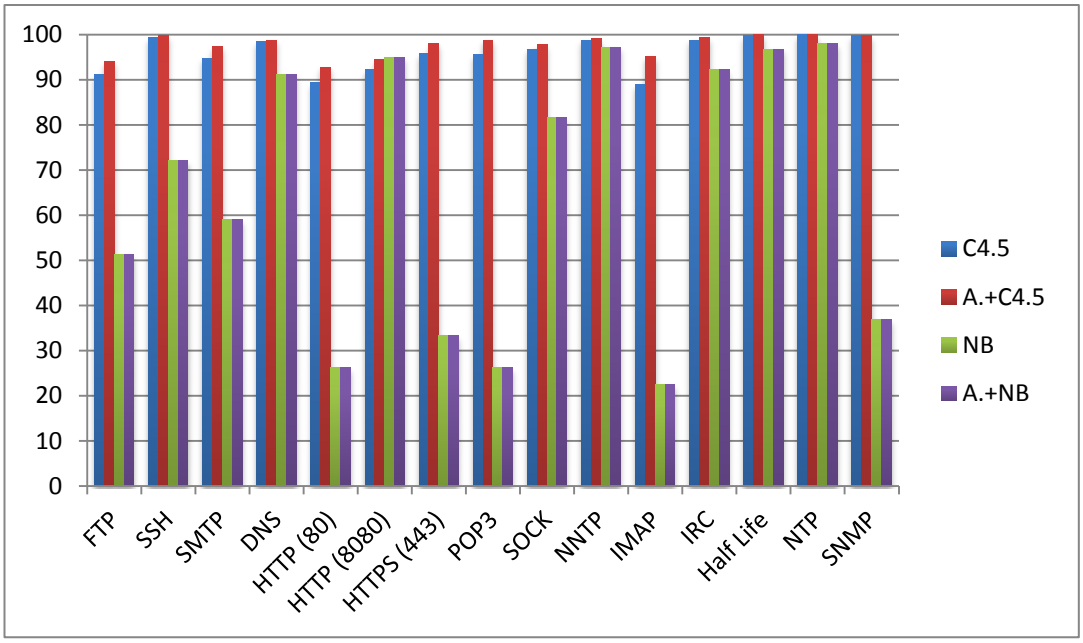
Tablo 8.8 Seçilmiş nitelik kümesi ile sınıflandırma algoritmaları kesinlik değerleri

	C4.5	**A.+C4.5**	**NB**	**A.+NB**
Maksimum	100	100	100	100
Minimum	89,6	92,4	45	45
Ağırlıklı Ortalama	96,38	97,55	73,05	73,05

Tablo 8.9 ve Tablo 8.10 de, Sınıflandırma algoritmalarının tüm nitelik kümesi kullanılarak oluşturduğu protokol sınıflarının duyarlılık bilgileri bulunmaktadır.

Tüm nitelik kümesi kullanılarak yapılan sınıflandırmanın duyarlılık analizleri bazı noktalarda kesinlik değerleri ile örtüşmektedir. Sınıflandırma algoritmalarında ortak olarak, FTP, HTTP(80) ve IMAP protokolü değerlerinin duyarlılık analizinde yine düşük olduğu gözlemlenmiştir. Half Life, NNTP ve NTP protokollerinin başarımları yine her iki algoritmada da oldukça yüksektir. Kesinlik değerlerinden farklı olarak, HTTP(8080), IRC ve DNS protokollerinin Naive Bayes başarım değerleri oldukça artmasına rağmen, SMTP, IMAP, POP3 ve SNMP protokol sınıflandırma duyarlılıkları oldukça azalmıştır. SSH protokolünde, C4.5 algoritması Adaboost+C4.5'e göre daha düşük duyarlılık değerinde kalmıştır. C4.5 algoritması yine Naive Bayes algoritmasına göre yüksek başarım göstermiştir.

Tablo 8.9 Tüm nitelik kümesi ile sınıflandırma algoritmaları duyarlılık grafiği



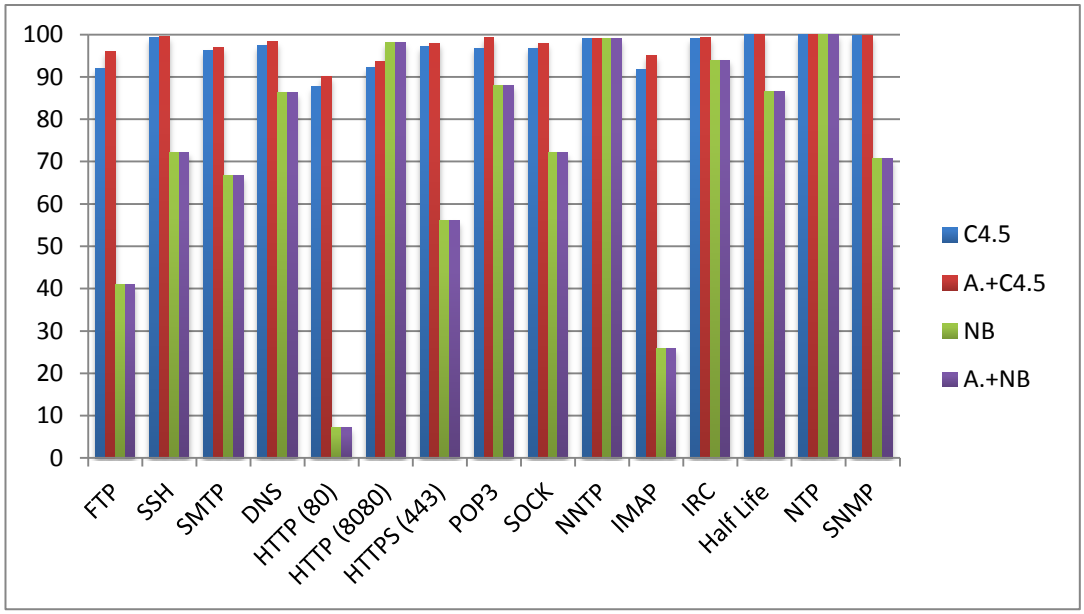
Tablo 8.10 Tüm nitelik kümesi ile sınıflandırma algoritmaları duyarlılık değerleri

	C4.5	**A.+C4.5**	**NB**	**A.+NB**
Maksimum	100	100	98	98
Minimum	89	92,8	22,6	22,6
Ağırlıklı Ortalama	95,99	97,71	65,31	65,31

Tablo 8.11 ve Tablo 8.12 da, sınıflandırma algoritmalarının seçilmiş nitelik kümesini kullanarak yapmış oldukları duyarlılık değerleri bulunmaktadır.

