KURUMSAL BİLGİSAYAR AĞLARINDAKİ TRAFİK BİLGİSİNİN AKILLI SİSTEMLER İLE SINIFLANDIRILMASI

Fatih ERTAM

Doktora Tezi
Yazılım Mühendisliği Anabilim Dalı
Danışman: Prof. Dr. Engin AVCI
HAZİRAN-2016

T.C
FIRAT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

KURUMSAL BİLGİSAYAR AĞLARINDAKİ TRAFİK BİLGİSİNİN AKILLI SİSTEMLER İLE SINIFLANDIRILMASI

DOKTORA TEZİ

Fatih ERTAM

(121137201)

Tezin Enstitüye Verildiği Tarih: 20 Nisan 2016
Tezin Savunulduğu Tarih: 03 Haziran 2016

Tez Danışmanı:	**Prof. Dr. Engin AVCI (F.Ü)**
Diğer Jüri Üyeleri:	**Prof. Dr. Mehmet Kemal LEBLEBİCİOĞLU (O.D.T.Ü)**
	Prof. Dr. İbrahim SOĞUKPINAR (G.T.Ü)
	Doç. Dr. Erkan TANYILDIZI (F.Ü)
	Doç. Dr. Arif GÜLTEN (F.Ü)

HAZİRAN-2016

ÖNSÖZ

Bu doktora tezi, Fırat Üniversitesi Fen Bilimleri Enstitüsü Yazılım Mühendisliği Anabilim Dalında hazırlanmıştır.

Tez çalışmasının tamamlanmasında her türlü desteğini gördüğüm başta tez danışmanım Sayın Prof. Dr. Engin AVCI olmak üzere, tez izleme komitesi üyeleri Sayın Doç.Dr. Erkan TANYILDIZI ve Sayın Doç.Dr. Arif GÜLTEN'e teşekkürlerimi sunarım.

Tez çalışmasının gerektirdiği uzun çalışma zamanlarından dolayı zaman zaman ihmal etmek zorunda kaldığım sevgili aileme, göstermiş oldukları anlayıştan dolayı ayrıca teşekkür ederim.

Fatih ERTAM

ELAZIĞ–2016

İÇİNDEKİLER

Sayfa No

ÖZET

İnternet kullanımının daha verimli hale getirilmesi amacıyla, internet üzerindeki verilerin sınıflandırılması özellikle kurumsal ağları yöneten ağ yöneticileri açısından önemli bir yer tutmaktadır. Son zamanlarda internet trafiğinin sınıflandırılmasına yönelik çalışmalar artmıştır. Bu çalışmalar ile ağ üzerinde servis kalitesinin artırılması, ağın verimli kullanılabilmesi, hizmet paketlerinin oluşturulması ve kullanıcılara sunulabilmesi amaçlanmaktadır. İnternet trafiğinin sınıflandırılması amacıyla kullanılan ilk sınıflandırma yöntemi port numaralarının kullanılmasına yönelik sınıflandırmadır. Bu sınıflandırma yöntemi, internetin ilk kullanım zamanları için etkili ve hızlı bir sınıflandırma yöntemi olmasına rağmen günümüzde geçerliğini yitirmiştir. Ağ trafiğinin sınıflandırılması için kullanılan bir diğer sınıflandırma yöntemi ise yük tabanlı sınıflandırma veya derin paket analizi olarak adlandırılmaktadır. Bu yaklaşım ağda akan paketler üzerindeki imzaları tanımaya çalışarak sınıflandırma yapma prensibine dayalıdır. Günümüzde yaygın olarak kullanılan ve bu tez çalışması için de seçilen bir diğer internet trafiğinin sınıflandırılması yöntemi ise makine öğrenmesi tabanlı yaklaşımlardır. Ağ üzerindeki akış bilgisinin toplanarak daha çok istatistiki yöntem ve algoritmaların kullanılması ile sınıflandırma yapılmasına dayanır. Tez çalışmasında, internet trafiğinin sınıflandırılması için daha önce hemen hemen hiç kullanılmamış olan Uç Öğrenme Makineleri (UÖM) yöntemleri kullanılmıştır. UÖM'nin başarımını karşılaştırabilmek amacıyla daha önce kullanılan makine öğrenmesi algoritmalarından Destek Vektör Makineleri (DVM), Naive Bayes (NB) ve Yapay Sinir Ağları (YSA) da veri setlerine uygulanarak karşılaştırılmıştır. UÖM algoritmaları ile yapılan sınıflandırma da diğer makine öğrenmesi algoritmalarına göre daha hızlı ve daha yüksek oranda başarım sağlandığı gözlenmiştir. Tez verilerine hem klasik hem de Çekirdek Tabanlı UÖM (ÇTUÖM) yaklaşımları uygulanarak sınıflandırma yapılmıştır. Sınıflandırıcıların her bir sınıfı için Alıcı İşlem Karakteristik (AİK) eğrileri oluşturulmuştur. Özellikle dalgacık fonksiyonunun kullanıldığı ÇTUÖM algoritmasında kullanılan parametrelerin iyi derecede seçimi için Genetik Algoritma (GA) tabanlı bir yazılım (GA-DF-ÇTUÖM) geliştirilmiştir.

Anahtar Kelimeler: İnternet Trafik Sınıflandırma, Makine Öğrenmesi, Uç Öğrenme Makineleri, Genetik Algoritma, Ağ Güvenliği.

SUMMARY

Intelligents Systems with Classification of Traffic Information in Corporate Computer Networks

The classification of data on the internet in order to make internet use more efficient has an important place especially for network administrators managing corporate networks. Studies for the classification of internet traffic have increased recently. By these studies, it is aimed to increase the quality of service on the network, use the network efficiently, create the service packages and offer them to the users. The first classification method used for the classification of the internet traffic was the classification for the use of port numbers. This classification method has already lost its validity although it was an effective and quick method of classification for the first usage times of the internet. Another classification method used for the classification of network traffic is called as payload-based classification or deep packet inspection. This approach is based on the principle of classification by identifying signatures on packets flowing on the network. Another method of classification of the internet traffic which is commonly used in our day and has been also selected for this study is the extreme learning machine based approaches. It is based on classifying by the use of more statistical methods and algorithms collecting the flow information on the network. For the classification of the internet traffic, extreme learning machines (ELM), which were hardly ever used in previous studies, were used. In order to compare the performance of ELM, support vector machines (SVM), Naive Bayes (NB) and artificial neural networks (ANN) from learning machine algorithms used before were also compared by applying to data set. It was observed that a faster and higher level performance was achieved in the classification made with ELM algorithms compared to other learning machine algorithms. Classification was made by applying both classical and kernel based ELM (KELM) approaches to data. The receiver operating characteristic (ROC) curves were created for each class of the classifiers. In particular, wavelet function which uses the KELM algorithm used parameters for the selection of a good degree of genetic algorithm (GA) based software (GA-WF-KELM) was developed.

Keywords: Internet Traffic Classification, Machine Learning, Extreme Learning Machine, Genetic Algorithm, Network Security.

ŞEKİLLER LİSTESİ

Sayfa No

TABLOLAR LİSTESİ

Sayfa No

KISALTMALAR LİSTESİ

CAT6 : CATegory 6
DNS : Domain Name Service
DVM : Destek Vektör Makinesi
FTP : File Transfer Protocol
GA : Genetik Algoritma
HTTP : Hyper Text Transfer Protocol
IANA : Internet Assigned Number Authority
IMAP : Internet Message Access Protocol
IP : Internet Protocol
NB : Naive Bayes
NTP : Network Time Protocol
P2P : Peer to Peer
POP3 : Post Office Protocol 3
RTT : Round Trip Time
SMTP : Simple Mail Transfer Protocol
SNMP : Simple Network Manegement Protocol
SSH : Secure SHell
TCP : Transmission Control Protocol
UDP : User Datagram Protocol
UÖM : Uç Öğrenme Makinesi
VoIP : Voice Over Internet Protocol
YSA : Yapay Sinir Ağları

1. GİRİŞ

Son zamanlarda internet, iş ve sosyal hayatımızın merkezine yerleşmeye başlamıştır. İnternetin yaygın kullanımı ile birlikte internet üzerinden akan bilgi de oldukça yüksek boyutlara ulaşmıştır. İnternet kullanımı hayatımızda önemli bir rol oynamasına karşın internet üzerinden geçen bilginin ne olduğunun tam olarak bilinmemesi özellikle kurumsal ağ yöneticileri için büyük bir problem oluşturmaktadır. Kontrolsüz gücün güç olmadığı kabul edilirse kurumsal ağlarda bu kadar büyük verilerin analiz edilmesi oldukça önemli hale gelmektedir. Yeni ağ mimarileri ile birlikte ağ protokollerinin ve uygulamalarının gelişmesi, internet üzerindeki bilginin analizini daha zor bir hale getirmiştir. Bu alanda yapılan çalışmalar internet veya ağ sınıflandırma kavramlarının ortaya çıkmasına sebep olmuştur.

Ağ trafiğinin sınıflandırılması; sistem ve ağ yöneticilerinin merakını tatmin etmenin yanında, birçok önemli uygulama hakkında bilgi sahibi olmalarına imkân sağlamaktadır. Bu analizler sonucunda ağın hizmet kalitesinin artırılması, yasal izleme yapılabilmesi, sızma girişimlerinin tespit edilmesi, güvenlik iyileştirmelerinin yapılması ve en önemlisi ağın verimli bir şekilde kullanılabilmesi sağlanabilecektir. İnternet servis sağlayıcı hizmetini sağlayan firmalar için de ağ trafiğinin analiz edilecek olması ağın hizmet kalitesi için uygulamaların ihtiyaç duyacağı politikaların belirlenmesi açısından önemli olacaktır.

1.1. Tezin Amacı

Bu tez çalışması ile ulaşılmak istenilen sonuçlara göre 6 temel amaç ortaya konulmaktadır. Bunlar:

- İnternet servis sağlayıcı olarak hizmet veren kurumların hizmet kalitesini gerçekleştirebilmeleri için politikalar üretebilmelerini sağlamak,
- İnternet kullanıcılarının eğilimlerinin tespit edilebilmesini sağlamak,
- Ağ yöneticilerinin, ağ kullanıcılarına özel bant genişliği ayarlayabilmesini, trafik şekillendirme yapabilmesini sağlamak,
- Ağ trafiğinin sınıflandırılabilmesi için daha önce hemen hemen hiç kullanılmamış makine öğrenmesi yöntemleri kullanılarak, bu algoritmaların başarımlarını ortaya koymak,

- GA teknikleri ile en uygun parametrelerin seçilerek makine öğrenmesi yöntemlerinin daha iyi sonuç verebilmesini sağlamak,
- Kurumsal bir üniversite kampüs alanında bulunan bir ağ sistemi üzerinden geçen verilerin alınarak sınıflandırma yapılabilmesini sağlamaktır.

1.2. Literatür Özeti

Ağ trafiğinin sınıflandırılması ile yapılan ilk çalışmalar, Internet Assigned Numbers Authority (IANA) tarafından kayıt altına alınmış, bilinen port numaralarının temel alındığı sınıflandırma yaklaşımı olarak karşımıza çıkmaktadır [1]. Bu çözüm, bilinen uygulamalar için kullanım kolaylığı olmasından dolayı oldukça verimli olmuştur. Bant genişliğini yüksek oranda tüketen Peer to Peer (P2P) uygulamalarının yaygın bir şekilde kullanımı ile beraber ağ yöneticilerinin varsayılan port numaralarını engellemeye başlaması ile bu sınıflandırma yöntemi verimliliğini kaybetmeye başlamıştır. Engellenen yazılımların farklı dinamik portları kullanmaya başlaması, bu yöntemin günümüzde geçerliliğini yitirmesine sebep olmuştur [2-4].

Port temelli sınıflandırmanın etkisinin hemen hemen bitmesi, bir tepki sonucu olarak araştırmacıların, ağ üzerinde akan paketlerin içerisindeki yükleri inceleyerek sınıflandırma yapılabilmesi ile ilgili çalışmalara ağırlık vermelerini sağlamıştır [3-9]. Derin paket analizi olarak da adlandırılan bu yöntem paket yükleri içindeki karakteristik imza veya desenlerin aranarak eşleştirilmesi yaklaşımına dayanmaktadır. Bu yöntem imzaların bilinmesi durumunda çok yüksek doğruluk oranında çalışmaktadır. Derin paket analizi yapabilen uygulamaların maliyetlerinin yüksek oluşu, güçlü donanımlar üzerinde çalışabilmesi ve belki de en önemlisi şifreli paketlerin içeriğinin tespit edilebilmesinin zorluğu bu yöntemin tercih edilmesinde karşılaşılan en önemli sorunlar olarak ortaya çıkmaktadır [10].

Makine öğrenmesi yöntemleri ile yapılan sınıflandırma, port ve paket yükü temelli sınıflandırma yöntemlerinde karşılaşılan sınırları ve zorlukları ortadan kaldırabilecek bir yaklaşım olarak karşımıza çıkmaktadır [11-29]. Nguyen ve arkadaşları [30], yaptıkları çalışma ile daha önce yapılan makine öğrenmesi tabanlı sınıflandırma çalışmalarını araştırmışlardır. Genellikle birçok makine öğrenmesi yöntemi, çevrimdışı olarak kayıt altına alınan paketleri; içerisindeki port numaraları, akış büyüklükleri gibi öz niteliklerden faydalanarak sınıflandırmaya çalışmaktadır. Bu özellikler daha sonra çevrimiçi ağ trafiğini tanımlamak için kullanılan bir model oluşturmak için kullanılmaktadır.

Son yıllarda makine öğrenmesi yöntemleri ile yapılan sınıflandırma yaklaşımlarında umut verici bir şekilde yüksek doğruluk oranlarına yaklaşıldığı görülmektedir [29-32]. Ağ trafiğinin sınıflandırılması amacıyla öncelikle alınan veriler içerisindeki öznitelikler belirlenerek sınıflandırma modelleri uygulanmaktadır. Moore ve arkadaşları [15] öznitelik sayısının en fazla seçilerek sunulduğu çalışmalardan birisini yapmışlardır. Bu çalışmada araştırmacılar paket büyüklüğü, geçen zaman vb. gibi 248 adet özniteliği bir kayıt dosyası içerisinden ayrıştırarak sınıflandırma yapmışlardır. İstatistiksel bilgilerin kullanılması ile yapılan bu sınıflandırma tekniğinde yüksek oranda başarım sağlandığı görülmektedir. Ayrıca araştırmacıların bu çalışmada geliştirdikleri açık kaynak kod tabanlı yazılımdan bu tez çalışmasında da faydalanılmıştır. Gerçek zamanlı ağ sınıflandırması yapılmak istendiğinde bu şekilde tüm verinin özelliklerinin çıkarılması tekniğinin kullanılması zor olmaktadır. Erken aşamada internet trafiğinin doğru bir şekilde sınıflandırılmasının önemli olması durumunda geçerli olacak bazı çalışmalar yapılmıştır [33, 34]. Qu ve arkadaşları [34] yapmış oldukları çalışma ile erken evrede ağ trafiğinin sınıflandırılmasının mümkün olduğunu göstermişlerdir. Erken evredeki bazı paketlere bakılarak sınıflandırma yapılabilmesinin zorluğuna karşın, Dainotti [33] erken evrede özellikleri ayıklamak için kullanılan paketlerin sayısını sınırlayan alt özellik çıkarımları yaparak sınıflandırma için kullanılacak birçok avantaj ortaya koymuştur. Hangi özelliklerin seçilerek daha verimli bir şekilde sınıflandırma yapılabileceği erken evrede yapılacak sınıflandırma için bir engel olarak ortaya çıkmaktadır. Bernaille ve arkadaşları [20] erken evre trafik tanımlama tekniğini sunmuşlardır. Yapmış oldukları çalışmada her bir TCP paketi içindeki ilk birkaç veri paketinin boyutunu kullanmışlardır. K-means kümeleme tekniğini uygulayarak yüksek oranda doğruluk oranı ile sınıflandırma yapma başarısı göstermişlerdir. Bir internet akışının erken evre paketlerinin trafik sınıflandırma için yeterli bilgi taşıdığını Este ve arkadaşları [35] yaptıkları çalışma ile öne sürmüşlerdir. Yaptıkları çalışmada gidiş dönüş süresi, paket boyutu, paket yönü gibi bilgileri analiz ederek erken evre için en etkili özelliklerin çıkarılmasını sağlamışlardır. Erken evre sınıflandırma için Huang ve arkadaşları [36] uygulama özelliklerini inceleyerek etkili bir sınıflandırma için kullanmışlardır. Erken evre için ilk on paketin büyüklüğü ve geçen süre temel alınarak yapılan sınıflandırma uygulamaları yüksek donanım gücü istemektedir [37]. Hullar ve arkadaşları [38] düşük hafıza kaynakları kullanılarak P2P trafiğinin erken evrede sınıflandırılması üzerine çalışmışlardır. Dainotti ve arkadaşları [33] erken evre için yüksek verimli melez bir sınıflandırma yöntemi uygulamışlardır. Nguyen ve arkadaşları [39] Voice over Internet

Protocol (VoIP) trafiğinin zamanında tanımlanması için alt akışlardan elde edilen istatistiksel özellikleri kullanmışlardır. Rizzi ve arkadaşları [40] erken dönem trafik tanımlaması için son derece verimli bulanık bir sistemi önermişlerdir.

Makine öğrenmesi yöntemlerinin tek başına kullanılması yerine yük tabanlı yaklaşımlar ile beraber kullanılarak sınıflandırma yapılması da bir diğer alternatif sınıflandırma önerisi olarak karşımıza çıkmaktadır [41-43]. Karaginnis ve arkadaşlarının [41] yapmış oldukları çalışma bu alanda bilinen en iyi çalışma olarak gösterilebilir. Bu yöntem sunucuda üretilen trafiğin davranışı üzerindeki sınıflandırmayı temel almaktadır. Bir diğer benzer çözüm ise bilinen port numaralarının kullanıldığı IP tabanlı sınıflandırma tekniğidir [44, 45]. Bu teknik bilinen uygulamalar için daha önceden kullanılan IP adreslerinden elde edilen bilgiyi kullanmaya dayanır.

Mevcut makine öğrenmesi yöntemleri kullanılarak çok yüksek oranlarda doğruluk değerleri yakalanabilmesi mümkün olmasına rağmen halen daha tüm ağ senaryosu için kullanılabilecek hazır bir yöntem bulunmamaktadır. Kurumsal ağ üzerindeki kullanıcı ve bağlantı sayısının çok yüksek oranlarda olması sınıflandırma yöntemlerinin pratik bir şekilde kullanılamamasına sebep olmaktadır. Bu tez çalışmasının ana önceliklerinden birisi ağ yöneticilerinin yapacağı sınıflandırma için ortaya çıkabilecek hız ve verimliliğin mümkün olan en yüksek seviyelere çıkarılabilmesidir. Bu amaçla geliştirilen ÇTUÖM yaklaşımı için kullanılan parametrelerin seçiminde GA tekniklerinden faydalanılmıştır.

1.3. Tezin Organizasyonu

Bu tez çalışması 7 ana bölümden oluşmaktadır. Birinci bölümde tezin hazırlanması için hangi sebeplerin etkili olduğundan, tezin hazırlandıktan sonraki kullanım amaçlarından, tez çalışmasının uygulama alanında daha önce yapılan akademik çalışmaların neler olduğundan bahsedildiği literatür özeti bulunmaktadır.

İkinci bölümde ağ trafiğinin sınıflandırılması ile ilgili yapılan çalışmalarda kullanılan yaklaşımlardan olan port tabanlı, yük tabanlı, sunucu davranışı tabanlı, IP tabanlı ve akış özelliklerini temel alan yaklaşımlardan bahsedilerek, tez çalışmasında kullanılmış olan makine öğrenmesi tabanlı yaklaşımların diğer yaklaşımlara göre avantajları vurgulanmıştır.

Üçüncü bölümde tez çalışması için benimsenen internet trafiğinin sınıflandırılması için kullanılan makine öğrenmesi kavramından bahsedilmektedir. Bu bölümde tez çalışmasında kullanılmış olan YSA, DVM, NB ve UÖM algoritmalarının çalışma prensiplerinden bahsedilmiştir. Kullanılan makine öğrenmesi yaklaşımlarının

karşılaştırılabilmesi amacıyla seçilen performans metriklerinden bahsedilmiştir. Grafiksel olarak karşılaştırma yapılabilmesi amacıyla alıcı işlem karakteristik eğrilerinin nasıl oluşturulduğuna değinilmiştir.

Dördüncü bölümde bu tez için en başarılı sınıflandırma sonuçlarının alındığı makine öğrenmesi yöntemlerinden UÖM ile ilgili ayrıntılı bilgi verilmiştir. Ayrıca bu bölümde tez çalışmasında kullanılmış olan farklı uç öğrenme algoritmaları ile ilgili bilgi verilmiş ve tez için kullanım alanlarından bahsedilmiştir.

Beşinci bölümde GA ile ilgili temel bilgilerden bahsedilmiştir. Tez çalışmasında kullanılan makine öğrenme yöntemlerinden birisi olan ÇTUÖM'de kullanılan parametrelerin seçiminde kullanılan GA yaklaşımının hız ve verimlilik açısından avantajları ortaya konulmuştur.

Altıncı bölümde bu tez çalışması için yapılan deneysel çalışma ve uygulamalar sonucunda elde edilen bulgular ayrıntılı bir şekilde ortaya konulmuştur. Verilerin hazırlanması ve kullanılabilir hale getirilebilmesi için yapılan işlemler sıralanmıştır. Kullanılan performans metriklerinden doğruluk değeri, belirleyicilik, kesinlik, duyarlılık ve F-ölçüsü her bir makine öğrenmesi yöntemi için bulunarak tez çalışmasının başarımı ortaya konulmuştur. Ayrıca her bir makine öğrenmesi yöntemi ile elde edilen sınıflandırma başarımına ait AİK eğrileri oluşturulmuştur. Sınıflandırıcıların, sınıflandırma yapmak için geçirdikleri süreler kayıt altına alınarak çalışma zamanları için de karşılaştırma yapılmıştır.

Son olarak yedinci bölümde tez çalışması ile elde edilen sonuçların akademik dünyaya olan katkıları ve ileriye yönelik olarak yapılacak olan iyileştirmeler ile internet dünyasına ve bu alanda çalışan servis sağlayıcı ve sistem yöneticilerine olan katkıları irdelenmiştir.

2. AĞ TRAFİĞİNİN SINIFLANDIRILMASI

İnternetin yaygın kullanımı ile beraber kurumsal ağlarda ortaya çıkan büyük verileri incelemek için geliştirilen ağ trafiğinin sınıflandırması ile ilgili yapılan yaklaşımlar içerisinde hangi yöntemin en iyisi olduğuna dair araştırmacılar içerisinde kesin bir tutarlılık bulunmamaktadır [30, 32, 46]. Özellikle herkesin erişebileceği açık bir veri tabanı olmaması, araştırmacıların farklı veriler üzerinde farklı uygulamalar ile çalışmaları ve bu çalışmalar için hazırladıkları kodları paylaşmamaları trafik sınıflandırma yöntemlerinin karşılaştırılmasında zorluk çıkarmaktadır. Kurumsal ağlardaki trafik bilgisinin sınıflandırılması ile ağ verileri üzerinden kullanıcı analizlerinin yapılması, ağ kaynaklarının yönetilmesi, planlanması ve anormallik tespitinin yapılabilmesi sağlanmaktadır [16, 47]. Ağ trafiğinin sınıflandırılması için kullanılan genel yöntemler şu şekilde sıralanabilir:

- Port tabanlı yaklaşım,
- Yük tabanlı yaklaşım,
- Sunucu davranışı tabanlı yaklaşım,
- IP tabanlı yaklaşım,
- Akış özelliklerine dayalı yaklaşım,

Alt başlıkları altında toplanmıştır.

2.1. Port Tabanlı Yaklaşım

İnternetin ilk kullanıldığı zamanlarda uygulamalara ait bilinen port numaralarının temel alınması ile gerçekleştirilen sınıflandırma yaklaşımıdır. Kullanıldığı zaman içerisinde çok kesin ve iyi sonuçlar vermiştir. IANA [1] tarafından atanan port numaraları 0 ile 65535 arasındaki numaralardır. Bu numaralar aşağıdaki gibi üç alt kategoride toplanabilir:

- İyi bilinen portlar (0-1023)
- Kayıtlı portlar (1024-49151)
- Özel portlar (49152-65535)

İyi bilinen portlar genel olarak bilinen uygulamaların kullandığı ve sistem yöneticilerinin de bildiği port numaralarıdır. Kayıtlı portlar yönetici izinleri gerektirmeyen portlardır. Özel portlar ise IANA tarafından kayıt edilmemiş olan portlardır. Tablo 2.1'de IANA tarafından atanmış olan birkaç port numarası ve genellikle bu port numarası üzerinden hizmet veren uygulama gösterilmektedir. Örnek olarak 23 numaralı port ile telnet

uygulamasının kullandığı port numarası, 53 numaralı port ile DNS uygulamasının kullandığı port numarası ifade edilmektedir.

Tablo 1.1. Bilinen bazı uygulamalar için IANA tarafından atanmış olan port numaraları ve uygulamalar

Atanmış port	Uygulama
21	FTP
22	SSH
23	Telnet
25	SMTP
53	DNS
80	HTTP
161	SNMP

Port tabanlı sınıflandırma yaklaşımı kısa bir zaman sonra geçerliliğini yitirmiştir [2-4]. Port tabanlı yapılan sınıflandırma yaklaşımlarını temel alan bazı çalışmalar ilk kullanım zamanlarında yüksek sayılabilecek oranda doğruluk değerini yakalamışlardır [4, 15, 48]. Günümüzde ise bu yaklaşımının temel alındığı çalışmalar ile çok daha düşük oranlarda doğruluk oranına ulaşabileceğimizi göstermektedir. Günümüzde kullanılan uygulamalarda, dinamik port numaralarının çok kullanılması belli bir port numarasına ait uygulamanın tespit edilebilmesini zorlaştırmaktadır. P2P uygulamalarının artması ile birlikte IANA tarafından kayıt altına alınmış olan portlar kullanılmamaya başlanmıştır. Ayrıca ağ yöneticileri gelebilecek saldırılara karşı bir önlem olması amacıyla bilinen port numaraları yerine farklı port numaralarını kullanmayı tercih etmektedir. Örneğin SSH için kullanılan port numarasını 22 yerine 2222 olarak değiştirerek bu bağlantı için bilinen port numarası üzerinden yapılabilecek saldırıların engellenmesi hedeflenmektedir. Bu durum port tabanlı sınıflandırma yaklaşımı için başka bir dezavantaj olarak karşımıza çıkmaktadır. Bu yetersizlikler sebebiyle araştırmacılar yeni sınıflandırma yaklaşımlarını araştırmak zorunda kalmışlardır.

2.2. Yük Tabanlı Yaklaşım

Port tabanlı yaklaşım ile sınırlı bir şekilde yapılabilen sınıflandırma işlemine alternatif olarak ağ trafiği içerisindeki paket yüklerinin analiz edilmesi ile sınıflandırma yapılan yaklaşım ortaya çıkmıştır [1-9, 49, 50]. Bu yaklaşım derin paket analizi olarak da ifade edilmektedir. Bu yaklaşım ile internet trafiği üzerindeki Transmission Control Protocol

(TCP) ve User Datagram Protocol (UDP) paketleri üzerindeki bilgi yükü analiz edilerek sınıflandırma işleminin yapılmasına çalışılmaktadır. Yük analizi için uygulamalara ait karakteristik imzaların olup olmadığının belirlenmesi gerekmektedir [4]. Karaginnis ve arkadaşlarının [5] P2P protokollerini anlayabilmek amacıyla yapmış oldukları çalışma için kullandıkları desen veya imza dizilerinden bazıları Tablo 2.2'de gösterilmektedir.

Tablo 2.2. P2P uygulaması için yapılan bir çalışmada kullanılan imza ifadeleri

P2P Protokolü	İmza
eDonkey 2000	0xe3190100000
eDonkey 2000	0xe53f0100000
Bittorrent	"0x13Bit"
Gnutella	"GNUT" "GIV"

Port tabanlı yaklaşım ile beraber derin paket analizi yöntemlerinin kullanıldığı bazı melez yaklaşımlarda araştırmacılar tarafından sunulmuştur [2, 3, 5]. Buna rağmen desen eşleştirme tekniği ile çalışan bu yaklaşımın birkaç dezavantajı bu yöntemin çok fazla tercih edilmemesine sebep oluşturmaktadır. Bunlardan ilki her paket üzerinde imza araması yapabilmek ve eşleştirebilmek için bu uygulamaların güçlü ve pahalı donanımlar üzerinde çalıştırılması ihtiyacının olmasıdır. İkinci büyük dezavantaj ise şifreli paketlerin içerisindeki imzaların tanınmasında yaşanan sıkıntılardır. Diğer bir dezavantajda uygulamaların güncellendikçe kullanmış oldukları imzaların değişebilmesi ve bu değişikliklerin her zaman takip edilememesinden kaynaklanan yanlış sınıflandırma ihtimalinin ortaya çıkmasıdır. Bazı çalışmalar [13] bu süreci hızlandırmaya çalışsa bile yeni imzaların oluşturulması uygulamaların çoğalması ile çok fazla zaman kaybına yol açmaktadır. Bu olumsuz yönlerine rağmen günümüzde hala bazı çalışmalarda bu tekniğin kullanıldığı görülmektedir [7-10, 49].

2.3. Sunucu Davranışı Tabanlı Yaklaşım

Yük tabanlı yaklaşımlar ile karşılaşılan sınırları aşabilmek amacıyla bir diğer sınıflandırma yaklaşımı olan sunucu davranışını temel alan yaklaşım ortaya atılmıştır. Karaginnis ve arkadaşları [41, 43] ile Xu ve arkadaşları [42] sunucu davranışına dayalı bir sınıflandırma yaklaşımı geliştirmişlerdir. Karaginnis ve arkadaşları yapmış olduğu çalışma ile sosyal, fonksiyonel ve uygulama olmak üzere üç temel sunucu davranışını temel alarak sınıflandırma yapmaya çalışmışlardır. Bu sınıflandırma tekniği ile %90 civarında doğruluk oranının yakalandığı görülmektedir. Bu başarım oranına rağmen iki yönlü trafik olduğu

durumlar için bu tekniğin kullanılarak sınıflandırma yapılmasında bazı sınırlar olduğu gözlenmiştir. Ayrıca davranışlara göre uygulamaların sınıflandırılmaya çalışıldığı bu teknikte tüm uygulamaların tespit edilebilmesi mümkün olmamaktadır. Farklı gruplar olmasına rağmen benzer davranışları gösteren P2P, VoIP gibi uygulamalar birbirleriyle karıştırılabilmekte ve bunun sonucu olarak yanlış sınıflandırma yapılabilmektedir.

2.4. IP Tabanlı Yaklaşım

IP Adresleri ürettikleri trafik ile çok önemli bilgileri taşıyabilmektedir. Bu yaklaşımın temel çalışma prensibi port tabanlı yaklaşım ile benzerdir. Bu yöntem ile IP adresi üzerindeki bilgi kullanılarak sınıflandırma işlemi yapılmaya çalışılmaktadır. Bu teknik iki şekilde kullanılmaktadır. Bunlardan birincisi bilinen IP adreslerini temel alarak sınıflandırma yapılmasıdır [44]. Bu şekilde Facebook, Twitter gibi popüler internet sitelerine ait IP adreslerinin bilinmesi, bu IP adreslerine yapılacak olan isteğin sınıflandırılabilmesine ve tanınmasına sebep olacaktır. İkinci teknik ise aynı zamanda servis tabanlı bir yöntem olarak bilinmektedir [45]. Bir servis içerisinde IP, port ve protokol bilgisi bulunur. Bu bilgilerin kullanılması ile hangi servisin veya uygulamanın kullanıldığı tespit edilmeye çalışılmaktadır.

2.5. Akış Özelliklerine Dayalı Yaklaşım

Ağ sınıflandırması için daha önce kullanılan yaklaşımlar ile ortaya çıkan dezavantajları kaldırabilmek amacıyla yeni bir alternatif olarak ağ üzerindeki trafiğin alınarak makine öğrenmesi yöntemleriyle analiz edilmesini temel alan bir sınıflandırma yaklaşımı ortaya çıkmıştır. Bu yöntemler ile daha çok çevrimdışı olarak kayıt altına alınan ağ akış verileri analiz edilerek sınıflandırma yapılmaktadır. Günümüzde ağ sınıflandırması için oldukça popüler olan tekniklerden birisi olarak kabul edilmektedir.

Makine öğrenmesini temel alan yaklaşımlar daha çok iki ana alanda yapılan çalışmalardır. Bu alanlar denetimli ve denetimsiz öğrenme olarak birbirlerinden ayrılmaktadır [30]. Denetimli öğrenme algoritmaları için veri madenciliği sınıflandırma analizi yöntemleri kullanılmaktadır. Denetimli öğrenme yapılabilmesi için akışlara ait sınıfların önceden bilinmesi gereklidir [11-18]. Denetimsiz öğrenme ise sınıfların bilinmediği veya bilinmesine ihtiyaç olmadığı durumlarda kümeleme yöntemleri kullanılarak yapılan sınıflandırma işlemidir [19-29].

Makine öğrenmesi yaklaşımlarına ait algoritmalarının kullanılması ile yapılan sınıflandırma iki temel adımda gerçekleştirilir. İlk adımda sınıflandırma için bir model oluşturulur, ikinci adımda ise daha çok istatistiki teknikler ve hesaplamalardan faydalanılarak sınıflandırma için gerekli işlemler uygulanır.

Akış tabanlı sınıflandırma yaklaşımında kullanılan makine öğrenmesi algoritmaları, akışa ait öznitelikleri kullanarak sınıflandırma yapmaya çalışmaktadır. Bu öznitelikler paketlerin iletimi için geçen süre, paketin büyüklüğü, paketin sayısı gibi özelliklerdir. Bu özellikler bazen istemci ile sunucu arasındaki tek yönlü akış bilgisine, bazen de her iki yönde de olan akış bilgisine bakılarak kullanılabilir.

3. MAKİNE ÖĞRENMESİ

Makine öğrenmesi kavramı ilk bakışta donanım ile ilgili olduğu izlenimi vermektedir. Çalışan bir bilgisayar sisteminin yazılım ve donanımdan oluştuğu düşünülse de çoğu zaman yazılımın kendisi makine olarak düşünülebilir [51, 52]. Bu yaklaşım temel alındığında öğrenen makinenin bir donanım olabileceği fakat makinelerin öğrenmesini sağlayan asıl unsurun makinenin ruhu olarak düşünebileceğimiz bilgisayar yazılımları olduğunu düşünebiliriz. Makine öğrenmesi kavramı yerine yapay öğrenme kavramının da kullanıldığı görülmektedir [53]. Makine öğrenmesi etkili ve sıfır hataya sahip tahmin yapabilen algoritmalardan oluşmaktadır [54]. Makine öğrenmesi tecrübe ile otomatik olarak gelişen bilgisayar yazılımlarının nasıl geliştirilebileceği ile ilgilenir [55]. Makine öğrenmesi ile performans geliştirilmesi için örnek veri veya tecrübeler kullanılarak bilgisayar programlanması yapılmaktadır [53]. Makine öğrenmesi yüz tanıma, elektronik posta filtrelemesi yapma, tıbbi teşhis gibi birçok alanda kullanılmaktadır. Kullanım alanlarından birisi de ağ trafiğinin sınıflandırılmasıdır. Araştırmacılar önceki dönemlerde sıklıkla makine öğrenmesi tekniklerini kullanarak ağ üzerinden akan trafiği sınıflandırmaya çalışmışlardır. Makine öğrenmesi algoritmalarının kullanıldığı ağ sınıflandırmasında genellikle denetimli, denetimsiz ve melez olarak kabul edilen yarı-denetimli öğrenme yöntemi tercih edilmiştir [56].

3.1. Denetimsiz Öğrenme

Denetimsiz öğrenme yönteminde verilere ait sınıf bilgileri bulunmaktadır. Kümeleme olarak da bilinen bu yöntem de amaç önceden bilinen sınıflar için bir eğitim yapmak değildir. Veriler üzerinde doğal kümeler keşfedilmeye çalışılır. Bu alanda araştırmacıların yapmış oldukları birçok araştırma mevcuttur [19-29]. McGregor ve arkadaşları [57] yaptıkları çalışmada IP trafik sınıflandırma için beklenti maksimizasyonu algoritmasını kullanmışlardır. Bu çalışma ile farklı uygulama tiplerine benzer gözlem özellikleri ile kümeleme işlemi yapılmıştır. Böylece HTTP, FTP, SMTP, IMAP, NTP ve DNS trafiği üzerinde sınıflandırma yapılmıştır. Zanter ve arkadaşları [28,29] AutoClass yaklaşımı kullanarak kümeleme işlemini yapmışlardır. Bernaille ve arkadaşları [20] basit K-Means algoritması ile trafik akışındaki ilk birkaç paketi kullanarak farklı tipteki uygulamalar için

sınıflandırma yapmışlardır. Erman ve arkadaşları [25] ağ üzerinde P2P trafiğini sınıflandırma amacıyla denetimsiz öğrenme yöntemini kullanmışlardır.

3.2. Yarı Denetimli Öğrenme

Yarı denetimli öğrenmede hem sınıfları belli olan veri hem de sınıfının ne olduğunun belli olmadığı veri öğrenme algoritmasına verilir. Genellikle etiketli olmayan, sınıfı belli olmayan, verinin kolay bir şekilde elde edilebildiği fakat verileri sınıflandırmanın, etiketlemenin, zor olduğu durumlarda kullanılmıştır. Etiketlenmiş verilerin sayısı, etiketlenmemiş olan verilerin sayısından çok daha fazla olması durumunda tercih edilir [58]. Melez bir öğrenme yöntemi olarak da kabul edilebilir. Erman ve arkadaşları [24] denetimli ve denetimsiz öğrenme yöntemini birleştirerek internet trafiğini sınıflandırmışlardır.

3.3. Denetimli Öğrenme

Denetimli öğrenmede verilere ait bir dizi etiketlenmiş, sınıflanmış, veri mevcuttur. Sınıflandırma, veri madenciliğinde kullanılan sınıflandırma analizi teknikleri kullanılarak yapılır. Ağ trafiğinin sınıflandırılmasında denetimli öğrenme yönteminin kullanıldığı birçok çalışma yapılmıştır [11-18]. Roughan ve arkadaşları [16] Doğrusal Diskriminant Analizi algoritmasını kullanarak imza tabanlı bir yaklaşım geliştirmişlerdir. Moore ve arkadaşları [15] Bayes analiz tekniklerini kullanarak sınıflandırma yapmışlardır. Zander ve arkadaşları [59, 60] C4.5, BayesNet; Agrawal ve arkadaşları [61-63] NB gibi denetimli öğrenme temelli makine öğrenmesi algoritmalarını kullanarak ağ trafiğini sınıflandırmışlardır.

3.4. Sınıflandırma

Her türlü girdi verisini sonlu sayıdaki ayrık bir kategoriye atamayı amaçlayan durumlar sınıflandırma problemi olarak ele alınmaktadır [64]. Sınıflandırma problemlerinde çıktı uzayındaki elemanlar sınıf olarak tanımlanır. Sınıflandırma için ortaya çıkan problemi çözmek için hazırlanan algoritmaya ise sınıflandırıcı adı verilmektedir [65]. Makine öğrenmesi yaklaşımları için YSA, DVM, k-en yakın komşu (K-NN), karar ağaçları, NB, UÖM gibi birçok sınıflandırma algoritması kullanılmaktadır. Sınıflandırma işlemini makine öğrenmesinin temel görevlerinden birisi olarak düşünebiliriz. Sınıflandırma, etiketlenmemiş bir verinin bilinen bir grup altına yerleştirilmesi amacıyla kullanılır [66].

