
GAZİ ÜNİVERSİTESİ
Gazi
1926

YAZILIM TANIMLI AĞLARDA AĞ TRAFİĞİNE DUYARLI BİR YAKLAŞIM İLE OTONOM SALDIRI TESPİT VE ÖNLEME MODELİ

Özgür TONKAL

DOKTORA TEZİ
BİLGİSAYAR MÜHENDİSLİĞİ ANA BİLİM DALI

GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

HAZİRAN 2022

ETİK BEYAN

Gazi Üniversitesi Fen Bilimleri Enstitüsü Tez Yazım Kurallarına uygun olarak hazırladığım bu tez çalışmasında;

- Tez içinde sunduğum verileri, bilgileri ve dokümanları akademik ve etik kurallar çerçevesinde elde ettiğimi,
- Tüm bilgi, belge, değerlendirme ve sonuçları bilimsel etik ve ahlak kurallarına uygun olarak sunduğumu,
- Tez çalışmasında yararlandığım eserlerin tümüne uygun atıfta bulunarak kaynak gösterdiğimi,
- Kullanılan verilerde herhangi bir değişiklik yapmadığımı,
- Bu tezde sunduğum çalışmanın özgün olduğunu,

bildirir, aksi bir durumda aleyhime doğabilecek tüm hak kayıplarını kabullendiğimi beyan ederim.

Özgür TONKAL

24 /06/2022

YAZILIM TANIMLI AĞLARDA AĞ TRAFİĞİNE DUYARLI BİR YAKLAŞIM İLE OTONOM SALDIRI TESPİT VE ÖNLEME MODELİ

(Doktora Tezi)

Özgür TONKAL

GAZİ ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

Haziran 2022

ÖZET

İnternet kullanımının artması ile birlikte bilişim ağları üzerindeki veri trafiği ve çeşitliliği katlanarak artış göstermektedir. Bu büyük veri trafiğini etkin bir şekilde yönetmek için Yazılım Tanımlı Ağ (YTA) ve Ağ Fonksiyonlarını Sanallaştırma (Ağ Fonksiyonlarını Sanallaştırma- AFS) gibi yeni nesil ağ teknolojileri kullanılmaktadır. YTA sahip olduğu merkezi yönetim anlayışı ile geleneksel ağ yaklaşımlarında karşılaşılan birçok zorluğun üstesinden gelse de mimari yapısı sebebiyle saldırganların hedefi olabilmektedir. Bu tez çalışmasında; YTA ortamında hacimsel saldırı trafiklerinin tespitine yönelik, YTA ve AFS' nin sağlamış olduğu avantajlar kullanılarak, trafik farkında otonom özelliklere sahip bir güvenlik modeli önerilmiştir. Önerilen model; YTA kontrolcüsü ile bütünleşik çalışan Trafik Sınıflandırma Servisi (TSS) ve Esnek Güvenlik Servisi (EGS) modüllerini içermektedir. TSS modülü ile trafik sınıflandırma ve süreklilik ölçümü işlemleri gerçekleştirilirken, EGS modülü ile saldırı tespit ve engelleme işlemleri gerçekleştirilir. EGS modülü içerisinde otonom olarak açılıp kapanma özelliğine sahip farklı Siber Güvenlik Alanları (SGA) yer almaktadır. SGA içerisinde; saldırı tespit ve engelleme işlemleri için eğitilmiş makine öğrenme modelleri kullanılmaktadır. Geliştirilen uygulama kullanılarak farklı senaryoların testleri gerçekleştirilmiştir. Senaryo kapsamında, YTA uygulama katmanında çalışan Web ve DNS sunucularını hedef alan saldırılar (HTTP, DNS ve ICMP flood saldırısı) gerçekleştirilmiştir. Önerilen modelin kullanıldığı güvenlik sistemi devreye alındıktan sonra ağ trafik gecikmeleri ve sistem kaynak kullanımları ölçülmüştür. Elde edilen test sonuçları anormallik tespitinde, önerilen modelin başarılı sonuçlar verdiğini göstermektedir.