Seçilmiş nitelik kümesi kullanılarak yapılan sınıflandırmanın duyarlılık analizi sonuçları yine tüm nitelik kümesi kullanılarak elde edilen sonuçlar ile genel olarak örtüşmektedir. C4.5 algoritması bütün sınıflarda yüksek başarım değeri almıştır. Fakat; HTTP(80) protokolünde ise yine kendisinin en düşük başarım yüzdesini almıştır. Naive Bayes algoritması, HTTP(80) , FTP ve DNS değerleri tüm nitelik kümesi duyarlılık değerlerlerine göre düşük başarım değerlerini almıştır. Naive Bayes algoritmasının, POP3, SNMP ve HTTPS protokol sınıflandırma duyarlılıkları tüm nitelik kümesi göre oldukça artmıştır. Naive Bayes algoritması HTTP(8080) protokolündeki değeri ile ilk defa bir kıyaslamada C4.5 algoritmasının başarım değerinden daha yüksek bir yüzde almıştır.

Tablo 8.11 Seçilmiş nitelik kümesi ile sınıflandırma algoritmaları duyarlılık grafiği



Tablo 8.12 Seçilmiş nitelik kümesi ile sınıflandırma algoritmaları duyarlılık değerleri

	C4.5	**A.+C4.5**	**NB**	**A.+NB**
Maksimum	100	100	100	100
Minimum	87,8	90	7,2	7,2
Ağırlıklı Ortalama	96,37	97,56	70,95	70,95

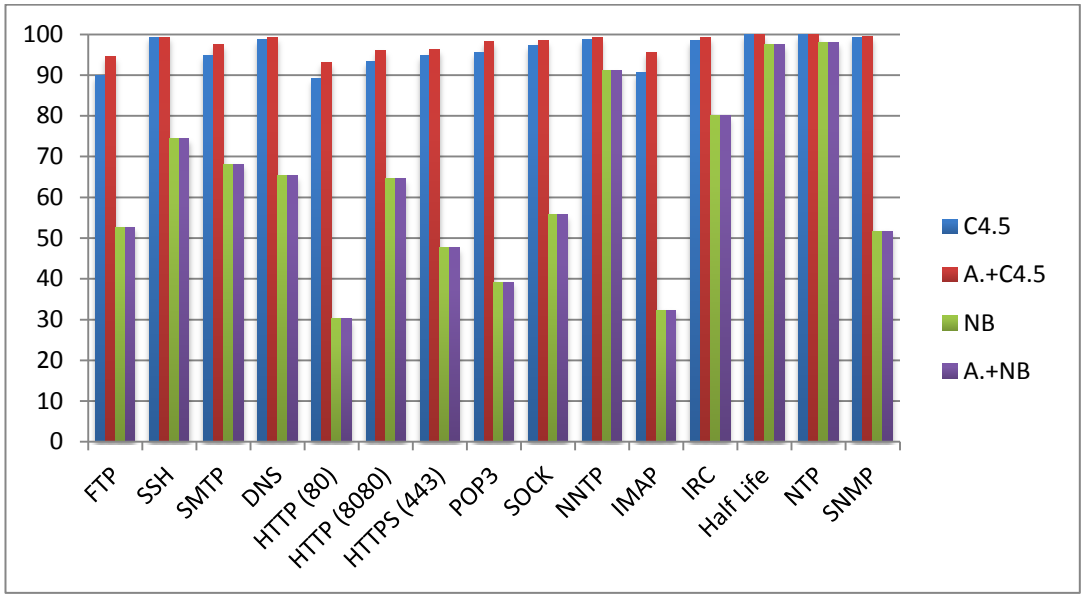
Seçilmiş nitelik kümesi kullanılarak yapılan duyarlılık ölçümlerinde tüm nitelik kümesine bakılarak yapılan duyarlılık ölçümlere göre daha yüksek başarım değerleri elde edildiği görülmüştür.

Tablo 8.13 ve Tablo 8.14 de, tüm nitelik kümesini kullanılarak incelenen veri seti üzerindeki duyarlılık ve kesinlik bilgilerinin harmonik ortalaması alınarak incelenen f-ölçütü bilgileri bulunmaktadır. Bu tablodaki verilere bakarak tüm nitelik kümesi ile yapılan sınıflandırma analizi hakkında daha doğru incelemeler yapılabilir.

Her iki algoritmada Half Life, NTP ve NNTP protokol sınıflarında yüksek başarım elde etmişlerdir. FTP, HTTP(80) ve IMAP protokol sınıflarında ise en düşük başarım yüzdelerini almışlardır. Ortak olarak yüksek başarım elde edilen protokol sınıfları

hariç geri kalan bütün sınıflarda C4.5 algoritması Naive Bayes e göre oldukça yüksek başarım yüzdesi yakalamıştır.

Tablo 8.13 Tüm nitelik kümesi ile sınıflandırma algoritmaları f-ölçütü grafiği



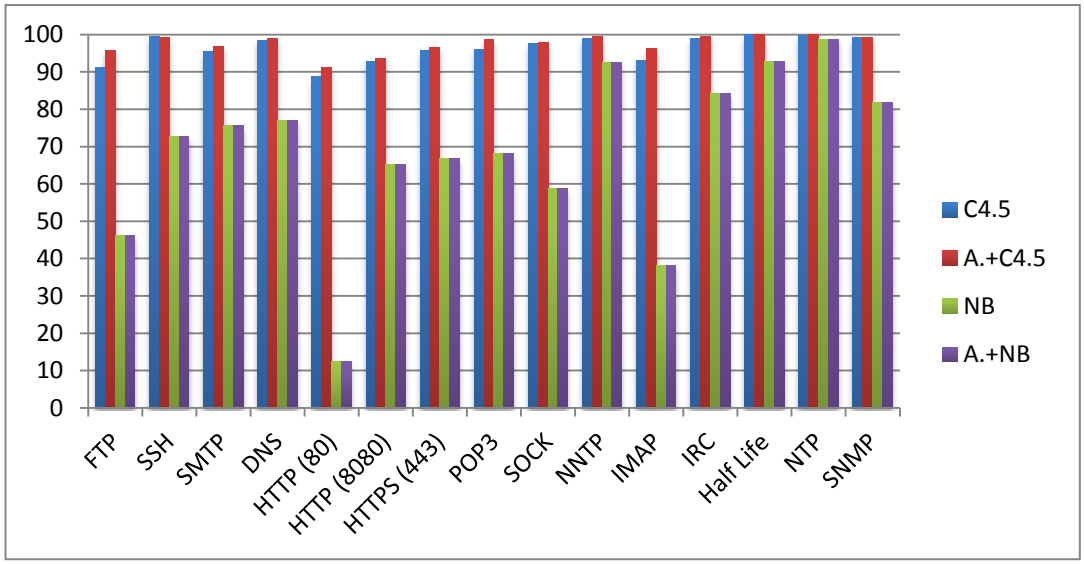
Tablo 8.14 Tüm nitelik kümesi ile sınıflandırma algoritmaları f-ölçütü değerleri

	C4.5	**A.+C4.5**	**NB**	**A.+NB**
Maksimum	100	100	98,1	98,1
Minimum	89,1	93,1	30,3	30,3
Ağırlıklı Ortalama	95,98	97,71	63,24	63,24

Protokol veri setinde son olarak incelenecek Tablo 8.15 ve Tablo 8.16 tablolarında, veri seti üzerinde seçilmiş nitelik kümesi kullanarak yapılan f-ölçütü değerleri bulunmaktadır.