3.5. Makine Öğrenmesi Süreçleri

Makine öğrenmesi yöntemlerinin kullanarak bir problemin çözülebilmesi için birkaç aşamadan geçilmesi gerekmektedir. Bu aşamalar genel olarak aşağıdaki adımlardan oluşmaktadır [67]:

- Problemin tanımlanması,
- Verilerin analiz edilmesi,
- Verilerin hazırlanması,
- Modelleme yapılması,
- Modellerin seçimi,
- Uygulama.

Bu tez çalışmasında kullanılan makine öğrenmesi yöntemleri içerisinden hangisinin tercih edilmesi gerektiği kanısına ulaşabilmek için bu sıralamada verilen süreçler değerlendirilmiştir.

3.5.1. Problemin Tanımı

Öğrenme probleminin iyi bir şekilde tanımlanarak bilinmesi, makine öğrenmesi uygulamaları ile yapılacak olan çözümler için yapılan ilk adımdır. İhtiyacın belirlenerek, makine öğrenmesi yöntemleri ile oluşturulacak olan modelin neyi amaçladığı ayrıntılı bir şekilde ortaya çıkarılmalıdır.

Bu tez çalışmasında ağ üzerindeki bilginin makine öğrenmesi yöntemleri ile sınıflandırılması ile amaçlanan faydalar tezin 1.1 bölümünde belirtilen, tezin amacı başlığı altında sıralanmıştır. Bu tez çalışması ile ağ verilerinin nasıl sınıflandırılacağına dair problem çözülmeye çalışılmaktadır.

3.5.2. Verinin Anlaşılması

Problemin tanımının iyi bir şekilde yapılmasının ardından yararlanılacak verinin ne olacağının ve nasıl toplanacağının belirlenmesi gerekmektedir. Veri toplama için üç tür yol haritası çıkarılabilir. Bunlardan birincisi internet üzerinden temin edilebilen, indirilip kullanılabilen verilerdir. Başka bir veri toplama yöntemi ise kişinin uydurma veriler kullanarak bir veri tabanı oluşturarak buradaki verileri kullanmasıdır. Diğer bir veri toplama yöntemi ise internet üzerinde var olan verilerin toplanma yöntemi kullanılarak gerçek ortamda çalışan bir sistem üzerinde yazılımlar aracılığıyla verilerin toplanmasıdır.

Bu tez çalışmasında kullanılacak verilerin toplanması için veri toplama yöntemlerinden ikisi üzerinde durulmuştur. Hem internet üzerinde var olan verilerden faydalanılmış [68, 69] hem de bu verilerin elde ediliş yöntemleri kullanılarak Fırat Üniversitesi ağındaki omurga anahtarlama cihazı üzerinden geçen veriler toplanarak kullanılmıştır. Gerçek verilerin toplanarak, aynı makine öğrenmesi algoritmalarıyla test edilmesi ileride gerçek zamanlı olarak yapılabilecek uygulamalar için bir altyapı oluşturabilecektir.

3.5.3. Verinin Hazırlanması

Veriler analiz edilmeden önce analiz için işlenmesi ve hazır hale getirilmesi gerekmektedir. Bu işlemler verinin özetlenmesi, temizlenmesi, dönüştürülmesi, indirgenmesi, ayrıklaştırılması şeklinde ele alınabilir [70]. Veri hazırlama için yapılacak işlemlerde standartlaşmış bir yöntem bulunmamaktadır, kullanılacak veri setlerine göre değişebilmektedir.

Hazır kullanılan veya sonradan toplanarak alınan veriler içerisindeki tüm sayısal değerler aynı aralıkta bulunmayabilir. Örneğin ağ iletişimi sırasında paketler arasında geçen süre 0 ile 1 arasında değişirken, verilerin byte olarak büyüklükleri 1000 ile 10000 arasında değişebilir. Bu durum niteliklerin almış olduğu değerler arasında bir uyumsuzluğa neden olabilir. Başka bir problem ise büyük değerlerin sınıflandırma sırasında baskın hale gelerek sınıflandırma başarımını olumsuz etkileyebilir. Bu sorunu giderebilmek amacıyla veriler üzerinde normalizasyon veya normalleştirme adı verilen işlem gerçekleştirilebilir [51]. Normalizasyon için farklı yöntemler kullanılabilir.

Bu tez çalışmasında kullanılan verilerin normalizasyonu için Denklem 3.1'de verilen ve aynı zamanda birlik-tabanlı normalizasyon adı olarak da adlandırılan yaklaşım benimsenmiştir.

$$v' = \frac{v - v_{min}}{v_{max} - v_{min}} \tag{3.1}$$

Özellik ölçeklendirme ile bu şekilde kullanılan veriler 0 ile 1 arasında değişen değerlere dönüştürülerek makine öğrenmesi algoritmaları için modellenen sisteme verilmiştir. Böylece bazı verilerin çok yüksek, bazı verilerin çok düşük değerlerde olması sebebiyle oluşabilecek sınıflandırma problemlerinin önüne geçilmesi ve daha doğru sınıflandırma yapılması amaçlanmıştır.

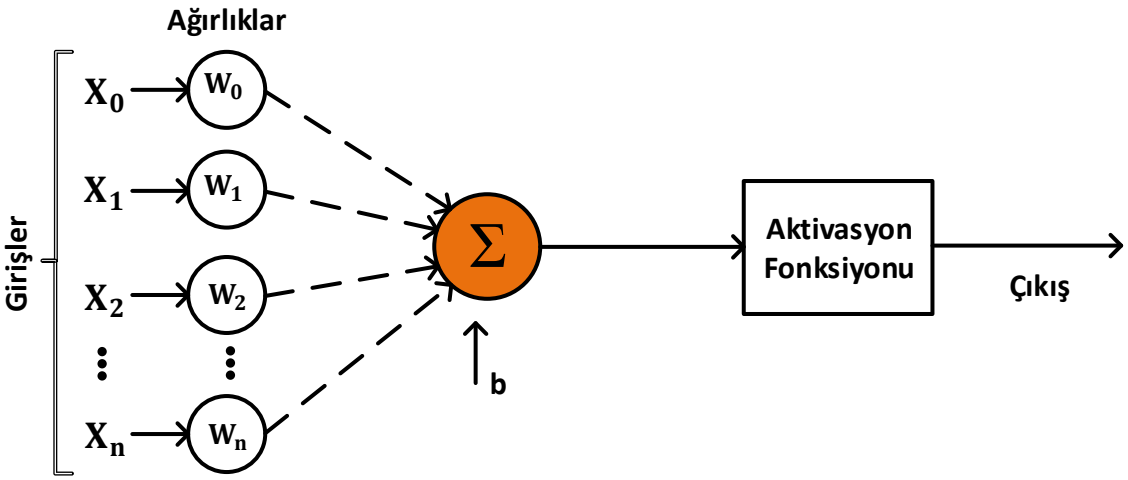
3.5.4. Modelleme

Verilerin elde edilip işlenecek şekilde hazırlanmasının ardından yapılacak olan bir sonraki adım verilerin makine öğrenmesi algoritmaları kullanarak modellenmesidir. Bu model ile veri içindeki örüntü yakalanmaya çalışılır [71]. Model, görevin niteliklerle tanımlanmış olan çıktısına, uygun ihtiyaç duyulan haritalamadır [72].

Bu tez çalışmasında ağ trafiğinin sınıflandırılması amacıyla daha önceki çalışmalarda kullanılan modellerin hepsi yerine başarılı performans gösterenler verilere uygulanarak sınıflandırma yapılmıştır. Bu amaçla daha önceki çalışmalarda başarılı sonuçlar veren NB ve DVM verilere uygulanmıştır. Yine bu tez çalışmasında yeni bir yaklaşım olarak UÖM tez verileri için bir model olarak seçilip daha önce yüksek oranda başarım alınan diğer makine öğrenmesi modelleriyle karşılaştırılmıştır. Temel bir sınıflandırıcı modeli olarak YSA sınıflandırıcı kullanılarak da sınıflandırma işlemi yapılmıştır. Faydalanılan modeller ve çalışma algoritmaları ile ilgili özet bilgi alt bölümlerde sunulmuştur.

3.5.4.1. Yapay Sinir Ağları

İnsan beyninin ileri özelliklerinin göz önüne alınması mantığına dayanan YSA kavramı, beynin olaylar karşısındaki tepkisinin modellenmesi sonucu ortaya çıkmış bir yaklaşımdır [73]. Biyolojik nöron hücrelerinin matematiksel modellerinin çıkartılması ile oluşturulan yapay nöron ilk olarak McCulloch ve Pitss tarafından yapılan çalışmalar ile 1943 yılında başlamıştır [74]. Yapay nöronların genel gösterimi Şekil 3.1'de verilmiştir. Burada $x = (x_0, x_1, \dots, x_n)$ giriş değerleri, $w = (w_0, w_1, \dots, w_n)$ ise ağırlık değerleri olarak kabul edilsin. i. giriş değeri ile i. ağırlık değerinin çarpılması ile $\sum_{i=0}^{n} w_i x_i$ toplamı elde edilmektedir. Bu toplama b eşik değeri eklenerek elde edilen değer, kullanılan aktivasyon fonksiyonundan geçirilerek yapay sinir ağının çıkışı elde edilir [75]. Kullanılan aktivasyon fonksiyonları doğrusal veya doğrusal olmayan fonksiyonlardan seçilebilir. Şekil 3.1'de gösterildiği gibi yapay nöron; girişler, ağırlıklar, toplama fonksiyonu, aktivasyon fonksiyonu ve çıkış olmak üzere 5 elemandan oluşmaktadır [76].



Şekil 3.1. YSA hücresinin genel gösterimi

YSA yapısı temel alındığında, ileri beslemeli YSA ve geri beslemeli YSA olmak üzere iki sınıfta incelenebilir. İleri beslemeli YSA'da hücreler katmanlar şeklinde yer almaktadır. Giriş katmanı tek yönlü bir şekilde çıkışa iletilmekte ve her katmana ait çıkış değeri diğer katmana giriş olarak verilmektedir. Böylece dışarıdan alınan bilgi değişikliğe uğramadan ara katmanda bulunan hücreler üzerinde ilerlemiş olur. Ağ çıkışının belirlenmesi için ara katmanlarda bilgi işlenir. Geri beslemeli YSA'da ise bir hücrenin çıkışı sadece kendisinden sonra gelen katmana çıkış olarak verilmek yerine kendisinden önceki veya kendisi ile aynı katmanda bulunan en az bir hücreye gecikmeli olarak verilir. Böylece geri besleme katmanlar arasında veya aynı katmanda bulunan hücreler arasında olmaktadır.

3.5.4.2. Naive Bayes

NB değişkenler arasında olan ilişkiyi temel alarak analiz eden, tahmine ve tanımlamaya dayalı bir sınıflandırma algoritmasıdır. NB sınıflandırıcı için amaç verinin öğrenilmesidir. Eğitimde kullanılacak olan veriler, modelin öğrenilebilmesi için her bir çıktının kaç kez meydana geleceğini hesaplamaya çalışır. Bu değere öncelikli olasılık değeri adı verilir. Yapılan hesaplamalar esnasında her bir bağımsız değişkenin bağımlı değişkenlere oranının birleşimi olayın meydana gelme sıklığını gösterir. Bu sıklık değeri veri kümesinden yapılacak tahmin için kullanılır [11, 15, 29, 30, 77].

NB sınıflandırıcısı kategorik verilerin sınıflandırılması işlemi için basit ve hızlı bir yöntemdir. İlgili eğitim setinde bulunan her bir bağımsız öznitelik arasındaki ilişkiyi ve her

ilişki içerisindeki şartlı olasılığın ortaya çıkarılması için analiz yapılır. Bağımsız değişkenlerin bağımlı değişkenler üzerindeki etkilerini bir araya getirip yeni bir durumu sınıflandırma yapabilmesi amacıyla tahmin işlemi kullanılmaktadır.

Bayes teoreminden faydalanılarak oluşturulmuş olan NB sınıflandırıcı yöntemiyle bir hedef niteliğin sınıf değerlerine ait olma olasılıkları bulunabilir [70]. NB sınıflandırıcın çalışma prensibi aşağıda özet şeklinde verilmiştir:

$X = (x_1, x_2, \dots, x_n)$ örnekler kümesini, $C = (c_1, c_2, \dots, c_n)$ veri kümesindeki sınıflar olarak kabul edilsin. Buradaki amaç $P(c_i|X)$ değerinin maksimize edilmesidir. Her c_i sınıfı için $P(c_i|X)$ değerinin maksimize edilmesi maksimum sonlu hipotez olarak adlandırılır. Bayes teoreminden Denklem 3.2 elde edilir:

$$P(c_i|X) = \frac{P(c_i)P(X|c_i)}{P(X)} \tag{3.2}$$

Burada $P(X)$ tüm sınıflar için eşit olduğundan dolayı bir sabit olarak kabul edilebilir. Amaç payın maksimize edilmesidir. Fazla sayıda örneğin bulunduğu veri setlerinde $P(X|c_i)$ değerinin hesaplanması zor olacağı için sınıf şartlı bağımsızlık varsayımı oluşacaktır [78]. Bu varsayıma göre örneklerin arasında birbirine bağımlı ilişki bulunmayacaktır. Bu sebeple $P(X|c_i)$ değeri Denklem 3.3'de ki haliyle yazılabilir:

$$P(X|c_i) = \prod_{t=1}^{n} P\,(x_t|c_t) = P(x_1|c_i)P(x_2|c_i) \dots\ P(x_n|c_i) \tag{3.3}$$

Ortaya çıkan $P(X|c_i)$ değerinin hesaplanması, örneğin kategorik veya sayısal olma durumuna göre değişebilir. Örneğin kategorik olması durumunda sınıf değerlerine göre hesaplanan tablodan $P(X|c_i)$ değeri bulunabilir. Örnek değeri x_t olan c_i sınıfına ait gözlemlerin sayısı, c_i sınıfına ait tüm gözlemlerin sayısına bölünmesiyle elde edilebilir. Örneklerin kategorik yerine nümerik olması durumunda niteliğin μ ortalaması ve σ standart sapması ile Gauss dağılımı gösterdiği düşünülerek Denklem 3.4 ve Denklem 3.5'de gösterildiği şekilde hesaplama yapılmaktadır.

$$(x_t|c_i) = g(x_t, \mu_{c_i}, \sigma_{c_i}) \quad \textbf{(3.4)}$$

$$g\left(x_t, \mu_{c_i}, \sigma_{c_i}\right) = \frac{1}{\sigma_{c_i}\sqrt{2\pi}}\, e^{-\frac{1}{2}\left(\frac{x_t - \mu_{c_i}}{\sigma_{c_i}}\right)^2} \quad \textbf{(3.5)}$$

3.5.4.3. Destek Vektör Makineleri

DVM sınıflandırma modelinin yüksek seviyeli genelleştirme özelliğinden dolayı makine öğrenmesi algoritmaları içerisinde oldukça yaygın bir kullanıma sahiptir. Ağ üzerindeki trafik bilgisinin sınıflandırılmasında sıklıkla kullanılmış ve başarılı olmuş bir yöntemdir [79-85].

DVM'de kullanılan algoritma istatistiki verilerin analiz edilmesi ve bu veriler üzerinde örüntü çıkarılmasını sağladığı için sınıflandırma ve regresyon problemlerinde kullanılmaktadır. Kullanılan algoritma temelde veri setinin boyutunda ve daha yüksek boyutlu bir uzayda yer alan nitelikleri en uygun şekilde ikiye ayıran bir düzlem oluşturmaktadır. Sınıflandırma problemlerinde kullanımı yeni örneklerin algoritma tarafından oluşturulmuş olan uzaya aktarılması ve bu uzayda bulunan düzleme göre konumlarına bakılarak sınıflandırma yapmaya çalışması ile gerçekleşir [86]. Bu makine öğrenmesi algoritmasının yaygın kullanımına neden olan avantajlardan en önemlisi deneysel risk düşürmesi yerine yapısal risk düşürmesinin kullanmasıdır. Bu şekilde öğrenme modelinin aşırı öğrenerek ezber yapması engellenmektedir. [87].

DVM daha çok iki sınıflı problemlerin çözümünde kullanılmasına rağmen ikiden fazla sınıflı problemlerin çözümünde de kullanılmaktadır. Bu işlemi gerçekleştirmek için kullanılan genel yaklaşım birden fazla sınıfı ikili sınıflar arasında gruplamaya çalışarak ayrıştırmaktır. Bu ayrıştırma işlemi için kullanılan yaklaşımlar bire karşı hepsi yöntemi, yönlendirilmiş çizge [88] ve çıktı kodlarının hata düzeltmesi şeklindedir [89].

Bire karşı hepsi yönteminde n tane sınıfı barındıran eğitim seti kullanılarak, n tane DVM sınıflandırıcısı oluşturulmaktadır. DVM algoritması tarafından oluşturulan i. model ve i. sınıfa ait olan örneklerin pozitif, diğer örneklerin ise negatif olarak etiketlenmesi ile eğitim sürdürülür. Bu teknik ile sınıflandırma fonksiyonu Denklem 3.6'da gösterildiği şekilde olmaktadır.

$$\begin{matrix}(\vec{w}^1)^T\emptyset(\vec{x})+b^1\\ \vdots\\ (\vec{w}^n)^T\emptyset(\vec{x}\,)+b^n\end{matrix} \quad \textbf{(3.6)}$$

Bu şekilde bir x vektörünün ait olduğu sınıf en yüksek değerin yakalandığı sınıflandırıcı fonksiyonu ile bulunacaktır:

$$Sınıf(\overrightarrow{x}) = arg_{i=1\ldots n}\max((w^i)^T\emptyset(\vec{x})+b^i) \quad \textbf{(3.7)}$$

Çıktı kodlarının hata düzeltmesi yönteminde çok sınıflı olan bir sınıflandırma problemi ikili sınıflandırma gruplarından oluşan kümelere dönüştürülmektedir. Eğitim ve sınıflandırma olmak üzere iki ana parçadan oluşan bu yöntemin ilk aşamasında S ikili sınıflandırıcı eğitim verisinde S ayrım olmak üzere eğitilmektedir. Eğitim setinin N tane sınıftan oluştuğu kabul edilirse kod matrisi ($M \in \{-1, 0, 1\}^{NxS}$) şeklinde sınıflar ve ayrımlar arasındaki ilişkiyi gösterir. Burada $m_{ns} = 1(m_{ns} = -1)$, n sınıfı f_s sınıflandırıcısının eğitimi için pozitif veya negatif örnek olarak kullanılmaktadır [90]. Yeni bir $\vec{x}$ örneğinin tahmin edilmesi aşamasında her sınıflandırıcı fonksiyonun çıktılarından oluşan bir vektör $\vec{f}\,(\vec{x}) = (f_1(\vec{x}), f_2(\vec{x}) \ldots f_s(\vec{x}))$ şeklinde hazırlanır. Bu vektöre en yakın olan sınıf, x girdisinin sınıfı olarak kabul edilir:

$$Sınıf(\overrightarrow{x}) = arg_{n\in N}\min d\,(m_q, \vec{f}\,(\vec{x})) \quad \textbf{(3.8)}$$

Buradaki M matrisi sadece ikili verilerden oluşmaktadır. Uzaklık fonksiyonu d için hamming uzaklığı kullanılabilir:

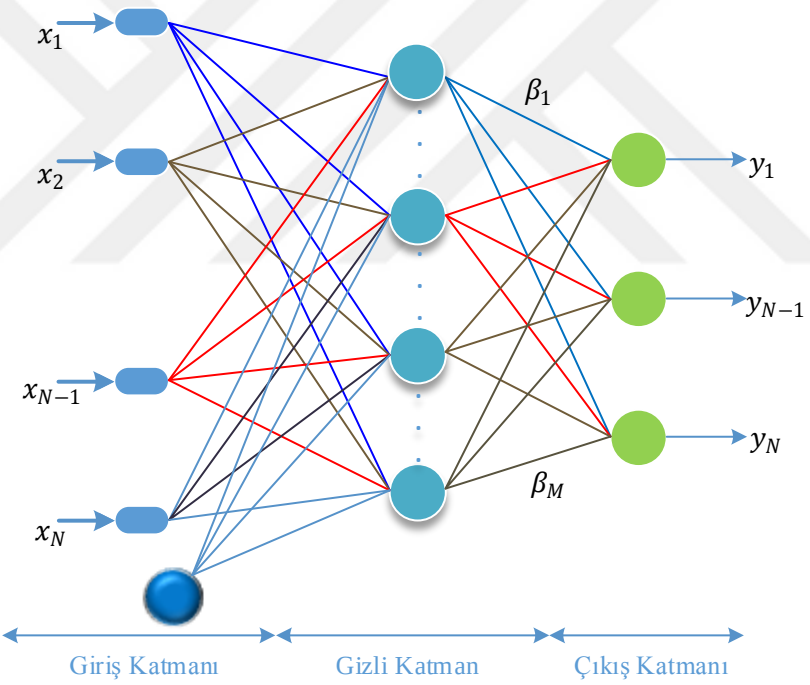
$$d(\vec{m}_n, \vec{f}) = \sum_{t=1}^{t}\frac{|m_{nt} - sign(f_t)|}{2} \quad \textbf{(3.9)}$$

DVM için kullanılan sınıflandırma algoritmasında kayıp tabanlı uzaklık fonksiyonu kullanılabilir:

$$d_L(\vec{m}_n, \vec{f}) = \sum_{t=1}^{t} L(m_{nt}, f_t) \quad \textbf{(3.10)}$$

3.5.4.4. Uç Öğrenme Makineleri

UÖM son zamanlarda veri madenciliği ve makine öğrenmesi modellerinde sıklıkla kullanılmaya başlanmıştır. Huang ve arkadaşları [91-99] tarafından önerilmiş hızlı çalışan ve doğru sınıflandırma oranı yüksek olan bir öğrenme tekniğidir. Araştırmacıların ilgisini çektiği için UÖM kullanılarak birçok çalışma gerçekleştirilmiştir [100-110]. Tek gizli katmanlı ileri beslemeli ağlarda kullanılan ağ ağırlıklarının gradyan temelli yenilenmesi yerine UÖM'de giriş ağırlıkları ve bias değerleri rastgele seçilerek çıkış ağırlıkları analitik bir yöntem ile bulunur. Bu teknik UÖM algoritmalarının hızlı çalışmasını sağlar. Bunun yanında geleneksel geri yayılım algoritması ile öğrenen ileri beslemeli ağlara göre daha iyi bir genelleme başarımına sahip olduğu söylenebilir. Şekil 3.2'de tek gizli katmanlı ileri beslemeli ağ için bir model gösterilmektedir.



Şekil 3.2. Tek gizli katmanlı ileri beslemeli ağ mimarisi

Şekil 3.2'de gösterildiği gibi kullanılan mimari için giriş çıkış ilişkisi x_i ve y_i şeklinde gösterildiği düşünülürse x_i , $x_i = [x_{i1}, x_{i2}, \dots, x_{in},]^T$ matrisi ile ifade edilir. Benzer şekilde y_i ise $y_i = [y_{i1}, y_{i2}, \dots, y_{im},]^T \in R^m$ matrisi ile ifade edilmektedir. N tane gizli sinir

hücresi içeren ve $g(x)$ aktivasyon fonksiyonuna sahip standart tek gizli katmanlı ağ mimarisi Denklem 3.11'de gösterildiği şekilde modellenebilir.

$$\sum_{i=1}^{M} \beta_i g(w_i . x_j + b_i) = o_j, j \in [1, N] \quad \textbf{(3.11)}$$

Burada $w_i = [w_{i1}, w_{i2}, \dots, w_{in},]^T$, i. gizli sinir hücresine ve giriş sinir hücresine bağlı olan ağırlık vektörünü, $\beta_i = [\beta_{i1}, \beta_{i2}, \dots, \beta_{im},]^T$ ise i. gizli sinir hücresine ve çıkış hücrelerine bağlı ağırlık vektörünü gösterir. b_i değeri i. gizli sinir hücresinin eşik değeridir. $o_j = [o_1, o_2, \dots, o_j]^T$ değeri UÖM ağının çıkışını göstermektedir.

Tek gizli katmanlı ileri beslemeli ağ yapısı ortalama olarak neredeyse hatasız bir duruma yaklaşabilir. Denklem 3.11'de verilen eşitlik Denklem 3.12'deki gibi kısaltılabilir. Burada o_j çıkışı y_j olarak alınmıştır.

$$H\beta = Y \quad \textbf{(3.12)}$$

Burada H gizli katman çıkış matrisini göstermektedir:

$$H = \begin{bmatrix} g(w_1 . x_1 + b_1) & \cdots & g(w_i . x_j + b_i) \\ \vdots & \ddots & \vdots \\ g(w_1 . x_j + b_i) & \cdots & g(w_i . x_j + b_i) \end{bmatrix}_{NxM} \quad \textbf{(3.13)}$$

Çıkış katmanı ağırlıkları:

$$\beta = \begin{bmatrix} \beta_1^T \\ \vdots \\ \beta_M^T \end{bmatrix}_{Mx1} \quad \textbf{(3.14)}$$

Çıkış vektörü:

$$Y = \begin{bmatrix} y_1^T \\ \vdots \\ y_N^T \end{bmatrix}_{Nx1} \quad \textbf{(3.15)}$$

Şeklinde ifade edilmektedir [110]. Burada w_i ile giriş katmanı ağırlıkları b_i ile gizli katman eşik değerleri rastgele seçilerek H analitik olarak hesaplanır. Geleneksel tek gizli katmanlı ileri beslemeli ağ yapısında ağın eğitilmesi gerekirken UÖM'de Denklem 3.12'de verilen doğrusal bir denklem elde edilmektedir [111]. Çıkış ağırlıkları $\beta = H^{\dagger}Y$ şekilde gösterilebilir.

Burada β, 3.15'den elde edilmiştir. $H^{\dagger}$, H çıkış matrisinin genelleştirilmiş Moore-Penrose matrisi olarak tanımlanmıştır [112,113]. UÖM'ye ait algoritma aşağıdaki şekilde üç adımda belirtilebilir:

- İlk aşamada $w_i = [w_{i1}, w_{i2}, \dots, w_{in},]$ giriş ağırlıkları ve gizli katman eşik değeri b_i rastgele seçilerek üretilir,
- İkinci aşamada H gizli katman çıkışı hesaplanır,
- Üçüncü aşamada β çıkış ağırlıkları hesaplanarak Y hedefine ulaşılır.

3.5.5. Model Değerlendirme Seçimi

Tez çalışması için verilere uygulanacak olan modellerin değerlendirilmesi amacıyla daha önce geliştirilmiş birçok performans değerlendirme yöntemi ve ölçüsü bulunmaktadır. Bu yöntem ve ölçümlerden faydalanılarak tez verileri ile yapılacak olan model çalışmasında hangi modelin seçilmesi gerektiğine dair fikir oluşturulması amaçlanmaktadır.

3.5.5.1. Model Performans Değerlendirme Yöntemleri

Ağ trafiğini modellemek için kullanılacak olan makine öğrenmesi algoritmasının göstereceği performans; algoritmanın seçimi ile birlikte kullanılan verinin eğitim ve test büyüklüğü, sınıfların dağılımı, sınıflandırmanın hatalı yapılması ile de doğrudan ilgilidir. Bu tür problemlerin ortadan kaldırılması için araştırmacılar bazı yöntemler geliştirmişlerdir [114].

Holdout yönteminde veriler eğitim ve test için iki parçaya ayrılır. Eğitim için ayrılan veri model oluşturmak için seçilen sınıflandırıcıyı eğitmek için kullanılır. Model için uygun değer bulunması bu adımda gerçekleştirilir. Test verileri ise kullanılan modelin ne kadar başarılı olacağını göstermek için kullanılır. Bu yöntemin bir diğer çalışma şekli ise tekrarlı kullanımdır. Bu şekilde farklı alt veri parçaları oluşturmak için her seferinde verinin belirli bir oranı eğitim seti için ayrılarak holdout yöntemi birkaç kez tekrar ettirilir. Tekrarlı durumda test için ayrılan veri parçaların üst üste binme olasılığı olduğu için çok tercih edilen

bir yöntem değildir. Üçlü ayırma yönteminde veriler eğitim, doğrulama ve test olmak üzere üç parçaya bölünerek model seçimi ve performans durumu aynı zamanda yapılır. Tabakalı örnekleme yöntemi çıktı için kullanılan sınıfların çok az olması durumunda sınıf oranlarının korunması amacıyla kullanılır. Çapraz geçerleme yöntemi k-kat ve birini dışarıda bırak çapraz geçerleme olarak iki farklı biçimde kullanılmaktadır. K-kat çapraz geçerleme yönteminde veriler k tane birbirine eşit parçaya ayrılır, her bir k parçadan her seferde bir tanesi test için, $k-1$ tanesi ise eğitim için kullanılmak üzere seçilir. Elde edilen k tane hata oranının ortalama değerinin büyük olması durumunda tahminci daha doğru sonuç verecek ve gerçek hata tahmincisinin sapması küçük olacaktır [115]. Çoğu zaman k değeri 10 olarak kullanılmıştır [116-121]. Çapraz geçerleme yönteminin diğer alt yöntemi olan birini dışarıda bırak uygulamasında ise her seferinde veriden sadece bir örnek test verisi için, verinin geri kalan kısmı ise eğitim verisi için kullanılmaktadır. Rastgele örnekleme yönteminde veri k parçaya ayrılır, veri değerleri istenilen orandan eğitim ve test veri setine aktarılır. Bootstrap örnekleme yönteminde ise veri setinin k tane örneği olduğu kabul edilirse veri setinden eğitim verisi için k defa rastgele örnek seçilir, fakat her seçimden sonra seçilen örnek veri setinde kalmaya devam eder. Bu sebeple aynı örnek eğitim veri setinde çok kez tekrar edebilir. Eğitim için veri seçme işlemi bittikten sonra bu sette olmayan asıl veri setindeki tüm örnekler test veri setine aktarılır.

3.5.5.2. Model Performans Değerlendirme Ölçüleri

Veri setine uygulanacak olan sınıflandırıcı modellerin performans karşılaştırılmasının yapılabilmesi için hata matrisinin oluşturulması gibi yöntemler geliştirilmiştir [122]. Tablo 3.1'de veri setinden alınan gerçek değerler ile tahmin edilen değerlerin arasındaki ilişki gösterilmektedir. Bu tabloda verilen değerlere göre hesaplanan performans kıstasları aşağıda açıklanmıştır [6, 13, 123-127].

Tablo 3.1. Performans değerlendirmesi için gerçek değerler ile tahmini değerlerin karşılaştırılması

Etiket	Gerçek Pozitif	Gerçek Negatif	Gerçek Toplam
Tahmini Pozitif	Doğru Pozitif (D_p)	Yanlış Pozitif (Y_p)	T_p
Tahmini Negatif	Yanlış Negatif (Y_n)	Doğru Negatif (D_n)	T_n
Tahmini Toplam	(p)	(n)	T

Tablo 3.1'deki değerlere göre doğru pozitif; gerçekte hem de tahminde pozitif olarak işaretlenen değerleri, yanlış pozitif; gerçekte negatif olmasına rağmen pozitif işaretlenen değerleri, yanlış negatif; gerçekte pozitif olmasına rağmen negatif olduğu tahmin edilen değerleri, doğru negatif ise hem gerçekte hem de tahminde negatif işaretlenen değerleri ifade etmektedir. Sınıflandırmanın doğruluğu Denklem 3.16'da gösterildiği şekilde hesaplanabilir.

$$Doğruluk\ Değeri\ (DOD) = \frac{D_p + D_n}{T} \tag{3.16}$$

Duyarlılık, sınıflandırıcıya ait pozitif sınıf etiketlerini tahmin etme etkinliği olarak adlandırılmaktadır. Duyarlılık değerinin hesaplanması Denklem 3.17'de gösterildiği gibi doğru sınıflandırılan pozitif örneklerin, doğru sınıflandırılan pozitif ve yanlış sınıflandırılan pozitif örneklerin toplamına oranıdır.

$$Duyarlılık\ (DUY) = \frac{D_p}{p} \tag{3.17}$$

Belirleyicilik, negatif sınıf etiketlerini tahmin etmeye çalışan sınıflandırıcının başarısıdır. Denklem 3.18'de gösterildiği gibi doğru sınıflandırılan negatif örneklerin, doğru sınıflandırılan negatif örnekler ile yanlış sınıflandırılan pozitif örneklerin toplamına oranıdır.

$$Belirleyicilik(BEL) = \frac{D_n}{n} \tag{3.18}$$

Yanlış pozitif oranı, gerçekte negatif olan fakat pozitif etiketli sınıflandırılmış örneklerin, tüm negatif etiketli örneklere oranını verir. Benzer şekilde gerçekte pozitif olan

fakat negatif olarak sınıflandırılmış örneklerin, tüm pozitif etiketlere örneklere oranı da yanlış negatif oranını verir.

$$Yanlış\ Pozitif\ Oranı(YPO) = \frac{Y_p}{n} \quad \textbf{(3.19)}$$

$$Yanlış\ Negatif\ Oranı(YNO) = \frac{Y_n}{p} \quad \textbf{(3.20)}$$

Pozitif kestirim oranı veya kesinlik; doğru sınıflandırılmış olan pozitif etiketli örneklerin, tahmin edilen toplam pozitif örneklere oranıdır. Benzer şekilde negatif kestirim oranı ise doğru sınıflandırılmış olan negatif sınıf etiketli örneklerin, tahmin edilen tüm negatif örneklere olan oranından bulunur.

$$Pozitif\ Kestirim\ Oranı(PKO) = \frac{D_p}{T_p} \quad \textbf{(3.21)}$$

$$Negatif\ Kestirim\ Oranı(NKO) = \frac{D_n}{T_n} \quad \textbf{(3.22)}$$

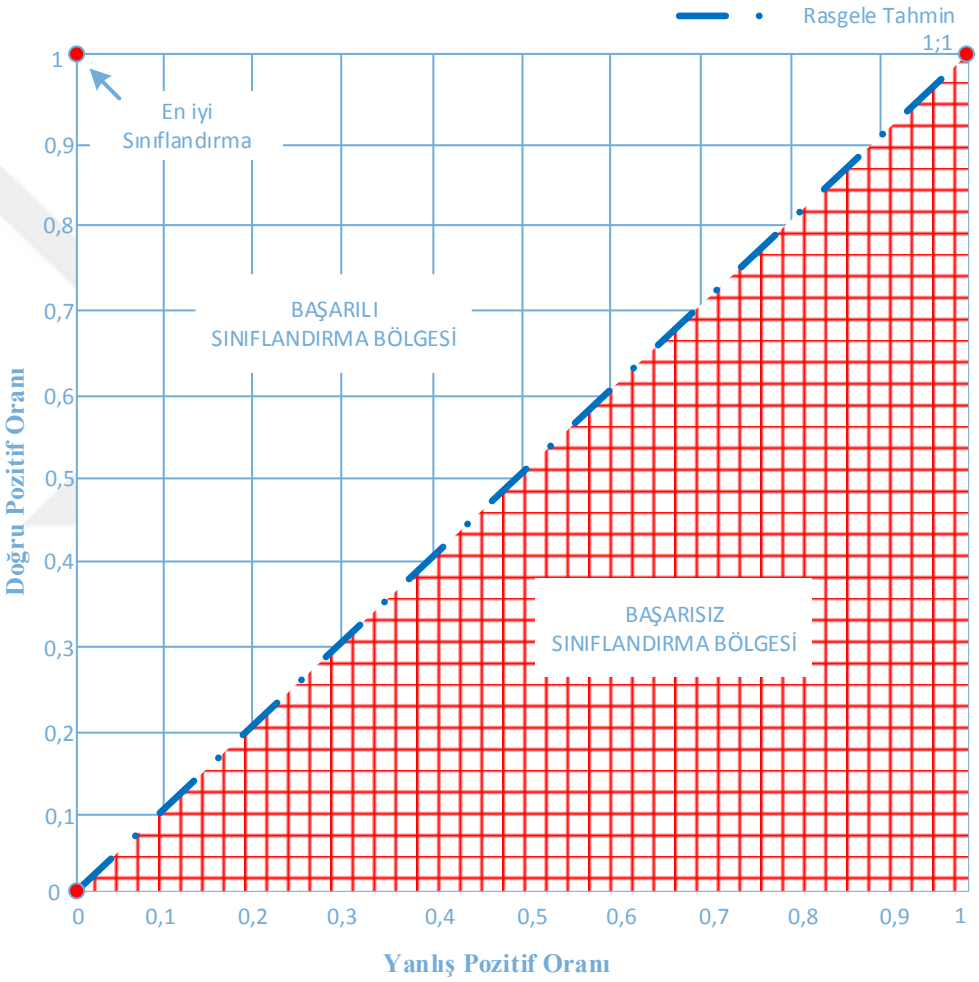
F-ölçüsü, pozitif kestirim oranı ve duyarlılık ölçülerinin uyumlu bir ortalamasıdır Denklem 3.23'de gösterildiği şekilde bulunur.

$$F - \text{Ölçüsü}\ (F) = \frac{2 * PKO * DUY}{PKO + DUY} \quad \textbf{(3.23)}$$

Yukarıda verilen performans değerlendirme ölçütleri için kullanılan hesaplamalar ikili sınıfa sahip veri setlerinde kullanılmak amacıyla geliştirilmiştir. Çoklu sınıfa sahip sınıflandırma problemlerinin çözümü için genel olarak ikili sınıflandırma için kullanılan yöntemlerin genelleştirilmiş halleri kullanılmaktadır [128-130]. Genel olarak çoklu sınıflandırıcı birden fazla ikili sınıflandırıcıya dönüştürülerek her bir ikili sınıflandırıcı için performans ölçüsü bulunmakta ve hesaplanan ölçülerin ortalaması alınarak çoklu sınıf performans ölçüleri elde edilmektedir. Örneğin m adet sınıfa sahip bir sınıflandırıcının doğruluk değeri Denklem 3.24'de gösterildiği şekilde hesaplanmaktadır.

$$Ortalama\ Doğruluk\ Değeri\ (DOD_{ort}) = \frac{\sum_{i=1}^{m}(\frac{D_{pi}+D_{ni}}{Ti})}{m} \quad \textbf{(3.24)}$$

Sınıflandırmanın doğruluğunun anlaşılması amacıyla AİK eğrileri oluşturulmuştur [131-133]. Bu grafikler y ekseninde D_p, x ekseninde ise Y_p değerlerinin gösterilmesi ile oluşturulurlar. Şekil 3.3'de bu eğrilerin gösterim grafiği verilmiştir [124].



Şekil 3.3. AİK uzayı

AİK uzayında D_p değerlerinin yüksek, Y_p değerlerinin ise düşük olduğu durumlar bir sınıflandırıcının iyi bir sınıflandırma yaptığını işaret etmektedir. Bu nedenle iyi bir sınıflandırıcının Şekil 3.3'de verilen grafiğin sol üst tarafında değerler alması beklenmektedir [53].