Bilim Kodu	:	92407
Anahtar Kelimeler	:	Yazılım Tanımlı Ağlar, Ağ Fonksiyonlarını Sanallaştırma, güvenlik modeli, trafik sınıflandırma, saldırı tespit, makine öğrenmesi
Sayfa Adedi	:	111
Danışman	:	Doç. Dr. Hüseyin POLAT

AUTONOMOUS ATTACK DETECTION AND MITIGATION MODEL BY NETWORK TRAFFIC AWARE APPROACH IN SOFTWARE DEFINED NETWORKS

(Ph. D. Thesis)

Özgür TONKAL

GAZİ UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

June 2022

ABSTRACT

With the increase in the use of the internet, the data traffic and diversity on the information networks increase exponentially. New generation network technologies such as Software Defined Networking (SDN) and Network Functions Virtualization (NFV) are used to effectively manage this big data traffic. Although SDN overcomes many difficulties encountered in traditional network approaches with its centralized management approach, it can be the target of attackers due to its architectural structure. In this study, an autonomous security model with a traffic awareness was proposed for the detection of volumetric attack traffic in the SDN environment, using the advantages provided by SDN and NFV. The proposed model includes Traffic Classification Service (TCS) and Flexible Security Service (FSS) modules that work integrated with the SDN controller. While traffic classification and continuity measurement operations are performed with the TCS module, intrusion detection and prevention operations are performed with the FSS module. There are different Cyber Security Areas (CSA) within the EGS module that have the feature of opening and closing autonomously. Trained machine learning models are used for intrusion detection and prevention in CSA. Tests of different scenarios were carried out using the developed application. Attacks targeting Web and DNS servers running in the SDN application layer (HTTP, DNS and ICMP flood attack) were designed in the scenarios. After commissioning the security system using the proposed model, network traffic delays and system resource usage were measured. The test results show that the proposed model gives successful results in anomaly detection.

Science Code : 92407

Key Words : Software Defined Networks, Network Functions Virtualization, security model, traffic classification, attack detection, machine learning

Page Number : 111

Supervisor : Assoc. Prof. Dr. Hüseyin POLAT

TEŞEKKÜR

Bu tezin hazırlanma sürecinde değerli bilgi ve deneyimleriyle bana yol gösteren, tez konusunun belirlenmesinden yazılmasına kadar tüm aşamalarda desteğini hiçbir zaman esirgemeyen, bana her zaman inanan saygıdeğer danışman hocam Doç. Dr. Hüseyin POLAT' a, tüm yükseköğrenim hayatım boyunca değerli yönlendirmeleriyle akademik kimliğimim oluşmasına ve mesleki gelişimime büyük katkı sağlayan kıymetli hocam Prof. Dr. O. Ayhan ERDEM' e, çalışmalarım süresince beni cesaretlendiren ve değerli katkıları ile bana destek olan komite üyesi değerli hocam Dr. Öğr. Üyesi Mustafa YENİAD' a, saygılarımı ve sonsuz teşekkürlerimi sunarım.

Yoğun doktora çalışmalarım süresince büyük fedakârlıklarda bulunarak her zaman yanımda olan sevgili eşim Esra EKMEKÇİ TONKAL' a, biricik kızlarım Fatma Çağla TONKAL ve Zeynep Ela TONKAL' a, ayrıca sonsuz destek ve anlayışları ile bugünlere gelmemde büyük emeği olan canım aileme ve dostlarıma teşekkür ediyorum.

İÇİNDEKİLER

Sayfa

Sayfa

Sayfa

ÇİZELGELERİN LİSTESİ

Çizelge **Sayfa**

ŞEKİLLERİN LİSTESİ

Şekil **Sayfa**

Şekil **Sayfa**

Şekil **Sayfa**

SİMGELER VE KISALTMALAR

Bu çalışmada kullanılmış simgeler ve kısaltmalar, açıklamaları ile birlikte aşağıda sunulmuştur.