Naive Bayes algoritması ile yapılan analiz sonucunda, FTP ve HTTP(80) protokol sınıfları hariç geri kalan bütün sınıfların f-ölçütü başarım değeri tüm nitelik kümesi değerlerine göre artmıştır. C4.5 ve Adaboost+C4.5 algoritmaları bütün sınıflarda Naive Bayes ve Adaboost+Naive Bayes'e göre yüksek başarım değerlerini almıştır. Fakat; HTTP(80) protokollerinin sınıflandırılmasında kendilerinin en düşük başarım değerini almıştır.

Tablo 8.15 Seçilmiş nitelik kümesi ile sınıflandırma algoritmaları f-ölçütü grafiği



Tablo 8.16 Seçilmiş nitelik kümesi ile sınıflandırma algoritmaları f-ölçütü değerleri

	C4.5	**A.+C4.5**	**NB**	**A.+NB**
Maksimum	100	100	98,8	98,8
Minimum	88,7	91,2	12,4	12,4
Ağırlıklı Ortalama	96,37	97,56	68,76	68,76

8.2 Kategorik Veri Seti Analizleri

Protokol veri setinde protokol türlerine ait sınıf değerleri sayısının yüksek olması ve protokol türlerinin birbirleri ile yakından ilişkili olması nedeniyle, ağ trafiği sınıflandırma işlemi oldukça hassas gerçekleştirildiği söylenebilir. Birbirlerine çok yakın protokol sınıflarının bir arada incelenmesi nedeniyle algoritmalar oluşturdukları sınıflara örnekler yerleştirmede ve algoritmanın öğrenme aşamasında sıkıntılar yaşadığı görülmüştür. Aslında farklı bir protokol yanlışlıkla kendisine benzeyen bir başka protokol sınıfına yerleştirilebilmiştir. Bu ve bunun gibi, hassas incelemelerin sonucunda oluşabilecek soru işaretlerinin ve yanlış anlaşılmaların giderilmesi için, protokol veri seti içerisindeki benzer trafik protokolleri bir araya getirilerek, yeni oluşturulan "kategorik veri seti" üzerinden ağ trafiği sınıflandırma

işlemi tekrar yapılmıştır. Tablo 8.17 deki gibi oluşturulan her bir kategoride bin kayıt bulunmaktadır.

Tablo 8.17 Kategorik veri seti içerik tablosu

Kategori türü	Protokol türleri (Protokol (Port #))
WWW	HTTP(80), HTTP(8080), HTTPS(443)
MAIL	SMTP(25), POP3(110), IMAP(143)
SERVIS	DNS(53), NTP(123)

Kategorik veri seti kullanarak ağ trafiğini sınıflandırma işlemi sırasında kullanılmayan veya yeterince ayırt edici olmayan niteliklerin çıkarımı işleminde kullanılan algoritmalara ait nitelik seçimi sonuçları Tablo 8.18 de gösterilmiştir.

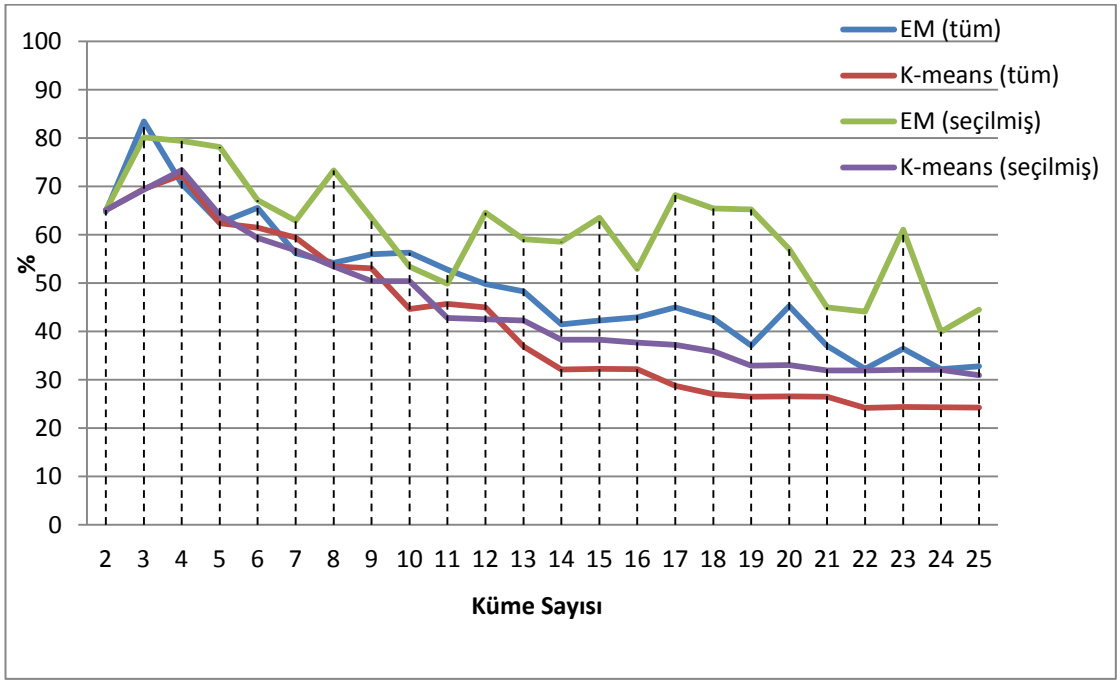
Tablo 8.18 Kategorik veri seti nitelik seçimi işlemi sonuçları