3.5.5.3. Modelin Uygulamaya Geçirilmesi

Tez çalışması için kullanılacak olan veri setinde tercih edilecek olan modeller önceki bölümlerde verilen performans değerlendirme ölçütleri ile kıyaslandıktan sonra sınıflandırma probleminin çözümüne geçilmektedir. Bu amaçla veri setinin iyi tanımlanması ve anlaşılması gerekmektedir. Eğer veri setinde eksiklikler veya bozukluklar varsa giderilerek analiz için hazırlanması gerekmektedir. Makine öğrenmesi algoritmaları veri üzerinde performans değerlendirme yöntemleri ile kıyaslanarak seçilmelidir.

Seçilen makine öğrenmesi algoritmasının en iyi performansı sağlayabilmesi amacıyla GA yöntemlerinden faydalanılacaktır. Yapılan uygulamalar Bölüm 6 içerisinde ayrıntılı olarak anlatılmıştır.

4. UÇ ÖĞRENME MAKİNELERİ

UÖM gizli katman ağırlıklarını rastgele alan hızlı öğrenen bir yapay sinir ağı modelidir. Huang ve arkadaşları [91, 92, 134] tarafından tek gizli katmanlı ileri beslemeli sinir ağlarının eğitimi amacıyla önerilmiştir. Geleneksel tek gizli katmanlı ileri beslemeli sinir ağlarında ağırlık ve eşik değerlerinin gradyan temelli öğrenme algoritmaları ile yenilenmesi gerekmektedir. İyi bir çözümün sağlanmasında, bu durum hem öğrenme sürecinin uzamasına hem de hatanın yerel bir noktaya takılmasına sebep olabilir. Hatanın yerel bir noktaya takılıp kalma problemi için momentum değerinde değişiklik yapılabilir, fakat bu değişikliğin öğrenme sürecine bir katkısı olmayacaktır [135]. Tek gizli katmanlı ileri beslemeli bir ağda ağırlıklar ve eşik değerleri ağın performansını etkilememektedir. UÖM'de giriş ağırlıkları ve eşik değeri rastgele üretilirken çıkış ağırlıkları hesaplanarak bulunmaktadır. Bu durum UÖM'nin geleneksel yöntemlere göre daha iyi sonuç vermesini ve hızlı çalışmasını sağlamaktadır [94, 95]. UÖM ile yapılan temel özelliklerinin kullanıldığı klasik veya farklı türevlerinin geliştirildiği birçok çalışma bulunmaktadır [96, 98, 101, 106, 136-144].

Bu tez çalışmasında uç öğrenme algoritmalarının geliştirilmiş yaklaşımlarından birisi olan ÇTUÖM ile ağ trafiğinde yer alan veriler analiz edilerek yüksek oranda doğruluk değerlerine ulaşılmıştır. Yapılan uygulamalar Bölüm 6'da ayrıntılı olarak verilmiştir.

4.1. Klasik Uç Öğrenme Makinesi

Giriş çıkış ilişkisi x_i ve y_i ise $x_i, x_i = [x_{i1}, x_{i2}, \dots, x_{in},]^T$ matrisi ile ifade edilir. Benzer şekilde y_i ise $y_i = [y_{i1}, y_{i2}, \dots, y_{im},]^T \in R^m$ matrisi ile ifade edilmektedir. N tane gizli sinir hücresi içeren ve $g(x)$ aktivasyon fonksiyonuna sahip standart tek gizli katmanlı ağ mimarisi Denklem 4.1'de gösterildiği şekilde modellenebilir.

$$\sum_{i=1}^{M} \beta_i g(w_i . x_j + b_i) = o_j, j \in [1, N] \quad \textbf{(4.1)}$$

Burada $w_i = [w_{i1}, w_{i2}, \dots, w_{in},]^T$, i. gizli sinir hücresine ve giriş sinir hücresine bağlı olan ağırlık vektörünü; $\beta_i = [\beta_{i1}, \beta_{i2}, \dots, \beta_{im},]^T$ ise i. gizli sinir hücresine ve çıkış

hücrelerine bağlı ağırlık vektörünü gösterir. b_i değeri i. gizli sinir hücresinin eşik değeridir. $o_j = [o_1, o_2, \dots, o_j]^T$ değeri UÖM ağının çıkışını göstermektedir.

Tek gizli katmanlı ileri beslemeli ağ yapısı ortalama olarak neredeyse hatasız bir duruma yaklaşabilir. Denklem 4.1'de verilen eşitlik, Denklem 4.2'de gösterildiği gibi kısaltılabilir. Burada o_j çıkışı y_j olarak alınmıştır.

$$H\beta = Y \tag{4.2}$$

Burada H gizli katman çıkış matrisini göstermektedir:

$$H = \begin{bmatrix} g(w_1.x_1 + b_1) & \cdots & g(w_i.x_j + b_i) \\ \vdots & \ddots & \vdots \\ g(w_1.x_j + b_i) & \cdots & g(w_i.x_j + b_i) \end{bmatrix}_{NxM} \tag{4.3}$$

Çıkış katmanı ağırlıkları:

$$\beta = \begin{bmatrix} \beta_1^T \\ \vdots \\ \beta_M^T \end{bmatrix}_{Mx1} \tag{4.4}$$

Çıkış vektörü:

$$Y = \begin{bmatrix} y_1^T \\ \vdots \\ y_N^T \end{bmatrix}_{Nx1} \tag{4.5}$$

Şeklinde ifade edilmektedir [91, 98]. Burada w_i ile giriş katmanı ağırlıkları b_i ile gizli katman eşik değerleri rastgele seçilerek H analitik olarak hesaplanır. Geleneksel tek gizli katmanlı ileri beslemeli ağ yapısında ağın eğitilmesi gerekirken UÖM'de Denklem 4.2'de verilen doğrusal bir denklem elde edilmektedir. Çıkış ağırlıkları $\beta = H^{\dagger}Y$ şeklinde gösterilebilinir.

Burada β, 4.5'de verilen ifade çözdürülerek elde edilmiştir. $H^{\dagger}$, H çıkış matrisinin genelleştirilmiş Moore-Penrose matrisi olarak tanımlanmıştır [112, 113, 145].

4.2. Çekirdek Tabanlı Uç Öğrenme Makinesi

Sınıflandırma problemlerinin çözümünde kullanılacak olan UÖM'nin optimize edilebilmesi amacıyla bazı yöntemler geliştirilmiştir. Eğitim hatasının azaltılması ve sinir ağının genel performansının yukarı çekilebilmesi için eğitim hatası ve çıkış ağırlıkları aynı anda minimize edilebilir.

$$Minimizasyon: \|H\beta - T\|^2, \|\beta\| \tag{4.6}$$

Denklem 4.6; Karush-Kuhn-Tucker koşuluna [146] göre en küçük kareler çözümü ile Denklem 4.7'de gösterildiği şekilde yazılabilir.

$$\beta = H^T \left[\frac{1}{C} + HH^T\right]^{-1} T \tag{4.7}$$

H Gizli katman çıkış matrisi, C düzenleme katsayısı ve T ise örneklerin beklenen çıkış matrisini ifade etmektedir. UÖM'nin öğrenme algoritması için çıkış fonksiyonu Denklem 4.8'de gösterildiği gibi ifade edilebilir.

$$f(x) = h(x)H^T \left[\frac{1}{C} + HH^T\right]^{-1} T \tag{4.8}$$

Eğer özellik haritalanması $h(x)$ bilinmiyorsa Mercer's koşullarına [147] göre UÖM'nin çekirdek matrisi Denklem 4.9'da gösterildiği şekilde tanımlanabilir.

$$M = HH^T : \; m_{ij} = \mathrm{h}(x_i)\mathrm{h}(x_j) = \mathrm{k}(x_i, x_j) \tag{4.9}$$

Böylece ÇTUÖM'nin $f(x)$ çıkış fonksiyonu Denklem 4.10'da gösterilen haliyle yazılabilir.

$$f(x) = [k(x, x_1), \ldots, k(x, x_N)] \left[\frac{1}{C} + M\right]^{-1} T \tag{4.10}$$

Yukarıda gösterilen $M = HH^T$ ve $k(x, y)$ tek gizli katmanlı ileri beslemeli sinir ağlarına ait çekirdek fonksiyonudur. Mercer koşullarını sağlayan doğrusal, gauss, polinom, üstel, dalgacık gibi birçok çekirdek tabanlı fonksiyon bulunmaktadır.

Bu tez çalışmasında dalgacık fonksiyonlu ÇTUÖM algoritması kullanılarak veriler sınıflandırılmıştır. Dalgacık tabanlı çekirdek fonksiyonu Denklem 4.11'de gösterildiği şekilde ifade edilebilir.

$$k(x,y) = \cos\left[a\frac{\|x-y\|}{b}\right]\exp\left[-\frac{\|x-y\|^2}{c}\right] \quad \textbf{(4.11)}$$

Dalgacık tabanlı uç öğrenme algoritmasında kullanılan a, b ve c parametreleri sınıflandırmanın başarılı bir şekilde yapılabilmesi için önemli bir rol oynamaktadır [148]. Bu tez çalışmasında parametrelerin en uygun olanın seçilebilmesi için GA yaklaşımından faydalanılarak bir uygulama geliştirilmiştir. Uygulamanın detayları Bölüm 6 da ayrıntılı olarak ifade edilmektedir.

4.3. Diğer Uç Öğrenme Makinesi Yaklaşımları

Klasik UÖM algoritmasının optimize edilerek tıpkı ÇTUÖM algoritması gibi farklı uç öğrenme algoritması çalışmaları da bulunmaktadır.

Aşamalı veya artımlı yaklaşım, küçük bir ağdan başlanarak adım adım belli bir durma değerine ulaşılana kadar yeni nöronlar eklenerek gizli katmanın büyütülmesi prensibine dayanır [94, 134]. Budamalı yaklaşım, artımlı yaklaşımın aksine öncelikle nöronların ihtiyaç duyduğundan daha büyük bir küme oluşturur. Tek katmanlı ileri beslemeli ağın kullanımı için M nöronların en iyi alt kümesini çıkarmaya çalışır [101]. Budamalı yaklaşımın farklı alt yöntemleri de mevcuttur [100, 149, 150]. Gizli nöronların alt kümesinin seçilmesi için farklı bir UÖM yaklaşımı da düzenlileştirme yaklaşımıdır [100, 104]. Belirtilen yaklaşımların dışında araştırmacıların geliştirmiş olduğu farklı UÖM yaklaşımları da mevcuttur [151-158]. Bu tez çalışmasında sınıflandırma yapmak için tercih edilen UÖM algoritmalarının geliştirilmeye açık olması bu tercihin yapılmasındaki en etkili sebeplerden birisidir.

5. GENETİK ALGORİTMA

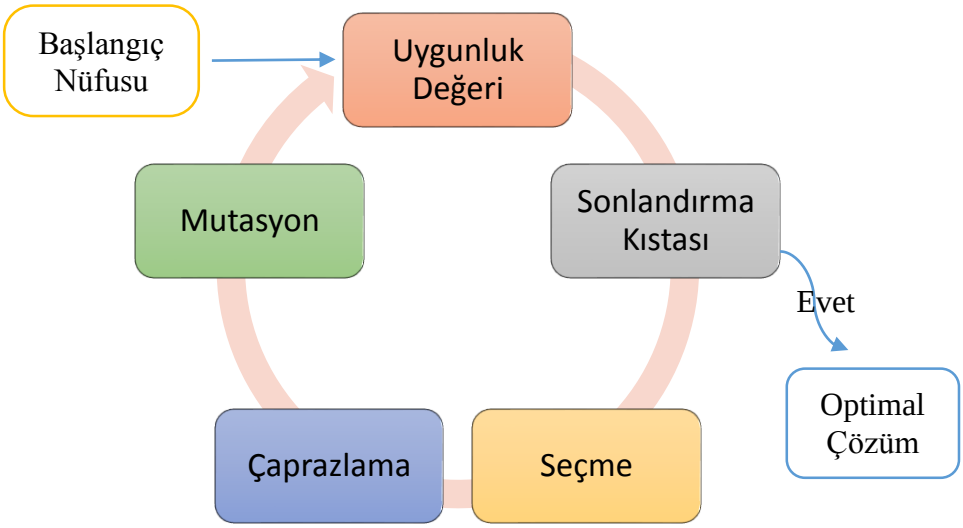
GA kavramı ile canlı ekosisteminde oluşan doğal değişiklikler, sosyal sistemlerde taklit etme ve psikolojik sonuçların değerlendirilmesi ile değişken yöntemlerin modellenmesini amaç edinir. Holland [159] yapmış olduğu çalışmalar ile teknolojik ilerlemelerde doğanın esin kaynağı olabileceğini kanıtlamıştır. Goldberg ise yapmış olduğu çalışmalar ile pratik alanda GA tekniklerinden faydalanılarak birçok soruna çözüm bulunabileceğini göstererek GA'nın kullanım alanının yaygınlaşmasına büyük bir katkıda bulunmuştur [160]. GA tekniklerinin makine öğrenmesi yaklaşımları içerisinde başarılı uygulama alanları bulunmaktadır [161-165]

Bu bölümde tez çalışması uygulamaları içerisinde de kullanılan ve en uygun değerlerin bulunmasında başarılı sonuçlar veren GA kavramı ile ilgili bilgi verilecektir.

5.1. Genetik Algoritmanın Tanımı

GA, her yinelenme sonucunda daha iyi sonuçlar elde edilebilmesi amacıyla sürekli gelişen nesillerin taklit edildiği bir yapay zekâ sistemi olarak kabul edilebilir. GA ile problemin çözümünde kullanılan işlemler çaprazlama, mutasyon ve seçme aşamalarından oluşmaktadır. Çaprazlama ile iki veya daha fazla bireye ait özellikler birleştirilerek daha iyi sonuç veren bireylerin oluşturulması amaçlanır. Mutasyon ile farklı adımların rastgele bir şekilde sıralanarak birleştirilmesi amaçlanır. Bireylerin seçimi ile daha iyi sonuç veren birey bir sonraki kuşağa aktarıldıktan sonra çözüm havuzu olarak kabul edilen diğer bireylere ait toplumdan çıkartılır. Son kalan toplumun uygunluk değeri ve kalitesi iyi olan bireyin, hesaplama içerisindeki en iyi çözüme yaklaşacak bir değer olması beklenmektedir [166].

GA; tam olarak çözüm yolu bilinmeyen veya bilinse bile bu işlemin çok fazla zaman alacağı düşünülen problemlerin sonuçlandırılmasında oldukça etkili bir yöntemdir. Şekil 5.1'de GA'nın genel akış diyagramı verilmiştir.



Şekil 5.1. GA genel akış diyagramı

5.2. Genetik Algoritmanın Bileşenleri

Şekil 5.1'de gösterildiği gibi GA'nın bulundurması gereken bileşenler mevcuttur. Öncelikle kromozom yapısı tasarlanarak problemin çözümü için bir tanım belirlenir. Başlangıç nüfusu çoğu zaman rastgele olarak belirlenir. Problem için uygunluk fonksiyonu belirlendikten sonra GA'nın parametrelerinden çaprazlama, mutasyon ve seçme kıstası belirlenir. Başlangıç nüfusundaki her bir birey için uygunluk fonksiyonu değeri hesaplanır. Seçim ile nüfusa yeni bireyler katılır. Nüfusa katılan yeni bireyler tekrar çaprazlama ve mutasyon operatörleri ile işlenir, yeni nüfustaki bireylerin uygunluk değeri hesaplanır. İşlemler bu şekilde çalıştırılarak durma kıstasına ulaşılıp ulaşılmadığı kontrol edilir. Durma kıstasına ulaşılmışsa istenilen en iyi çözüme ulaşıldığı kabul edilerek işlem sonlandırılır [167-169].

5.2.1. Kromozom Tasarımı

GA ile çözüm aranan problemlerde her bir çözümü gösterecek şekilde bir dizi kodlama kullanılmaktadır. Kodlamanın yapılma şekli çözümün iyi olması için önemli bir etkendir. İkili, tam sayılı, ondalık sayılı, harfli kodlama gibi çeşitli kodlama teknikleri bulunmaktadır. Kromozom tasarımı için kodlama işleminin problemin yapısına uygun bir şekilde olması çözüm için önemlidir. Bu tez çalışmasında ikili kodlama tekniğinden faydalanılmıştır.

5.2.2. İlk Nüfusun Oluşturulması

GA'nın temel prensibinde tek bir çözüm yerine bir çözüm kümesi ile çalışmak yatmaktadır. Bu sebeple başlangıçta çözüm kümesinin nasıl seçileceğinin iyi tanımlanması gereklidir. Genelde rastgele olarak seçilmektedir. İkili kodlama tekniğinde 0 ve 1 sayıları eşit alınırken diğer kodlama tekniklerinde normal dağılım gözetilerek nüfus oluşturulmaya çalışılır.

5.2.3. Uygunluk Fonksiyonunun Seçimi

GA için problemin amaç fonksiyonu, ulaşılmak istenilen uygunluk fonksiyonudur. Toplum içerisindeki kaliteli bireylerin bir sonraki nesillere aktarılması için daha önceden oluşturulmuş olan bireyin uygunluk durumunun kontrol edilmesi işlemi yapılır. Uygunluk fonksiyonu, bireylerin uygunluk seviyesini belirleyebilmek amacıyla değerlendirmenin yapılması için kalitenin ölçülmesini sağlar. Böylece uygunluk fonksiyonu değeri iyi olan çözüm kaliteli ve istenilen çözüme en yakın çözüm olan kromozomlar olacaktır. Uygunluk fonksiyonunun çalışmanın başında belirlenmesi ile nüfus içerisindeki tüm bireylerin uygunluk değerleri hesaplanmış olur. GA'nın temel amacı olarak uygunluk değeri en yüksek olan bireyler sonraki nesillere aktarılmaya çalışılır.

5.2.4. Genetik Operatörler

GA yöntemi ile kaliteli nesil oluşturulabilmesi amacıyla bireyler bir dizi işlemden geçirilmektedir. GA tekniğinde seçim, çaprazlama ve mutasyon olarak adlandırılan üç temel operatör bulunmaktadır.

5.2.4.1. Uygun Bireylerin Seçimi

Uygunluk değeri iyi olan bireylerin yaşaması ilkesine göre seçme işlemi gerçekleştirilir. Başlangıç kuşağında bulunan çözümler içerisinden en uygun olan kromozomlardan bazıları seçilerek sonraki kuşağa aktarılır. Yeni kuşakta bir önceki kuşaktan gelmiş olan kromozomlar dışındaki diğer kromozomlar genetik teknikler kullanılarak oluşturulur. Uygunluk fonksiyonu ile ölçüldüğünde değeri iyi çıkan kromozomların bir sonraki kuşağa aktarılma ihtimali de yüksek olmaktadır. Bu şekilde yeni oluşacak olan kuşağa kötü kabul edilen bireyler taşınmamış olunur. İyi kabul edilen bireylerin yeni kuşağa taşınması ve bu iyi etiketli bireylerden başka bireylerin üretilmesi

istenilen en iyi çözüme kavuşulmasını sağlayacaktır. Bir sonraki kuşağa aktarılacak olan kromozomların seçimi için birkaç yöntem kullanılmaktadır. Bunlar turnuva, rulet çarkı ve elitist gibi yöntemlerdir.

Turnuva yönteminde seçim politikasında nüfus içinden rastgele seçilen k sayıda kromozom içerisindeki en iyi olarak düşünülenler ebeveyn olarak seçilir. Turnuvanın büyüklüğü fazla olursa daha az uygunluk değerindeki kromozomlarım seçilme şansı da küçülecektir [170].

Rulet tekeri yönteminde nüfus içerisindeki tüm kromozomların kalitesi belirlenir. Kalitesi yüksek olan kromozomun seçilme olasılığı da yüksek olacaktır. Bu ihtimaller göz önüne alınarak tüm kromozomlar ihtimal değeri büyüklüğüne göre rulet çemberine yerleştirilir. En yüksek ihtimalli bireyler rulet çarkında daha geniş bir alanı kapatır, böylece seçilme ihtimalini artırmış olur. Bu teker yeni nüfusun birey sayısı tamamlanıncaya kadar dönerek yeni bireylerin seçimi gerçekleştirilir [170].

Elitist seçim yönteminde nüfusun en iyi bir veya iki bireyi korunarak, geri kalanlar yeni bireyler ile değiştirilir. Böylece en iyi bireylerin hayatta kalması amaçlanmaktadır. Bu yöntem ile en iyi uyum değerinde olan bireyin genetik operatörlerin kullanılması ile yok olmasına engel olunmaya çalışılır [168].

5.2.4.2. Çaprazlama

Çaprazlama operatörü ile melez yapıların üretilmesine denk gelecek bir özellik GA'ya kazandırılmış olunur. Seçme işlemi ile bir sonraki topluma aktarılması istenilen kromozomlardan bir çift rastgele seçilerek bu çiftten yeni bireyler elde edebilmek için çaprazlama işlemi yapılır. Bu sayede toplumda bulunmayan yeni bireyler de sonraki kuşaklara aktarılarak çeşitlilik sağlanmış olunur. Çaprazlama yöntemi birkaç şekilde yapılabilmektedir.

Çaprazlama tekniklerinden ilki, bir nokta çaprazlaması olarak adlandırılır. Amaç tesadüfi bir şekilde seçilen bir tamsayının seçilen birinci ve ikinci kromozom için hangi gene denk geldiğinin tespit edilmesi ve bu iki kromozom arasındaki genlerin belirtilen tam sayıya denk gelen genden itibaren değiştirilerek yeni bireylerin elde edilmesidir.

Bir diğer çaprazlama tekniği iki nokta çaprazlamasıdır. Rastgele seçilen iki tam sayının birinci ve ikinci kromozom için hangi gene denk geldiği bulunarak iki kromozom arasındaki birinci ve ikinci tam sayının arasında olan genler birbiriyle yer değiştirilerek yeni bireyler oluşturulur.

Çaprazlama tekniklerinden düzgün çaprazlama olarak adlandırılan teknikte rastgele bir bitlik bir sayı üretilir. Üretilen bit 1 ise birinci, değilse ikinci kromozomdan gen alınarak yeni bireyler oluşturulur.

5.2.4.3. Mutasyon

Tıpkı doğada gerçekleşen mutasyon olayı gibi GA için de çeşitlilik ile doğru sonuçlara ulaşmamız sağlanır. Yeni nüfusta bulunan çözümlere ait her bit kontrol edilerek mutasyon oranına göre bit değeri değiştirilir. Mutasyon operatörü kullanılmayan bir GA için en iyi çözüm ancak çözüm bireylerinin ilk toplumda olması durumda bulunabilir. Mutasyon ile daha önce görülmemiş yeni çözüm değerlerinin bulunması sağlanır. Kromozom üzerinde rastgele seçilen bir genin değiştirilmesi ile sağlanacak olan mutasyon işlemi GA'nın yerel iyi çözümler ile boğuşmasını önleyerek genel iyi çözümlerin aranmasını sağlar.

5.3. Genetik Algoritmanın İşleyişi

GA ile bir problem çözülürken tek bir çözüm yerine bir çözüm kümesi ile çalışılır. Diğer çözüm algoritmalarına göre GA'nın güçlü bir yanı olan bu durum çözümler arasında farklar oluşmasını sağlar. Nüfus olarak adlandırılan bu çözüm kümesi ilk önce rastgele bir şekilde oluşturulur. Nüfus içerisindeki bireylerin bir kısmı iyi bir kısmı nispeten kötü çözümü ifade eder. Bir nüfustan diğerine geçilirken daha iyi çözümlerin bulunması GA'dan beklenendir. Kromozomlar bir sonraki nüfusa geçerken seçim, çaprazlama ve mutasyon operatörleri ile çeşitliliğe uğramış olur. Temel olarak yeni bireyler önceki bireylerden türetilmiş olmasına rağmen çeşitlilik ile algoritmanın arama uzayındaki birçok çözümü gözden geçirmesi sağlanır. GA ile elde edilen çözüm problemin en iyi genel çözümü olmayabilir fakat beklenti iyi bir çözüm olmasıdır.

6. AĞ TRAFİĞİNİN SINIFLANDIRILMASINDA GELİŞTİRİLEN UYGULAMALAR

Tez çalışmasının bu bölümünde önceki bölümlerde anlatılmış olan makine öğrenmesi yöntemlerinden YSA, NB, DVM ve UÖM algoritmaları ile ilgili uygulamalar yapılarak karşılaştırılmış ve en iyi sonuçların alındığı makine öğrenmesi yöntemi olarak UÖM algoritmalarının kullanılmasına karar verilmiştir. Seçilen UÖM algoritmaları da hem klasik hem de çekirdek tabanlı yaklaşımlar ile test edilmiştir. Klasik UÖM'ye göre daha performanslı çalıştığı görülen dalgacık fonksiyonunun kullanıldığı ÇTUÖM algoritması üzerinde geliştirme yapılmıştır. Dalgacık fonksiyonu için kullanılan parametrelerin seçiminde en uygun değerlerin seçiminin yapılabilmesi amacıyla GA tekniğinden faydalanılarak bir uygulama geliştirilmiştir. Tez çalışmasının 3.5 bölümünde bahsedilmiş olan makine öğrenmesi süreçlerinin işletilerek, uygulamaların nasıl sonuçlandırıldığı ve başarım ölçütlerinin neler olduğu ayrıntılı olarak irdelenmiştir.

6.1. Problemin Tanımı ve Verilerin Anlaşılması

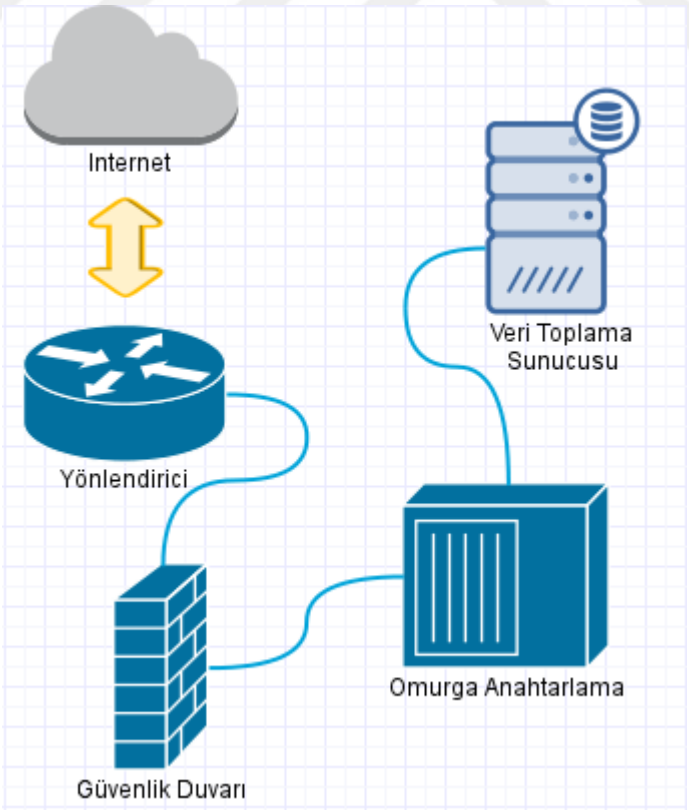
Nesnelerin interneti kavramının hayatımıza sıklıkla girmeye başlaması ile birlikte kullanıcı başına internete bağlı olan cihaz sayısı her geçen gün biraz daha artmaktadır. Internet kullanımının artması ile beraber internet üzerindeki verilerin sınıflandırılmasının ve internet kullanımının daha verimli hale getirilmesinin özellikle kurumsal ağları yöneten ağ yöneticileri için önemi daha da artmıştır. Son zamanlarda internet trafiğinin sınıflandırılmasına yönelik çalışmalar yaygınlaşmıştır. Bu çalışmalar sayesinde ağ yöneticileri, sınıflandırma ile elde edilecek olan bilgi ile ağ üzerinde servis kalitesinin artırılmasını, ağın verimli kullanılabilmesini, hizmet paketlerinin oluşturulmasını ve kullanıcılara sunulabilmesini gerçekleştirebilmektedir. Ağ sınıflandırılması yapılabilmesi için bilginin alınma probleminin çözülmesi ve tanımlanması gerekmektedir. Ağ üzerinde akan bilginin nasıl toplanacağı ve nasıl sınıflandırılacağına çözüm aranmalıdır.

Bu tez çalışmasında ağdaki bilginin alınması için hem hazır verilerden hem de Fırat Üniversitesi yönlendirici ve ana omurga anahtarlama cihazı üzerinden geçen kampüs kullanıcılarına ait veriler toplanarak analiz yapılmıştır. Toplanan verilerin analizi için sınıflandırmayı amaçlayan makine öğrenmesi yöntemlerinin uygulanması ve başarılı olan yöntemin geliştirilmesine karar verilmiştir.

6.2. Verilerin Hazırlanması

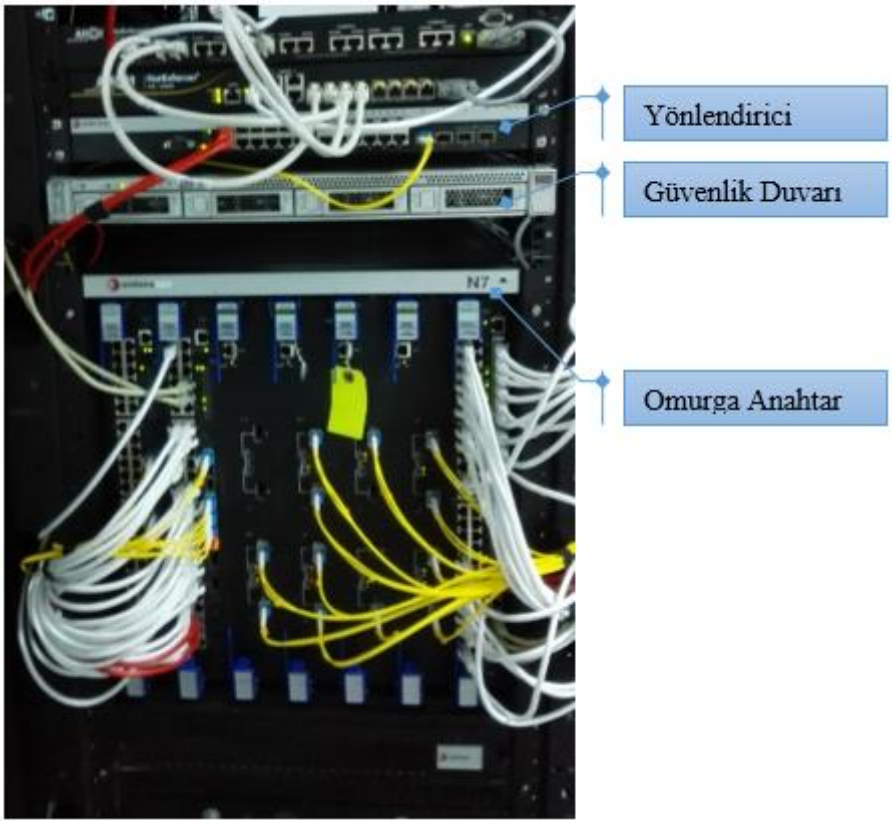
Ağ üzerindeki kullanıcılardan alınan internet kullanım bilgileri için üç farklı veri hazırlama yöntemi kullanılmıştır. Moore ve arkadaşlarının [68] Cambridge Üniversitesi kampüsünden aldıkları veriler bu tez çalışmasında da kullanılmıştır. Başka bir veri hazırlama yöntemi olarak Fırat Üniversitesi sistem odasında bulunan omurga anahtarlama cihazı üzerindeki veriler paket kayıt dosyaları şeklinde kaydedilmiştir. Kaydedilen verilerin sınıflandırılarak makine öğrenmesi yöntemleri ile analiz edilebilmesi amacıyla Moore ve arkadaşlarının verileri sınıflandırma için kullandıkları açık kaynak kodlu yazılım kullanılmıştır. Diğer bir veri hazırlama yöntemi olarak da çeşitli ağ kaynakları üzerinden kaydedilmiş verilerin sunulduğu CAIDA [69] veri tabanına üye olunarak buradan sınıflandırılmamış paket kayıt dosyaları alınarak incelenmiştir.

Fırat Üniversitesi internet ağındaki verilerin toplanabilmesi amacıyla Şekil 6.1'de kurumsal ağlar için genel ağ topolojisi gösterilmektedir. Veri toplama sunucusu bu yapıya eklenerek veriler toplanmıştır.



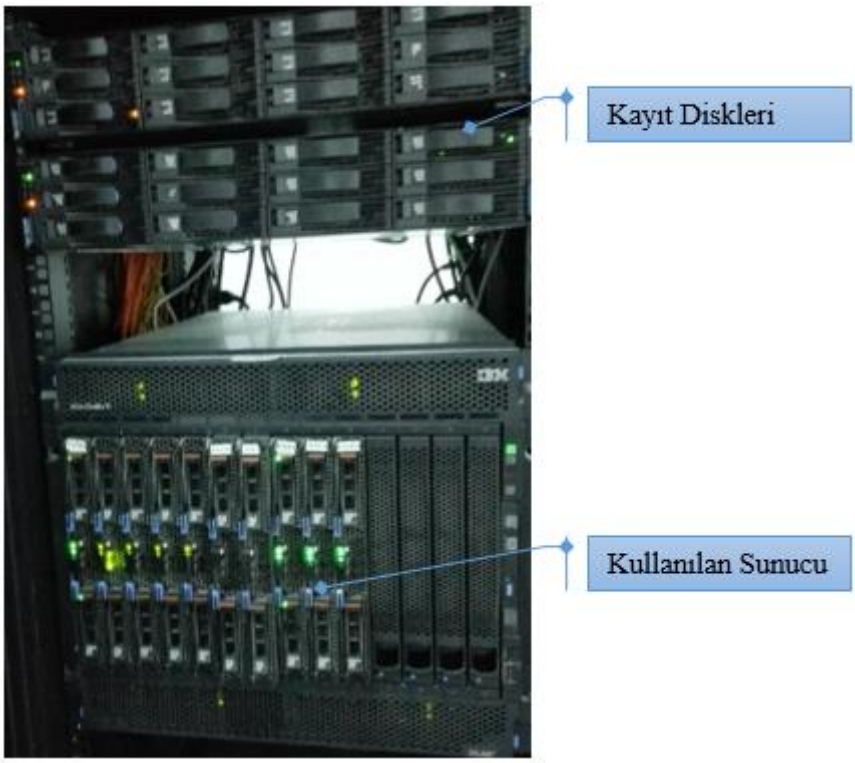
Şekil 6.1. Fırat Üniversitesi ağı üzerindeki verilerin toplanması amacıyla hazırlanan sistem

Kurumsal ağlar için ağ diyagramı olarak düşünebileceğimiz Şekil 6.1'de kullanıcılar bağlı oldukları omurga veya kenar anahtarlama cihazları üzerinden güvenlik duvarı ve yönlendirici cihazlarını geçerek internete çıkmaktadırlar. Şekil 6.2'de kullanılan sisteme ait çalışma ortamı gösterilmektedir.



Şekil 6.2. Fırat Üniversitesi yönlendirici, güvenlik duvarı ve omurga anahtarlama cihazı

Güvenlik duvarı ile omurga cihaz arasında kullanılan veri taşıma kablosunun bağlı olduğu yuva üzerinden geçen veri yine aynı omurga üzerindeki başka bir yuvaya port aynalama adı verilen yöntemle aktarılmıştır. Aktarılan yuva üzerindeki veriler bir sunucu bilgisayara aktarılarak veriler toplanmıştır.



Şekil 6.3. Fırat Üniversitesi ağında verilerin toplanması amacıyla kullanılan sunucu bilgisayar

Verilerin toplanabilmesi ve makine öğrenmesi teknikleriyle analiz edilebilmesi amacıyla depolama alanı ve donanım gücü oldukça yüksek olan IBM Marka BladeCenter HS22 sistem modeline sahip bir sunucu kullanılmıştır. Kullanılan sunucu Şekil 6.3'de verilen şase üzerindeki dokuzuncu yuvadaki bilgisayardır. Bu sunucu bilgisayara ait teknik özellikler şu şekildedir:

- 2 adet işlemciye sahiptir. Her işlemci Intel Xeon X5570 2.93 GHz 4 çekirdeklidir.
- 96 GB RAM donanımına sahiptir (Kullanılabilir değeri 92,7 GB).
- 110 GB toplam sanal hafızaya sahiptir (Kullanılabilir değeri 107 GB).
- 2 adet sabit diske sahiptir. Her sabit disk 300 GB boyutundadır.
- 2 adet 1 GB ethernet portuna sahiptir

Veri toplamak ve analiz etmek için kullanılan sunucu bilgisayarın ethernet portlarından birisi omurga anahtarlama cihazı üzerinde güvenlik duvarı üzerinden geçen portun, port aynalama yöntemiyle aktarıldığı yuvaya Cat-6 kablo ile bağlanmıştır. Böylece bu sunucu bilgisayarın Ethernet portu üzerinde dinleme yapılmış ve veriler paket kayıt dosyaları halinde disk üzerinde saklanmıştır. Verilerin kaydedilmesi amacıyla açık kaynak

kodlu tcpdump [171] ve yine lisans gerektirmeden kullanılabilen wireshark [172] yazılımlarından faydalanılmıştır.

Kayıt altına alınan paket kayıt dosyalarının analiz edilebilmesi amacıyla açık kaynak kodlu fullstats [173] ve NetMATE [174] yazılımlarından faydalanılmıştır. Bu yazılımlar ile veriler içerisindeki öz nitelikler alınmıştır. Bu yazılımlar aynı zamanda CAIDA veri tabanından alınan işlenmemiş veriler üzerinde de uygulanarak öznitelikler çıkarılarak incelenmiştir.

Bu tez çalışmasında kullanılan veri alma yöntemleri içerisinden Moore ve arkadaşlarının kullanmış olduğu Cambridge Üniversitesi kampüsünden alınan verilerin modellenerek makine öğrenmesi teknikleri ile sınıflandırma başarımının karşılaştırılmasına karar verilmiştir. Bu seçimdeki en önemli etken daha önce bu veriler kullanılarak yapılan çalışmalar ile tez çalışmasında kullanılan yöntemlerin karşılaştırma yapılabilmesini sağlamaktır. Önemli etkenlerden birisi de farklı sınıflara ait akışlara sahip olan verilerin kullanılmasıdır. Ayrıca bu tez çalışmasının ana amacı olan ağ üzerindeki verilerin doğru sınıflandırılabilmesi amacıyla başarılı bir makine öğrenme tekniğinin uygulanması için herkesin ulaşabileceği, var olan verilerin kullanılması daha güvenli sonuçlar alınmasına zemin hazırlayacaktır.

Veri setinin işlenebilir hale getirilebilmesi amacıyla Bölüm 3.5.3'de anlatıldığı gibi ihtiyaç duyulması halinde verinin özetlenmesi, temizlenmesi, dönüştürülmesi, indirgenmesi ve ayrıklaştırılması gerekmektedir. Kullanılan veri setinde 324.277 adet veri akışı bulunmaktadır. Bu veriler 13 sınıfa ayrılmıştır. Bu sınıflar web, veri tabanı, e-posta, P2P, atak, oyun, interaktif, servis, medya, ftp kontrol, ftp pasif ve ftp veri ve yönetim şeklindedir Bu sınıfların tanımlanması amacıyla 12 öz nitelik değeri kullanılmıştır. Tablo 6.1'de seçilen öz nitelikler ve açıklamaları gösterilmektedir.