Simgeler	**Açıklamalar**
ms	Milisaniye
sn	Saniye

Kısaltmalar	**Açıklamalar**
AFS	Ağ Fonksiyonlarını Sanallaştırma
CLI	Command Line Interface (Komut Satırı Arayüzü)
CPU	Central Processing Unit (Merkezi İşlem Birimi)
DDoS	Distributed Denial of Service (Dağıtılmış Servis Hizmet Reddi)
DoS	Denial of Service (Servis Hizmet Reddi)
DNS	Domain Name Service (Alan Adı Sistemi)
E2E	End to End (Uçtan Uca)
EGS	Esnek Güvenlik Servisi
HTTP	Hyper Text Transfer Protocol (Üst Metin Aktarım Protokolü)
HTTPS	Hyper Text Transfer Protocol Secure (Güvenli Üst Metin Aktarım Protokolü)
ICMP	Internet Control Message Protocol (İnternet Kontrol Mesaj Protokolü)
IDS	Intrusion Detection System (Saldırı Tespit Sistemi)
IoT	Internet of Things (Nesnelerin İnterneti)
IP	Integer Programming (Tam Sayılı Programlama)

Kısaltmalar	Açıklamalar
IPS	Intrusion Prevention System (Saldırı Önleme Sistemi)
ISP	Internet Service Provider (İnternet Servis Sağlayıcısı)
JSON	Javascript Object Notation (JavaScript Nesnesi Gösterimi)
MİB	Merkezi İşlem Birimi
NAT	Network Address Translation (Ağ Adres Çeviricisi)
NETCONF	Network Configuration Protocol (Ağ Yapılandırma Protokolü)
ONF	Open Networking Foundation (Açık Ağ Kurumu)
PDR	Packet Delivery Ratio (Paket Teslimat Oranı)
QoS	Quality of Service (Servis Kalitesi)
SDN	Software Defined Networking (Yazılım Tanımlı Ağ)
SGA	Siber Güvenlik Alanı
SNMP	Simple Network Management Protocol (Basit Ağ Yönetim Protokolü)
SOAP	Simple Object Access Control (Basit Nesne Erişim Protokolü)
SOM	Self Organizing Map (Öz Düzenleyici Haritalar)
SSH	Secure Shell (Güvenli Çekirdek)
TCP	Transmission Control Protocol (İletim Kontrol Protokolü)
TSS	Trafik Sınıflandırma Servisi
TLS	Transport Layer Security (Taşıma Katmanı Güvenliği)
UDP	User Datagram Protocol (Kullanıcı Veribloğu İletişim Kuralları)
YTA	Yazılım Tanımlı Ağlar

Kısaltmalar	Açıklamalar
XML	Extensible Markup Language (Genişletilebilir İşaretleme Dili)
W3C	World Wide Web Consortium (Dünya Çapında Ağ Konsorsiyumu)

1. GİRİŞ

Internet teknolojilerinin kullanımının artması ile bilişim ağları üzerindeki veri trafiği her geçen gün katlanarak artış göstermektedir. Bu büyük veri trafiğini yönetmek için, geleneksel ağ mimarisi üzerinde farklı uygulama ve protokollerin kullanıldığı çok sayıda cihaz birbirine bağlı olarak kullanılmaktadır. Bu yüzden geleneksel bilişim ağları daha büyük, daha karmaşık ve heterojen hale gelmektedir. Geleneksel ağların etkin bir şekilde yönetilmesi, ölçeklenebilmesi ve güvenliğinin sağlanması ağ yöneticileri ve hizmet sağlayıcılar için gittikçe zorlaşmaktadır.

Geleneksel büyük ağların etkin bir şekilde yönetilmesinde, ağ hizmet kalitesinin sağlanmasında ve güvenli iletişim yollarının kurulmasında karşılaşılan zorluklara çözüm getirebilmek için yeni nesil ağ teknolojilerine ihtiyaç duyulmaktadır. Yazılım Tanımlı Ağ (YTA) ve Ağ Fonksiyonları Sanallaştırma (AFS) gibi yeni ağ teknolojileri, günümüzde kullanılan geleneksel ağların ihtiyaçlarını karşılamada etkili çözümler olarak kabul edilmektedir.