Korelasyon-tabanlı (CFS)	**En iyi ilk ileri yönlü**	proto, min_fpktl, max_fpktl, std_fpktl, std_bpktl, fpsh_cnt, total_bhlen
	En iyi ilk ileri yönlü	proto, min_fpktl, max_fpktl, std_fpktl, std_bpktl, fpsh_cnt, total_bhlen
	Hızlı arama ileri yönlü	proto, min_fpktl, max_fpktl, std_fpktl, std_bpktl, fpsh_cnt, total_bhlen
	Hızla arama geri yönlü	proto, min_fpktl, max_fpktl, std_fpktl, std_bpktl, fpsh_cnt, total_bhlen
Tutarlılık-tabanlı (CON)	**En iyi ilk ileri yönlü**	max_fpktl, std_fpktl, max_bpktl fpsh_cnt
	En iyi ilk ileri yönlü	max_active, min_idle, max_idle, sflow_fbytes, sflow_bpackets, sflow_bbytes, fpsh_cnt, bpsh_cnt, total_fhlen
	Hızlı arama ileri yönlü	Bütün nitelik kümesi
	Hızla arama geri yönlü	total_fpackets, mean_fpktl, max_fpktl, mean_bpktl

CFS algoritması kategorik veri seti nitelik kümesi incelemesinde de protokol veri seti incelemesinde olduğu gibi bütün arama yöntemlerinde yedi tür aynı baskın nitelikleri bularak oldukça başarılı bir performans göstermiştir. CON algoritması ise yine daha önce olduğu gibi kullandığı her arama yöntemi altında farklı sonuçlar vermesi nedeniyle kullanılan veri setine ait yeterince tutarlı sonuçlar vermediği ortaya çıkmıştır. Bu nedenle; Seçilmiş nitelik kümesi kullanılarak protokol veri seti üzerinde yapılacak ağ trafiği analizde yine CFS algoritması kullanılmıştır.

EM ve K-means algoritmalarının kategorik veri seti üzerinde tüm nitelik kümesi ve seçilmiş nitelik kümesi kullanılarak yapılan analiz sonuçları Tablo 8.19 da gösterilmiştir.

Tablo 8.19 Kategorik veri seti küme sayısı - doğruluk oranı ilişki grafiği



K-means ve EM algoritmalarının, tüm nitelik kümesi ve seçilmiş nitelik kümesi kullanılarak yapılan incelemelerinde en uygun küme aralığında (oluşturulan sınıf sayısı ve komşuluğu) algoritmaların doğruluk oranlarının en yüksek olduğu ve en uygun küme aralığından uzaklaştıkça algoritmanın doğruluk değerlerinin hızla düşmeye başladığı görülmüştür.

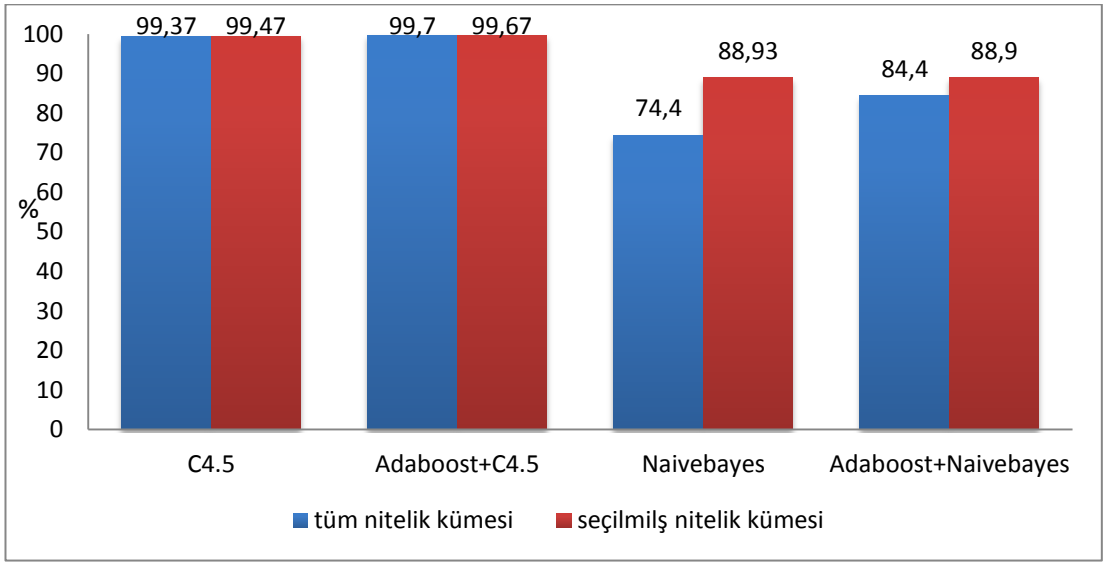
K-means algoritmasının hem tüm nitelik kümesi hem de seçilmiş nitelik kümesi kullanılarak yapılan incelemelerinin her ikisinde en yüksek doğruluk oranı (%72,43) eşit çıkmıştır ve bu en yüksek doğruluk oranına küme sayısı dört olduğunda ulaşmıştır.

EM algoritması kendi belirlediği küme sayısı olarak, tüm nitelik kümesini kullanarak kategorik veri setini incelenmesinde uygun küme sayısı yedidir ve bu kümelere ait doğruluk oranını %56,2 olarak bulmuştur. Nitelik seçimini kullanarak incelenmesinde ise küme sayısını yine yedi olarak belirlemiş fakat doğruluk oranını %62,93 bulmuştur.

EM algoritması, küme sayısı araştırmacı tarafından belirtilerek ve tüm nitelik kümesi kullanılarak yapılan incelemede, en yüksek doğruluk oranına (%83,47) küme sayısı üç olduğunda ulaşmıştır. Seçilmiş nitelik kümesi kullanılarak yapılan incelemede de en yüksek doğruluk oranına (%80,17) küme sayısı üç olduğunda ulaşmıştır.