Tablo 6.1. Veri setinde kullanılan öz nitelikler ve açıklamaları

Öz Nitelik Numarası	Açıklaması
1	Sunucu portu
2	İstemci portu
3	İstemciden sunucuya doğru TCP veri yükünün en azından 1 byte kullanımının olduğu tüm paketlerin sayısı
4	İstemciden sunucuya doğru tüm paketlerin sayısı
5	Sunucudan istemciye doğru tüm paketlerin sayısı
6	İstemciden sunucuya doğru bağlantı ömrü boyunca gözlenen en küçük parça boyutu
7	Sunucudan istemciye doğru bağlantının ömrü boyunca gözlenen ortalama boyutu
8	İstemciden sunucuya doğru ilk veride gönderilen toplam byte sayısı
9	Sunucudan istemciye doğru ilk veride gönderilen toplam byte sayısı
10	İstemciden sunucuya doğru RTT örneklerinin toplam sayısı
11	İstemciden sunucuya doğru IP paketlerindeki toplam byte miktarının ortalama değeri
12	Sunucudan istemciye doğru ethernet paketindeki byte farkı

Kullanılan sınıflardan bazıları mevcut veri setindeki akışlar içerisinden çıkarılamadığı için hiç kullanılmamıştır. Bazı sınıflar ise çok az kullanılmıştır. Bu sınıflar elenerek en fazla kullanılan 7 sınıf veri analizi için seçilmiştir. FTP ile ilgili sınıflar bağlantı sınıfı altında toplanmıştır. Kullanılan sınıflar ve açıklamaları Tablo 6.2'de gösterilmektedir.

Tablo 6.2. Veri setinde kullanılan sınıflar ve açıklamaları

No	Sınıf Adı	Açıklama
1	Atak	ddos
2	P2P	Emule, bittorrent
3	E-Posta	POP3, SMTP
4	Web	http, https
5	Servis	DNS
6	Bağlantı	FTP, SSH
7	Veri tabanı	Mysql, mssql

Veri setinde tekrar eden veri olup olmadığı kontrol edilerek, tekrar eden veriler veri setinden çıkarılmıştır. Baskın bir sınıfın olmaması amacıyla her sınıftan 1000 adet akış veri setinden rastgele ve tekrarlanmayacak şekilde seçilerek alınmıştır. Toplam 7000 adet veri makine öğrenmesi için kullanılacak eğitim seti için seçilmiştir. Geriye kalan verilerden yine rastgele ve tekrarlanmayacak şekilde her bir sınıf için 750 adet toplam 5250 adet veri test için kullanılmak üzere seçilmiştir.

Verilerin normalizasyonu için Denklem 3.1'de verilen ve aynı zamanda birlik-tabanlı normalizasyon adı da verilen denklem kullanılmıştır. Özellik ölçeklendirme ile bu şekilde kullanılan veriler 0 ile 1 arasında değişen değerlere dönüştürülerek makine öğrenmesi algoritmaları için modellenen sisteme verilmiştir. Böylece bazı verilerin çok yüksek bazı verilerin çok düşük değerlerde olması sebebiyle oluşabilecek sınıflandırma problemlerinin önüne geçilmesi ve daha doğru sınıflandırma yapılması amaçlanmıştır.

6.3. Modelleme

Bu tez çalışmasının amacı olan ağ üzerindeki trafiğin sınıflandırılması için en doğru sonucun alınabilmesi amacıyla bu alanda gerçekleştirilmiş olan daha önceki çalışmalarda başarılı sonuçlar alınan modeller ile daha önce kullanılmayan modeller oluşturularak bir birleriyle karşılaştırılmıştır. Veri seti için seçilen eğitim ve test verileri makine öğrenmesi algoritmalarına aynı şekilde verilerek kıyaslanmıştır.

Ağ trafiğinin sınıflandırılmasında daha önce yapılan çalışmalar neticesinden en başarılı sınıflandırmalar olarak göze çarpan NB ve DVM algoritmaları veri setlerine uygulanmıştır. Daha önce ağ sınıflandırılması için yapılan çalışmalarda kullanılmamış olan yeni gelişen bir sınıflandırma algoritması olarak kabul edebileceğimiz UÖM algoritmaları da bu veri setlerine uygulanarak veriler modellenmiştir. Temel bir sınıflandırma modeli olması dolayısıyla YSA algoritması da sınıflandırma yapmak için kullanılmış ve diğer sınıflandırıcılar ile karşılaştırılmıştır. Algoritmaların çalışma yöntemleri Bölüm 3.5.4'de ayrıntılı olarak ele alınmıştır. Aynı zamanda uç öğrenme algoritmasının farklı bir yaklaşımı ÇTUÖM algoritması ile veriler analiz edilmiştir. Uç öğrenme algoritmaları ile ilgili ayrıntılı bilgi Bölüm 4 de verilmiştir.

Algoritmalara ait performans değerlendirme ölçüleri için bölüm 3.5.5'de ayrıntıları verilen performans metriklerinden faydalanılmıştır. En etkili değerlendirme kıstası olarak doğruluk değeri (DOD) kullanılmıştır. Sınıflandırıcıların performansları karşılaştırılırken doğruluk değerinin yanında duyarlılık, belirleyicilik, kesinlik, F-Ölçüsü gibi performans metriklerinden faydalanılmıştır.

Algoritmaların çalıştırılması için hazırlanan kodlar MATLAB 2015a [175] yazılımı ile hazırlanmıştır. NB ve DVM'ye ait kodlar aynı zamanda WEKA [176] yazılımı ile test edilmiştir. Grafiklerin oluşturulmasında algoritmaların hazırlandığı yazılım ile birlikte Microsoft Excel 2016 programından faydalanılmıştır.

6.4. Bulgular

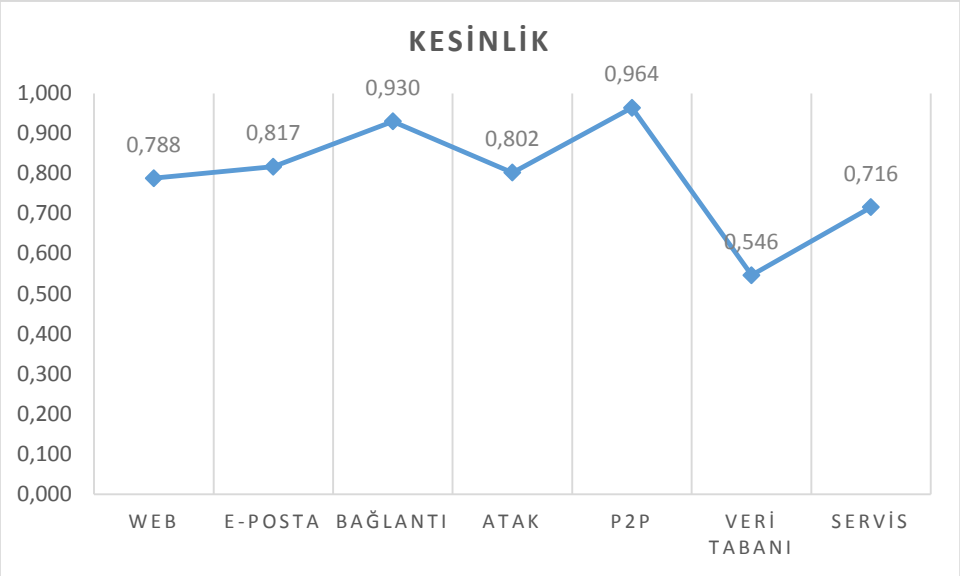
Bu bölümde veri setine uygulanan makine öğrenmesi algoritmalarından alınan sonuçlar performans değerlendirme kıstaslarına göre karşılaştırılmıştır. Bu amaçla YSA, NB, DVM, klasik UÖM ve ÇTUÖM algoritmaları ayrı ayrı verilere uygulanmıştır. Sonuçların karşılaştırılabilmesi amacıyla aynı eğitim ve test verileri makine öğrenmesi algoritmalarına verilmiştir.

Veriler analiz edilirken sınıflandırma algoritmasının sonucunda elde edilen doğru pozitif (D_p), doğru negatif (D_n), yanlış pozitif (Y_p) ve yanlış negatif (Y_n) değerleri kullanılarak; ortalama doğruluk değeri (DOD_{ort}), F-Ölçüsü (F), Duyarlılık (DUY), Belirleyicilik (BEL) metrikleri göz önüne alınmıştır. Analizler yapılırken sınıflandırıcı algoritması test verilerinin üzerinde on kez çalıştırılarak çıkan sonuçların ortalaması alınmıştır.

6.4.1. Yapay Sinir Ağları ile Elde Edilen Bulgular

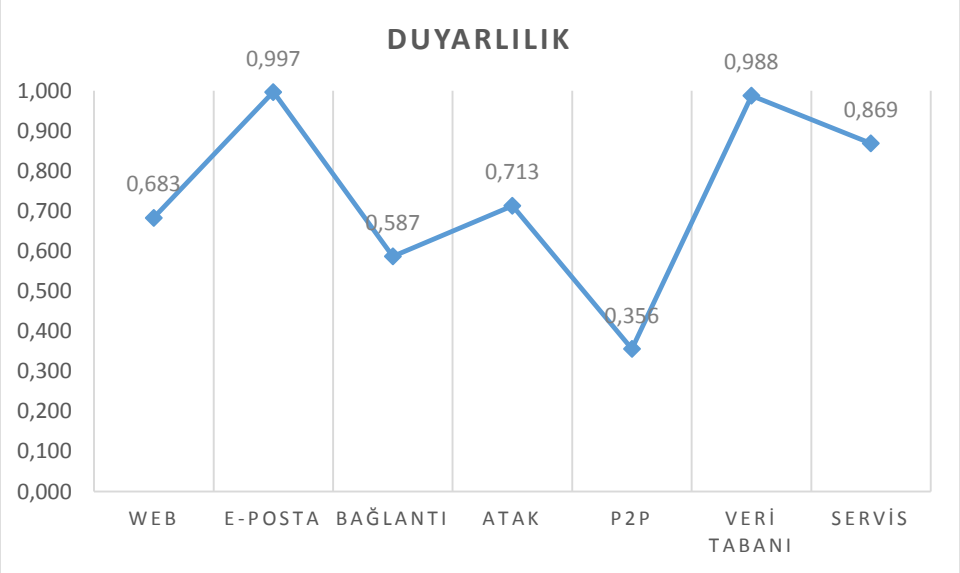
YSA sınıflandırıcısı ile yapılan sınıflandırma ile test verilerinden 3895 tanesi doğru, 1355 tanesi yanlış sınıflandırılmıştır. Sınıflandırıcının ortalama çalışma süresi 24 saniyedir. Ortalama doğruluk değeri (DOD_{ort}) yüzdesi % 74,19 olarak bulunmuştur. Sınıflar içerisinde en yüksek doğru sınıflandırma değeri %99.70 ile E-Posta sınıfına aittir. En düşük doğru sınıflandırma yüzdesi ise %35,60 ile P2P sınıfına aittir.

Pozitif kestirim oranı veya kesinlik olarak ifade edilen değerlerin YSA ile yapılan sınıflandırma sonucunda elde edilen değerler Şekil 6.4'de gösterilmektedir.



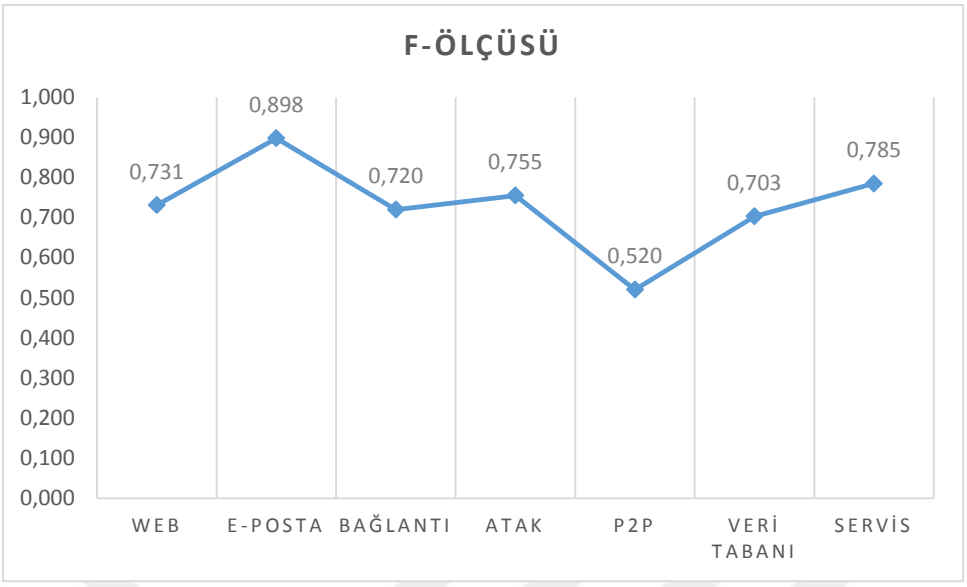
Şekil 6.4. YSA sınıflandırıcına ait kesinlik metrik değerleri

Kesinlik değerlerine göre en yüksek sonuç ile 0,964 ile P2P sınıfına aittir. En düşük sonuç ise 0,546 ile Veri Tabanı sınıfına aittir. Şekil 6.5'de duyarlılık değerleri verilmiştir.



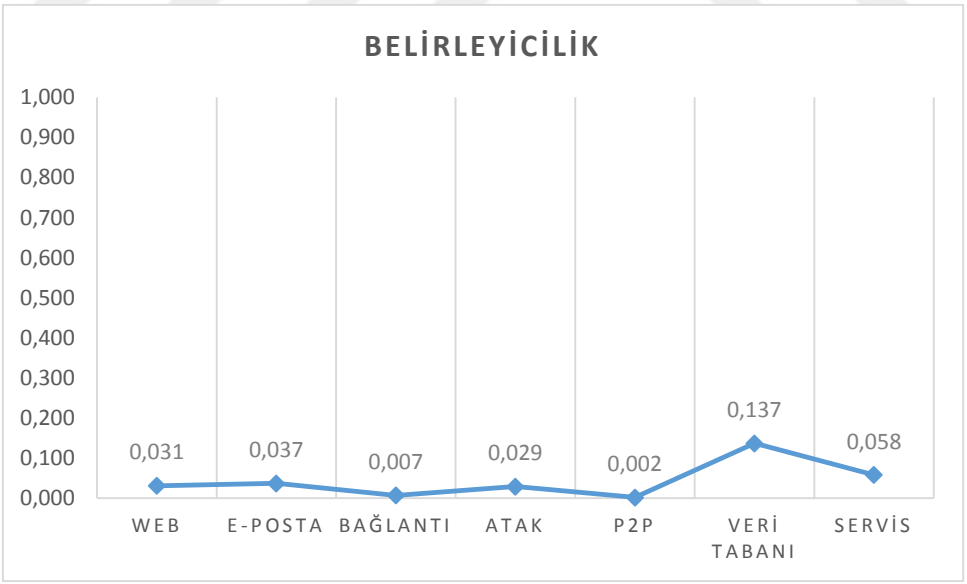
Şekil 6.5. YSA sınıflandırıcına ait duyarlılık metrik değerleri

Duyarlılık değerlerine göre en yüksek sonuç 0,988 ile Veri Tabanı sınıfına aittir. En düşük sonuç ise 0,355 ile P2P sınıfına aittir. F-Ölçüsü metriğine ait değerler Şekil 6.6'da gösterilmiştir.



Şekil 6.6. YSA sınıflandırıcına ait F-Ölçüsü metrik değerleri

F-Ölçüsüne metriği için en yüksek değer 0,898 ile E-Posta sınıfında, en düşük değer ise 0,520 ile P2P sınıfında gözlenmiştir. Belirleyicilik metrik değerlerine ait veriler Şekil 6.7'de gösterilmektedir.



Şekil 6.7. YSA sınıflandırıcına ait belirleyicilik metrik değerleri

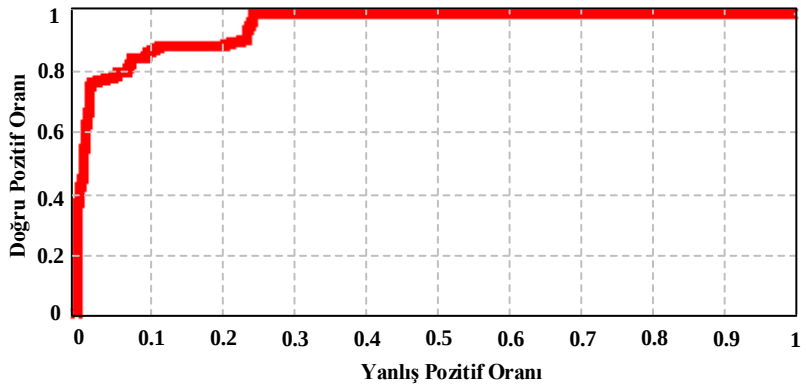
Belirleyicilik metrik değerlerine bakıldığında en düşük değerin 0,002 ile P2P sınıfına ait olduğu görülmektedir. En yüksek değerin ise 0,137 ile Veri Tabanı sınıfına ait olduğu görülmektedir.

YSA sınıflandırıcının test verilerine uygulanması ile elde edilen performans metrik değerleri Tablo 6.3'de gösterilmektedir.

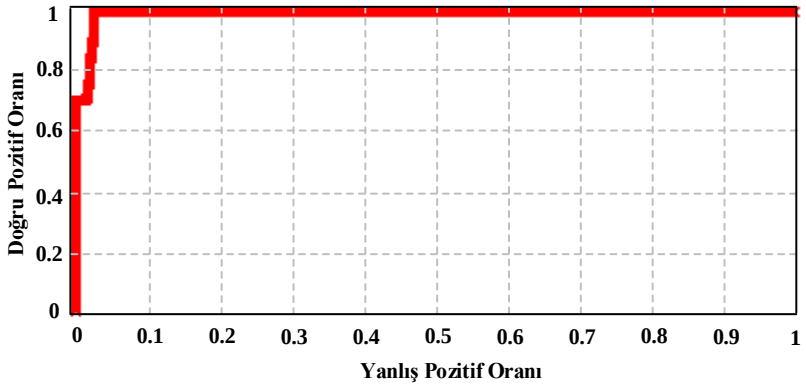
Tablo 6.3. YSA sınıflandırıcısı ile elde edilen performans değerleri

SINIF	Performans Metrik Değerleri			
	Belirleyicilik	Kesinlik	Duyarlılık	F-Ölçüsü
WEB	0,031	0,788	0,683	0,731
E-POSTA	0,037	0,817	0,997	0,898
BAĞLANTI	0,007	0,930	0,587	0,720
ATAK	0,029	0,802	0,713	0,755
P2P	0,002	0,964	0,356	0,520
VERİ TABANI	0,137	0,546	0,988	0,703
SERVİS	0,058	0,716	0,869	0,785
ORTALAMA	0,043	0,795	0,742	0,730

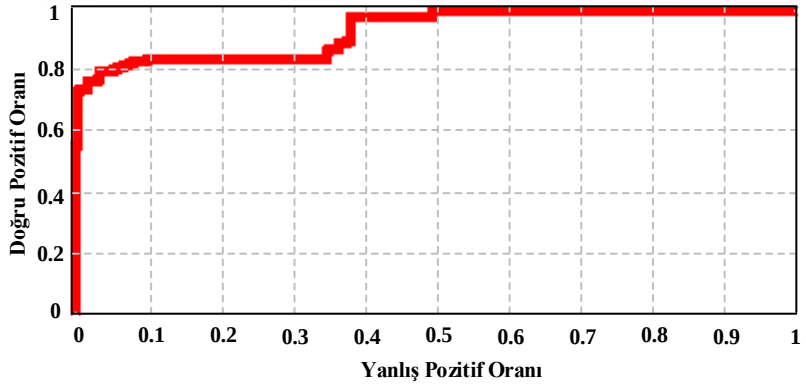
YSA kullanılarak yapılan sınıflandırıcı için her bir sınıfa ait AİK eğri grafikleri Şekil 6.8 ile Şekil 6.14 arasında gösterilmektedir. AİK eğri alanları Web sınıfı için 0,956; E-Posta sınıfı için 0,993; Bağlantı sınıfı için 0,929; Atak sınıfı için 0,908; P2P sınıfı için 0,941; Veri Tabanı sınıfı için 0,994; Servis sınıfı için 0,947 olarak bulunmuştur. AİK eğri alanının ortalama değeri ise 0,953 çıkmıştır.



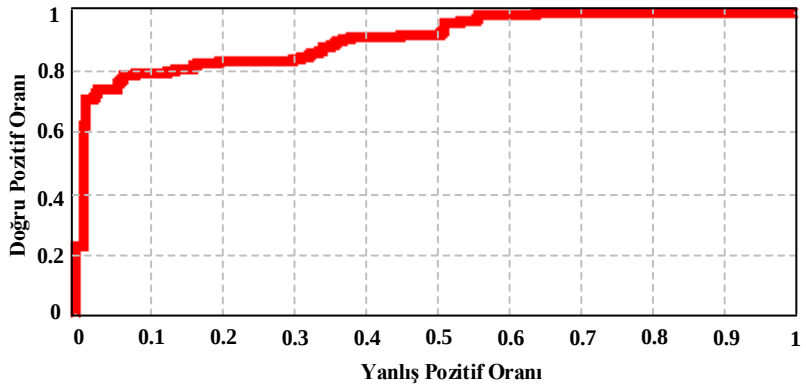
Şekil 6.8. YSA sınıflandırıcısının WEB sınıfına ait AİK eğrisi



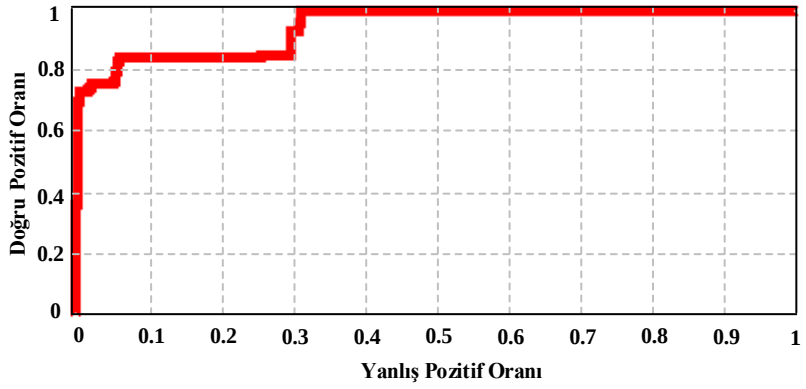
Şekil 6.9. YSA sınıflandırıcısının E-POSTA sınıfına ait AİK eğrisi



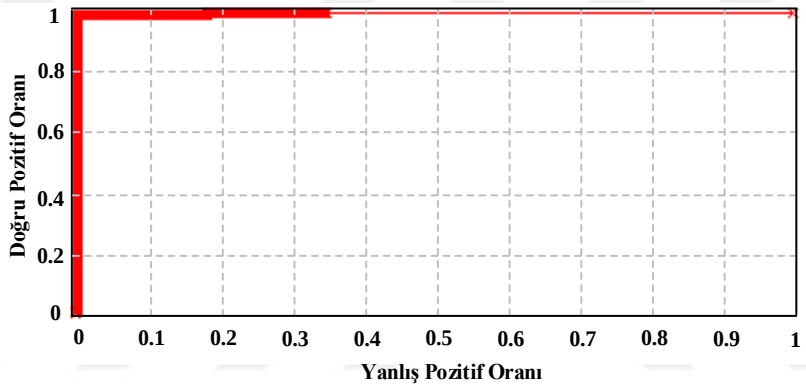
Şekil 6.10. YSA sınıflandırıcısının BAĞLANTI sınıfına ait AİK eğrisi



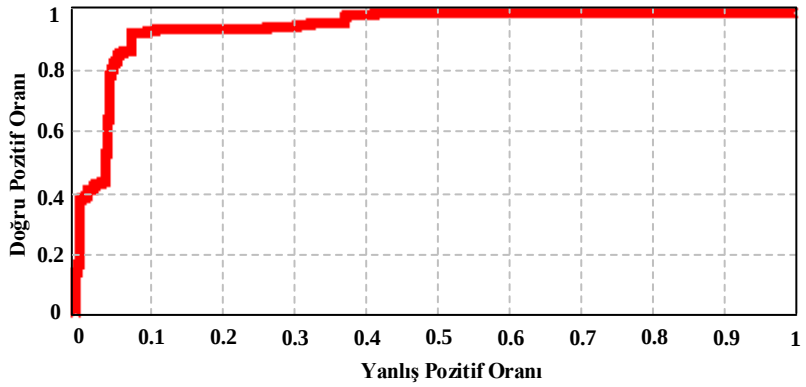
Şekil 6.11. YSA sınıflandırıcısının ATAK sınıfına ait AİK eğrisi



Şekil 6.12. YSA sınıflandırıcısının P2P sınıfına ait AİK eğrisi



Şekil 6.13. YSA sınıflandırıcısının VERİ TABANI sınıfına ait AİK eğrisi

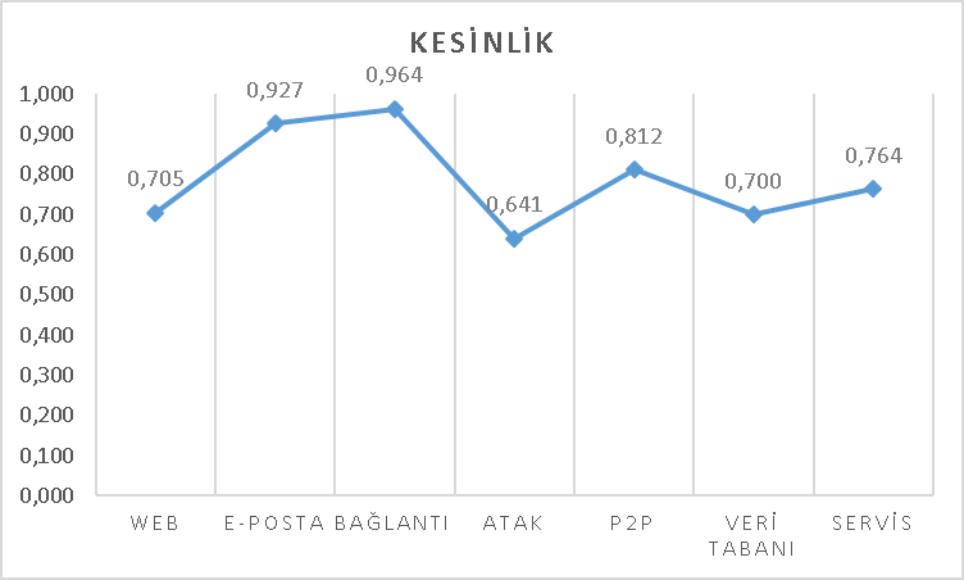


Şekil 6.14. YSA sınıflandırıcısının SERVİS sınıfına ait AİK eğrisi

6.4.2. Naive Bayes Sınıflandırıcı ile Elde Edilen Bulgular

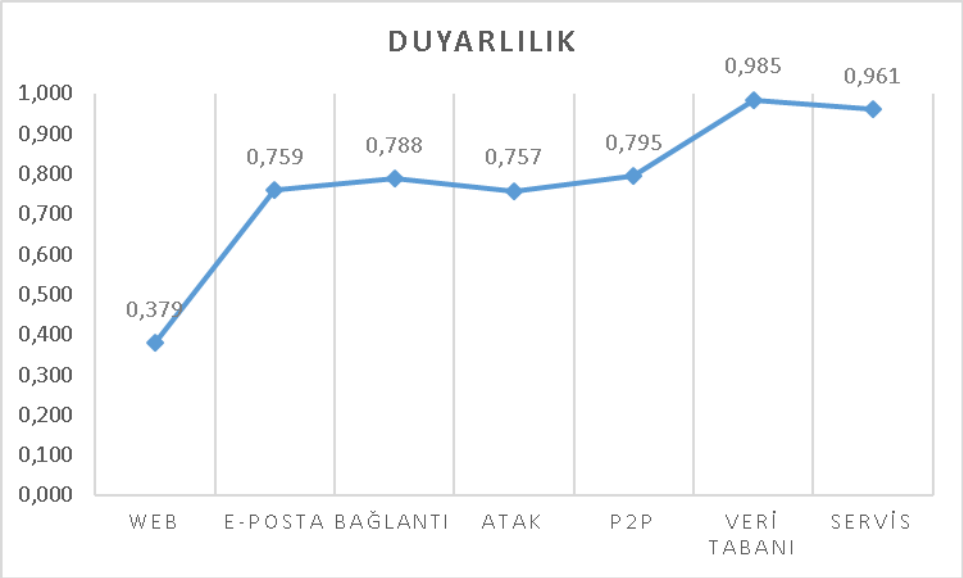
NB sınıflandırıcısı ile yapılan sınıflandırma ile test verilerinden 4068 tanesi doğru, 1182 tanesi yanlış sınıflandırılmıştır. Sınıflandırıcının ortalama çalışma süresi 120 milisaniyedir. Ortalama doğruluk değeri (DOD_{ort}) yüzdesi % 77,49 olarak bulunmuştur. Sınıflar içerisinde en yüksek doğru sınıflandırma değeri %98,50 ile veri tabanı sınıfına aittir. En düşük doğru sınıflandırma yüzdesi ise %37,90 ile Web sınıfına aittir.

Yapılan sınıflandırma çalışmalarında sınıflandırıcının sınıflandırmayı ne kadar doğru yaptığının yanında Kesinlik ve Duyarlılık değerleri de göz önüne alınarak karşılaştırma yapılmaktadır. Sınıflandırma metriklerinden kesinlik veya doğru kestirim oranı olarak verilen metrik değerleri Şekil 6.15'de gösterilmiştir.



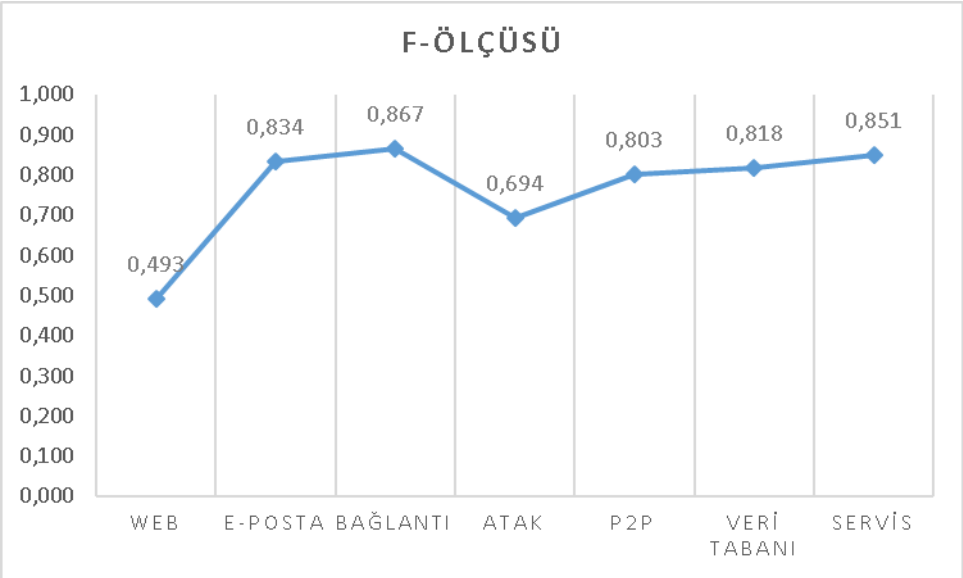
Şekil 6.15. NB sınıflandırıcına ait kesinlik metrik değerleri

Kesinlik değerlerine göre en yüksek değer ile 0,964 ile Bağlantı sınıfına aittir. En düşük değer ise 0,641 ile Atak sınıfına aittir. Şekil 6.16'da Duyarlılık değerleri verilmiştir.



Şekil 6.16. NB sınıflandırıcına ait duyarlılık metrik değerleri

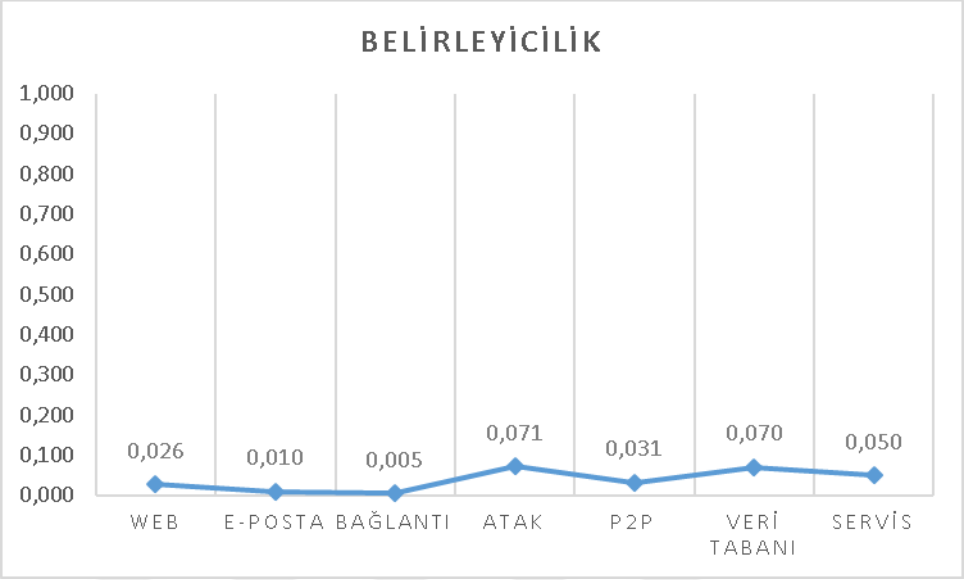
Duyarlılık değerlerine göre en yüksek sonuç 0,985 ile Veri Tabanı sınıfına aittir. En düşük sonuç ise 0,379 ile Web sınıfına aittir. Kesinlik ve Duyarlılık metrikleri beraber kullanılarak Denklem 3.23'de gösterildiği gibi F-Ölçüsünün bulunmasında kullanılmaktadır. F-Ölçüsüne ait metrik değerleri Şekil 6.17'de gösterilmiştir.



Şekil 6.17. NB sınıflandırıcına ait F-Ölçüsü metrik değerleri

F-Ölçüsüne ait sınıflandırma değerleri E-Posta, Bağlantı, P2P, Veri Tabanı ve Servis sınıflarında bir birlerine yakın olarak 0,8 değerinin üzerinde çıkmıştır. Web sınıfında ise en

düşük F-Ölçüsü değeri olarak 0.493 görülmektedir. Bir diğer sınıflandırma metriği olarak Belirleyicilik metrik değerleri hesaplanmıştır. Bu değerler Şekil 6.18'de gösterilmektedir.



Şekil 6.18. NB sınıflandırıcına ait belirleyicilik metrik değerleri

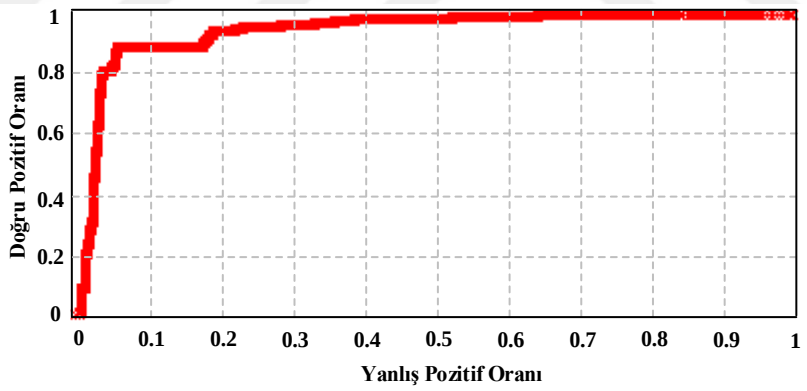
Negatif sınıf etiketlerini tahmin etmeye çalışan sınıflandırıcının başarısı olarak kabul edebileceğimiz Belirleyicilik metrik değerleri analiz edildiğinde en düşük değerin 0,005 ile Bağlantı sınıfına, en yüksek değerlerin ise 0,071 ile Atak ve 0,07 ile Veri tabanı sınıflarına ait oldukları görülmüştür.

NB sınıflandırıcının test verilerine uygulanması ile elde edilen performans metrik değerleri Tablo 6.4'de gösterilmektedir.

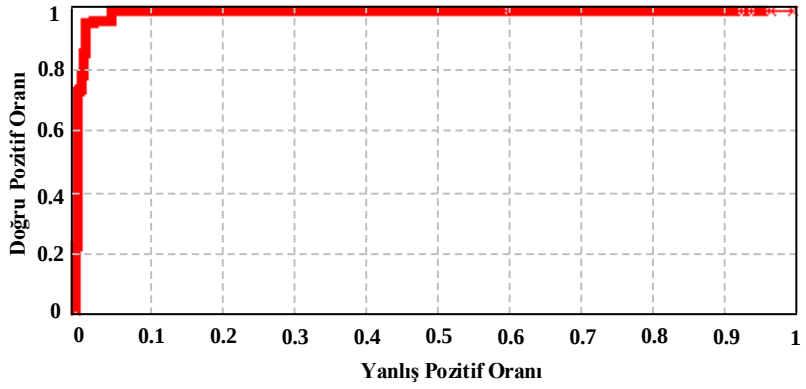
Tablo 6.4. NB sınıflandırıcısı ile elde edilen performans değerleri

SINIF	Performans Metrik Değerleri			
	Belirleyicilik	Kesinlik	Duyarlılık	F-Ölçüsü
WEB	0,026	0,705	0,379	0,493
E-POSTA	0,010	0,927	0,759	0,834
BAĞLANTI	0,005	0,964	0,788	0,867
ATAK	0,071	0,641	0,757	0,694
P2P	0,031	0,812	0,795	0,803
VERİ TABANI	0,070	0,700	0,985	0,818
SERVİS	0,050	0,764	0,961	0,851
ORTALAMA	0,038	0,788	0,775	0,766

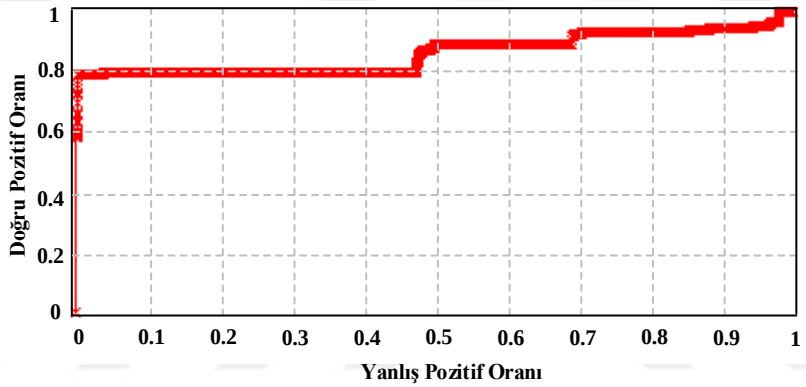
NB yaklaşımı kullanılarak yapılan sınıflandırıcı için her bir sınıfa ait AİK eğri grafikleri Şekil 6.19 ile Şekil 6.25 arasında gösterilmektedir. Grafiklerde yatay eksen yanlış pozitif oranını, dikey eksen ise doğru pozitif oranını göstermektedir. AİK eğri alanları Web sınıfı için 0,942; E-Posta sınıfı için 0,991; Bağlantı sınıfı için 0,857; Atak sınıfı için 0,917; P2P sınıfı için 0,936; Veri Tabanı sınıfı için 0,979; Servis sınıfı için 0,973 olarak bulunmuştur. AİK eğri alanının ortalama değeri ise 0,942 çıkmıştır.