YTA, vaat ettiği merkezi yönetim ve doğrudan programlanabilir yapısı ile yeni bir ağ mimarisi olarak ortaya çıkmıştır [1]. Bu yeni ve esnek ağ teknolojisinden faydalanmak isteyen Google, IBM, Cisco, Juniper gibi büyük şirketler yatırımlarını YTA teknolojileri üzerine yapmışlardır. YTA alanındaki pazar payının 2024 yılına kadar 70,41 milyar dolara ulaşacağı öngörülmektedir [2]. YTA ağ mimarisi, veri (data), kontrol (control) ve uygulama (application) adı verilen üç ayrı ağ düzleminden oluşmaktadır. YTA'da, geleneksel ağ mimarisinden farklı olarak kontrol düzlemi, veri düzleminden ayrıldığı için merkezi bir kontrolcü üzerinden ağ doğrudan programlanabilir. Kontrol düzlemi, yüksek performanslı bir sunucuya taşınmakta ve ağın yönetimi merkezi bir kontrolcü yazılımı üzerinden gerçekleştirilmektedir. Veri düzlemi ise OpenFlow protokolü destekli yönlendirici veya anahtarlayıcı cihazlar üzerinde bırakılmakta ve sadece paketlerin iletiminden sorumludur. Uygulama düzleminde ise yük dengeleme, güvenlik duvarı, trafik izleme, saldırı tespit sistemi, derin paket inceleme gibi farklı işlevleri olan ağ servisleri yer almaktadır.

YTA'da veri iletişiminde kullanılan birçok protokol olmakla birlikte OpenFlow protokolü bunlar arasında en başarılı ve yaygın kullanılan protokoldür [3]. YTA, OpenFlow protokolü

aracılığı ile veri akışını sağlamaktadır. YTA kontrolcüsü, OpenFlow protokolünü kullanarak ağdaki akış bilgilerini toplar ve özelleştirilmiş kurallar yazarak anahtarlar cihazlarına iletir. Böylece YTA ağ yöneticilerine herhangi bir fiziksel donanıma ihtiyaç duymadan yazılımsal olarak trafik analizi imkânı vermektedir. Kontrol düzlemine yerleştirilen yazılım modülleri aracılığı ile kontrolcü trafik analizi yaparak ağ kaynaklarının yönetimi, ağ güvenliğinin sağlanması ve hizmet kalitesinin (Quality of Service-QoS) artırılmasına yönelik etkin politikalar oluşturabilmektedir [4].

AFS, ağ topolojisini oluşturan; anahtarlar, yönlendiriciler, güvenlik duvarları ve yük dengeleyiciler gibi özel görevlere sahip ağ cihazlarını sanallaştırma çözümleri ile yazılım uygulamaları haline getiren ağ mimarisi konseptidir [5]. Bu mimari çözüm, enerji tasarrufu, yük optimizasyonu, ağ ölçeklenebilirliği gibi avantajlar sağlamasının yanında önemli ölçüde ağ maliyetlerini düşürmektedir. AFS'nin esnek mimarisi, bir veya daha çok sanal makine üzerinde farklı ağ fonksiyonlarının sanallaştırılmasına imkân vererek ağ yöneticilerini ve hizmet sağlayıcılarını her ağ işlevi için özel donanım bağımlılığından kurtarmaktadır.

YTA ve AFS farklı standartlara sahip teknolojiler olsa da her ikisi de açık kaynak yazılıma ve ağ donanımına geçişi savundukları için birçok ortak noktaya sahiptir [6]. Birlikte kullanıldıklarında tamamlayıcı teknolojiler olarak etkin çözümler üretebilmektedirler. AFS, YTA kontrolcüsünü sanallaştırarak kontrolcünün optimum konumlara dinamik olarak geçişine izin verirken, YTA sağlamış olduğu programlanabilir ağ yönetim yapısı ile AFS' ler arasında etkin bir trafik yönetimi sağlayabilmektedir [7].

YTA ve AFS teknolojilerinin birlikte kullanılmasıyla oluşturulan ağlarda; trafik sınıflandırma, yük dengeleme, güvenlik duvarı, saldırı tespit sistemleri ve derin paket inceleme gibi ağ fonksiyonları bir veya daha çok sunucu üzerinde sanallaştırılarak yazılım uygulamaları haline getirilmektedir. Böylece daha az yatırım ve işletim giderleri ile ağ yapıları oluşturularak son kullanıcılara verilen hizmetin kalitesi arttırılmaktadır. Yeni ağ fonksiyonlarının sisteme eklenmesi yazılım güncelleştirmeleri veya iyileştirmeleri ile kolayca gerçekleştirilebilmektedir [8].