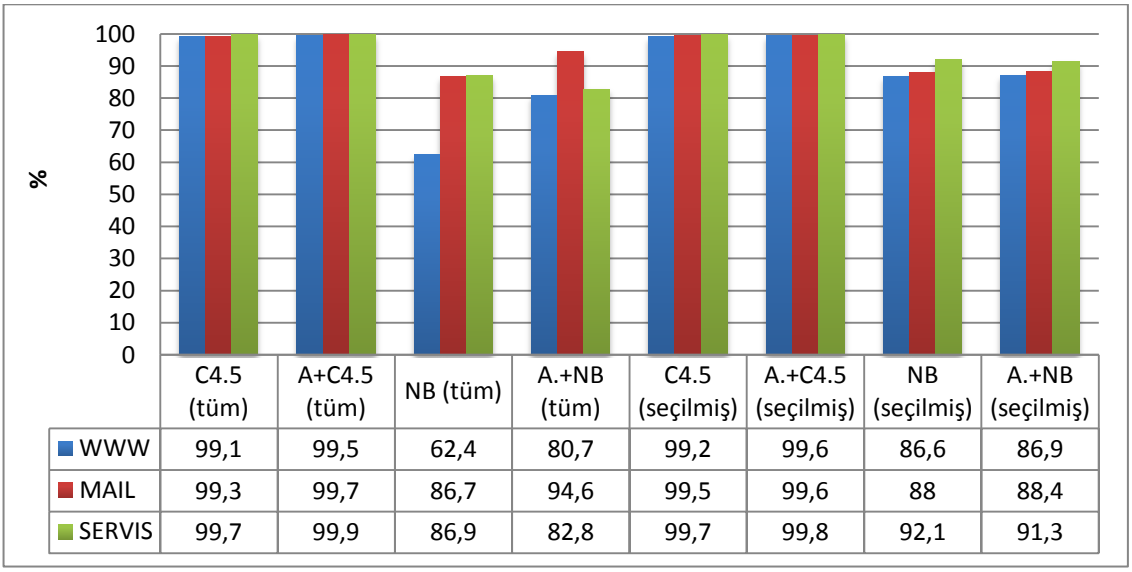
Bu durumda; Nitelik seçimi işleminde EM algoritmasında doğruluk değeri üzerinde çok küçük bir düşüş oluşturduğu söylenebilir. K-means algoritmasında ise herhangi bir değişiklik oluşturmamıştır.

Tablo 8.20 Kategorik veri seti sınıflandırma algoritmaları doğruluk grafiği



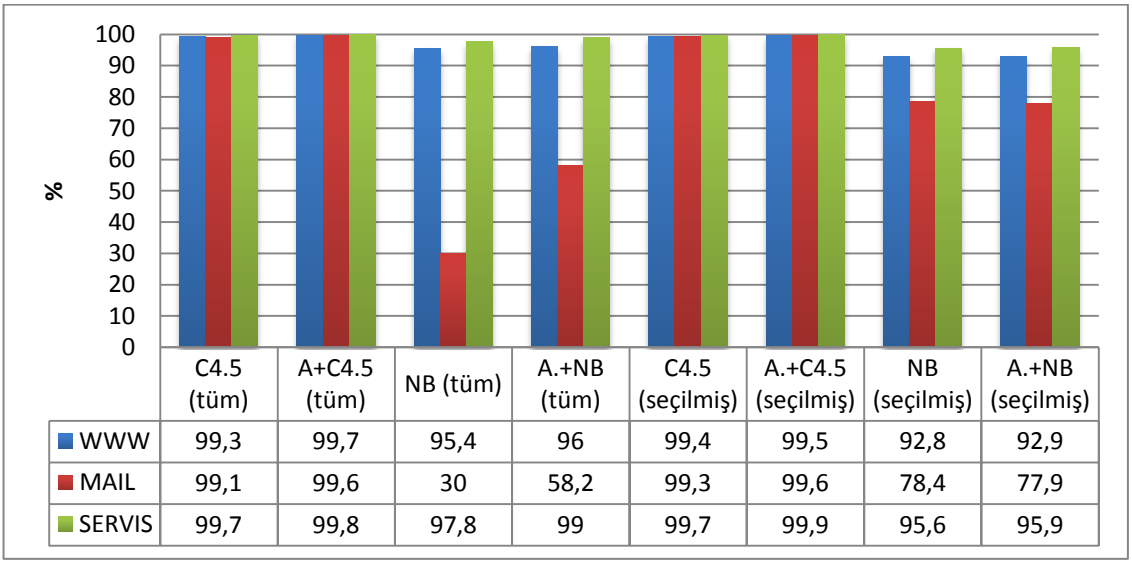
Tablo 8.20 den incelenebileceği gibi, hem nitelik seçme işlemi hem de Adaboost algoritması ile boosting işlemi Naive Bayes algoritmasının doğruluk oranında önemli artma sağlamıştır. C4.5 algoritması hem yalnız başına hem de Adaboost algoritması ile kullanımda daha önce protokol veri setinde olduğu gibi çok yüksek doğruluk değeri göstermiştir.

Tablo 8.21 Kategorik veri seti sınıflandırma algoritmaları kesinlik grafiği



	C4.5 (tüm)	A+C4.5 (tüm)	NB (tüm)	A.+NB (tüm)	C4.5 (seçilmiş)	A.+C4.5 (seçilmiş)	NB (seçilmiş)	A.+NB (seçilmiş)
WWW	99,1	99,5	62,4	80,7	99,2	99,6	86,6	86,9
MAIL	99,3	99,7	86,7	94,6	99,5	99,6	88	88,4
SERVIS	99,7	99,9	86,9	82,8	99,7	99,8	92,1	91,3

Tablo 8.22 Kategorik veri seti sınıflandırma algoritmaları duyarlılık grafiği



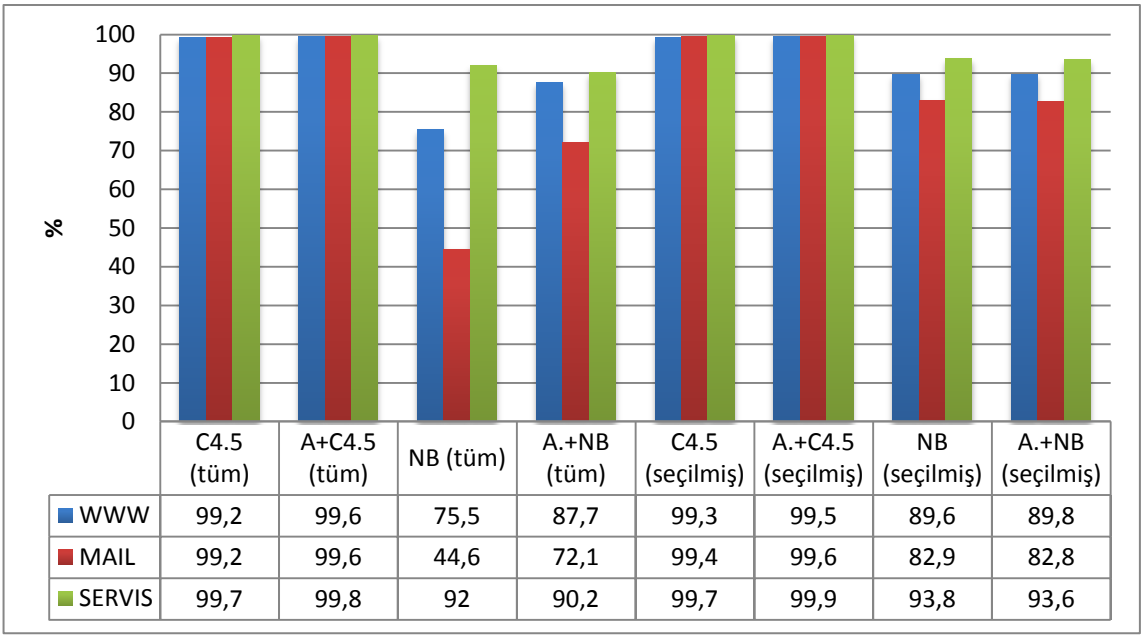
	C4.5 (tüm)	A+C4.5 (tüm)	NB (tüm)	A.+NB (tüm)	C4.5 (seçilmiş)	A.+C4.5 (seçilmiş)	NB (seçilmiş)	A.+NB (seçilmiş)
WWW	99,3	99,7	95,4	96	99,4	99,5	92,8	92,9
MAIL	99,1	99,6	30	58,2	99,3	99,6	78,4	77,9
SERVIS	99,7	99,8	97,8	99	99,7	99,9	95,6	95,9