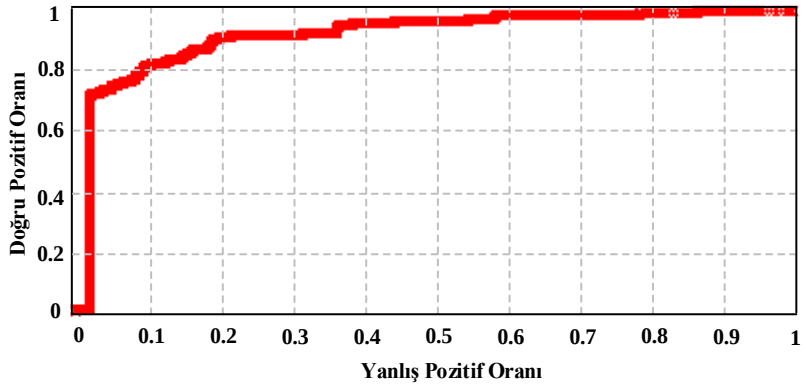
Şekil 6.19. NB sınıflandırıcısının WEB sınıfına ait AİK eğrisi



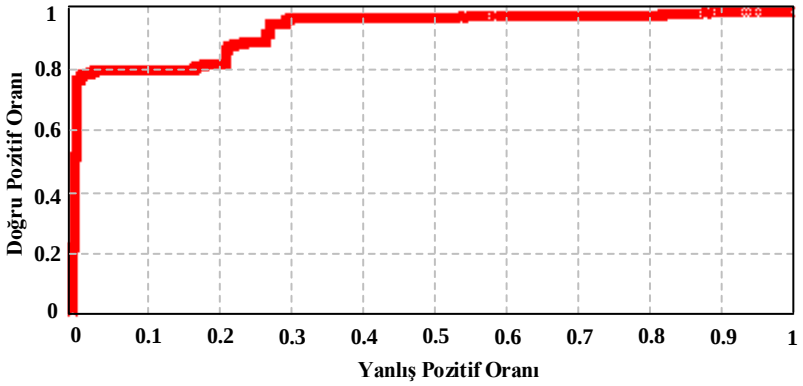
Şekil 6.20. NB sınıflandırıcısının E-POSTA sınıfına ait AİK eğrisi



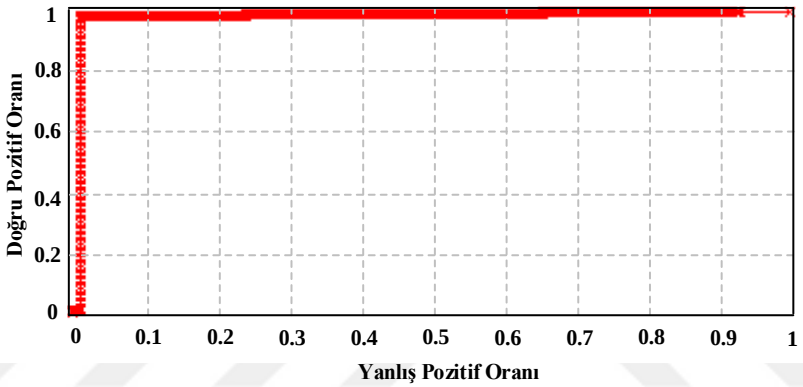
Şekil 6.21. NB sınıflandırıcısının BAĞLANTI sınıfına ait AİK eğrisi



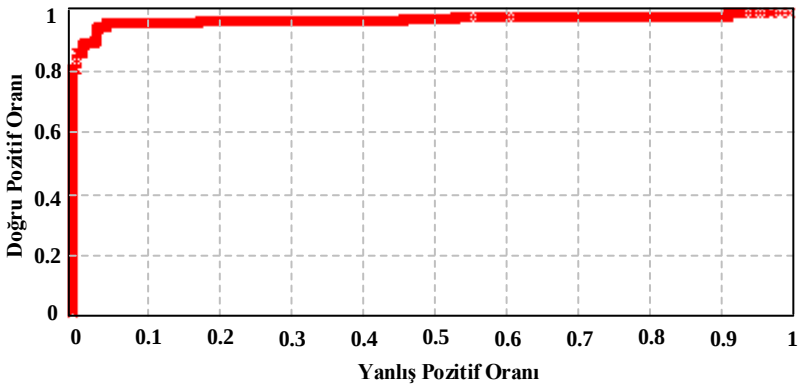
Şekil 6.22. NB sınıflandırıcısının ATAK sınıfına ait AİK eğrisi



Şekil 6.23. NB sınıflandırıcısının P2P sınıfına ait AİK eğrisi



Şekil 6.24. NB sınıflandırıcısının VERİ TABANI sınıfına ait AİK eğrisi

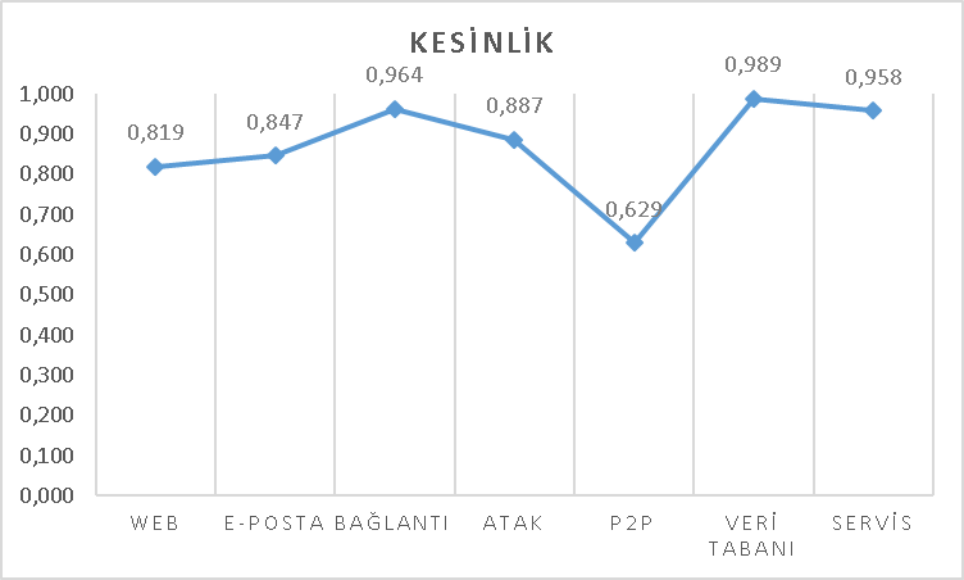


Şekil 6.25. NB sınıflandırıcısının SERVİS sınıfına ait AİK eğrisi

6.4.3. Destek Vektör Makineleri ile Elde Edilen Bulgular

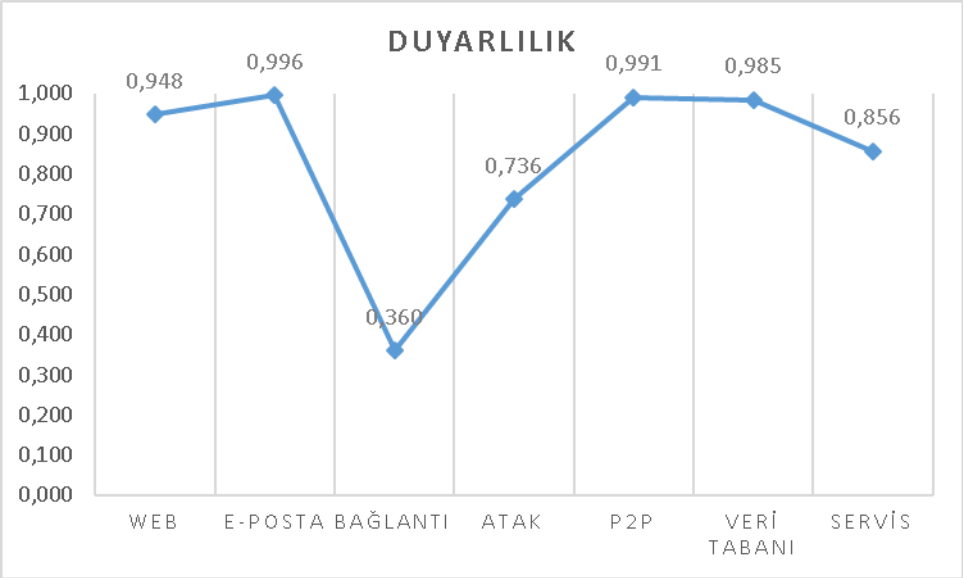
Çok sınıflı DVM sınıflandırıcısı ile yapılan sınıflandırma ile test verilerinden 4404 tanesi doğru, 846 tanesi yanlış sınıflandırılmıştır. Sınıflandırıcının ortalama çalışma süresi 200 milisaniyedir. Ortalama doğruluk değeri (DOD_{ort}) yüzdesi % 83,88 olarak bulunmuştur. Sınıflar içerisinde en yüksek doğru sınıflandırma değeri %99.60 ile E-Posta sınıfına aittir. En düşük doğru sınıflandırma yüzdesi ise %36,00 ile Bağlantı sınıfına aittir.

Pozitif kestirim oranı veya kesinlik olarak ifade edilen değerlerin DVM ile elde edilen sonuçları Şekil 6.26'da gösterilmektedir.



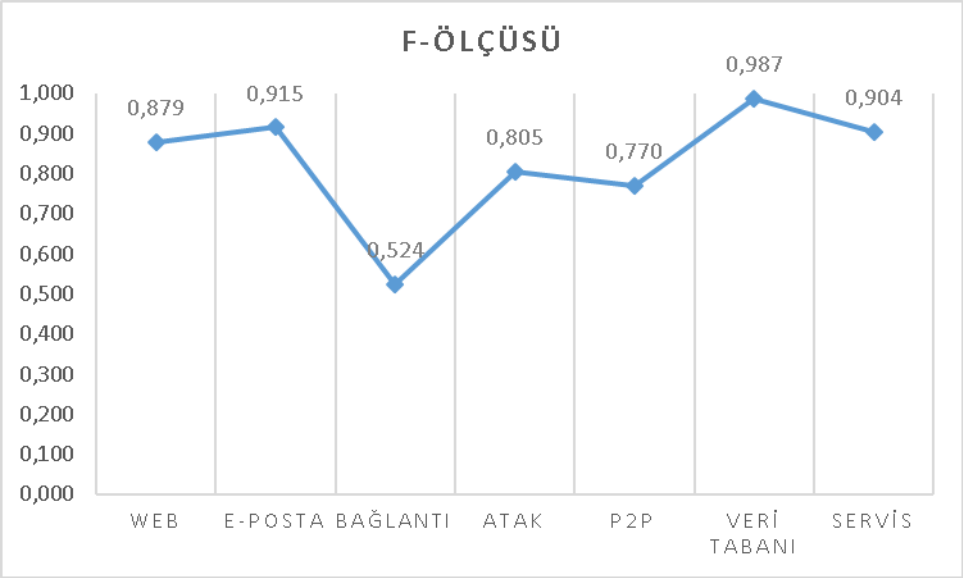
Şekil 6.26. Çok Sınıflı DVM sınıflandırıcına ait Kesinlik metrik değerleri

Kesinlik değerlerine göre en yüksek sonuç ile 0,964 ile Bağlantı sınıfına aittir. En düşük sonuç ise 0,629 ile P2P sınıfına aittir. Şekil 6.27 de Duyarlılık değerleri verilmiştir.



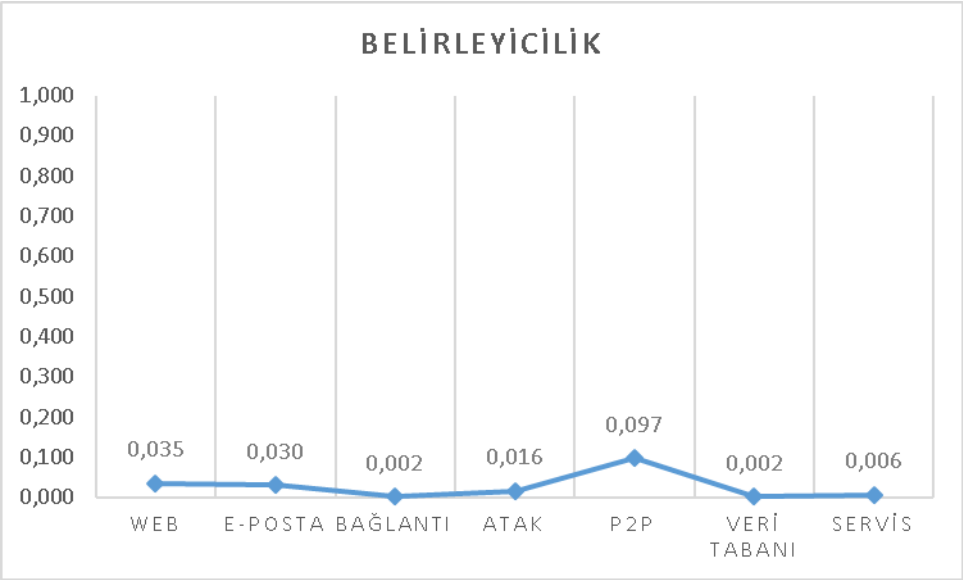
Şekil 6.27. Çok Sınıflı DVM sınıflandırıcına ait Duyarlılık metrik değerleri

Duyarlılık değerlerine göre en yüksek sonuç 0,996 ile E-Posta sınıfına aittir. En düşük sonuç ise 0,360 ile Bağlantı sınıfına aittir. F-Ölçüsüne ait metrik değerleri Şekil 6.28'de gösterilmiştir.



Şekil 6.28. Çok Sınıflı DVM sınıflandırıcına ait F-Ölçüsü metrik değerleri

F-Ölçüsüne ait sınıflandırma için en yüksek değer 0,987 ile Veri Tabanı sınıfında, en düşük değer ise 0,524 ile Bağlantı sınıfında gözlenmiştir. Belirleyicilik metrik değerlerine ait veriler Şekil 6.29'da gösterilmektedir.



Şekil 6.29. Çok Sınıflı DVM sınıflandırıcına ait Belirleyicilik metrik değerleri

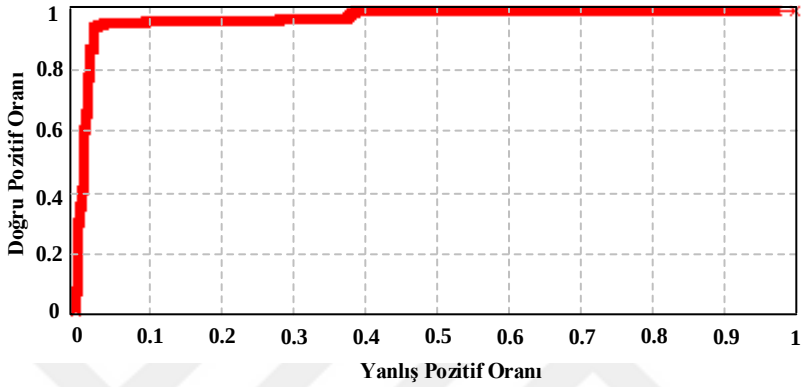
Belirleyicilik metrik değerleri analiz edildiğinde en düşük değerin 0,002 ile Bağlantı ve Veri Tabanı sınıflarına ait olduğu görülmektedir. En yüksek değerin ise 0,097 ile P2P sınıfına ait olduğu görülmektedir.

Çok sınıflı DVM sınıflandırıcının test verilerine uygulanması ile elde edilen performans metrik değerleri Tablo 6.5'de gösterilmektedir.

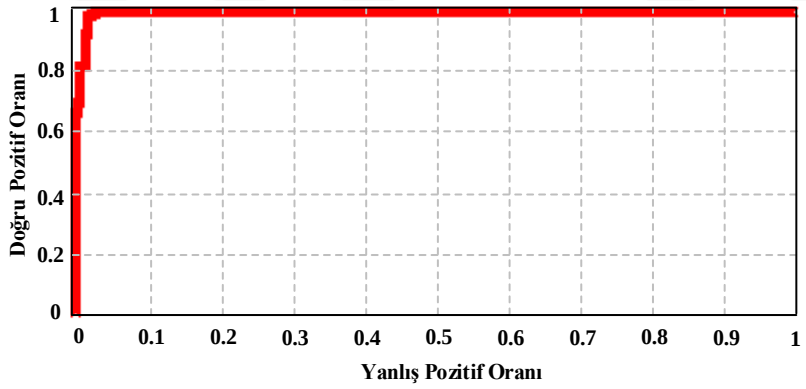
Tablo 6.5. DVM sınıflandırıcısı ile elde edilen performans değerleri

SINIF	Performans Metrik Değerleri			
	Belirleyicilik	Kesinlik	Duyarlılık	F-Ölçüsü
WEB	0,035	0,819	0,948	0,879
E-POSTA	0,030	0,847	0,996	0,915
BAĞLANTI	0,002	0,964	0,360	0,524
ATAK	0,016	0,887	0,736	0,805
P2P	0,097	0,629	0,991	0,770
VERİ TABANI	0,002	0,989	0,985	0,987
SERVİS	0,006	0,958	0,856	0,904
ORTALAMA	0,027	0,870	0,839	0,826

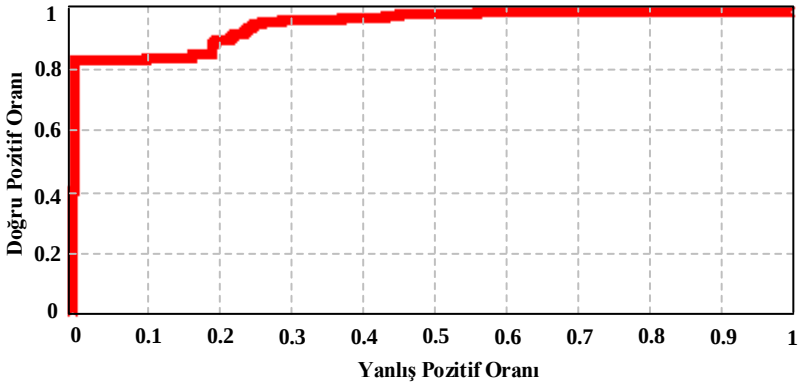
Çok sınıflı DVM ile yapılan sınıflandırıcı için her bir sınıfa ait AİK eğri grafikleri Şekil 6.30 ile Şekil 6.36 arasında gösterilmektedir. Grafiklerde yatay eksen yanlış pozitif oranını, dikey eksen ise doğru pozitif oranını göstermektedir. AİK eğri alanları Web sınıfı için 0,972; E-Posta sınıfı için 0,995; Bağlantı sınıfı için 0,955; Atak sınıfı için 0,872; P2P sınıfı için 0,988; Veri Tabanı sınıfı için 0,992; Servis sınıfı için 0,966 olarak bulunmuştur. AİK eğri alanının ortalama değeri ise 0,963 çıkmıştır.



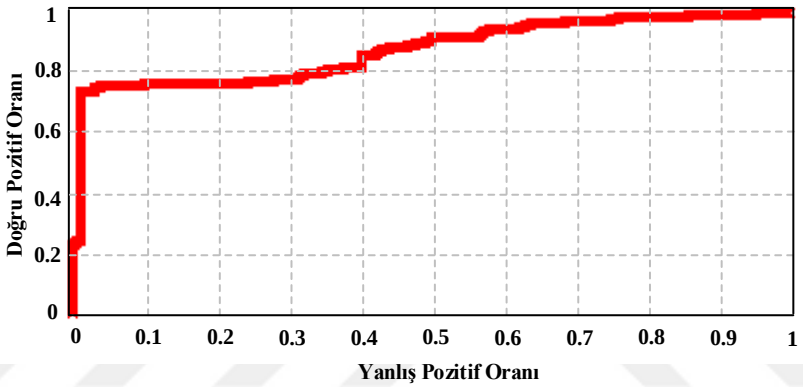
Şekil 6.30. Çok sınıflı DVM sınıflandırıcısının WEB sınıfına ait AİK eğrisi



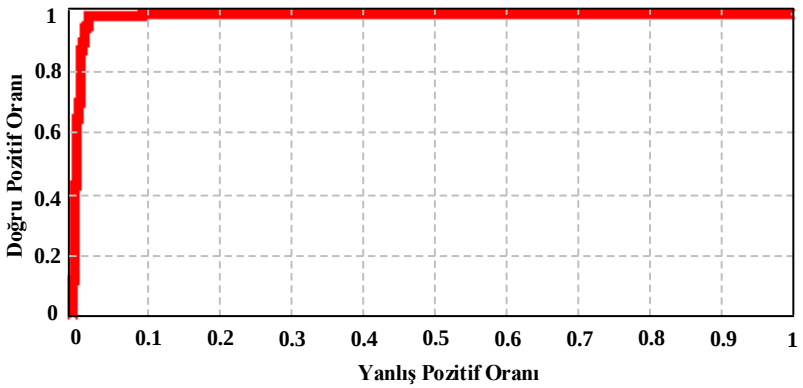
Şekil 6.31. Çok sınıflı DVM sınıflandırıcısının E-POSTA sınıfına ait AİK eğrisi



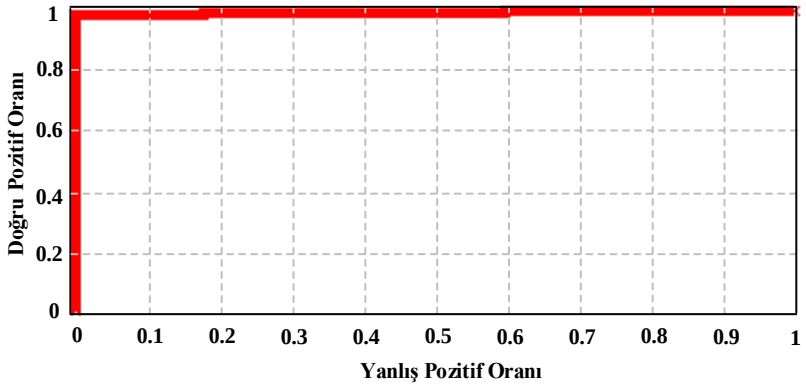
Şekil 6.32. Çok sınıflı DVM sınıflandırıcısının BAĞLANTI sınıfına ait AİK eğrisi



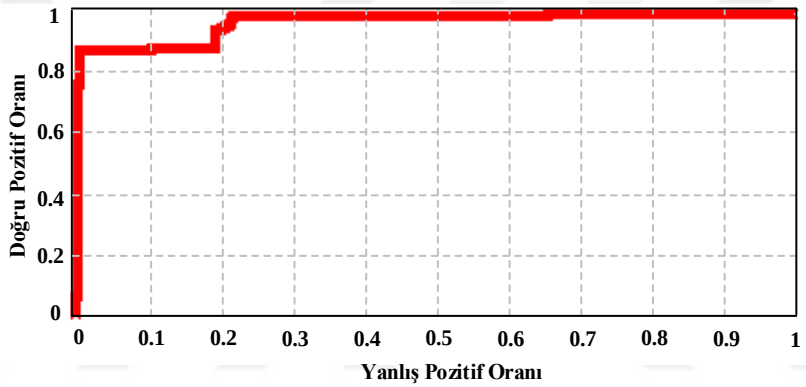
Şekil 6.33. Çok sınıflı DVM sınıflandırıcısının ATAK sınıfına ait AİK eğrisi



Şekil 6.34. Çok sınıflı DVM sınıflandırıcısının P2P sınıfına ait AİK eğrisi



Şekil 6.35. Çok sınıflı DVM sınıflandırıcısının VERİ TABANI sınıfına ait AİK eğrisi

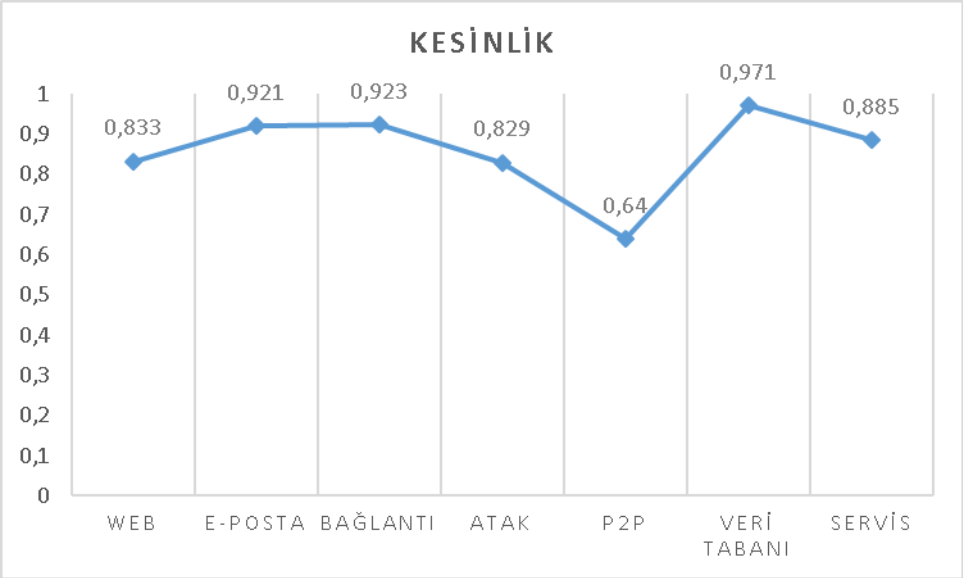


Şekil 6.36. Çok sınıflı DVM sınıflandırıcısının SERVİS sınıfına ait AİK eğrisi

6.4.4. Klasik Uç Öğrenme Makineleri ile Elde Edilen Bulgular

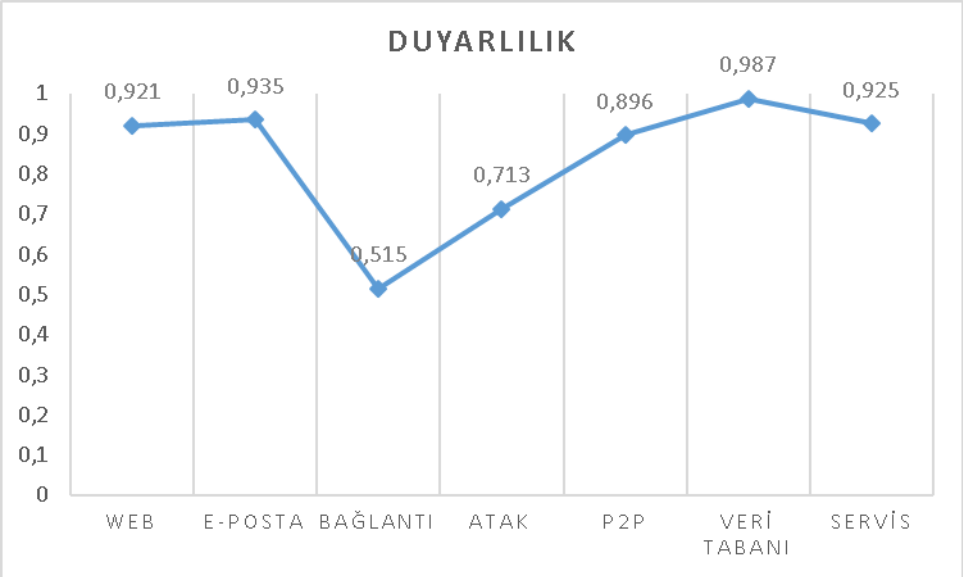
Klasik UÖM'ye verilecek test değerleri için algoritmanın aktivasyon fonksiyonu tanjant sigmoid, gizli katman hücre sayısı ise 40 seçilmiştir. Bu sınıflandırıcı ile yapılan sınıflandırma ile test verilerinden 4419 tanesi doğru, 831 tanesi yanlış sınıflandırılmıştır. Sınıflandırıcının ortalama çalışma süresi 83,22 milisaniyedir. Ortalama doğruluk değeri (DOD_{ort}) yüzdesi % 84,17 olarak bulunmuştur. Sınıflar içerisinde en yüksek doğru sınıflandırma değeri % 98.70 ile Veri Tabanı sınıfına aittir. En düşük doğru sınıflandırma yüzdesi ise % 51,50 ile Bağlantı sınıfına aittir.

Klasik UÖM algoritması sonuçlarına göre kesinlik metriğinin aldığı değerler Şekil 6.37'de gösterilmektedir.



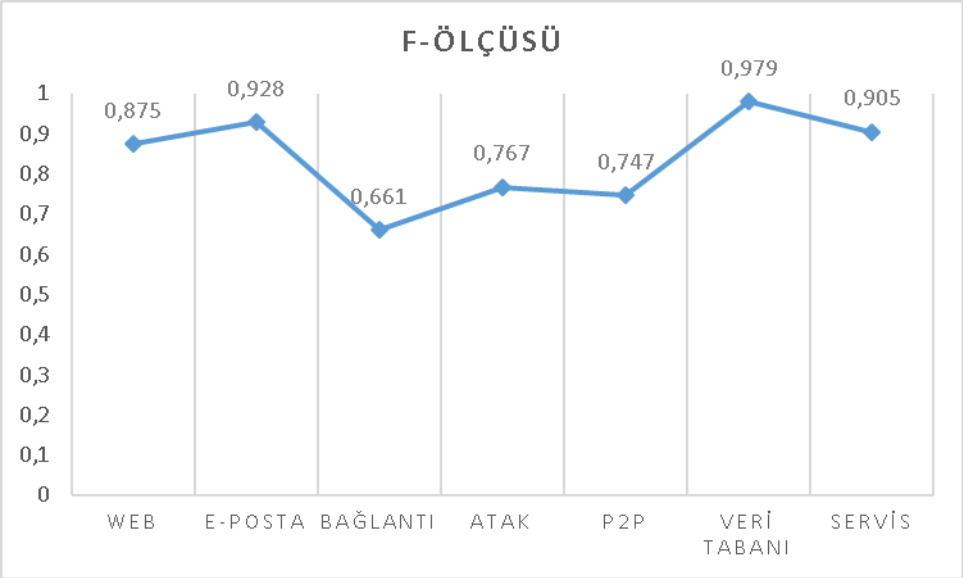
Şekil 6.37. Klasik UÖM sınıflandırıcına ait Kesinlik metrik değerleri

Kesinlik değerlerine göre en yüksek sonuç ile 0,971 ile Veri Tabanı sınıfına aittir. En düşük sonuç ise 0,640 ile P2P sınıfına aittir. Şekil 6.38'de Duyarlılık değerleri verilmiştir.



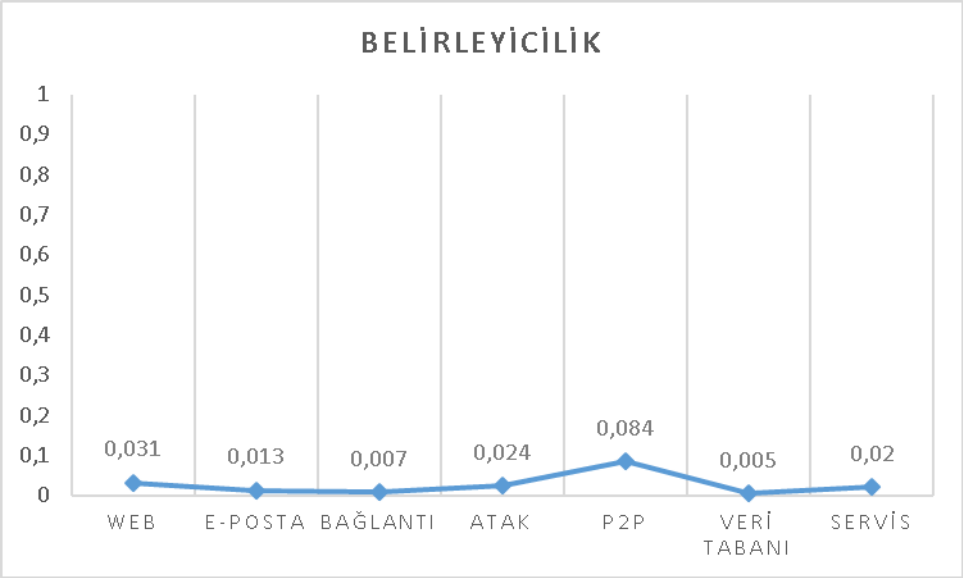
Şekil 6.38. Klasik UÖM sınıflandırıcına ait Duyarlılık metrik değerleri

Duyarlılık değerlerine göre en yüksek sonuç 0,987 ile Veri Tabanı sınıfına aittir. En düşük sonuç ise 0,515 ile Bağlantı sınıfına aittir. F-Ölçüsüne ait metrik değerleri Şekil 6.39'da gösterilmiştir.



Şekil 6.39. Klasik UÖM sınıflandırıcına ait F-Ölçüsü metrik değerleri

F-Ölçüsüne ait sınıflandırma için en yüksek değer 0,979 ile Veri Tabanı sınıfında, en düşük değer ise 0,661 ile Bağlantı sınıfında gözlenmiştir. Belirleyicilik metrik değerlerine ait veriler Şekil 6.40'da gösterilmektedir.



Şekil 6.40. Klasik UÖM sınıflandırıcına ait Belirleyicilik metrik değerleri

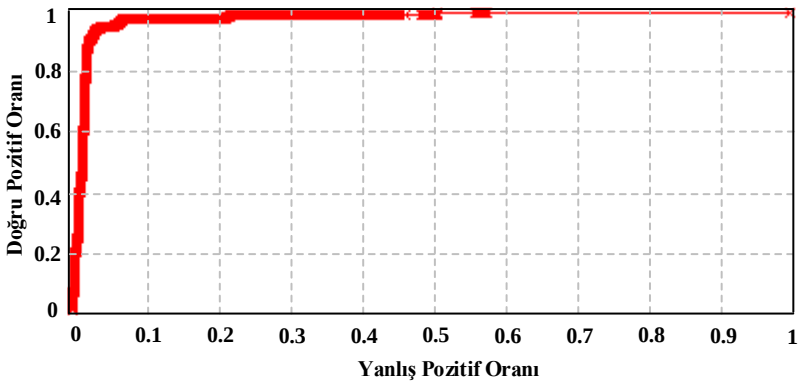
Belirleyicilik metrik değerleri analiz edildiğinde en düşük değerin 0,005 ile Veri Tabanı sınıfında olduğu görülmektedir. En yüksek değerin ise 0,084 ile P2P sınıfına ait olduğu görülmektedir. Tanjant sigmoid aktivasyon fonksiyonu ve 40 adet gizli katman hücre

sayısı kullanılan klasik UÖM sınıflandırıcının test verilerine uygulanması ile elde edilen performans metrik değerleri Tablo 6.6'da gösterilmektedir.

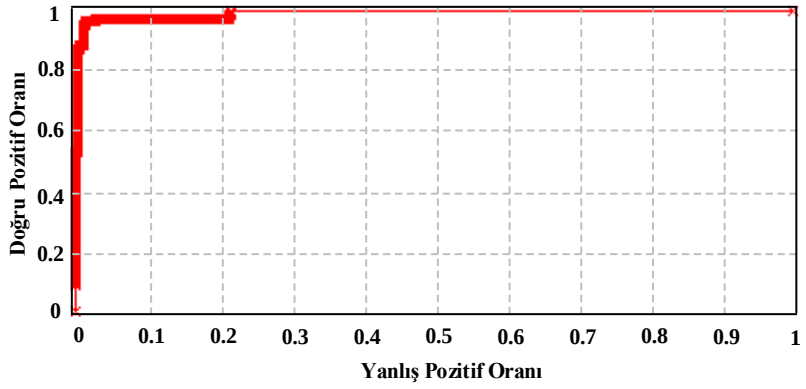
Tablo 6.6. Tanjant sigmoid aktivasyon fonksiyonunun kullanıldığı klasik UÖM sınıflandırıcısı ile elde edilen performans değerleri

SINIF	Performans Metrik Değerleri			
	Belirleyicilik	Kesinlik	Duyarlılık	F-Ölçüsü
WEB	0,031	0,833	0,921	0,875
E-POSTA	0,013	0,921	0,935	0,928
BAĞLANTI	0,007	0,923	0,515	0,661
ATAK	0,024	0,829	0,713	0,767
P2P	0,084	0,640	0,896	0,747
VERİ TABANI	0,005	0,971	0,987	0,979
SERVİS	0,020	0,885	0,925	0,905
ORTALAMA	0,026	0,858	0,842	0,837

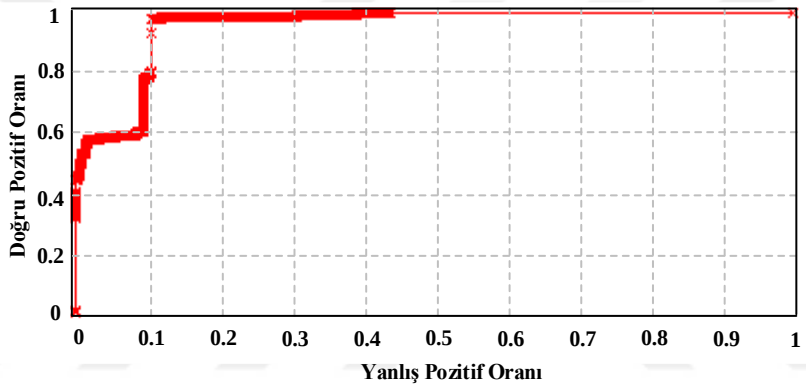
Klasik UÖM yaklaşımı kullanılarak yapılan sınıflandırıcı için her bir sınıfa ait AİK eğri grafikleri Şekil 6.41 ile Şekil 6.47 arasında gösterilmektedir. Grafiklerde yatay eksen yanlış pozitif oranını, dikey eksen ise doğru pozitif oranını göstermektedir. AİK eğri alanları Web sınıfı için 0,975; E-Posta sınıfı için 0,990; Bağlantı sınıfı için 0,951; Atak sınıfı için 0,925; P2P sınıfı için 0,980; Veri Tabanı sınıfı için 0,996; Servis sınıfı için 0,969 olarak bulunmuştur. AİK eğri alanının ortalama değeri ise 0,970 bulunmuştur.