YTA, geleneksel ağlara göre daha kolay ölçeklenebilme ve esnek ağ yönetimi gibi avantajlar sağlamasına rağmen mimari yapısı sebebi ile güvenlik açısından bazı zafiyetlere sahiptir. YTA güvenliğini tehdit eden birçok saldırı vektörü vardır [1]. Bu saldırı vektörleri farklı

YTA düzlemlerini ve iletişim yollarını hedef almaktadır. Saldırılar merkezi kontrolcüyü devre dışı bırakarak tüm ağ yönetimini ele almak ya da YTA düzlemleri arasındaki iletişim yolları üzerinde darboğazlar oluşturarak ağ iletişimini kesmeye yönelik gerçekleştirilebilir. Bütün ağın yönetimini sağlayan merkezi kontrolcüyü ve iletişim yollarını hedef alan saldırganlar, çok sayıda kaynağı belli olmayan ağ paketleri göndererek sistem kaynaklarını tüketip (işlemci, giriş/çıkış, bant genişliği, soketler ve bellek vb.) ağı hizmet veremez hale getirebilirler. Bu ağ saldırılarının tespit edilmesi için tüm ağ trafiğinin izlenerek, normal trafik ile saldırı trafiğinin ayrıştırılması gerekmektedir [9]. Ancak yüksek miktarda ağ trafik verisinden dolayı geleneksel ağ savunma yaklaşımları yetersiz kalabilmektedir. Ağ iletişiminde her trafik türünün farklı akış özellikleri olabilmektedir. Örneğin web trafiği ile ses trafiği (bant genişlikleri, gecikme oranları gibi) farklı özelliklere sahiptir. Bu trafiklerin yönlendirilmesinde, trafiklere özgü yaklaşımların uygulanması hizmet kalitesinin arttırılarak etkin bir ağ ağ yönetimi politikası oluşturulmasını sağlar. Ayrıca, legal olmayan trafiklerin tespitinde farklı trafik türlerinin farklı akış özelliklerinin kullanılması gerekebilir. Bundan dolayı YTA'da saldırılara karşı etkin bir ağ savunma sistemi oluşturmak için trafik farkında bir güvenlik yaklaşımı oluşturmak önemli bir konudur.

Geleneksel ağlarda kullanılan siber savunma sistemleri belirli bir olgunluk seviyesine ulaşmıştır. Ancak YTA yeni bir mimari yaklaşım olduğu için hacimsel saldırılara karşı savunması nispeten hala zayıf bir aşamadadır. YTA uygulamaları varsayılan olarak, güvenlik duvarı ve saldırı tespit sistemleri gibi siber savunma sistemlerini içinde barındıran, bir mimari yapıya sahip değildir. YTA kontrolcüsü, ağda bulunan diğer güvenlik ürünleri ile entegre olarak saldırıların engellenmesini amaçlamaktadır. Bu çözüm, ağdaki trafik ile ilgili doğru karar verme oranını azaltmanın yanı sıra sonuca ulaşma süresini artırmaktadır. Çünkü kontrolcü, ilgili trafiği ağdaki güvenlik duvarı ve saldırı tespit sistemlerine göndererek, gelecek sonuca göre trafik hakkında bir karar verecektir. YTA kontrolcüsü ve siber savunma sistemleri arasında sürekli trafik akışının olması, gereksiz yere bu bileşenler arasında yüksek boyutlarda trafiğin sürekli olarak dolaşmasına neden olacaktır. Bu durum, ağdaki paket teslim oranını (Packet Delivery Ratio-PDR) düşürmekte ve uçtan uca gecikme (E2E-Delay) süresini artırmaktadır. Sonuç olarak YTA mimarisinin kendi içinde tümleşik bir siber savunma sistemini barındırmaması, güvenlik açısından olumsuz bir etki oluşturmaktadır.

Bu çalışmada, YTA ortamında hacimsel saldırı trafiklerinin tespitine yönelik, YTA ve AFS' nin sağlamış olduğu avantajlar kullanılarak, trafik farkında otonom özelliklere sahip bir

güvenlik modeli önerilmiştir. Bu model; YTA kontrolcüsü üzerine yerleştirilen ve kontrolcü ile bütünleşik çalışan Trafik Sınıflandırma Servisi (TSS) ve Esnek Güvenlik Servisi (EGS) modüllerini içermektedir. TSS modülü ile trafik sınıflandırma ve trafik sürekliliğini ölçme işlemleri gerçekleştirilirken, EGS modülü ile saldırı tespit ve engelleme işlemleri gerçekleştirilir. EGS modülü içerisinde otonom olarak açılıp kapanma özelliğine sahip farklı Siber Güvenlik Alanları (SGA) yer almaktadır. SGA içerisinde, saldırı tespit ve engelleme işlemleri için makine öğrenme modelleri kullanılmaktadır.