Tablo 8.21 ve Tablo 8.22 de; Tüm nitelik kümesi ve seçilmiş nitelik kümesi kullanılarak kategorik veri seti üzerinden algoritmaların ve oluşturdukları sınıfların kesinlik ve duyarlılık değerleri bulunmaktadır. Duyarlılık işleminde WWW ve SERVIS sınıflarında her iki algoritma da minimum %92 başarı göstermişlerdir. MAIL sınıfında ise; C4.5 algoritması yüksek duyarlılık başarısı göstermesine rağmen, Naive Bayes algoritması çok düşük başarım göstermiştir. Kesinlik değerlerinde ise; Naive Bayes algoritması WWW sınıfında düşük başarım göstermiştir. Her iki sınıflandırma metriğinde de bu başarım Adaboost ve nitelik seçimi yöntemi kullanılarak oldukça geliştirilebildiği gözlemlenmektedir.

MAIL sınıfta düşük başarım gösterme nedeni olarak, IMAP ve POP3 protokollerinin WWW sınıfından oldukça etkilenmesinden kaynaklandığı görülmektedir. POP3 protokolü daha önceki çalışmalarda da belirtildiği gibi [39] baskın olmayan bir protokoldür. Bu nedenle; Kesinlik ve duyarlılık ölçümlerinde kendi sınıfı ve diğer sınıflar arasında başarım düşüklüğü yaratabileceği görülmektedir.

Tablo 8.23 de, Tablo 8.22 ve Tablo 8.21 değerleri ile oluşturulan f-ölçütü verileri bulunmaktadır. Daha önce tezde değinildiği gibi f-ölçütü bireysel olarak kesinlik ve duyarlılığın incelenmesinden daha doğru sonuçlar vermektedir. Mevcut veri seti sonuçlarında ise, duyarlılık tablosunda anlatımlar ile f-ölçütü sonuçları uyuşmaktadır. f-ölçütü sonuçlarına göre başarımın duyarlılık başarımlarına göre biraz daha arttığı da dikkat edilmesi gereken bir noktadır.

Tablo 8.23 Kategorik veri seti sınıflandırma algoritmaları f-ölçütü grafiği



	C4.5 (tüm)	A+C4.5 (tüm)	NB (tüm)	A.+NB (tüm)	C4.5 (seçilmiş)	A.+C4.5 (seçilmiş)	NB (seçilmiş)	A.+NB (seçilmiş)
WWW	99,2	99,6	75,5	87,7	99,3	99,5	89,6	89,8
MAIL	99,2	99,6	44,6	72,1	99,4	99,6	82,9	82,8
SERVIS	99,7	99,8	92	90,2	99,7	99,9	93,8	93,6

9. SONUÇLAR VE GELECEK ÇALIŞMALAR

Yapılan tez çalışmasında, port-tabanlı ve yük-tabanlı sınıflandırmalarının günümüzde etkisini yitirmesi nedeniyle ortaya çıkan makine öğrenme tabanlı sınıflandırma kullanılmış ve makine öğrenme algoritmaları karşılaştırılmıştır.

Gerçekleştirilen işlemin daha kolay anlaşılabilmesi için; Veri madenciliği ve makine öğrenme kavramları hakkında bilgiler verilmiştir. Ağ trafiği analizini anlayabilmek için, internet üzerinden iletişimin nasıl gerçekleştiği hakkında bilgiler verilmiştir.

Sıklıkla kullanılan ağ trafiği sınıflandırma metrikleri anlatılmıştır. Sınıflandırma ve kümeleme algoritmaları kendi aralarında algoritma başarım metrikleri kullanılarak karşılaştırılmışlardır.

Kümeleme algoritmaları, en uygun küme sayısı aralığında maksimum doğruluk değerine ulaştığı, EM algoritmasının kendi oluşturduğu küme sayısında yeterince yüksek doğruluk değerine ulaşamadığı gözlemlenmiştir. Bu durum, kümeleme

işlemini yapan araştırmacının en uygun küme sayısı hakkında ön bilgiye sahip olması gerektiği sonucunu doğrulamıştır.

Gelecekteki çalışmalar içerisinde, K-means ve EM algoritmalarının araştırmacı tarafından karar verilen küme sayılarını, algoritmaların kendilerinin, etkin bir şekilde belirleyebileceği şekilde geliştirilmeler yapılabilir.

Sınıflandırma algoritmalarının karşılaştırmalarında kullanılan duyarlılık ve kesinlik metriklerinin bazen birbirleri ile ters sonuçlar verebileceği, bu yüzden karşılaştırma yaparken doğruluk ve f-ölçütü değerlerinin incelenmesinin daha doğru olduğu gözlemlenmiştir.

Sınıflandırma algoritmalarının performanslarını yükseltmek için kullanılan yükseltme algoritması Adaboost'un, C4.5 ve Naive Bayes algoritmalarının öğrenme yeteneklerini çoğu durumda geliştirdiği gözlemlenmiştir. C4.5 algoritması bütün metriklerde Naive Bayes algoritmasına göre daha başarılı olduğu sonucu çıkarılmıştır.

Nitelik seçme yöntemleri ve kullandıkları arama yöntemleri incelenmiştir. CFS algoritmasının CON algoritmasına göre çok daha tutarlı sonuçlar vermesi nedeniyle CFS algoritması incelenmiş ve CON algoritmasının veri seti üzerindeki etkisi gelecek çalışmalara bırakılmıştır. CFS algoritması kullanılarak sadeleştirilen nitelik kümesinin, algoritmaların başarımlarını arttırdığı ve algoritmaların çalışma zamanlarında ciddi kazanımlar sağladığı gözlemlenmiştir.