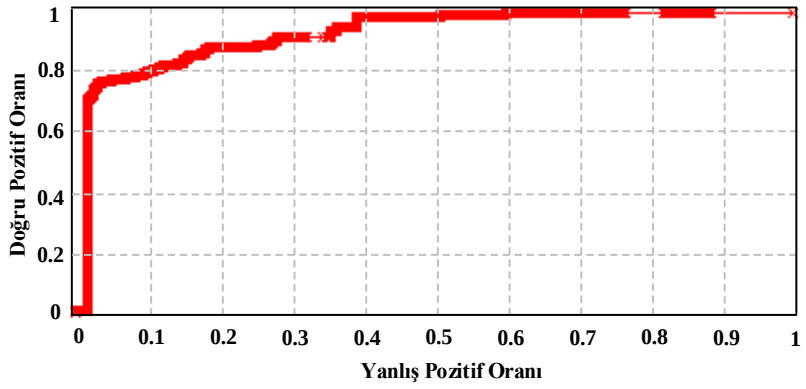
Şekil 6.41. Klasik UÖM sınıflandırıcısının WEB sınıfına ait AİK eğrisi



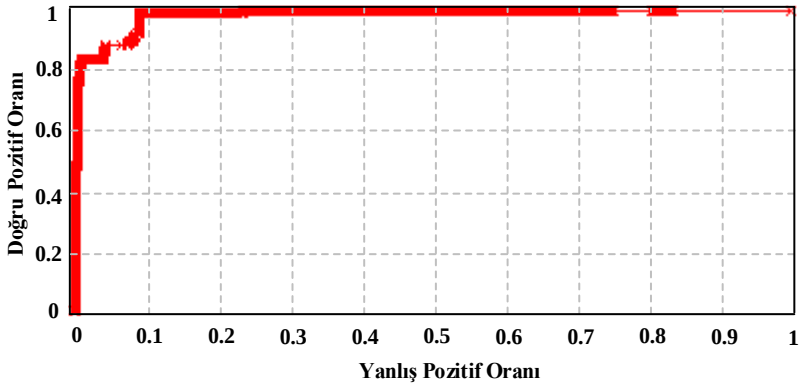
Şekil 6.42. Klasik UÖM sınıflandırıcısının E-POSTA sınıfına ait AİK eğrisi



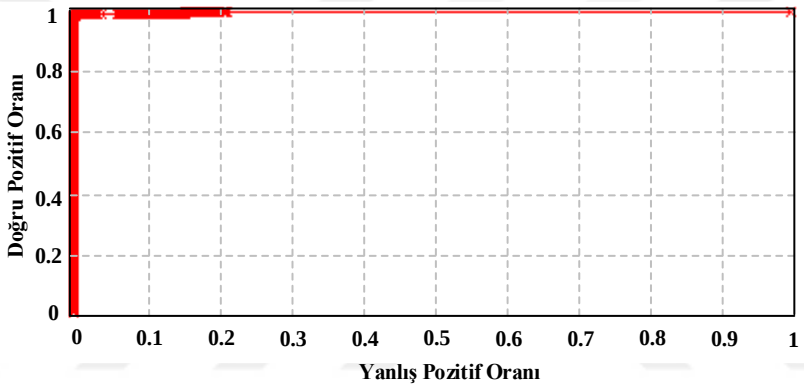
Şekil 6.43. Klasik UÖM sınıflandırıcısının BAĞLANTI sınıfına ait AİK eğrisi



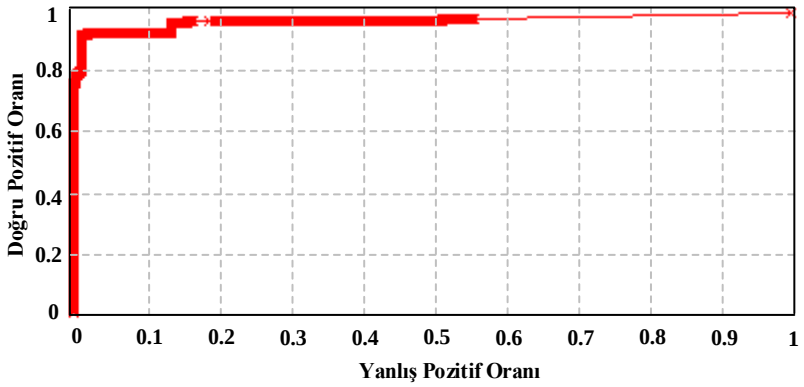
Şekil 6.44. Klasik UÖM sınıflandırıcısının ATAK sınıfına ait AİK eğrisi



Şekil 6.45. Klasik UÖM sınıflandırıcısının P2P sınıfına ait AİK eğrisi



Şekil 6.46. Klasik UÖM sınıflandırıcısının VERİ TABANI sınıfına ait AİK eğrisi

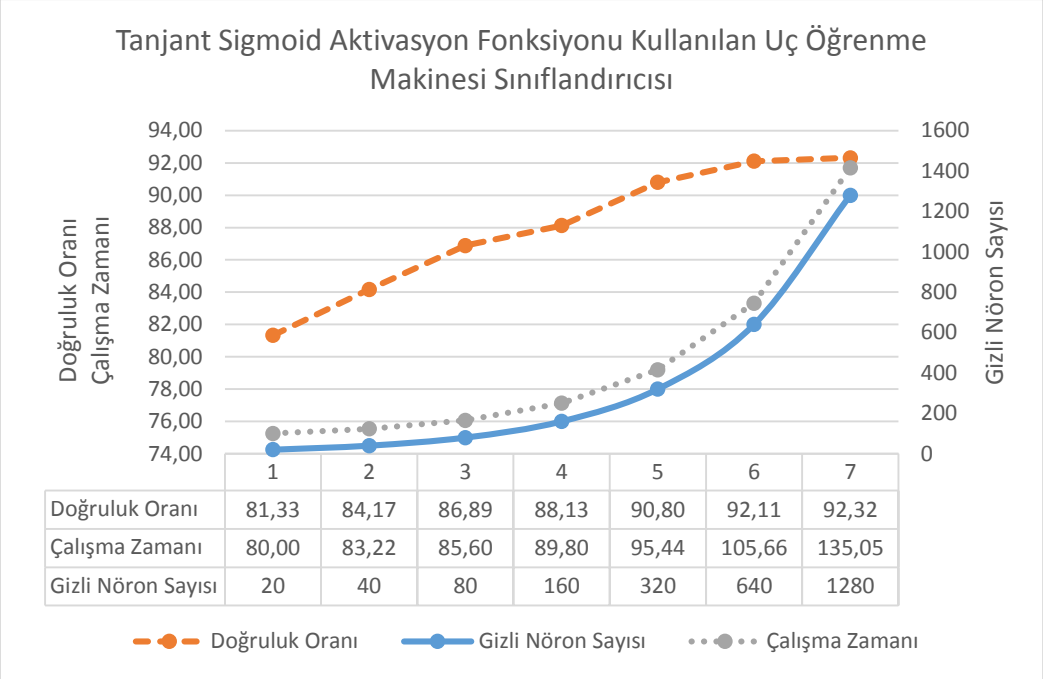


Şekil 6.47. Klasik UÖM sınıflandırıcısının SERVİS sınıfına ait AİK eğrisi

Tanjant Sigmoid aktivasyon fonksiyonu için farklı gizli katman hücre sayılarının seçilmesiyle elde edilen sınıflandırma başarımları ve bu sınıflandırma için harcanan süre

değerleri Şekil 6.48 de gösterilmiştir. Gizli katman hücre sayısı 20 değerinden başlatılarak her seferinde iki katı alınarak artırılmıştır.

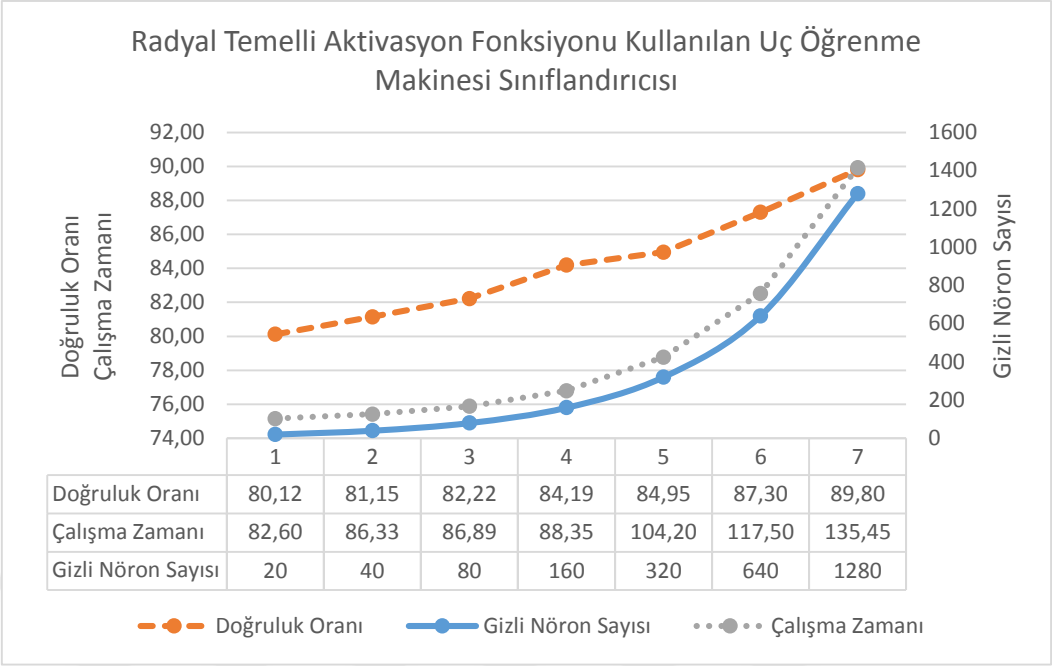
Grafik incelendiğinde katman hücre sayısının artışı ile birlikte doğruluk oranlarında da artış gözlenmiştir. Çalışma süresi de katman hücre sayısının artması ile birlikte artmıştır. 640 gizli katman hücre sayısının alındığı sınıflandırıcı ile 1280 gizli katman hücre sayısının alındığı sınıflandırıcının (DOD_{ort}) değerleri arasında büyük bir değişiklik olmadığı gözlenmektedir. Bu sebeple daha fazla gizli katman hücre sayısı artırılarak inceleme yapılmamıştır. 640 gizli katman hücre sayısının kullanıldığı sınıflandırıcının doğruluk yüzdesi % 92.11 olarak bulunmuştur.



	1	2	3	4	5	6	7
Doğruluk Oranı	81,33	84,17	86,89	88,13	90,80	92,11	92,32
Çalışma Zamanı	80,00	83,22	85,60	89,80	95,44	105,66	135,05
Gizli Nöron Sayısı	20	40	80	160	320	640	1280

Şekil 6.48. Tanjant sigmoid aktivasyon fonksiyonunun farklı katman hücre sayılarındaki doğruluk ve çalışma zamanı değerleri

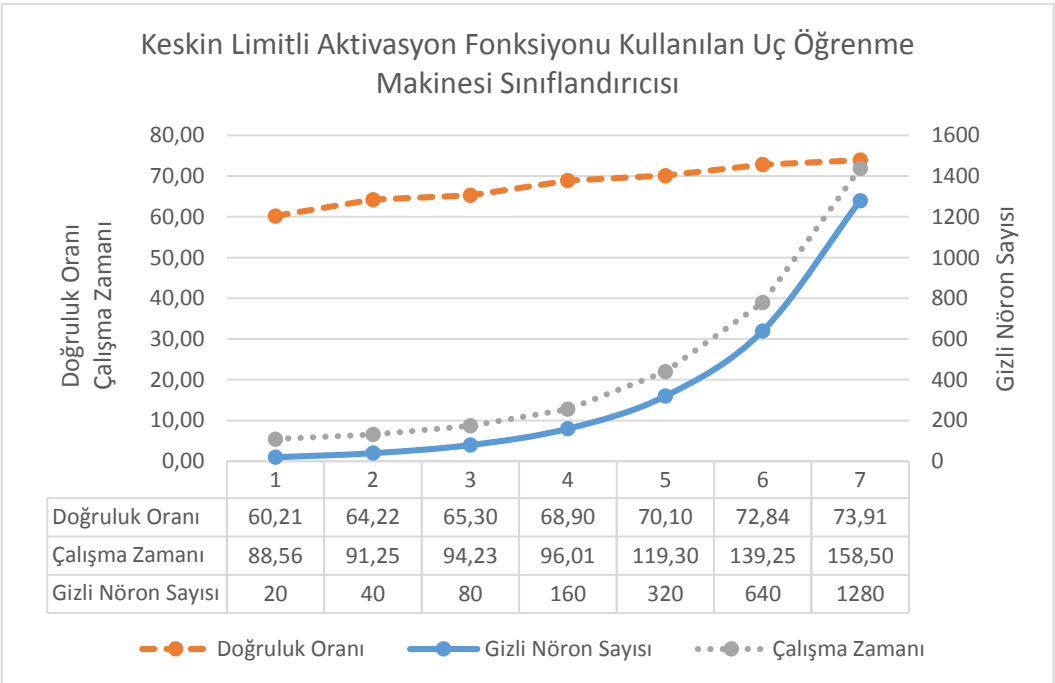
Klasik UÖM ile elde edilen analizlerde farklı aktivasyon fonksiyonları da kullanılarak başarım oranları gözlenmiştir. Radyal temelli aktivasyon fonksiyonun farklı katman hücre sayılarındaki doğruluk yüzdeleri Şekil 6.49'da gösterilmektedir.



	1	2	3	4	5	6	7
Doğruluk Oranı	80,12	81,15	82,22	84,19	84,95	87,30	89,80
Çalışma Zamanı	82,60	86,33	86,89	88,35	104,20	117,50	135,45
Gizli Nöron Sayısı	20	40	80	160	320	640	1280

Şekil 6.49. Radyal temelli aktivasyon fonksiyonunun farklı katman hücre sayılarındaki doğruluk ve çalışma zamanı değerleri

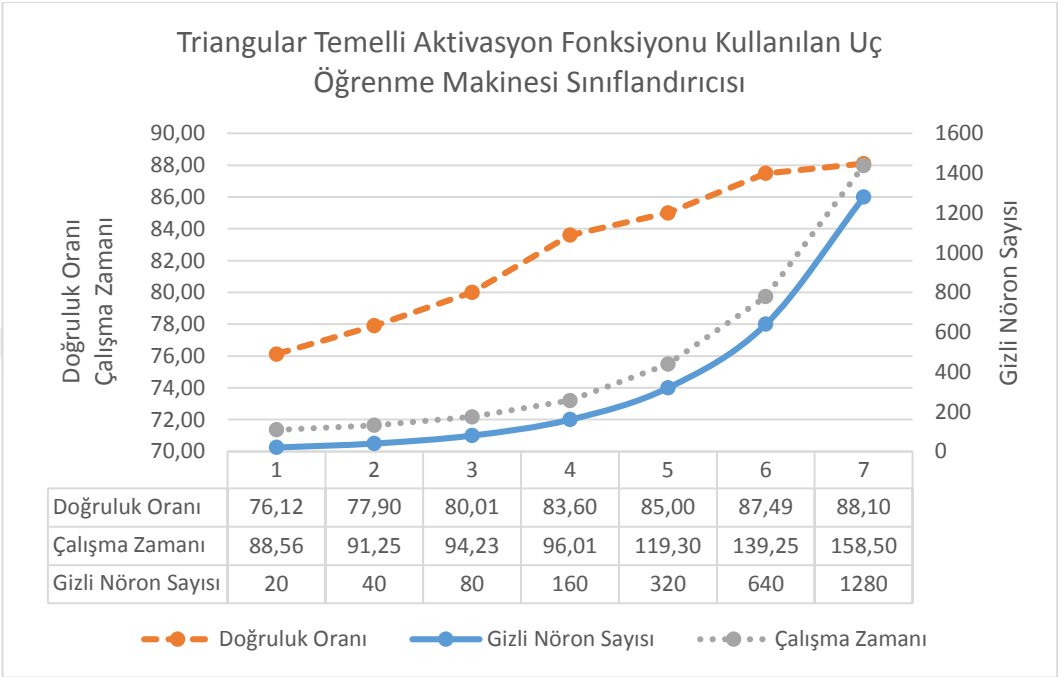
Keskin limitli aktivasyon fonksiyonunun seçimi ile gizli katman hücre sayılarının değiştirilmesiyle elde edilen sınıflandırma başarımına ait veriler Şekil 6.50'de gösterilmiştir.



	1	2	3	4	5	6	7
Doğruluk Oranı	60,21	64,22	65,30	68,90	70,10	72,84	73,91
Çalışma Zamanı	88,56	91,25	94,23	96,01	119,30	139,25	158,50
Gizli Nöron Sayısı	20	40	80	160	320	640	1280

Şekil 6.50. Keskin limitli aktivasyon fonksiyonunun farklı katman hücre sayılarındaki doğruluk ve çalışma zamanı değerleri

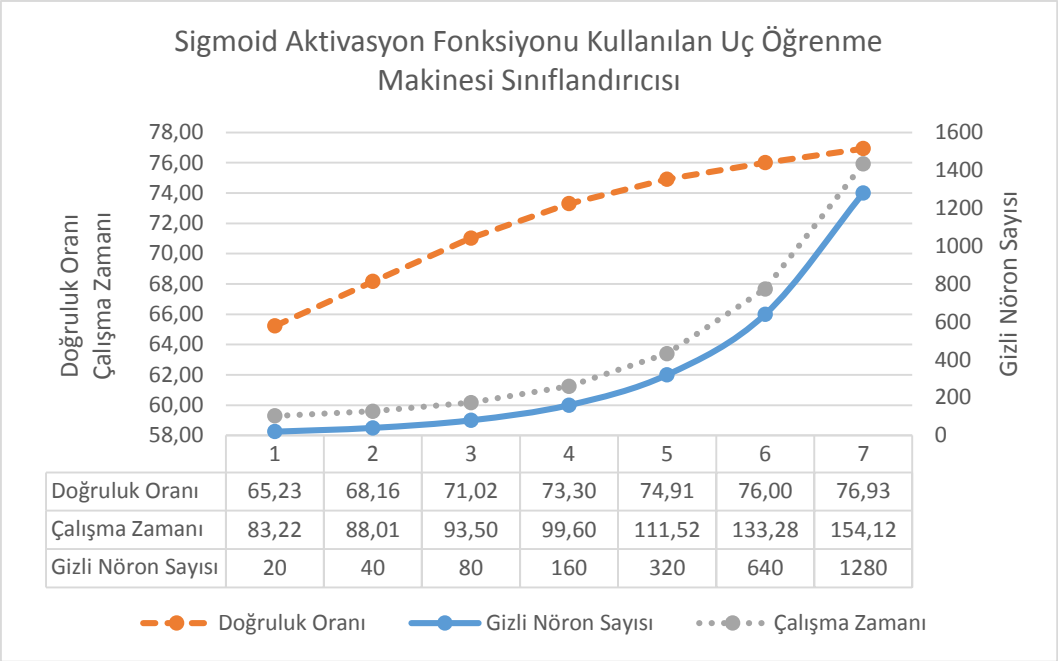
Keskin limitli aktivasyon fonksiyonu kullanılarak yapılan sınıflandırmanın başarım oranının düşük olduğu gözlenmiştir. Triangular temelli aktivasyon fonksiyonu seçilerek yapılan sınıflandırmaya ait sonuçlar Şekil 6.51'de gösterilmektedir.



	1	2	3	4	5	6	7
Doğruluk Oranı	76,12	77,90	80,01	83,60	85,00	87,49	88,10
Çalışma Zamanı	88,56	91,25	94,23	96,01	119,30	139,25	158,50
Gizli Nöron Sayısı	20	40	80	160	320	640	1280

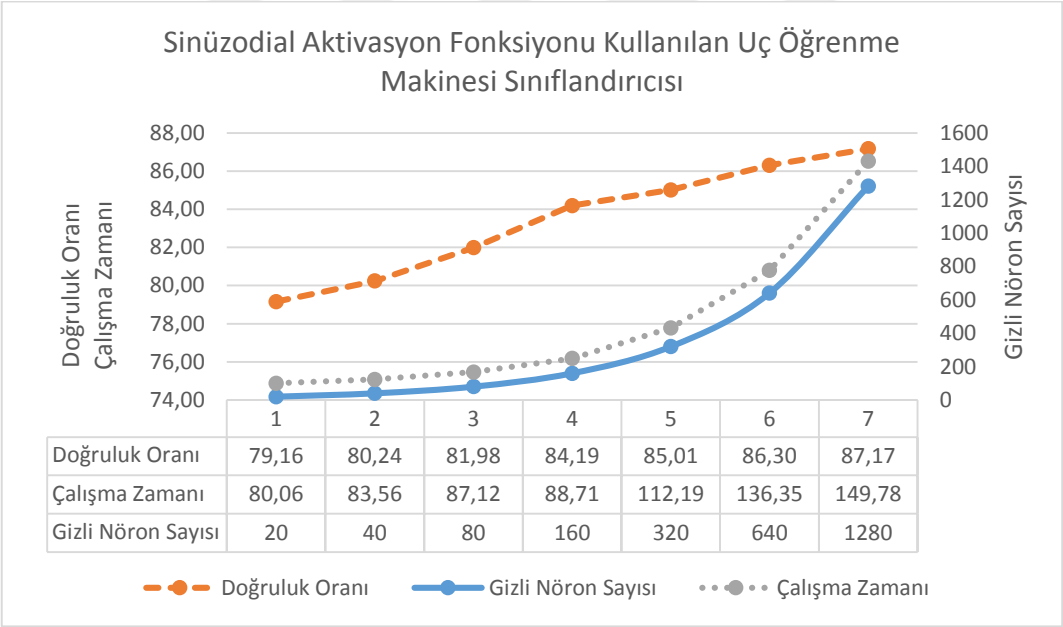
Şekil 6.51. Triangular temelli aktivasyon fonksiyonunun farklı katman hücre sayılarındaki doğruluk ve çalışma zamanı değerleri

Yaygın olarak kullanılan diğer aktivasyon fonksiyonlarından olan sigmoid ve sinüzodal aktivasyon fonksiyonları da gizli katman hücre sayıları 20 ile 1280 arasında her seferinde iki katı alınacak şekilde seçilerek kullanılmıştır. Sigmoid aktivasyon fonksiyonunun kullanılması ile elde edilen başarım değerlerine ait veriler Şekil 6.52'de verilmiştir. Sinüzodial aktivasyon fonksiyonunun kullanılması ile elde edilen başarım değerlerine ait veriler Şekil 6.53'de verilmiştir.



	1	2	3	4	5	6	7
Doğruluk Oranı	65,23	68,16	71,02	73,30	74,91	76,00	76,93
Çalışma Zamanı	83,22	88,01	93,50	99,60	111,52	133,28	154,12
Gizli Nöron Sayısı	20	40	80	160	320	640	1280

Şekil 6.52. Sigmoid aktivasyon fonksiyonunun farklı katman hücre sayılarındaki doğruluk ve çalışma zamanı değerleri



	1	2	3	4	5	6	7
Doğruluk Oranı	79,16	80,24	81,98	84,19	85,01	86,30	87,17
Çalışma Zamanı	80,06	83,56	87,12	88,71	112,19	136,35	149,78
Gizli Nöron Sayısı	20	40	80	160	320	640	1280

Şekil 6.53. Sinüzoidal aktivasyon fonksiyonunun farklı katman hücre sayılarındaki doğruluk ve çalışma zamanı değerleri

Kullanılan aktivasyon fonksiyonları ve seçilen gizli katman hücre sayılarına göre klasik UÖM algoritmasına ait doğruluk değerleri ve çalışma zamanları Tablo 6.7'de ayrıntılı olarak verilmiştir.

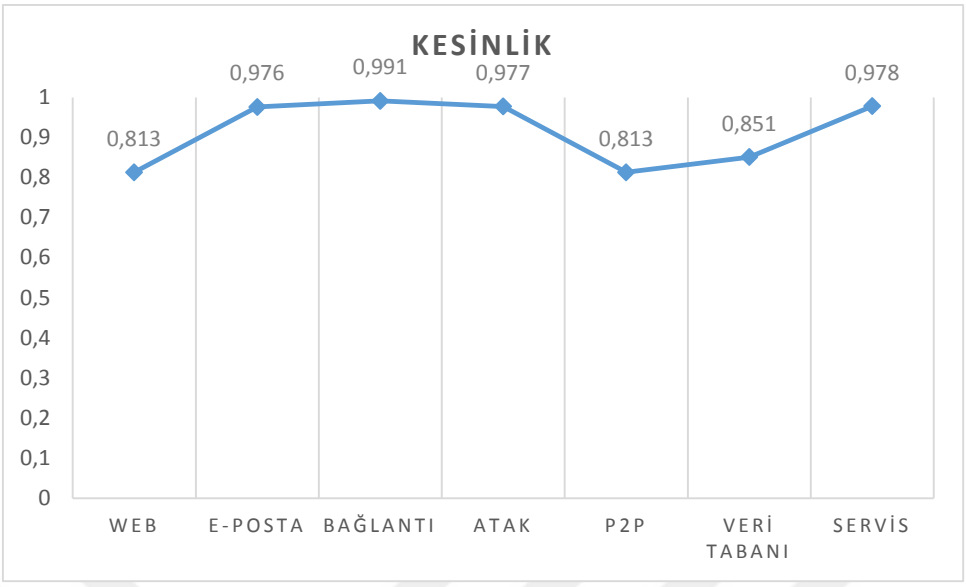
Tablo 6.7. Farklı aktivasyon fonksiyonları ve gizli katman hücre sayılarına göre doğruluk değeri yüzdeleri ve çalışma süreleri

Gizli Katman hücre Sayısı	Aktivasyon Fonksiyonu											
	Tanjant Sigmoid		Radyal Temelli		Sigmoid		Sinüzoidal		Keskin Limitli		Triangular Temelli	
	DOD (%)	Süre (ms)	DOD (%)	Süre (ms)	DOD (%)	Süre (ms)	DOD (%)	Süre (ms)	DOD (%)	Süre (ms)	DOD (%)	Süre (ms)
20	81,33	80,00	80,12	82,60	65,23	83,22	79,16	80,06	60,21	88,56	76,12	88,56
40	84,17	83,22	81,15	86,33	68,16	88,01	80,24	83,56	64,22	91,25	77,90	91,25
80	86,89	85,60	82,22	86,89	71,02	93,50	81,98	87,12	65,30	94,23	80,01	94,23
160	88,13	89,80	84,19	88,35	73,30	99,60	84,19	88,71	68,90	96,01	83,60	96,01
320	90,80	95,44	84,95	104,20	74,91	111,52	85,01	112,19	70,10	119,30	85,00	119,30
640	92,11	105,66	87,30	117,50	76,00	133,28	86,30	136,35	72,84	139,25	87,49	139,25
1280	92,32	135,05	89,80	135,45	76,93	154,12	87,17	149,78	73,91	158,50	88,10	158,50

6.4.5. Çekirdek Tabanlı Uç Öğrenme Makineleri ile Elde Edilen Bulgular

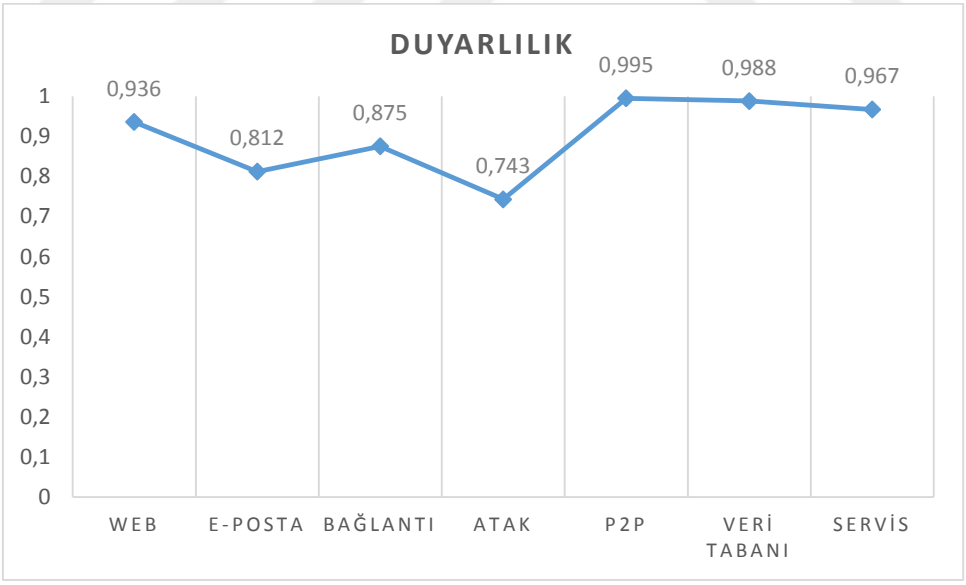
UÖM algoritmalarının farklı bir yaklaşımı olan ÇTUÖM yaklaşımı için dalgacık fonksiyonu seçilerek test verileri sınıflandırıcıya verilmiştir. Denklem 4.11 de gösterilen a, b ve c parametreleri 1 alınarak sınıflandırma algoritması çalıştırılmıştır. Sınıflandırıcı ile yapılan sınıflandırma ile test verilerinden 4736 tanesi doğru, 514 tanesi yanlış sınıflandırılmıştır. Sınıflandırıcının ortalama çalışma süresinin 155 milisaniye olduğu görülmüştür. Ortalama doğruluk değeri (DOD_{ort}) yüzdesi % 90,20 olarak bulunmuştur. Sınıflar içerisinde en yüksek doğru sınıflandırma değeri % 99,50 ile P2P sınıfına aittir. En düşük doğru sınıflandırma yüzdesi ise % 74,30 ile Atak sınıfına aittir.

Dalgacık fonksiyonunun kullanıldığı ÇT UÖM ait Kesinlik performans değerleri Şekil 6.54'de gösterilmektedir.



Şekil 6.54. Dalgacık aktivasyon fonksiyonlu ÇTUÖM sınıflandırıcına ait Kesinlik metrik değerleri

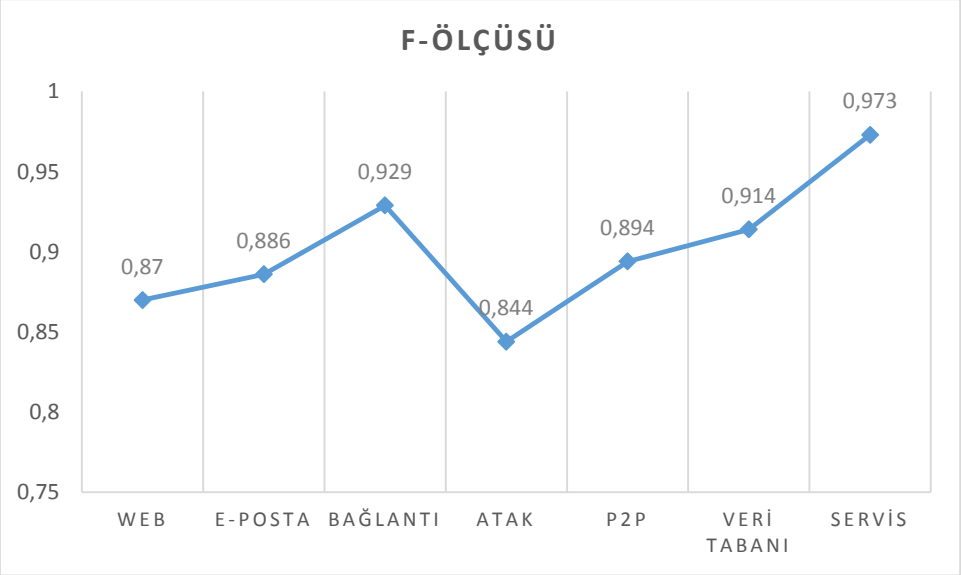
Kesinlik değerlerine göre en yüksek sonuç ile 0,977 ile Atak sınıfına aittir. En düşük sonuç ise 0,813 ile Web ve P2P sınıflarına aittir. Şekil 6.55'de Duyarlılık değerleri verilmiştir.



Şekil 6.55. Dalgacık aktivasyon fonksiyonlu ÇTUÖM sınıflandırıcına ait Duyarlılık metrik değerleri

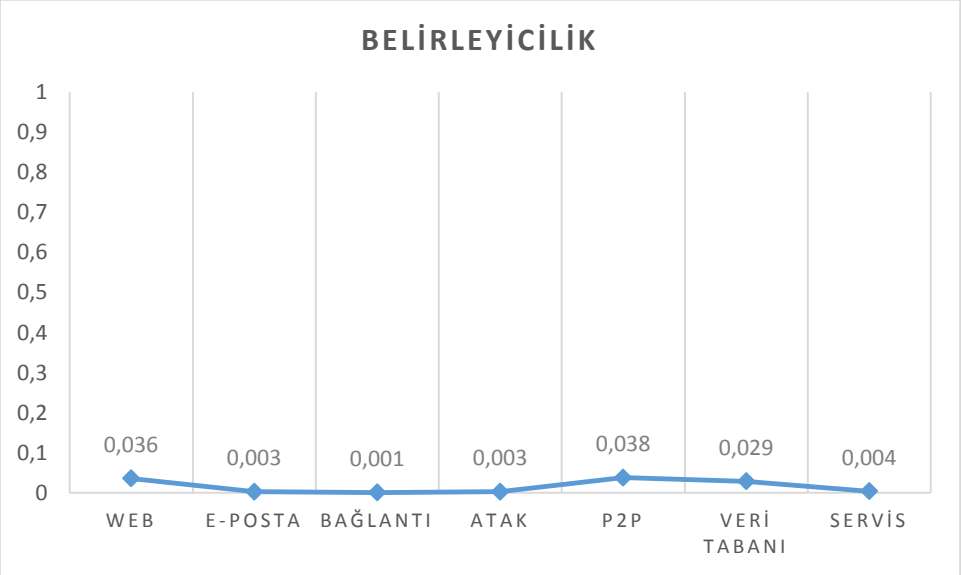
Duyarlılık değerlerine göre en yüksek sonuç 0,995 ile P2P sınıfına aittir. En düşük sonuç ise 0,743 ile Atak sınıfına aittir. F-Ölçüsüne ait metrik değerleri Şekil 6.56'da

gösterilmiştir. F-Ölçüsüne ait sınıflandırma değerleri içerisinde en yüksek değere 0,973 ile Servis sınıfı ulaşmıştır. En düşük değer ise 0,844 ile Atak sınıfına aittir.



Şekil 6.56. Dalgacık aktivasyon fonksiyonlu ÇTUÖM sınıflandırıcına ait F-Ölçüsü metrik değerleri

Bir diğer sınıflandırma metriği olarak Belirleyicilik metrik değerleri hesaplanmıştır. Belirleyicilik metriğine ait değerler Şekil 6.57'de gösterilmektedir.



Şekil 6.57. Dalgacık aktivasyon fonksiyonlu ÇTUÖM sınıflandırıcına ait Belirleyicilik metrik değerleri

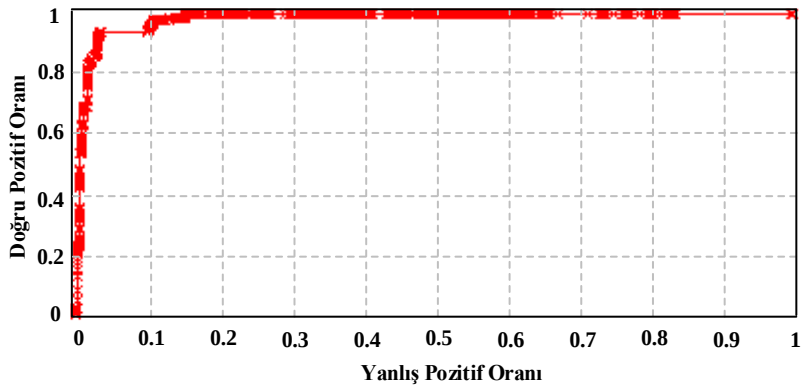
Negatif sınıf etiketlerini tahmin etmeye çalışan sınıflandırıcının başarısı olarak kabul edebileceğimiz Belirleyicilik metrik değerleri analiz edildiğinde en düşük değerin 0,001 ile bağlantı sınıfına, en yüksek değerlerin ise 0,038 ile P2P sınıfına ait olduğu görülmüştür.

Dalgacık aktivasyon fonksiyonlu ÇTUÖM sınıflandırıcının test verilerine uygulanması ile elde edilen performans metrik değerleri Tablo 6.8'de gösterilmektedir.

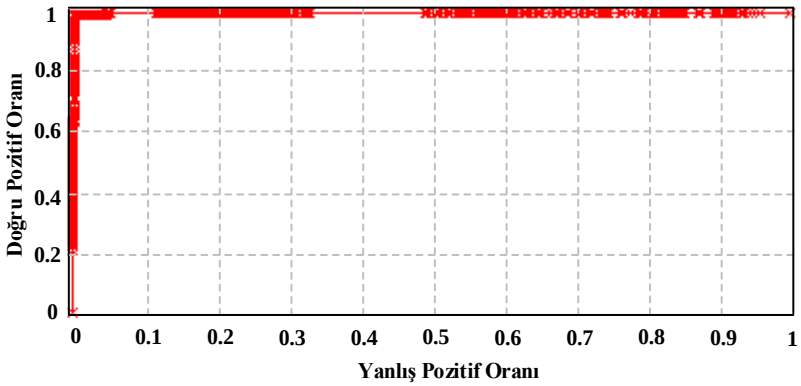
Tablo 6.8. Dalgacık aktivasyon fonksiyonlu ÇTUÖM sınıflandırıcısı ile elde edilen performans değerleri

SINIF	Performans Metrik Değerleri			
	Belirleyicilik	Kesinlik	Duyarlılık	F-Ölçüsü
WEB	0,036	0,813	0,936	0,87
E-POSTA	0,003	0,976	0,812	0,886
BAĞLANTI	0,001	0,991	0,875	0,929
ATAK	0,003	0,977	0,743	0,844
P2P	0,038	0,813	0,995	0,894
VERİ TABANI	0,029	0,851	0,988	0,914
SERVİS	0,004	0,978	0,967	0,973
ORTALAMA	0,016	0,914	0,902	0,902

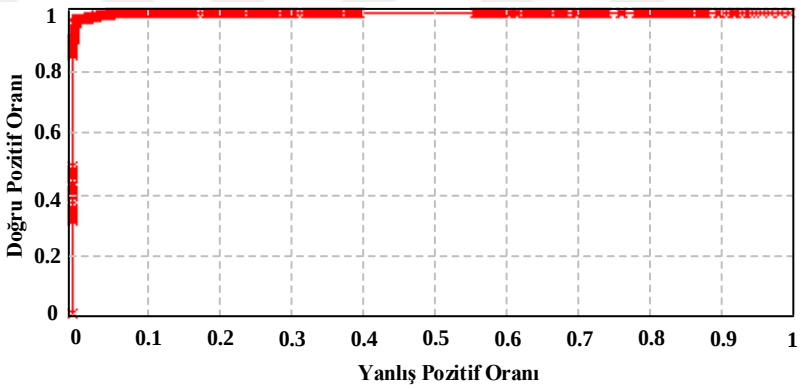
Dalgacık aktivasyon fonksiyonlu ÇTUÖM sınıflandırıcısı için her bir sınıfa ait AİK eğri grafikleri Şekil 6.58 ile Şekil 6.64 arasında gösterilmektedir. Grafiklerde yatay eksen yanlış pozitif oranını, dikey eksen ise doğru pozitif oranını göstermektedir. AİK eğri alanları Web sınıfı için 0,980; E-Posta sınıfı için 0,998; Bağlantı sınıfı için 0,998; Atak sınıfı için 0,963; P2P sınıfı için 0,998; Veri Tabanı sınıfı için 0,998; Servis sınıfı için 0,997 olarak bulunmuştur. AİK eğri alanının ortalama değeri ise 0,990 çıkmıştır.