Bu çalışmada önerilen yaklaşımın testleri, ağ sanallaştırma teknolojileri kullanılarak oluşturulan bir deneysel uygulama topolojisi üzerinde gerçekleştirilmiştir. Ayrıca, test aşamalarının gerçekleştirilmesi için bir uygulama arayüzü de geliştirilmiştir. "EGS Teknoloji" olarak isimlendirdiğimiz bu uygulama arayüzü ile ağ trafiği izleme, trafik süreklilik ölçümü ve trafik analiz işlemleri gerçekleştirilmektedir.

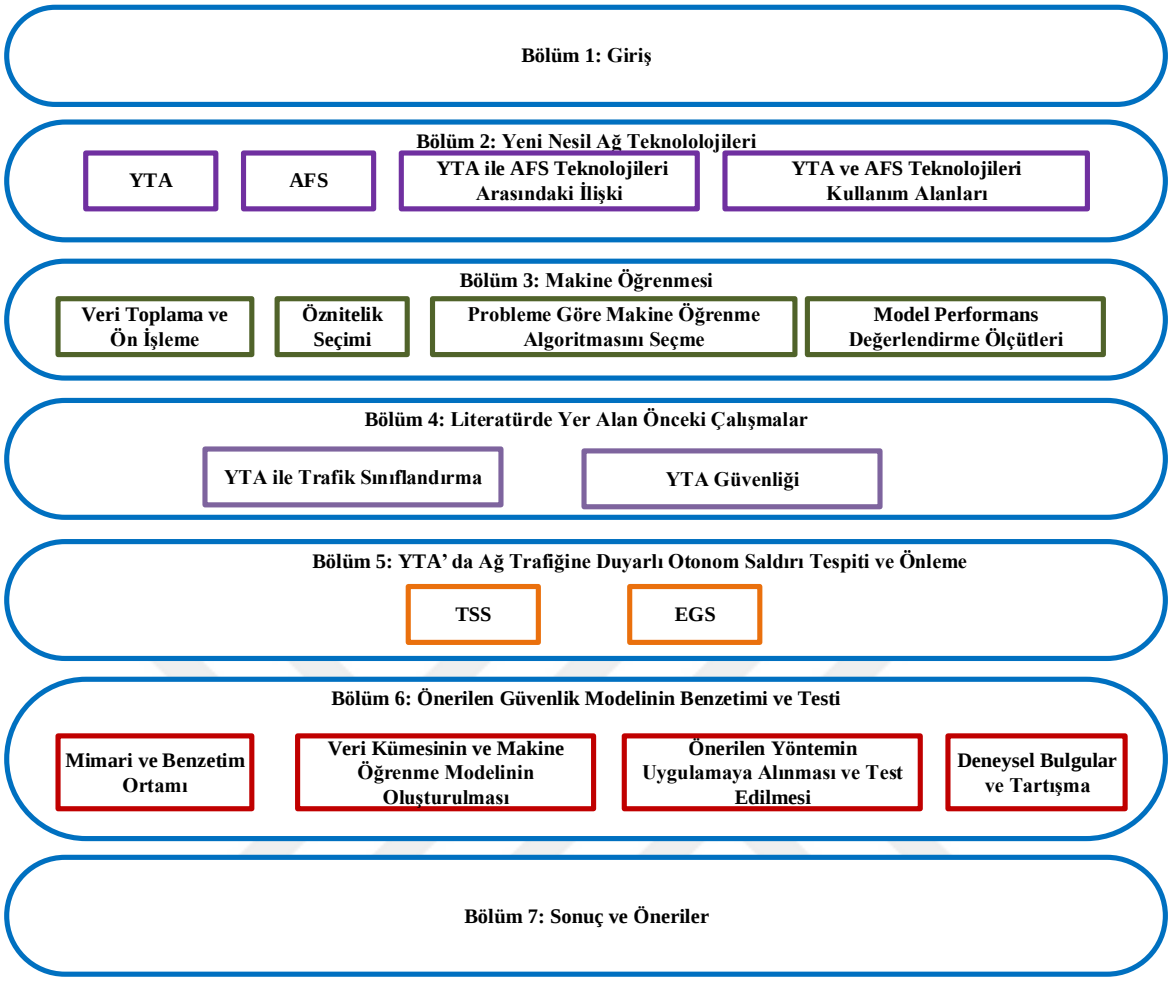
Geliştirilen uygulama kullanılarak farklı senaryoların testleri gerçekleştirilmiştir. Bu senaryolar kapsamında, YTA uygulama düzleminde çalışan Web ve DNS sunucularını hedef alan hacimsel saldırılar (HTTP, DNS ve ICMP flood saldırıları) gerçekleştirilmiştir. Önerilen yaklaşımın kullanıldığı koruma sistemi devreye alındıktan sonra ağ trafik gecikmeleri ve sistem kaynak kullanımları ölçülmüştür. Elde edilen deneysel test sonuçları, anormallik tespitinde önerilen yaklaşımın başarılı sonuçlar verdiğini göstermektedir.