Ayrıca; Ağ trafiği analizi gerçekleştirilirken analiz sırasında dikkat edilmesi gereken önemli bir nokta ortak baskın niteliklerin ortaya çıkarılmasıdır. CFS algoritması ile yapılan nitelik seçimi işleminde de, hem kategorik veri setine ait 9 nitelik türünde hem de protokol veri setine ait 7 nitelik türünde ortak olarak çıkarılmış 5 tür (*min_fpktl, max_fpktl, std_fpktl, std_bpktl, fpsh_cnt*) nitelik olduğu gözlemlenmiştir. Bu durumda sonuç olarak, kullanılan ham veri seti üzerinde bu 5 türün baskın nitelik olduğu söylenebilir.

KAYNAKLAR

[1] Witten I. & Frank E., Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations, *Morgan Kaufmann*, 2000.

[2] Herbert A. Simon, Why should machines learn?, *Carnegie-Mellon University*, Departments of Computer Science and Psychology, USA, 1983.

[3] Taiwo Oladipupo Ayodele, Types of Machine Learning Algorithms, New Advance in Machine Learning, *Portsmouth University*, England, Edited by Yagang Zhang, 2010.

[4] Ryszad S. Michalski, Ivan Bratko, Miroslav Kubat, Machine Learning and Data Mining: Methods and Applications, Wiley, 1998.

[5] "Data Mining Definition and Origins" erişim adresi: http://www.dataminingarticles.com/data-mining-introduction.html, erişim tarihi: Ağustos 2012.

[6] Pang-Ning Tan, Michael Steinbach & Vipin Kumar, Introduction to Data Mining, *Pearson*, 2005.

[7] J. Han & M. Kamber, Data Mining Concepts and techniques, *Morgan Kaufmann*, 2006.

[8] X. Wu, V. Kumar, J. R. Quinlan, J. Ghosh, Q. Yang, H. Motoda, G. J. McLachlan, A. Ng, B. Liu, P. S. Yu, Z. Zhou, M. Steinbach, D. J. Hand, D. Steinberg, Survey Paper: Top 10 algorithms in Data Mining, Springer-Verlag, 2007.

[9] "IEEE ICDM" erişim adresi: http://www.cs.uvm.edu/~icdm/, erişim tarihi: Ağustos 2012.

[10] B. M. Leiner, V. G. Cerf, D. D. Clark, R. E. Kahn, L. Kleinrock, D. C. Lynch, J. Postel, L. G. Roberts, S. S. Wolff, The Past and Future History of the Internet, Springer-Verlag, 1997.

[11] "ITU ICT Data and Statistics database" erişim adresi: http://www.itu.int/ITU-D/ict/statistics/ , erişim tarihi: Ağustos 2012.

[12] A. S. Tanenbaum, Computer Networks, 4. Ed., *Pearson* 2003.

[13] "IANA" erişim adresi: http://www.iana.org/assignments/port-numbers , erişim tarihi: Ağustos 2012.

[14] T. Karagiannis, A. Broido, M. Faloutsos, K. Claaffy, Transport Layer Identification of P2P Traffic, IMC'04 Proceeding of the 4th SIGCOMM Conference on Internet measurement, 121-134, ACM New York, New York, U.S.A, 2004.

[15] F. Dehghani, N. Movahhedinia, M. R. Khayyambashi, Real-time Traffic Classification Based on Statistical and Payload Content Features, IEEE ISA, 1-4, Wuhan, China, 2010.

[16] A. W. Moore & K. Papagiannaki, Toward the Accurate Identification of Network Applications, SpringerLink, PAM Lecture Notes in Computer Science, Vol. 3431, 41-54, 2005.

[17] J. Erman, A. Mahanti, M. Arlitt, Internet Traffic Identification Using Machine Learning, IEEE GLOBECOM, San Francisco, CA, 1-6, 2006.

[18] L. Yingqiu, L. Wei, L. Yunchun, Network Traffic Classificaion Using K-means Clustering, IEEE IMSCCS, Iowa City, IA, 360-365, 2007.

[19] R. Xu, Survey of Clustering Algortihms, IEEE Transactions on Neural Networks, Vol. 16, 645-678, 2005.

[20] S. S. Shinde & S. P. Abhang, State Of Art Survey Of Network Traffic Classification, IJCA Proceedings on International Conference in Computational Intelligence (ICCIA2012), New York, USA, 2012.

[21] J. Erman, M. Arlitt, A. Mahanti, Traffic Classification Using Clustering Algorithms, MineNet'06 Proceedings of the 2006 SIGCOMM workshop on Mining network data, 281-286, ACM New York, New York, USA, 2006.

[22] T. S. Korting, C4.5 algorithm and Multivariate Desicion Trees, Image Processing Division, National Institute for Space Research (INPE), SP, Brazil.

[23] K. Singh & S. Agrawal, Comparative Analysis of Five Machine Learning Algorithms for IP Traffic Classification, International Conference on Emerging Trends in Networks and Computing Communications (ETNCC), Udaipur, Rajasthan, India, 33-38, 2011.

[24] T. T. T. Nguyen & G. Armitage, A Survey of Techniques for Internet Traffic Classification Using Machine Learning, IEEE Communications Surveys and Tutorials, Vol. 10, Issue 4, 56-76, 2008.

[25] T. Fawcett, An Introduction to ROC Analysis, Pattern Recognation Letters, Vol 27, Issue 8, 861-874, 2006.

[26] T. C. W. Landgrebe, P. Paclik, R. P. W. Duin, A. P. Bradley, Precision-Recall Operating Characteristic (P-ROC) Curves in imprecise enviroments, 18th

International Conference on Pattern Recognition (ICPR 2006), Vol 4., 123-127, Hong Kong, 2006.

[27] M. Halkidi, Y. Batistakis, M. Vazirgiannis, Cluster Validity Methods : Part I, ACM SIGMOD Record, Vol. 31, Issue 2, 40-45, ACM New York, USA, 2002.

[28] M. Halkidi, Y. Batistakis, M. Vazirgiannis, Cluster Validity Methods : Part II, ACM SIGMOD Record, Vol. 31, Issue 3, 19-27, ACM New York, USA, 2002.

[29] A. Callado, C. Kamienski, G. Szabo, B. P. Gero, J. Kelner, S. Fernandes, D. Sadok, A Survey on Internet Identification, IEEE Communications Survey and Tutorials, Vol. 11, Issue 3, 37-52, 2009.