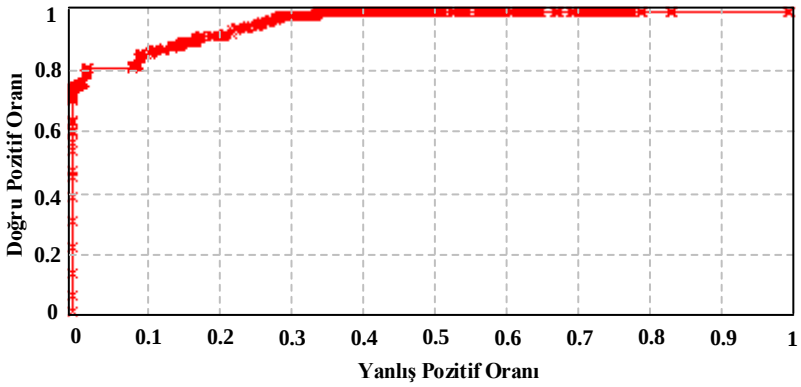
Şekil 6.58. Dalgacık aktivasyon fonksiyonlu ÇTUÖM sınıflandırıcısının WEB sınıfına ait AİK eğrisi



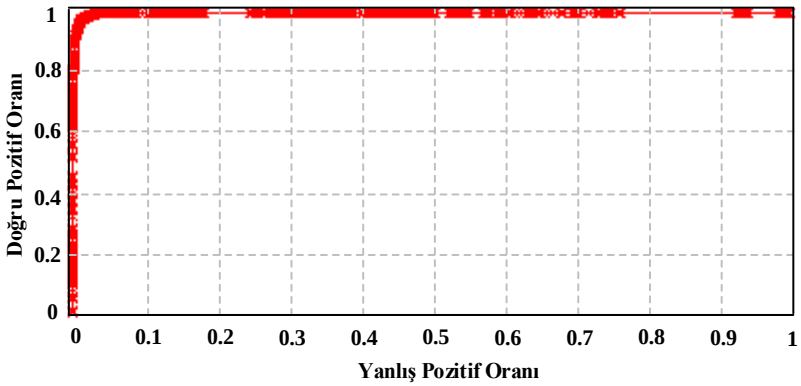
Şekil 6.59. Dalgacık aktivasyon fonksiyonlu ÇTUÖM sınıflandırıcısının E-POSTA sınıfına ait AİK eğrisi



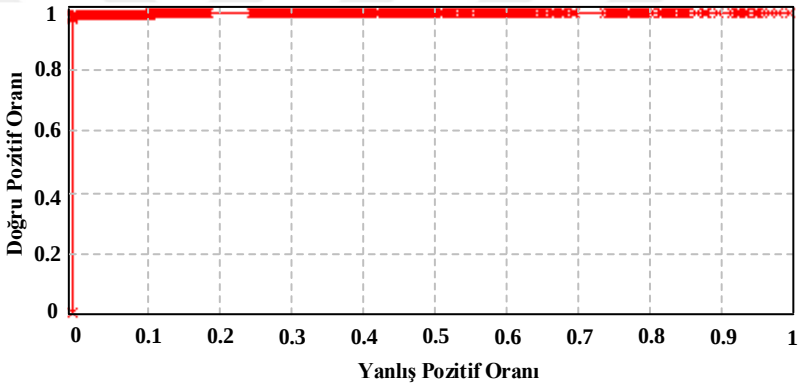
Şekil 6.60. Dalgacık aktivasyon fonksiyonlu ÇTUÖM sınıflandırıcısının BAĞLANTI sınıfına ait AİK eğrisi



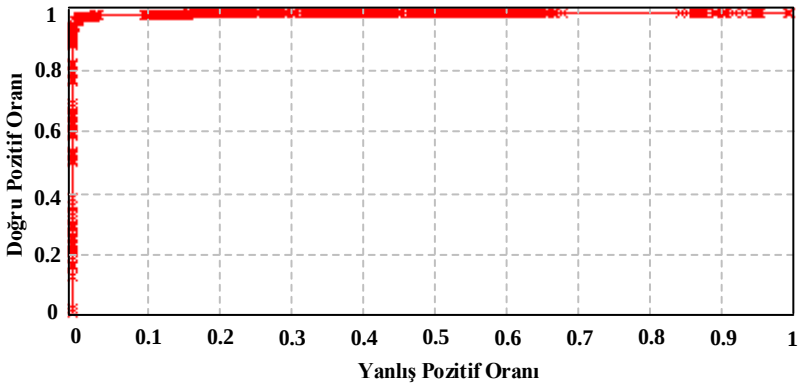
Şekil 6.61. Dalgacık aktivasyon fonksiyonlu ÇTUÖM sınıflandırıcısının ATAK sınıfına ait AİK eğrisi



Şekil 6.62. Dalgacık aktivasyon fonksiyonlu ÇTUÖM sınıflandırıcısının P2P sınıfına ait AİK eğrisi



Şekil 6.63. Dalgacık aktivasyon fonksiyonlu ÇTUÖM sınıflandırıcısının VERİ TABANI sınıfına ait AİK eğrisi



Şekil 6.64. Dalgacık aktivasyon fonksiyonlu ÇTUÖM sınıflandırıcısının SERVİS sınıfına ait AİK eğrisi

ÇTUÖM algoritması için kullanılan aktivasyon fonksiyonu olarak dalgacık yerine polinom, radyal temelli, doğrusal gibi aktivasyon fonksiyonları seçilerek başarım oranları gözlenmiştir. Yapılan analizlerde aynı veriler için polinom aktivasyon fonksiyonun seçimi ile (DOD_{ort}) değerinin 0,76 çıktığı görülmüştür. Doğrusal aktivasyon fonksiyonun seçimi ile polinom aktivasyon fonksiyonu ile elde edilen sonuca yakın bir değer olarak 0,75 değeri görülmüştür. Radyal temelli aktivasyon fonksiyonu ise çekirdek tabanlı aktivasyon fonksiyonuna en çok yaklaşan aktivasyon fonksiyonu olmuştur. Radyal temelli aktivasyon fonksiyonuna ait (DOD_{ort}) değerinin 0,88 değerini aldığı görülmüştür.

Dalgacık fonksiyonu için kullanılan parametrelerin değerleri değiştirilerek sınıflandırıcı algoritmanın çalıştırılması ile (DOD_{ort}) değerlerinin de değiştiği gözlenmiştir. Başlangıçta her üç parametre değeri 1 alınırken daha sonra parametrelere hem birbiriyle eşit olarak hem de farklı değerler olacak şekilde rastgele sayılar verilerek gözlenmiştir. Elde edilen sonuçlar Tablo 6.9'da gösterilmektedir.

Tablo 6.9. Farklı parametre değerleri kullanılan dalgacık aktivasyon fonksiyonlu ÇTUÖM sınıflandırıcısı ile elde edilen doğruluk değerleri ortalamaları

Dalgacık Fonksiyonu Parametreleri			(DOD_{ort}) %
a	b	c	
1	1	1	90,20
100	10	1	82,15
1	10	100	74,67
100	1	10	49,05
10	10	10	80,04
0,1	0,1	0,1	92,32
1	1	0,1	92,67
10	5	1	89,31
0,01	0,01	0,01	95,04
0,01	1	0,01	94,80
1	1	0,01	94,38

Tablo 6.9 incelendiğinde parametrelerin almış oldukları değerlerin azalması durumunda doğruluk değerlerine ait başarım yüzdelerinin arttığı gözlenmektedir. Yapılan analizde değerlerin küçülmesinin çalışma sürelerini artırmış olduğu görülmüştür. Doğru bir sınıflandırıcı için çalışma zamanı da önemli olacağı için parametreleri doğrudan küçük seçmek yerine en uygun değerlerinin bulunması büyük önem kazanmaktadır. Dalgacık fonksiyonunda kullanılan c parametresinin başlangıç değerinden daha küçük bir değer

seçilerek a ve b parametre değerlerinin sabit bırakılması ile aynı şekilde küçültülmesi arasında çok önemli bir kazanç olmadığı gözlenmiştir.

Parametrelerin rastgele değiştirilmesi yerine, bu işlemi en uygun değerleri bulacak şekilde bir algoritmadan geçirilerek dalgacık aktivasyon fonksiyonlu ÇTUÖM algoritmasına verilmesi için GA'dan faydalanılarak bir uygulama hazırlanmıştır. Geliştirilen yöntem ile ilgili ayrıntılar tez çalışmasının 6.5 bölümünde yer almaktadır.

6.4.6. Kullanılan Makine Öğrenmesi Yöntemlerinin Karşılaştırılması

Ağ trafiği verilerinin makine öğrenmesi yaklaşımlarıyla analiz edilmesi amacıyla çeşitli makine öğrenmesi yöntemlerinden faydalanılmıştır. Bu yöntemlerden NB ve DVM daha önceki araştırmacılar tarafından sıklıkla uygulanan ve başarılı sınıflandırma sonuçları veren yöntemlerdendir. Uç öğrenme makineleri ise diğer öğrenme yöntemlerine göre daha yeni olan ve hala gelişmekte olan bir makine öğrenmesi yöntemidir. Ağ trafiği verilerine hem klasik UÖM sınıflandırıcı ile hem de UÖM algoritmalarının farklı bir yaklaşımı olan ÇTUÖM sınıflandırıcı uygulanarak analizler yapılmıştır. Aynı zamanda temel bir sınıflandırıcı olan YSA, verilerin sınıflandırılmasında kullanılmıştır. Bu beş yaklaşım ile elde edilen belirleyicilik metriklerine ait veriler Tablo 6.10'da karşılaştırılarak sunulmuştur.

Tablo 6.10. Makine öğrenmesi yöntemleri ile elde edilen belirleyicilik performans metriklerinin karşılaştırılması

SINIF	Sınıflandırıcı				
	YSA	NB	DVM	Klasik UÖM	ÇTUÖM
WEB	0,031	0,026	0,035	0,031	0,036
E-POSTA	0,037	0,010	0,030	0,013	0,003
BAĞLANTI	0,007	0,005	0,002	0,007	0,001
ATAK	0,029	0,071	0,016	0,024	0,003
P2P	0,002	0,031	0,097	0,084	0,038
VERİ TABANI	0,137	0,070	0,002	0,005	0,029
SERVİS	0,058	0,050	0,006	0,020	0,004
ORTALAMA	0,043	0,038	0,027	0,026	0,016

Negatif sınıf etiketlerinin başarısını belirlemeye çalışan belirleyicilik performans değerleri içerisinde en küçük ortalama değer ÇTUÖM'ye ait sınıflandırıcıdan elde edilmiştir. DVM ve klasik UÖM ile elde edilen ortalama değerler birbirlerine yakın çıkmıştır. Sınıflar içerisindeki en küçük değer Bağlantı sınıfına ait olup ÇTUÖM sınıflandırıcısından elde

edilmiştir. En büyük değer ise P2P sınıfına ait olup DVM sınıflandırıcından elde edilmiştir. Ortalama değerler içerisinde en büyük değere YSA sınıflandırıcısı ile ulaşılmıştır.

Sınıflandırıcılar ile elde edilen kesinlik performans metriklerine ait veriler Tablo 6.11'de karşılaştırılarak sunulmuştur.

Tablo 6.11. Makine öğrenmesi yöntemleri ile elde edilen kesinlik performans metriklerinin karşılaştırılması

SINIF	Sınıflandırıcı				
	YSA	NB	DVM	Klasik UÖM	ÇTUÖM
WEB	0,788	0,705	0,819	0,833	0,813
E-POSTA	0,817	0,927	0,847	0,921	0,976
BAĞLANTI	0,930	0,964	0,964	0,923	0,991
ATAK	0,802	0,641	0,887	0,829	0,977
P2P	0,964	0,812	0,629	0,64	0,813
VERİ TABANI	0,546	0,700	0,989	0,971	0,851
SERVİS	0,716	0,764	0,958	0,885	0,978
ORTALAMA	0,795	0,788	0,870	0,858	0,914

Pozitif öngörü veya pozitif kestirim olarak da adlandırılabilen kesinlik performans metriğine ait veriler ile doğru sınıflandırılmış olan pozitif etiketli örneklerin, tahmin edilen toplam pozitif örneklerinin oranı tespit edilmektedir. Ortalama değer içerisinde dalgacık aktivasyon fonksiyonunun kullanıldığı ÇTUÖM sınıflandırıcısı öne çıkmaktadır. Klasik UÖM ve DVM ile yapılan sınıflandırma analizinin belirleyicilik performans metriğindekine benzer şekilde birbirlerine yakın değerlerde çıktığı gözlenmiştir. Sınıflar içerisinde en yüksek değer Bağlantı sınıfı ile ÇTUÖM sınıflandırıcından elde edilmiştir. İkinci en yüksek değer ise DVM sınıflandırıcısından elde edilen Veri Tabanı sınıfına ait veridir. En küçük değer ise klasik UÖM kullanılarak P2P sınıfından elde edilen veridir.

Sınıflandırıcılar ile elde edilen duyarlılık performans metriklerine ait veriler Tablo 6.12'de karşılaştırılarak sunulmuştur.

Tablo 6.12. Makine öğrenmesi yöntemleri ile elde edilen duyarlılık performans metriklerinin karşılaştırılması

SINIF	Sınıflandırıcı				
	YSA	NB	DVM	Klasik UÖM	ÇTUÖM
WEB	0,683	0,379	0,948	0,921	0,936
E-POSTA	0,997	0,759	0,996	0,935	0,812
BAĞLANTI	0,587	0,788	0,360	0,515	0,875
ATAK	0,713	0,757	0,736	0,713	0,743
P2P	0,356	0,795	0,991	0,896	0,995
VERİ TABANI	0,988	0,985	0,985	0,987	0,988
SERVİS	0,869	0,961	0,856	0,925	0,967
ORTALAMA	0,742	0,775	0,839	0,842	0,902

Pozitif sınıf etkinliklerinin tahmin edilmesini amaçlayan duyarlılık performans metriği sınıflandırıcıların performanslarının karşılaştırılmasında en çok dikkat edilen doğruluk değerini vermektedir. En iyi sınıflandırma sonucunun ÇTUÖM ile alındığı görülmektedir. DVM ve klasik UÖM yaklaşımlarının birbirlerine yakın olduğu gözlenmektedir. YSA ve NB sınıflandırıcısının ise diğer sınıflandırıcılara göre daha kötü performans gösterdiği görülmektedir. Sınıflar içerisinde % 99 üzerinde doğru sınıfta olduğu tespit edilen sınıf P2P sınıfıdır. Hem ÇTUÖM'de hem de DVM'de yüksek oranda doğru sınıflandırıldığı görülmektedir. NB sınıflandırıcısı ile Web sınıfı, YSA ile P2P sınıfı, DVM sınıflandırıcısı ile de Bağlantı sınıfı en kötü oranda doğru sınıflandırılan sınıflar olarak görülmektedir.

Sınıflandırıcılar ile elde edilen F-Ölçüsü performans metriklerine ait veriler Tablo 6.13'de karşılaştırılarak sunulmuştur. Duyarlılık ve kesinlik metriklerinin uyumlu bir oranı olarak kabul edebileceğimiz F-Ölçüsü performans metriği ile elde edilen sonuçlar incelendiğinde en iyi sonucun ÇTUÖM sınıflandırıcısından elde edildiği görülmektedir.

Tablo 6.13. Makine öğrenmesi yöntemleri ile elde edilen F-Ölçüsü performans metriklerinin karşılaştırılması

SINIF	Sınıflandırıcı				
	YSA	NB	DVM	Klasik UÖM	ÇTUÖM
WEB	0,731	0,493	0,879	0,875	0,870
E-POSTA	0,898	0,834	0,915	0,928	0,886
BAĞLANTI	0,720	0,867	0,524	0,661	0,929
ATAK	0,755	0,694	0,805	0,767	0,844
P2P	0,520	0,803	0,770	0,747	0,894
VERİ TABANI	0,703	0,818	0,987	0,979	0,914
SERVİS	0,785	0,851	0,904	0,905	0,973
ORTALAMA	0,730	0,766	0,826	0,837	0,902

Sınıflandırıcıların karşılaştırılmasında performans metrikleri kadar sınıflandırıcının çalışma süresinin de önemi büyüktür. Kullanılan sınıflandırıcılar içerisinde UÖM yaklaşımlarının DVM ve NB sınıflandırıcısına göre daha hızlı sınıflandırma yaptığı görülmektedir. UÖM yaklaşımlarından klasik UÖM'de kullanılan gizli katman hücre sayısının artırılması ile çalışma zamanı için geçen sürenin doğru orantılı olarak arttığı gözlenmektedir. Gizli katman hücre sayısının artırılması ile doğruluk oranlarında bir artış olduğu fakat belli bir değerden sonra bu artışın çok fazla olmayarak çok küçük oranlarda değiştiği veya sabit kalmaya devam ettiği görülmüştür. Bu sebeple gizli katman hücre sayısının çok fazla artırılarak çalışma zamanının artırılmasının sınıflandırıcının maliyetini artırmasına rağmen kazanç sağlamadığı görülmüştür. ÇTUÖM algoritmalarında kullanılan dalgacık aktivasyon fonksiyonuna ait parametrelerinin değerlerinin azaltılması ile çalışma zamanlarının arttığı ve doğruluk oranında iyileşme olduğu gözlenmiştir. Bu parametre değerlerinin çok fazla düşürülerek uzun zaman içerisinde sınıflandırılma yapılmasının kazanç sağlamadığı belli bir değerden sonra doğruluk değerlerinde ciddi artışların olmadığı gözlenmiştir. YSA ile yapılan sınıflandırıcının çalışma zamanı diğer sınıflandırıcılara göre belirgin ölçüde yavaş çıkmıştır.

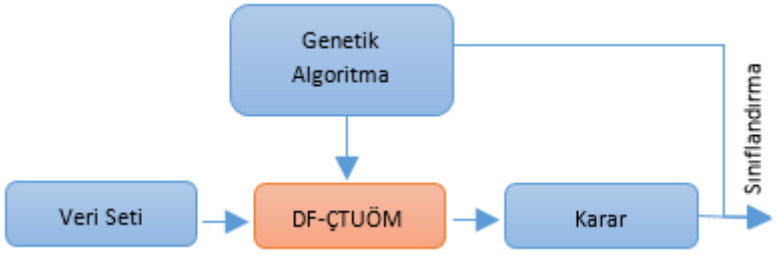
6.5. Genetik Algoritma ile Dalgacık Fonksiyonu Kullanılan Çekirdek Tabanlı Uç Öğrenme Makinesi

Ağ trafiği verilerinin sınıflandırılması amacıyla uygulanan makine öğrenmesi yaklaşımları içerisinde en iyi sonuçların dalgacık fonksiyonunun kullanıldığı ÇTUÖM ile elde edildiği görülmektedir. Dalgacık fonksiyonunda kullanılan parametrelerin seçimi sınıflandırıcının başarımı için çok önemli rol oynamaktadır. Bu parametrelere ait değerlerin değiştirilmesi ile sınıflandırıcının daha iyi bir performans göstermesi sağlanabilir.

Tezin bu bölümünde hazırlanan uygulama ile dalgacık aktivasyon fonksiyonunda kullanılan parametrelerin sınıflandırıcı için en uygun değerlerde kullanılabilmesi için deneme yanılma yöntemiyle seçilmesi yerine, GA yaklaşımıyla en iyi değerlere getirilmesi amaçlanmaktadır. GA-DF-ÇTUÖM yaklaşımı ile başarılı bir şekilde veriler yüksek doğruluk oranlarında sınıflandırılmıştır.

GA-DF-ÇTUÖM yöntemi üç aşamadan oluşmaktadır. Birinci aşamada ağ trafiğinin sınıflandırılması için kullanılacak olan veriler elde edilir. GA-DF-ÇTUÖM yönteminin ikinci aşamasında GA'dan yararlanılarak ÇTUÖM için dalgacık aktivasyon fonksiyonuna ait parametrelerin en iyi değerlerinin seçilebilmesi amaçlanır. Üçüncü aşamada ise karar

verilerek sınıflandırma işlemi gerçekleştirilir. Şekil 6.65'de hazırlanan sistemin blok diyagramı gösterilmektedir.



Şekil 6.65. Optimal sınıflandırma sistemi için geliştirilen GA-DF-ÇTUÖM yöntemine ait blok diyagram

GA yapısında 12 bitten oluşan bir birey oluşturulmuştur. Bu bitler şu amaçlarla kullanılmaktadır:

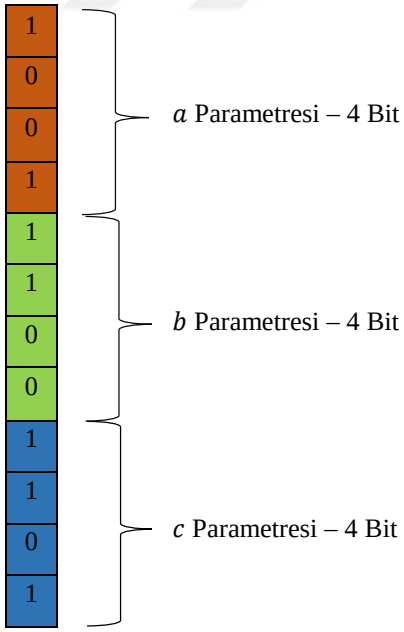
- İlk dört bit dalgacık çekirdek fonksiyonunda kullanılan a parametresinin 1 ile 16 arasında değer alabilmesi için kullanılmaktadır.
- İkinci dört bit dalgacık çekirdek fonksiyonunda kullanılan b parametresinin 1 ile 16 arasında değer alabilmesi için kullanılmaktadır.
- Üçüncü dört bit dalgacık çekirdek fonksiyonunda kullanılan c parametresinin 1 ile 16 arasında değer alabilmesi için kullanılmaktadır.

Tablo 6.14'de dalgacık çekirdek fonksiyonunda kullanılan parametrelerin almış olduğu değerler ve GA'da kullanılan ikili kodlamaları gösterilmektedir. Parametre değerleri bulunurken, bu değerlerin 1 ile 16 arasında tamsayı alınması yerine, daha önce Bölüm 6.4.4'de yapılan deneysel çalışmada 0 ile 1 arasındaki değerlerde daha iyi sonuç verdiğinin görülmesi sebebiyle değerler 1000 ile bölünerek ikinci bir en uygun değere getirme işlemi daha yapılmıştır. Daha küçük değerlere normalize edilmesi doğruluk değerinde küçük sayılacak değerlerde başarım oranını artırmasına karşın çalışma süresini çok fazla uzattığı için tercih edilmemiştir. ÇTUÖM'de gizli hücre katman sayısının bilinmesine gerek yoktur [93, 148]. Bundan dolayı GA ile ÇTUÖM içerisinde kullanılan dalgacık aktivasyon fonksiyonuna ait parametre değerleri hesaplanmıştır.

Tablo 6.14. Dalgacık çekirdek fonksiyonunun aldığı değerler ve bit kodlaması

a, b ve c Parametre Değerleri	Bit Kodlaması
1	0 0 0 0
2	0 0 0 1
3	0 0 1 0
4	0 0 1 1
5	0 1 0 0
6	0 1 0 1
7	0 1 1 0
8	0 1 1 1
9	1 0 0 0
10	1 0 0 1
11	1 0 1 0
12	1 0 1 1
13	1 1 0 0
14	1 1 0 1
15	1 1 1 0
16	1 1 1 1

Nüfusun oluşturulması için kullanılan 12 bitlik veriye ait örnek Şekil 6.66'da gösterilmektedir.



Şekil 6.66. GA ile geliştirilen uygulamada bireylerin seçilmesine ait bir örnek

İlk nüfusun belirlenmesinde rastgele bir şekilde 40 birey oluşturulmuştur. Böylece uç öğrenme makinesi sınıflandırıcısında mümkün olan en iyi sınıflandırma performansının elde edilmesi amaçlanmaktadır. Bu uygulamada 7000 adet veri eğitim için, 5250 adet veri ise test için kullanılmıştır. Bu uygulama on kez çalıştırılıp ortalama değerler hesaplanarak GA-DF-ÇTUÖM yönteminin performansı belirlenmeye çalışılmıştır.

GA'da ortalama doğruluk değeri uygunluk fonksiyonu olarak kullanılmıştır. GA-DF-ÇTUÖM uygulamasının test süresince dalgacık çekirdek fonksiyonuna ait parametrelerin GA tekniği ile en uygun değerlerinin bulunmasına çalışılmıştır.

GA-DF-ÇTUÖM yönteminin performansının analiz edilebilmesi için ayrıntıları bölüm 3.5.5.2'de verilen doğruluk değerlerinin ortalaması (DOD_{ort}), performans metriğinden faydalanılmıştır. Çalışma sonucunda (DOD_{ort}) değerinin yüzdesi % 95 üzerinde bulunmuştur. GA ile geliştirilen yaklaşım ile 1, 5 ve 8 parametre değerleri elde edilmiştir. Bu değerler 1000 ile bölünerek ikinci bir en uygun değere getirme işlemi uygulanarak a, b ve c parametrelerinin her seferinde farklı bir değer gelecek şekilde sırasıyla verilmiştir. GA sadece dalgacık fonksiyonuna ait parametre değerlerinin tespiti için kullanılmıştır. ÇTUÖM'nin sınıflandırma yapması için tekrar kullanılmasına gerek kalmamış, elde edilen veriler doğrudan sınıflandırıcıya verilmiştir. DF içerisinde kullanılan parametre değerlerinden c 'ye ait değerlerin küçülmesinin doğruluk oranına daha iyi derecede katkı yaptığı gözlemlenmiştir. Diğer parametre değerleri olan a ve b Denklem 4.11'den görüleceği üzere birbirini sadeleştirdiği için bu parametre değerlerine ikinci normalizasyon işleminin uygulanmasının sınıflandırıcının başarım değerini etkilemediği gözlemlenmiştir.

Bu tezin en önemli amacı olan ağ trafiğinin sınıflandırılmasının en uygun makine öğrenmesi algoritmaları ile sağlanabilmesi amacıyla kullanılan GA-DF-ÇTUÖM yönteminin kullanıldığı akıllı sistem ile elde edilen bulguların daha önceki kullanılan makine öğrenmesi yaklaşımlarından elde edilen bulgulara göre daha iyi sonuç verdiği gözlenmiştir. Önerilen sınıflandırma sistemi doğrudan özellik vektörüne uygulanabilmesi, hızlı eğitim ve test zamanına sahip olabilmesi gibi avantajlara sahiptir. Bu avantajları ile gerçek zamanlı çalışmalar için kullanılabilecek bir altyapıya sahiptir. Bu yaklaşım, sınıflandırma ve sınıflandırıcıya verilecek olan parametrelerin en uygun değerlerden seçilmesi olarak iki aşamadan oluşmaktadır. Ağ üzerinden alınan veri seti dalgacık aktivasyon fonksiyonunun kullanıldığı uç öğrenme makinesi sınıflandırıcısına girdi olarak verilmektedir.

ÇTUÖM sınıflandırıcısının iyi bir performans sergileyebilmesi için kullanılan dalgacık fonksiyonu içerisindeki parametrelerin uygun değerlerde seçilmesine ihtiyaç

vardır. En iyi değerlerin tespit edilebilmesini gerçekleştirmek için deneme yanılma yöntemi yerine akıllı bir sistem ile çözülebilmesi amacıyla GA tekniğinden faydalanılmıştır. GA dalgacık fonksiyonuna ait parametrelerin en uygun değerlerde seçilebilmesinde kullanılmıştır.

Tablo 6.15'de hem klasik uç öğrenme makinesi hem de GA-DF-ÇTUÖM yöntemi kullanılarak elde edilen verilerin uygulandığı UÖM yaklaşımı ile elde edilen doğruluk değeri yüzdeleri gösterilmektedir. En iyi sonucun dalgacık fonksiyonunda kullanılan a, b ve c parametrelerine sırasıyla 8, 5 ve 1 değerlerinin 1000 ile bölünerek verilmesi ile elde edildiği gözlenmiştir.

Tablo 6.15. Ağ trafiğinin sınıflandırılması amacıyla geliştirilen GA-DF-ÇTUÖM yöntemi ile klasik UÖM'ye ait parametre ve doğruluk yüzdelerinin karşılaştırılması

Kullanılan Yöntem	Aktivasyon Fonksiyonu	Dalgacık Fonksiyonu Parametresi (a)	Dalgacık Fonksiyonu Parametresi (b)	Dalgacık Fonksiyonu Parametresi (c)	Gizli Katman hücre Sayısı	Doğruluk Değeri (DOD_{ort}) Yüzdesi
Klasik UÖM	Tanjant Sigmoid	-	-	-	20	81,33
Klasik UÖM	Tanjant Sigmoid	-	-	-	40	84,17
Klasik UÖM	Tanjant Sigmoid	-	-	-	80	86,89
Klasik UÖM	Tanjant Sigmoid	-	-	-	160	88,13
Klasik UÖM	Tanjant Sigmoid	-	-	-	320	90,80
Klasik UÖM	Tanjant Sigmoid	-	-	-	640	92,11
Klasik UÖM	Radyal Temelli	-	-	-	20	80,12
Klasik UÖM	Radyal Temelli	-	-	-	40	81,15
Klasik UÖM	Radyal Temelli	-	-	-	80	82,22
Klasik UÖM	Radyal Temelli	-	-	-	160	84,19
Klasik UÖM	Radyal Temelli	-	-	-	320	84,95
Klasik UÖM	Radyal Temelli	-	-	-	640	87,30
Klasik UÖM	Sigmoid	-	-	-	20	65,23
Klasik UÖM	Sigmoid	-	-	-	40	68,16
Klasik UÖM	Sigmoid	-	-	-	80	71,02
Klasik UÖM	Sigmoid	-	-	-	160	73,30
Klasik UÖM	Sigmoid	-	-	-	320	74,91
Klasik UÖM	Sigmoid	-	-	-	640	76,00
Klasik UÖM	Sinüzodial	-	-	-	20	79,16
Klasik UÖM	Sinüzodial	-	-	-	40	80,24
Klasik UÖM	Sinüzodial	-	-	-	80	81,98
Klasik UÖM	Sinüzodial	-	-	-	160	84,19
Klasik UÖM	Sinüzodial	-	-	-	320	85,01
Klasik UÖM	Sinüzodial	-	-	-	640	86,30
Klasik UÖM	Keskin Limitli	-	-	-	20	60,21
Klasik UÖM	Keskin Limitli	-	-	-	40	64,22
Klasik UÖM	Keskin Limitli	-	-	-	80	65,30
Klasik UÖM	Keskin Limitli	-	-	-	160	68,90
Klasik UÖM	Keskin Limitli	-	-	-	320	70,10
Klasik UÖM	Keskin Limitli	-	-	-	640	72,84
Klasik UÖM	Triangular Temelli	-	-	-	20	76,12
Klasik UÖM	Triangular Temelli	-	-	-	40	77,90
Klasik UÖM	Triangular Temelli	-	-	-	80	80,01
Klasik UÖM	Triangular Temelli	-	-	-	160	83,60
Klasik UÖM	Triangular Temelli	-	-	-	320	85,00
Klasik UÖM	Triangular Temelli	-	-	-	640	87,49
DF-ÇTUÖM	Dalgacık	0,10	0,10	0,10	-	92,32
DF-ÇTUÖM	Dalgacık	0,01	0,01	0,01	-	95,04
DF-ÇTUÖM	Dalgacık	0.03	0.01	0.08	-	93,45
DF-ÇTUÖM	Dalgacık	0.02	0.01	0.08	-	93,81
DF-ÇTUÖM	Dalgacık	0.02	0.01	0.10	-	93,28
GA-DF-ÇTUÖM	Dalgacık	0.008	0.001	0.005	-	95,24
GA-DF-ÇTUÖM	Dalgacık	0,008	0,005	0,001	-	96,57
GA-DF-ÇTUÖM	Dalgacık	0,005	0,008	0,001	-	96,53
GA-DF-ÇTUÖM	Dalgacık	0,005	0,001	0,008	-	94,93
GA-DF-ÇTUÖM	Dalgacık	0,001	0,008	0,005	-	96,25
GA-DF-ÇTUÖM	Dalgacık	0,001	0,005	0,008	-	95,18

7. SONUÇLAR VE TARTIŞMA

İnternet trafik bilgisinin sınıflandırılması son zamanlarda popülerliği artan bir konu haline gelmiş ve önemi artmıştır. Özellikle kurumsal bilgisayar ağlarındaki trafik bilgisinin sınıflandırılması ağın performanslı bir şekilde kullanıcıları tarafından kullanılabilmesine olanak tanımakta ve ağ yöneticisine; verilen hizmet kalitesinin artırılabilmesi, istenmeyen trafiklerin engellenebilmesi, ağa sızma girişimlerinin tespit edilebilmesi, ağ üzerinde istenilen güvenlik politikalarının oluşturulabilmesi gibi avantajlar için imkân sağlamaktadır. Ağ trafiğinin analiz edilerek sınıflandırılması amacıyla daha önceki çalışmalarda çeşitli yaklaşımlar öne sürülmüştür. Bu tez çalışmasında makine öğrenmesi yaklaşımlarının güçlü yanları göz önüne alınarak sınıflandırma için bu yaklaşımlardan faydalanılmıştır.

Bu tez çalışması ağ trafiğinin sınıflandırılması amacıyla daha önce yapılan makine öğrenmesi tabanlı çalışmalara göre aşağıdaki yönleri ile farklılık göstermektedir:

- Tez çalışmasında sadece hazır bir veri seti kullanılmamıştır. Bu tez çalışması için veriler elde edilirken üç farklı veri toplama sistemi geliştirilmiştir. İlk önce sınıfları belli olan Cambridge Üniversitesine ait veriler analiz edilmiştir. İkinci alternatif olarak Fırat Üniversitesi sistem odası üzerinde ağ verilerinin dinlenerek sınıflandırılabilmesi amacıyla bir sistem geliştirilmiş ve sınıflandırılmamış bir şekilde ham veriler kaydedilmiştir. Üçüncü alternatif olarak CAIDA veri tabanına üye olunarak ham halde hazır bulunan veri kayıt dosyaları analiz edilmiştir. Ham verilerden öz niteliklerin çıkarılması ve sınıf etiketlerinin verilmesi için Cambridge Üniversitesi kampüs alanından verilerin sınıflandırılması amacıyla kullanılan açık kaynak kod tabanlı yazılımdan faydalanılmıştır.
- Daha önceki çalışmalarda sıklıkla kullanılan ve başarılı sonuçlar alınan NB ve DVM yaklaşımları ile daha önce ağ trafiğinin sınıflandırılmasında hemen hemen hiç kullanılmamış yeni gelişen bir makine öğrenmesi yaklaşımı olan UÖM veri setlerine uygulanarak performans karşılaştırması yapılmıştır. Aynı zamanda bu sınıflandırıcıların temel çalışma prensibine dayanan YSA kullanılarak da sınıflandırma yapılmıştır.
- Makine öğrenmesi yaklaşımlarının birbiriyle karşılaştırabilmesi amacıyla genellikle doğruluk değeri metriği kullanılırken bu çalışmada doğruluk değeri metriğinin yanında belirleyicilik, duyarlılık, F-Ölçüsü, kesinlik metrikleri ile

beraber AİK eğrileri çizilerek karşılaştırma yapılmıştır. Karşılaştırma yapılırken hem genel olarak hem de her bir sınıf için ayrı ayrı performans metriklerinden faydalanılarak karşılaştırma yapılmıştır. Elde edilen veri setlerinden Cambridge Üniversitesi kampüsünden alınan veriler tezin 6.4 bulgular bölümünde yapılan değerlendirmeler için kullanılmıştır. Fırat Üniversitesi ve CAIDA veri tabanından alınan veriler yerine bu veri setinin seçilmesindeki en büyük etken herkesin ulaşabileceği ve farklı sınıfların elde edilebileceği bir veritabanı olmasıdır.

- Sınıflandırıcı performansları karşılaştırılırken sadece performans metrikleri ile yetinilmemiş aynı zamanda sınıflandırıcıların bu sınıflandırmayı ne kadar sürede yaptıkları da analiz edilmiştir.
- Makine öğrenmesi yaklaşımlarından uç öğrenme makinesi yaklaşımı ile alınan değerlere ait performansların daha yüksek olduğu gözlendikten sonra farklı gizli katman hücre sayılarının seçilmesi ve farklı aktivasyon fonksiyonlarının kullanılmasının sınıflandırma başarıma etkisi analiz edilmiştir.
- Klasik uç öğrenme yaklaşımları ile alınan başarılı sonuçların yanında farklı bir uç öğrenme yaklaşımı olan ÇTUÖM yaklaşımı ile ayrıca analiz yapılmıştır. Bu analizler için farklı aktivasyon fonksiyonları ve aktivasyon fonksiyonlarına ait farklı parametre değerleri ile sınıflandırıcıya test verileri verilerek önceki yaklaşımlar ile karşılaştırma çalışmaları yapılmıştır.
- Dalgacık fonksiyonu yapısında kullanılan ve ÇTUÖM yaklaşımında kullanılan parametre değerlerinin en uygun değerlerde seçilmesinin sınıflandırıcının başarımında oldukça önemli olduğu gözlenmiştir.
- Dalgacık aktivasyon fonksiyonunun kullanıldığı ÇTUÖM algoritması için GA tekniklerinden faydalanılarak dalgacık fonksiyonunda kullanılan parametrelerin en uygun değerleri alabilmesi için akıllı bir yöntem geliştirilmiştir. GA-DF-ÇTUÖM adı verilen yöntem sayesinde dalgacık fonksiyonunda kullanılan a, b ve c parametre değerlerinin eniyileme yöntemi ile seçilebilmesi sağlanmıştır.
- İkinci bir eniyileme yaklaşımı olarak 1 ile 16 arasında tamsayı olarak bulunan dalgacık fonksiyonuna ait parametre değerleri daha önce hazırlanan ÇTUÖM yaklaşımındaki küçük değerlerde daha uygun sonuçlar alındığının gözlenmesi sebebiyle 1000'e bölünerek normalize edilmiştir. Daha küçük değerlerin alınması çalışma zamanını çok artırdığı fakat yüksek bir performans elde edilemediği için tercih edilmemiştir.

Tez kapsamında veri setindeki veriler sınıflandırıcılara verilmeden önce baskın sınıfların oluşmaması ve sınıflandırıcının ezber yapmaması için tüm değerlerin 0 ile 1 arasına normalize edilmiştir. Sınıflandırıcıya verilen 5250 adet test verisi sonuçları tek bir sefer yerine on kez çalıştırılarak değerlerin ortalaması alınmıştır. Kullanılan tüm sınıflandırıcılarda bu işlem tekrarlanmıştır.

YSA ve NB ile yapılan sınıflandırma yaklaşımlarında DVM ve UÖM yaklaşımlarına göre daha kötü sonuçlar elde edildiği gözlenmiştir. Özellikle YSA ile hazırlanan sınıflandırıcının çalışma süresi bu ağ trafiğinin gerçek zamanlı sınıflandırılabilmesi için ihtiyaç duyulacak sürelere göre çok daha uzun çıkmaktadır. DVM ile klasik UÖM yaklaşımına yakın sonuçlar elde edilmesine rağmen klasik UÖM yaklaşımında daha başarılı sınıflandırma sonuçları elde edilmiştir. Özellikle tanjant sigmoid ve radyal temelli aktivasyon fonksiyonlarının seçildiği UÖM yaklaşımlarında başarım oranlarının daha fazla arttığı gözlenmiştir. Keskin limitli aktivasyon fonksiyonunun kullanıldığı UÖM yaklaşımında ise en kötü sonuçların elde edildiği gözlenmektedir. Bu uygulama çalışmasında, en iyi performansı verecek gizli katman hücre sayısını bulabilmek için, gizli katman hücre sayısı 20'den başlanarak her seferinde iki kat artırılmıştır. Gizli katman hücre sayıları sırasıyla 20, 40, 80, 160, 320, 640 ve 1280 değerleri seçilmiştir. Sınıflandırıcının başarımının gizli katman hücre sayısının artmasına paralel olarak arttığı gözlenmesine karşın çalışma süresinin de uzadığı görülmüştür. 1280 gizli katman hücresi kullanımında ise 640 gizli katman hücresi ile yapılan analiz çalışmasına bir miktar başarım artışı olduğu gözlenmesine rağmen geçen çalışma süresinin de yüksek olması sebebiyle kazancın istenilen düzeyde olmadığı gözlemlenmiştir.