Tezin organizasyonu

Tezin organizasyonu Şekil 1.1' de verilmiş olup bölüm içerikleri aşağıda özetlenmiştir:

Tezin 2. bölümünde Yazılım Tanımlı Ağ ve Ağ İşlevi Sanallaştırma kavramları detaylı bir şekilde ele alınarak, iki teknoloji arasındaki benzer noktalar ve farklılıklar incelenmiştir. Ayrıca bu iki teknolojinin kullanılabileceği uygulama alanları açıklanmıştır.

Tezin 3. bölümünde makine öğrenmesi kavramı ele alınarak makine öğrenmesi modeli geliştirme aşamalarına genel bir bakış sunulmuştur.



Şekil 1.1. Tezin organizasyonu

Tezin 4. bölümünde, YTA trafik sınıflandırma yöntemleri ile birlikte bu alanda yaygın kullanılan saldırı tespit ve önleme yaklaşımları detaylı olarak incelenmiştir. Tezin 5. bölümünde önerilen yaklaşımın tüm bileşenleri açıklanmıştır. Tezin 6. bölümünde önerilen yaklaşımın test edilmesi için oluşturulan mimari ve benzetim ortamı hakkında detaylar ve test sonuçları verilmiştir. Son bölümde elde edilen deneysel sonuçlar yorumlanarak bu çalışmanın literatüre katkısı tartışılmış ve çalışma süresince karşılaşılan açık araştırma konularına yönelik önerilerde bulunulmuştur.