[30] H. Kim, M. Fomenkov, K. Claffy, N. Brownlee, D. Barnan, M. Faloutsos, Comparison of Internet Traffic Classification Tools, In Workshop on Application Classification and Identification, Boston, MA, USA, 2007.

[31] K. Xu, F. Wang, L. Giu, Network-Aware Behavior Clustering of Internet End Hosts, Proceedings IEEE INFOCOM, 2078-2086, Shanghai, China, 2011.

[32] M. Roughan, S. Sen, O. Spatscheck, N. Duffield, Class-of-Service Mapping for QoS: A Statistical Signature-based Approach to IP Traffic Classification, IMC'04 Proceeding of the 4th SIGCOMM Conference on Internet measurement, 135-148, ACM New York, New York, USA, 2004.

[33] X. Zang, A. Tangpong, G. Kesidis, D. J. Miller, Botnet Detection Through Fine Flow Classification, *The Pennsylvania State University*, CSE Dept. Technical Report No. CSE11-001, 2011.

[34] I. Ismail, M. N. Marsono, S. M. Nor, Detecting Worms Using Data Mining Techniques: Learning in Presence of Class Noise, 6th International Conference on Signal-Image Technology and Internet Based Systems, 187-194, Kuala Lumpur, Malaysia, 2010.

[35] "CAIDA" erişim adresi: http://www.caida.org/research/trafficanalysis/tcpudpratio/ , erişim tarihi: Ağustos 2012.

[36] J. Cai, Z. Zhang, X. Song, An Analysis of UDP Traffic Classification, 12th IEEE International Conference on Commnication Technology, 116-119, Nanjing, China, 2010.

[37] A. Moore, D. Zuev, M. Crogan, Discriminators for use in flow-based classification, Queen Mary University of London, CSE Dept., Technical Report No. RR-05-13, 2005.

[38] T. Karagiannis, K. Papagiannaki, M. Faloutsos, BLINC: Multilevel Traffic Classification in the Dark, in Proceeding of the Special Interest Group on Data Communication Conference, Philadelphia, USA, 2005.

[39] L. Bernaille, R. Teixeira, I. Akodkenou, A. Soule, K. Salamatian, Traffic Classification On The Fly, ACM Special Interest Group on Data Communication Computer Communication Review, Vol. 36, Issue 2, 2006.

[40] L. Parsons, E. Haque, H. Liu, Subspace Clustering for High Dimensional Data: A Review, ACM SIGKDD Explorations Newsletter – Special issue on learning form imbalanced datasets, Vol. 6, Issue 1, 90-105, ACM New York, New York, USA, 2004.

[41] M. A. Hall, G. Holmes, Benchmarking Attribute Selection Techniques for Discrete Class Data Mining, IEEE Transactions on Knowledge and Data Engineering, Vol. 15, Issue 6, 1437-1447, 2003.

[42] N. Williams, S. Zender, G. Armitage, Evaluating Machine Learning Algorithms for Automated Network Application Identification, Centre for Advanced Internet Architectures (CAIA), Technical Report 060410B, 2006.

[43] N. Williams, S. Zender, G. Armitage, A Preliminay Performance Comparison of Five Machine Learning Algorithms for Practical IP Traffic Flow Classification, ACM SIGCOMM Computer Communication Review, Vol. 36, Issue 5, 5-16, ACM New York, New York, USA, 2006.

[44] M. A. Hall, Correlation-based Feature Selection for Machine Learning, PhD Thesis, *University of Waikato*, Dept. of Computer Science, Hamilton, New Zealand, 1998.

[45] M. Dash, H. Lau, Consistency-based search in feature selection, Elsevier Artifical Intelligence 151, 155-176, 2003.

[46] M. Goebel, L. Gruenwald, A Survey of Data Mining and Knowledge Discovery Software Tools, ACM SIGKDD Explorations Newsletter, Vol. 1, Issue 1, 20-33, ACM New York, New York, USA, 1999.

[47] "NLANR" erişim adresi: http://www.caida.org/projects/nlanr/ , erişim tarihi: Ağustos 2012.

[48] "WAND Group Internet Traffic Storage" erişim adresi: http://wand.net.nz/wits/ , erişim tarihi: Ağustos 2012.

[49] "NetMate" erişim adresi: http://sourceforge.net/projects/netmate-meter/ , erişim tarihi: Ağustos 2012.

[50] N. Brownlee, NeTraMet & NeMac Reference Manual, University of Auckland, 1999.

[51] "WEKA" erişim adresi:
http://www.cs.waikato.ac.nz/ml/weka/ , erişim tarihi: Ağustos 2012.

[52] T. Karagiannis, A. Broido, N. Brownlee, K. Claffy, M. Faloutsos, Is P2P dying or just hiding?, in IEEE Proceeding of Globecom, 2004.

[53] C. Colman, What to do about P2P?, Network Computer Magazine, Vol. 12, No. 6, 2003.

[54] S. Zander, T. Nguyen, G. Armitage, Automated Traffic Classification and Application Identification using Machine Learning, Proceeding of the IEEE Conference on Local Computer Networks 30th Anniversary, 250-257, Sydney, NSW, 2005.

[55] S. Agrawal, K. Singh, Performance Evaluation of Five Machine Learning Algorithms and Three Feature Selection Algorithms for IP Traffic Classification, IJCA Special Issue on Evolution in Networks and Computer Communications, No. 1, 25-32, 2011.

[56] S. Agrawal, K. Singh, Feature Exraction based IP Traffic Classification using Machine Learning, Proceeding of the International Conference on Advances in Computing and Artifical Intelligence, 208-212, ACM New York, NY, USA, 2011.

[57] "Libpcap" erişim adresi:
http://www.tcpdump.org/ , erişim tarihi: Ağustos 2012.

ÖZGEÇMİŞ

Kişisel Bilgiler

Soyadı, Adı	: YİĞİDİM, Hüseyin Ahmet
Uyruğu	: T.C.
Doğum Tarihi ve Yeri	: 23.12.1984, Ankara
Telefon	: 0 (312) 292 5537
Faks	: 0 (312) 292 5555
E-Posta	: hayigidim@hotmail.com

Eğitim

Derece	**Eğitim Birimi**	**Mezuniyet Tarihi**
Y. Lisans	TOBB ETÜ Bilgisayar Mühendisliği	2012 (Beklenen)
Lisans	Selçuk Üniversitesi Matematik	2008

İş Deneyimi	**Yer**	**Görev**
2010 -	Türkiye Ekonomi Politikaları Araştırma Vakfı (TEPAV)	Bilişim Uzmanı

Yabancı Dil

İngilizce (İleri Seviye), Fransızca (Başlangıç), Almanca (Başlangıç)