Bir sınıflandırıcının başarımında, doğru sınıflandırma oranı kadar doğru sınıflandırmayı mümkün olan en kısa zamanda yapabilmesi de önemlidir. Çalışma zamanı olarak UÖM yaklaşımlarının DVM ve NB algoritmalarına göre daha hızlı çalıştığı tespit edilmiştir.

Uygulanan makine öğrenmesi yaklaşımları içerisinde en başarılı sonuçlar DF-ÇTUÖM yaklaşımı ile elde edilmiştir. Seçilen aktivasyon fonksiyonunda kullanılan parametrelerin değiştirilmesi ile doğru sınıflandırılan değerlerin oranı artırılabilmektedir. Bu seçimlerin akıllı bir yöntem ile yapılabilmesi amacıyla geliştirilen GA-DF-ÇTUÖM yöntemi kullanılarak yapılan çalışma sonucunda (DOD_{ort}) değerinin yüzdesi en iyi % 96,57 olarak bulunmuştur. Bu uygulama çalışmasında kullanılan dalgacık çekirdek fonksiyonuna ait

parametreler 1, 5 ve 8 olarak GA tarafından belirlenmiştir. Bu değerler 1000 e bölünerek normalize edilmiş ve ÇTUÖM sınıflandırıcısında kullanılan DF parametrelerine gönderilmiştir.

Bu tez çalışması temel alınarak ileride gerçekleştirilmesi planlanan ve geliştirilmesinin kurumsal bilgisayar ağlarındaki internet trafiğinin akıllı sistemler ile sınıflandırılması amacına katkıda bulunacağı düşünülen çalışmalar aşağıda belirtilmiştir:

- Bu tez çalışmasında kullanılan veriler elde edilip, analiz için uygun hale getirildikten sonra sınıflandırıcılara verilmiştir. Uç öğrenme makinesinin hızlı çalışabilme avantajı göz önüne alınarak gerçek zamanlı çalışmalar için de kullanılabilecek şekilde analiz yapılabilmesi,
- Uç öğrenme makinesi algoritmalarını temel alan farklı yaklaşımların da verilerin analizlerinde uygulanarak performanslarının değerlendirilmesi,
- Yapılan analizler için hazırlanan yazılımların sonuçlarının görülebileceği bir yazılım ara yüzü oluşturularak kurumsal ağları yöneten ağ ve sistem yöneticilerinin kullanımına sunulabilmesi,
- Analiz edilen tüm verilerin analiz sonuçları ve zamanları ile beraber bir veri tabanında tutularak daha sonra yapılan analizler ile karşılaştırılabilmesinin sağlanması ve böylece ağ yöneticilerinin geçmişe dönük güvenlik politikalarını belirleyebilmeleri,
- Kurumsal ağlar üzerinde kullanılan ağ cihazları ile bütünleşik çalışabilecek ve veri alışverişinde bulunabilecek şekilde geliştirilen sınıflandırma algoritmalarının çalışabileceği donanım tabanlı sistemlerin geliştirilebilmesi,
- Günümüzün popüler çalışma alanlarından birisi olan büyük veri olarak adlandırabileceğimiz çok yüksek miktardaki verilerin işlenebilmesi için hızlı çalışan UÖM yaklaşımlarının geliştirilmesidir.

Sonuç olarak tez çalışmasında kurumsal bilgisayar ağlarından alınan internet trafik bilgisinin analiz edilerek sınıflandırılabilmesi ve bu sınıflandırılan verilerin özellikle ağ yöneticilerinin güvenlik politikaları geliştirmesinde ve böylece hizmet kalitesinin daha verimli bir şekilde sunulmasına katkı da bulunması amaçlanmıştır. Bu amacın gerçekleştirilebilmesi için makine öğrenmesi yaklaşımlarından faydalanılarak akıllı bir sistem geliştirilmiştir. Geliştirilen bu sistemin kullanılması ile özellikle kullanıcı başına düşen internete bağlı cihaz sayısının her geçen gün arttığı dijital dünyada internet trafiğinin

sınıflandırılması ve kullanıcıların internetten mümkün olan en iyi şekilde faydalanabilmeleri hedeflenmiştir. İnternet trafiğinin sınıflandırılması için daha önceki çalışmalarda hemen hemen hiç kullanılmamış olan UÖM yaklaşımları kullanılmıştır. UÖM'nin başarımını karşılaştırabilmek amacıyla daha önce kullanılan makine öğrenmesi algoritmalarından DVM, NB ve YSA yaklaşımları da elde edilen verilere uygulanarak karşılaştırılmıştır. UÖM algoritmaları ile yapılan sınıflandırma da diğer makine öğrenmesi algoritmalarına göre daha hızlı çalıştığı ve daha yüksek oranda doğruluk değerlerine ulaşıldığı gözlenmiştir. Tez verilerine hem klasik UÖM hem de ÇTUÖM yaklaşımları uygulanarak sınıflandırma yapılmıştır. Sınıflandırıcıların tespit ettiği her bir sınıf için AİK eğrileri oluşturulmuştur. Özellikle dalgacık fonksiyonunun kullanıldığı ÇTUÖM algoritmasında kullanılan parametrelerin iyi derecede seçimi için GA-DF-ÇTUÖM uygulaması geliştirilmiştir.

KAYNAKLAR

[1] **IANA**, http://www.iana.org/assignments/port-numbers. 01.12.2014

[2] **Karagiannis, T., Broido, A., Brownlee, N., Claffy, K., ve Faloutsos, M.** (2003). File-sharing in the Internet: A characterization of P2P traffic in the backbone.University of California, Riverside, USA, Tech. Rep.

[3] **Karagiannis, T., Broido, A., Brownlee, N., Claffy, K. C., ve Faloutsos, M.** (2004). Is p2p dying or just hiding? [p2p traffic measurement]. Global Telecommunications Konferansı, 2004. GLOBECOM'04. IEEE, **3**, 1532-1538.

[4] **Moore, A. W.** (2005). Toward the accurate Identification of network applications, in poc. 6. passive active measurement. Workshop (PAM) 3431, 41-54.

[5] **Karagiannis, T., Broido, A., ve Faloutsos, M.** (2004). Transport layer identification of P2P traffic. In Proceedings of the 4th ACM SIGCOMM conference on Internet measurement, 121-134.

[6] **Sen, S., Spatscheck, O., ve Wang, D.** (2004). Accurate, scalable in-network identification of p2p traffic using application signatures. In Proceedings of the 13th international conference on World Wide Web, ACM, 512-521.

[7] **L7 Fillter**, http://l7-filter.sourceforge.net. 01.12.2014

[8] **PACE**, https://www.ipoque.com/products/pace. 01.12.2014

[9] **nDPI**, http://www.ntop.org/products/deep-packet-inspection/ndpi. 01.12.2014

[10] **Carela Español, V.** (2014). Network traffic classification: from theory to practice, *Doktora Tezi*, Universitat Politecnica de Catalunya BarcelonaTech, Department d'Arquitectura de Computadors, Barcelona.

[11] **Auld, T., Moore, A. W., ve Gull, S. F.** (2007). Bayesian neural networks for internet traffic classification. *Neural Networks*, IEEE Transactions on, **18/1**, 223-239.

[12] **Crotti, M., Dusi, M., Gringoli, F., ve Salgarelli, L.** (2007). Traffic classification through simple statistical fingerprinting. ACM SIGCOMM Computer Communication Review, **37/1**, 5-16.

[13] **Haffner, P., Sen, S., Spatscheck, O., ve Wang, D.** (2005). ACAS: automated construction of application signatures. In Proceedings of the 2005 ACM SIGCOMM workshop on Mining network data, ACM. 197-202.

[14] **Jiang, H., Moore, A. W., Ge, Z., Jin, S., ve Wang, J.** (2007). Lightweight application classification for network management. In Proceedings of the 2007 SIGCOMM workshop on Internet network management, 299-304. ACM.

[15] **Moore, A. W., ve Zuev, D.** (2005). Internet traffic classification using bayesian analysis techniques. In ACM SIGMETRICS Performance Evaluation Review, **33/1**, 50-60. ACM.

[16] **Roughan, M., Sen, S., Spatscheck, O., ve Duffield, N.** (2004). Class-of-service mapping for QoS: a statistical signature-based approach to IP traffic classification. In Proceedings of the 4th ACM SIGCOMM conference on Internet measurement, 135-148. ACM.

[17] **Zuev, D., ve Moore, A. W.** (2005). Traffic classification using a statistical approach. In Passive and Active Network Measurement, 321-324. Springer Berlin Heidelberg.

[18] **Szabo, G., Szabó, I., ve Orincsay, D.** (2007). Accurate traffic classification. In World of Wireless, Mobile and Multimedia Networks, 2007. WoWMoM 2007. 1-8.

[19] **Bernaille, L., Teixeira, R., ve Salamatian, K.** (2006). Early application identification. In Proceedings of the 2006 ACM CoNEXT conference, 1-6. ACM.

[20] **Bernaille, L., Teixeira, R., Akodkenou, I., Soule, A., ve Salamatian, K.** (2006). Traffic classification on the fly. ACM SIGCOMM Computer Communication Review, **36/2**, 23-26.

[21] **Bernaille, L., ve Teixeira, R.** (2007). Early recognition of encrypted applications. In Passive and Active Network Measurement,165-175. Springer Berlin Heidelberg.

[22] **Dainotti, A., De Donato, W., Pescape, A., ve Rossi, P. S.** (2008). Classification of network traffic via packet-level hidden Markov models. InGlobal Telecommunications Conference, 2008. IEEE GLOBECOM 2008. 1-5.

[23] **Erman, J., Arlitt, M., ve Mahanti, A.** (2006). Traffic classification using clustering algorithms. In Proceedings of the 2006 SIGCOMM workshop on Mining network data, 281-286. ACM.

[24] **Erman, J., Mahanti, A., Arlitt, M., Cohen, I., ve Williamson, C.** (2007). Offline/realtime traffic classification using semi-supervised learning. Performance Evaluation, **64/9**, 1194-1213.

[25] **Erman, J., Mahanti, A., Arlitt, M., ve Williamson, C.** (2007). Identifying and discriminating between web and peer-to-peer traffic in the network core. InProceedings of the 16th international conference on World Wide Web, 883-892. ACM.

[26] **Erman, J., Mahanti, A., ve Arlitt, M.** (2007). Byte me: a case for byte accuracy in traffic classification. In Proceedings of the 3rd annual ACM workshop on Mining network data, 35-38. ACM.

[27] **Soule, A., Salamatia, K., Taft, N., Emilion, R., ve Papagiannaki, K.** (2004). Flow classification by histograms: or how to go on safari in the internet. ACM SIGMETRICS Performance Evaluation Review, **32/1**, 49-60.

[28] **Zander, S., Nguyen, T., ve Armitage, G.** (2005). Self-learning IP traffic classification based on statistical flow characteristics. In Passive and Active Network Measurement, 325-328. Springer Berlin Heidelberg.

[29] **Zander, S., Nguyen, T., ve Armitage, G.** (2005). Automated traffic classification and application identification using machine learning. In Local Computer Networks, 250-257.

[30] **Nguyen, T. T., ve Armitage, G.** (2008). A survey of techniques for internet traffic classification using machine learning. Communications Surveys and Tutorials, IEEE, **10/4**, 56-76.

[31] **Li, W., ve Moore, A. W.** (2007). A machine learning approach for efficient traffic classification. In Modeling, Analysis, and Simulation of Computer and Telecommunication Systems, 2007. MASCOTS'07, 310-317.

[32] **Callado, A., Kamienski, C., Szabó, G., Gerö, B. P., Kelner, J., Fernandes, S., ve Sadok, D.** (2009). A survey on internet traffic identification. Communications Surveys & Tutorials, **11(3)**, 37-52.

[33] **Dainotti, A., Pescape, A., ve Claffy, K. C.** (2012). Issues and future directions in traffic classification. Network, IEEE, **26(1)**, 35-40.

[34] **Qu, B., Zhang, Z., Guo, L., ve Meng, D.** (2012). On accuracy of early traffic classification. 7. International Conference, In Networking, Architecture and Storage (NAS), 348-354.

[35] **Este, A., Gringoli, F., ve Salgarelli, L.** (2009). On the stability of the information carried by traffic flow features at the packet level. ACM SIGCOMM Computer Communication Review, **39(3)**, 13-18.

[36] **Huang, N. F., Jai, G. Y., ve Chao, H. C.** (2008). Early identifying application traffic with application characteristics. In Communications, 2008. ICC'08. IEEE International Conference, 5788-5792.

[37] **Huang, N. F., Jai, G. Y., Chao, H. C., Tzang, Y. J., ve Chang, H. Y.** (2013). Application traffic classification at the early stage by characterizing application rounds. Information Sciences, **232**, 130-142.

[38] **Hullár, B., Laki, S., ve György, A.** (2011). Early identification of peer-to-peer traffic. In Communications (ICC), 2011 IEEE International Conference on IEEE. 1-6.

[39] **Nguyen, T. T., Armitage, G., Branch, P., ve Zander, S.** (2012). Timely and continuous machine-learning-based classification for interactive IP traffic. IEEE/ACM Transactions on Networking (TON), **20(6)**, 1880-1894.

[40] **Rizzi, A., Colabrese, S., ve Baiocchi, A.** (2013). Low complexity, high performance neuro-fuzzy system for internet traffic flows early classification. In Wireless Communications and Mobile Computing Conference (IWCMC), 77-82.

[41] **Karagiannis, T., Papagiannaki, K., ve Faloutsos, M.** (2005). BLINC: multilevel traffic classification in the dark. In ACM SIGCOMM Computer Communication Review, ACM. **35/4**, 229-240.

[42] **Xu, K., Zhang, Z. L., ve Bhattacharyya, S.** (2005). Profiling internet backbone traffic: behavior models and applications. In ACM SIGCOMM Computer Communication Review ACM. **35/4**, 169-180.

[43] **Karagiannis, T., Papagiannaki, K., Taft, N., ve Faloutsos, M.** (2007). Profiling the end host. In Passive and Active Network Measurement186-196. Springer Berlin Heidelberg.

[44] **Mori, T., Kawahara, R., Hasegawa, H., ve Shimogawa, S**. (2010). Characterizing traffic flows originating from large-scale video sharing services. In Traffic MonitoriL-10ng and Analysis, 17-31. Springer Berlin Heidelberg.

[45] **Yoon, S. H., Park, J. W., Park, J. S., Oh, Y. S., ve Kim, M. S.** (2009). Internet application traffic classification using fixed IP-port. In Management Enabling the Future Internet for Changing Business and New Computing Services 21-30. Springer Berlin Heidelberg.

[46] **Kim, H., Fomenkov, M., Claffy, K., Brownlee, N., Barman, D., ve Faloutsos, M.** (2007). Comparison of internet traffic classification tools. IMRG WACI.

[47] **Xu, K., Wang, F., ve Gu, L.** (2011). Network-aware behavior clustering of Internet end hosts. In INFOCOM, 2011 Proceedings IEEE, 2078-2086.

[48] **Madhukar, A., ve Williamson, C**. (2006). A longitudinal study of P2P traffic classification. In Modeling, Analysis, and Simulation of Computer and Telecommunication Systems, 2006. MASCOTS 2006. 14th IEEE International Symposium, IEEE, 179-188.

[49] **Alcock, S., ve Nelson, R.** (2012). Libprotoident: Traffic Classification Using Lightweight Packet Inspection. WAND Network Research Group, Tech. Rep.

[50] **Antoniades, D., Polychronakis, M., Antonatos, S., Markatos, E. P., Ubik, S., ve Øslebø, A.** (2006). Appmon: An application for accurate per application network traffic characterization. In IST Broadband Europe 2006 Conference.

[51] **Kartal, E.** (2015). Sınıflandırmaya Dayalı Makine Öğrenmesi Teknikleri ve Kardiyolojik Risk Değerlendirmesine İlişkin Bir Uygulama, *Doktora Tezi*, İstanbul Üniversitesi, Fen Bilimleri Enstitüsü, İstanbul.

[52] **Nilsson, N. J.** (2009). The quest for artificial intelligence. Cambridge University Press.

[53] **Alpaydin, E.** (2014). Introduction to machine learning. MIT press.

[54] **Mohri, M., Rostamizadeh, A., ve Talwalkar, A.** (2012). Foundations of machine learning. MIT press.

[55] **Mitchell, T. M**. (1997). Machine learning. WCB.

[56] **Yiğidim, H.A.** (2012). Makine Öğrenme Algoritmalarını Kullanarak Ağ Trafiğinin Sınıflandırılması, *Yüksek Lisans Tezi*, TOBB Ekonomi ve Teknoloji Üniversitesi, Fen Bilimleri Enstitüsü, Ankara.

[57] **McGregor, A., Hall, M., Lorier, P., ve Brunskill, J.** (2004). Flow clustering using machine learning techniques. In Passive and Active Network Measurement, 205-214. Springer Berlin Heidelberg.

[59] **Williams, N., Zander, S., ve Armitage, G.** (2006). Evaluating machine learning algorithms for automated network application identification. Center for Advanced Internet Architectures, CAIA, Technical Report B, 60410, 2006.

[60] **Williams, N., Zander, S., ve Armitage, G.** (2006). A preliminary performance comparison of five machine learning algorithms for practical IP traffic flow classification. ACM SIGCOMM Computer Communication Review, **36/5**, 5-16.

[61] **Singh, K., ve Agrawal, S.** (2011). Comparative analysis of five machine learning algorithms for IP traffic classification. In Emerging Trends in Networks and Computer Communications (ETNCC), 2011 International Conference on IEEE. 33-38.

[62] **Singh, K., ve Agrawal, S.** (2011). 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 (1), 25-32.

[63] **Singh, K., ve Agrawal, S.** (2011). Feature extraction based IP traffic classification using machine learning. In Proceedings of the International Conference on Advances in Computing and Artificial Intelligence, 208-212. ACM.

[64] **Bishop, C. M.** (2006). Pattern recognition and machine learning. springer.

[65] **Camastra, F., ve Vinciarelli, A.** (2008). Machine learning for audio, image and video analysis. Advanced Information and Knowledge Processing, 83-89.

[66] **Harrington, P.** (2012). Machine learning in action, 28-178. Manning.

[67] **Shearer, C.** (2000). The CRISP-DM model: the new blueprint for data mining. Journal of data warehousing, **5/4**, 13-22.

[68] **Moore, A., Zuev, D., ve Crogan, M.** (2005). Discriminators for use in flow-based classification. Queen Mary and Westfield College, Department of Computer Science.

[69] **CAIDA**, https://data.caida.org/datasets/passive-2015/

[70] **Han, J., ve Kamber, M.** (2000). Data mining: concepts and techniques (the Morgan Kaufmann Series in data management systems).

[71] **Vellido, A., Martin-Guerroro, J. D., ve Lisboa, P.** (2012). Making machine learning models interpretable. In Proceedings of the 20th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN). Bruges, Belgium, 163-172.

[72] **Flach, P.** (2004). The many faces of ROC analysis in machine learning. ICML Tutorial.

[73] **Çetin, O.** (2014). Yapay Sinir Ağlarının Uyarlanabilir Donanımsal Yapılarda Gerçeklenmesi, *Doktora Tezi*, Sakarya Üniversitesi, Fen Bilimleri Enstitüsü, Sakarya.

[74] **McCulloch, W. S., ve Pitts, W.** (1943). A logical calculus of the ideas immanent in nervous activity. The bulletin of mathematical biophysics, **5(4),** 115-133.

[75] **Demirel, H.** (2010). Peltier Elemanlarla Hipotermiya Uygulamasında Sıcaklık Kontrolü ve Yapay Sinir Ağları ile Modellenmesi, *Doktora Tezi*, Gazi Üniversitesi, Bilişim Enstitüsü, Ankara.

[76] **Öztemel, E.** (2003). Yapay Sinir Ağları, Papatya Yayıncılık, İstanbul.

[77] **Kim, H., Claffy, K. C., Fomenkov, M., Barman, D., Faloutsos, M., ve Lee, K.** (2008). Internet traffic classification demystified: myths, caveats, and the best practices. In Proceedings of the 2008 ACM CoNEXT, 11. ACM.

[78] **Değerli, O.** (2012). Naive Bayes ile Blog İçeriklerinin Sınıflandırılması, *Yüksek Lisans Tezi*, Gazi Üniversitesi, Bilişim Enstitüsü, Ankara.

[79] **Este, A., Gringoli, F., ve Salgarelli, L.** (2009). Support vector machines for TCP traffic classification. *Computer Networks*, **53/14**, 2476-2490.

[80] **Peng, X., Qiong, L., ve Sen, L.** (2009). Internet traffic classification using support vector machine [j]. Journal of computer research and development, 3, 010.

[81] **Li, Z., Yuan, R., ve Guan, X.** (2007). Accurate classification of the internet traffic based on the svm method. In Communications, 2007. ICC'07. IEEE International Conference on IEEE. 1373-1378.

[82] **Wang, R., Liu, Y., Yang, Y., ve Zhou, X.** (2006). Solving the app-level classification problem of P2P traffic via optimized support vector machines. InIntelligent Systems Design and Applications, 2006. ISDA'06. Sixth International Conference on IEEE. **2**, 534-539.

[83] **Yuan, R., Li, Z., Guan, X., ve Xu, L.** (2010). An SVM-based machine learning method for accurate internet traffic classification. *Information Systems Frontiers*, **12/2**, 149-156.

[84] **Gómez Sena, G., ve Belzarena, P.** (2009). Early traffic classification using support vector machines. In Proceedings of the 5th International Latin American Networking Conference, 60-66. ACM.

[85] **Yang, Y. X., Wang, R., Liu, Y., ve Zhou, X. Y.** (2007). Solving p2p traffic identification problems via optimized support vector machines. In Computer Systems and Applications, 2007. AICCSA'07. IEEE/ACS International Conference on IEEE. 165-171.

[86] **Vapnik, V.** (2013). The nature of statistical learning theory. Springer Science ve Business Media.

[87] **Çatak, F.** (2014). Bulut bilişim sistemlerinde eşle/indirge yöntemi uygulanarak veri madenciliği yazılım çatısının geliştirilmesi, *Doktora Tezi*, İstanbul Üniversitesi, Enformatik Bölümü, İstanbul.

[88] **Platt, J. C., Cristianini, N., ve Shawe-Taylor, J.** (1999). Large Margin DAGs for Multiclass Classification. In nips, **12**, 547-553.

[89] **Dietterich, T. G., ve Bakiri, G.** (1995). Solving multiclass learning problems via error-correcting output codes. *Journal of artificial intelligence research*, 263-286.

[90] **Passerini, A., Pontil, M., ve Frasconi, P.** (2004). New results on error correcting output codes of kernel machines. *Neural Networks*, IEEE Transactions on, **15/1**, 45-54.

[91] **Huang, G. B., Zhu, Q. Y., ve Siew, C. K.** (2006). Extreme learning machine: theory and applications. Neurocomputing**, 70/1**, 489-501.

[92] **Huang, G. B., Zhu, Q. Y., ve Siew, C. K.** (2004). Extreme learning machine: a new learning scheme of feedforward neural networks. In Neural Networks, 2004. Proceedings. 2004 IEEE International Joint Conference on IEEE, **2**, 985-990.

[93] **Huang, G. B., Zhou, H., Ding, X., ve Zhang, R.** (2012). Extreme learning machine for regression and multiclass classification. Systems, Man, and Cybernetics, *Part B: Cybernetics*, IEEE Transactions on, **42/2**, 513-529.

[94] **Huang, G. B., ve Chen, L.** (2007). Convex incremental extreme learning machine. *Neurocomputing*, **70/16**, 3056-3062.

[95] **Huang, G. B., ve Chen, L.** (2008). Enhanced random search based incremental extreme learning machine. *Neurocomputing*, **71/16**, 3460-3468.

[96] **Huang, G. B., Ding, X., ve Zhou, H.** (2010). Optimization method based extreme learning machine for classification. *Neurocomputing*, **74/1**, 155-163.

[97] **Li, M. B., Huang, G. B., Saratchandran, P., ve Sundararajan, N.** (2005). Fully complex extreme learning machine. *Neurocomputing*, **68**, 306-314.

[98] **Zhu, Q. Y., Qin, A. K., Suganthan, P. N., ve Huang, G. B.** (2005). Evolutionary extreme learning machine. *Pattern recognition*, **38/10**, 1759-1763.

[99] **Huang, G. B., Wang, D. H., ve Lan, Y.** (2011). Extreme learning machines: a survey. *International Journal of Machine Learning and Cybernetics*, **2/2**, 107-122.

[100] **Miche, Y., Sorjamaa, A., Bas, P., Simula, O., Jutten, C., ve Lendasse, A.** (2010). OP-ELM: optimally pruned extreme learning machine. *Neural Networks*, IEEE Transactions on, **21/1**, 158-162.

[101] **Rong, H. J., Ong, Y. S., Tan, A. H., ve Zhu, Z.** (2008). A fast pruned-extreme learning machine for classification problem. Neurocomputing, **72/1**, 359-366.

[102] **Lan, Y., Soh, Y. C., ve Huang, G. B.** (2009). Ensemble of online sequential extreme learning machine. *Neurocomputing*, **72/13**, 3391-3395.

[103] **Zhang, R., Huang, G. B., Sundararajan, N., ve Saratchandran, P.** (2007). Multicategory classification using an extreme learning machine for microarray gene expression cancer diagnosis. IEEE/ACM Transactions on *Computational Biology and Bioinformatics* (TCBB), **4/3**, 485-495.

[104] **Deng, W., Zheng, Q., ve Chen, L.** (2009). Regularized extreme learning machine. In Computational Intelligence and Data Mining, 2009. CIDM'09. IEEE Symposium on IEEE, 389-395.

[105] **Huang, G. B., Li, M. B., Chen, L., ve Siew, C. K.** (2008). Incremental extreme learning machine with fully complex hidden nodes. *Neurocomputing*, **71/4**, 576-583.

[106] **Cao, J., Lin, Z., Huang, G. B., ve Liu, N.** (2012). Voting based extreme learning machine. Information Sciences, **185/1**, 66-77.

[107] **Huang, G. B., Liang, N. Y., Rong, H. J., Saratchandran, P., ve Sundararajan, N.** (2005). On-Line Sequential Extreme Learning Machine. *Computational Intelligence*, 232-237.

[108] **Avci, E., ve Coteli, R.** (2012). A new automatic target recognition system based on wavelet extreme learning machine. *Expert Systems with Applications*, **39/16**, 12340-12348.

[109] **Avci, E.** (2013). A new method for expert target recognition system: Genetic wavelet extreme learning machine (GAWELM). *Expert Systems with Applications*, **40/10**, 3984-3993.

[110] **Alcin, O. F., Sengur, A., Ghofrani, S., ve Ince, M. C.** (2014). GA-SELM: Greedy algorithms for sparse extreme learning machine. *Measurement*, **55**, 126-132.

[111] **Alçin, Ö. F., Şengür, A., ve İnce, M. C.** (2015). İleri-Geri Takip Algoritması Tabanlı Seyrek Aşırı Öğrenme Makinesi. *Journal of the Faculty of Engineering & Architecture of Gazi University*, **30/1**.

[112] **Luo, J., Vong, C. M., ve Wong, P. K.** (2014). Sparse bayesian extreme learning machine for multi-classification. *Neural Networks and Learning Systems*, IEEE Transactions on, **25/4**, 836-843.

[113] **Banerjee, K. S.** (1973). Generalized inverse of matrices and its applications, *Technometrics*, **15/1**, 197-197.

[114] **Dua, S., ve Chowriappa, P.** (2012). Data mining for bioinformatics. CRC Press.

[115] **Saharidis, G. K. D., Androulakis, I. P., ve Ierapetritou, M. G.** (2011). Model building using bi-level optimization. *Journal of Global Optimization*, **49/1**, 49-67.

[116] **Martinasek, Z., Hajny, J., ve Malina, L.** (2014). Optimization of power analysis using neural network. In Smart Card Research and Advanced Applications, 94-107. Springer International Publishing.

[117] **Remesan, R., ve Mathew, J.** (2014). Hydrological Data Driven Modelling: A Case Study Approach, **1**. Springer.

[118] **Srivastava, R. (Ed.).** (2013). Research Developments in Computer Vision and Image Processing: Methodologies and Applications: Methodologies and Applications. IGI Global.

[119] **Guil-Reyes, F., ve Daza-Gonzalez, M. T.** (2011). Summarizing frequent itemsets via pignistic transformation. In Progress in Artificial Intelligence, **297-310**. Springer Berlin Heidelberg.

[120] **Olson, D. L., ve Delen, D.** (2008). Advanced data mining techniques. Springer Science ve Business Media.

[121] **Baker, D. R.** (2007). A Hybrid Approach to Expert and Model Based Effort Estimation. ProQuest.

[122] **Japkowicz, N., ve Shah, M.** (2011). Evaluating learning algorithms: a classification perspective. Cambridge University Press.

[123] **Hamel, L.** (2009). Model Assessment with ROC Curves.

[124] **Fawcett, T.** (2006). An introduction to ROC analysis. *Pattern recognition letters*, **27/8**, 861-874.

[125] **Forman, G.** (2003). An extensive empirical study of feature selection metrics for text classification. *The Journal of machine learning research*, **3**, 1289-1305.

[126] **Caruana, R., ve Niculescu-Mizil, A.** (2004). Data mining in metric space: an empirical analysis of supervised learning performance criteria. InProceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining, 69-78. ACM.

[127] **Hill, M.** (2007). U.S. Patent No. 7,292,531. *Washington, DC: U.S. Patent and Trademark Office*.

[128] **Wu, T. F., Lin, C. J., ve Weng, R. C.** (2004). Probability estimates for multi-class classification by pairwise coupling. *The Journal of Machine Learning Research*, **5**, 975-1005.

[129] **Sokolova, M., ve Lapalme, G.** (2009). A systematic analysis of performance measures for classification tasks. *Information Processing ve Management*, **45/4**, 427-437.

[130] **Felkin, M.** (2007). Comparing classification results between n-ary and binary problems. In Quality Measures in Data Mining, 277-301. Springer Berlin Heidelberg.

[131] **Hanley, J. A., ve McNeil, B. J.** (1982). The meaning and use of the area under a receiver operating characteristic (ROC) curve. *Radiology*, **143/1**, 29-36.

[132] **Bradley, A. P.** (1997). The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern recognition*, **30/7**, 1145-1159.

[133] **Pencina, M. J., D'Agostino, R. B., D'Agostino, R. B., ve Vasan, R. S.** (2008). Evaluating the added predictive ability of a new marker: from area under the ROC curve to reclassification and beyond. Statistics in medicine, **27/2**, 157.

[134] **Huang, G. B., Chen, L., ve Siew, C. K.** (2006). Universal approximation using incremental constructive feedforward networks with random hidden nodes.Neural Networks, IEEE Transactions on, **17/4**, 879-892.

[135] **Zhou, H.** (2013). Extreme learning machine for classification and regression, *Doktora Tezi*, Nanyang Technologıcal Unıversıty, Singapur.

[136] **Miche, Y., Sorjamaa, A., ve Lendasse, A.** (2008). OP-ELM: theory, experiments and a toolbox. In Artificial Neural Networks-ICANN 2008, 145-154. Springer Berlin Heidelberg.

[137] **Liang, N. Y., Huang, G. B., Saratchandran, P., ve Sundararajan, N**. (2006). A fast and accurate online sequential learning algorithm for feedforward networks. *Neural Networks, IEEE Transactions on*, **17/6**, 1411-1423.

[138] **Cao, J., Lin, Z., ve Huang, G. B.** (2010). Composite function wavelet neural networks with extreme learning machine. *Neurocomputing*, **73/7**, 1405-1416.

[140] **MartíNez-MartíNez, J. M., Escandell-Montero, P., Soria-Olivas, E., MartíN-Guerrero, J. D., Magdalena-Benedito, R., ve GóMez-Sanchis, J.** (2011). Regularized extreme learning machine for regression problems. *Neurocomputing*, **74/17**, 3716-3721.

[141] **Savitha, R., Suresh, S., ve Sundararajan, N.** (2012). Fast learning circular complex-valued extreme learning machine (CC-ELM) for real-valued classification problems. *Information Sciences*, **187**, 277-290.

[142] **Soria-Olivas, E., Gómez-Sanchis, J., Martín, J. D., Vila-Francés, J., Martínez, M., Magdalena, J. R., ve Serrano, A. J.** (2011). BELM: Bayesian extreme learning machine. *Neural Networks, IEEE Transactions on*, **22/3**, 505-509.

[143] **Tang, X., ve Han, M.** (2009). Partial Lanczos extreme learning machine for single-output regression problems. *Neurocomputing*, **72/13**, 3066-3076.

[144] **Yin, J., Dong, F., ve Wang, N.** (2009). Modified Gram-Schmidt Algorithm for Extreme Learning Machine. In Computational Intelligence and Design, 2009. ISCID'09. Second International Symposium on IEEE, **2**, 517-520.

[145] **Albert, A.** (1972). Regression and the Moore-Penrose pseudoinverse. Elsevier.

[146] **Qi, L., ve Jiang, H.** (1997). Semismooth Karush-Kuhn-Tucker equations and convergence analysis of Newton and quasi-Newton methods for solving these equations. Mathematics of Operations Research, **22/2**, 301-325.

[147] **Cortes, C., ve Vapnik, V.** (1995). Support-vector networks. *Machine learning*, **20/3**, 273-297.

[148] **Li, B., Rong, X., ve Li, Y.** (2014). An Improved Kernel Based Extreme Learning Machine for Robot Execution Failures. The Scientific World Journal, 2014.

[149] **Wang, N., Han, M., Dong, N., ve Er, M. J.** (2014). Constructive multi-output extreme learning machine with application to large tanker motion dynamics identification. *Neurocomputing*, **128**, 59-72.

[150] **Heeswijk, M.** (2015). Advances in Extreme Learning Machines, *Doktora Tezi*, Aolto Üniversity, Helsinki.

[151] **Frénay, B., ve Verleysen, M.** (2010). Using SVMs with randomised feature spaces: an extreme learning approach. In ESANN.

[152] **Frénay, B., ve Verleysen, M.** (2011). Parameter-insensitive kernel in extreme learning for non-linear support vector regression. *Neurocomputing*, **74/16**, 2526-2531.

[153] **Parviainen, E., Riihimäki, J., Miche, Y., ve Lendasse, A.** (2010). Interpreting Extreme Learning Machine as an Approximation to an Infinite Neural Network. In KDIR, 65-73.

[154] **Liu, X., Wang, L., Huang, G. B., Zhang, J., ve Yin, J.** (2015). Multiple kernel extreme learning machine. *Neurocomputing*, **149**, 253-264.

[155] **Kasun, L. L. C., Zhou, H., Huang, G. B., ve Vong, C. M.** (2013). Representational learning with ELMs for big data. IEEE Intelligent Systems, **28/6**, 31-34.

[156] **Tapson, J., de Chazal, P., ve van Schaik, A.** (2015). Explicit computation of input weights in extreme learning machines. In Proceedings of ELM-2014, **1**, 41-49. Springer International Publishing.

[157] **McDonnell, M. D., Tissera, M. D., van Schaik, A., ve Tapson, J.** (2014). Fast, simple and accurate handwritten digit classification using extreme learning machines with shaped input-weights.

[158] **Huang, G., Song, S., Gupta, J. N., ve Wu, C.** (2014). Semi-supervised and unsupervised extreme learning machines. *Cybernetics*, IEEE Transactions on, **44/12**, 2405-2417.

[159] **Holland, J. H.** (1975). Adaption in natural and artificial systems. Ann Arbor MI: The University of Michigan Press.

[160] **Goldberg, D. E., ve Holland, J. H.** (1988). Genetic algorithms and machine learning. *Machine learning*, **3/2**, 95-99.

[161] **Schaffer, J. D.** (1985). Some experiments in machine learning using vector evaluated genetic algorithms. Vanderbilt Univ., Nashville, TN (USA).

[162] **Lee, C. Y., Piramuthu, S., ve Tsai, Y. K.** (1997). Job shop scheduling with a genetic algorithm and machine learning. International Journal of production research, **35/4**, 1171-1191.

[163] **Yang, J., ve Honavar, V.** (1998). Feature subset selection using a genetic algorithm. Feature extraction, construction and selection, 117-136. Springer US.

[164] **Booker, L. B., Goldberg, D. E., ve Holland, J. H.** (1989). Classifier systems and genetic algorithms. *Artificial intelligence*, **40/1**, 235-282.

[165] **Grefenstette, J. J.** (1993). Genetic algorithms and machine learning. In Proceedings of the sixth annual conference on Computational learning theory, 3-4. ACM.

[166] **Baş, M.** (2015). Sarmal ve Çevik Yazılım Geliştirme Çizelgesinin Sezgisel Yöntemlerle Optimizasyonu ve Karşılaştırması, *Yüksek Lisans Tezi*, Bahçeşehir Üniversitesi, Fen Bilimleri Enstitüsü, Bilgi Teknolojileri, İstanbul.

[167] **Horat, B.** (2014). Genetik Algoritma ile Ankara İlinde Mobese (Kent Güvenlik Yönetim Sistemi) Yerleştirme Optimizasyonu, *Yüksek Lisans Tezi*, Gazi Üniversitesi, Bilişim Enstitüsü, Ankara.

[168] **Coşkun, A.** (2006). Genetik Algoritma Kullanılarak Kimyasal Maddelerin Deriden Geçiş Katsayılarının ve Molekül Yapılarının Bulunması, *Doktora Tezi*, Gazi Üniversitesi Fen Bilimleri Enstitüsü, Ankara.

[169] **Akpınar, F.** (2009). Yerleştirme Rotalama Problemi İçin Bir Genetik Algoritma, *Yüksek Lisans Tezi*, İstanbul Teknik Üniversitesi Fen Bilimleri Enstitüsü, İstanbul.

[170] **Kevran, M. M.** (2009). Çoklu Sensör Konumlandırma Problemlerinin Genetik Algoritmalar İle Çözülmesi, *Yüksek Lisans Tezi*, Hava Harp Okulu Havacılık ve Uzay Teknolojileri Enstitüsü, İstanbul.

[171] **TCPDUMP**, http://www.tcpdump.org. 01.12.2014

[172] **WIRESHARK**, https://www.wireshark.org. 01.12.2014

[173] **FULLSTATS**, http://www.cl.cam.ac.uk/research/srg/netos/brasil/index.html. 01.12.2014

[174] **NetMATE**, https://dan.arndt.ca/nims/calculating-flow-statistics-using-netmate. 01.12.2014

[175] **MATLAB**, http://www.mathworks.com/products/matlab. 01.12.2014

[176] **WEKA**, http://www.cs.waikato.ac.nz/ml/weka. 01.12.2014

ÖZGEÇMİŞ

Fatih ERTAM
Fırat Üniversitesi
Enformatik Bölümü
E- mail: fatih@ertam.com

Eğitim

DOKTORA **(2012–2016) Fırat Üniversitesi** /Elazığ, Türkiye
Fırat Üniversitesi Fen Bilimleri Enstitüsü, Yazılım Mühendisliği Anabilim Dalı, Yazılım Mühendisliği

YÜKSEK LİSANS **(2003–2005) Fırat Üniversitesi** /Elazığ, Türkiye
Fırat Üniversitesi Fen Bilimleri Enstitüsü, Elektronik ve Bilgisayar Eğitimi Anabilim Dalı, Bilgisayar Sistemleri

LİSANS **(1996–2000) Fırat Üniversitesi** /Elazığ, Türkiye
Fırat Üniversitesi Teknik Eğitim Fakültesi Bilgisayar Öğretmenliği