2. YENİ NESİL AĞ TEKNOLOJİLERİ

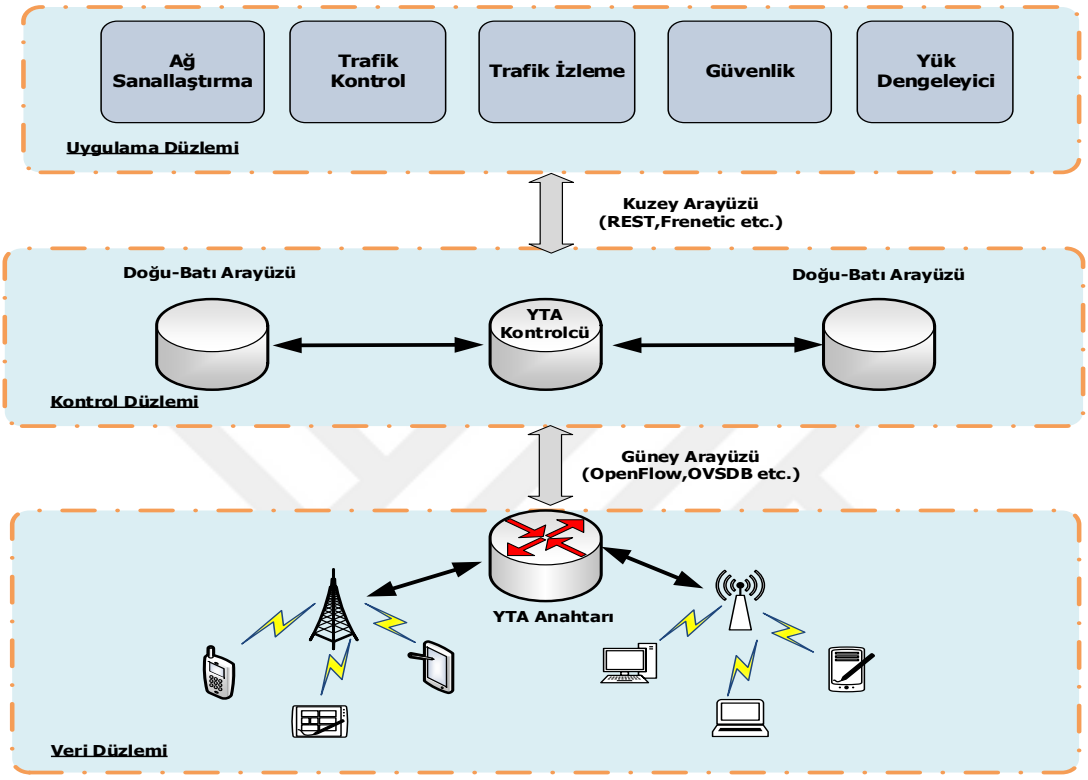
Büyük bir artış gösteren veri trafiğinin yönetilmesinde uygun maliyetli ağ teknolojilerinin kullanılması ağ ve hizmet sağlayıcılar için önemli bir konu olarak görülmektedir. Bugünün alt yapısında daha uygun maliyetli ve daha güvenilir çözüm sunan yeni nesil ağ yaklaşımı iki temel özelliğe sahiptir; donanım bağımsız çalışma ve açık kaynak geliştirme olanağı. Yazılım tanımlı ağ mimari yaklaşımı daha esnek ve dinamik bir ağ yönetim anlayışı ile donanım bağımlı geleneksel mimari yaklaşımın yerini almaktadır. YTA ve AFS, bu yeni mimari yaklaşımı destekleyen iki yazılım teknolojisi olarak ön plana çıkmaktadır.

YTA merkezi yönetim anlayışı ile büyük ve karmaşık ağların etkin bir şekilde yönetilmesini sağlarken AFS, sanallaştırma çözümleri ile yeni servislerin geliştirilme ve test edilme maliyetlerininin düşürülmesini sağlar. YTA ve AFS birbirinden farklı teknolojiler olarak ortaya çıkmış olsa da birlikte kullanıldıklarında, ağ operatörleri ve hizmet sağlayıcılara önemli avantajlar sağlamaktadır.

2.1. Yazılım Tanımlı Ağlar

Geleneksel ağ mimari yaklaşımının merkeziyetçi ve karmaşık yapısının yarattığı kısıtlamalarının ortadan kaldırılması için YTA mimarisinin kullanımı gittikçe yaygınlaşmaktadır. Bu yeni mimari yaklaşımda, geleneksel ağ mimarisinde bütünleşik olarak bulunan veri ve kontrol düzlemleri birbirinden ayrılmaktadır [10]. Geleneksel ağlarda farklı ağ cihazlarında dağıtık olarak yer alan kontrol fonksiyonu YTA mimarisinde merkezi bir kontrolcü üzerinde toplanmıştır. YTA mimarisindeki ağ anahtarları ise kontrolcüden aldığı kural tanımlarına göre veri iletim görevlerini yerine getiren akıllı olmayan cihazlara dönüşmüşlerdir. Böylece ağ yöneticileri, YTA ile daha özel veya özelleştirilmiş gereksinimlere uyacak şekilde ağ programlama esnekliğine kavuşmuşlardır. Ayrıca kontrolcü, anahtarların ağın yalnızca kısmi bir görünümüne sahip olduğu geleneksel ağlara kıyasla, ağın küresel bir görünümüne sahip olmuştur. Bu durum, kontrolcünün ağ kaynaklarını etkin bir şekilde kullanabileceği kararlar almasına olanak tanımaktadır. YTA ile ağ yönetiminin basitleştirilmesi ve otonom bir yapıya kavuşturulması hedeflenmektedir.

YTA, sağlamış olduğu merkezi ve esnek yönetim yaklaşımı ile veri merkezlerinde, kurumsal ağlarda, omurga ağlarında ve kablosuz ağlarda yaygın olarak kullanılmaktadır [3]. YTA mimarisinde, Şekil 2.1'de gösterildiği gibi uygulama, kontrol ve veri düzlemi olmak üzere üç düzlem bulunmaktadır [11].



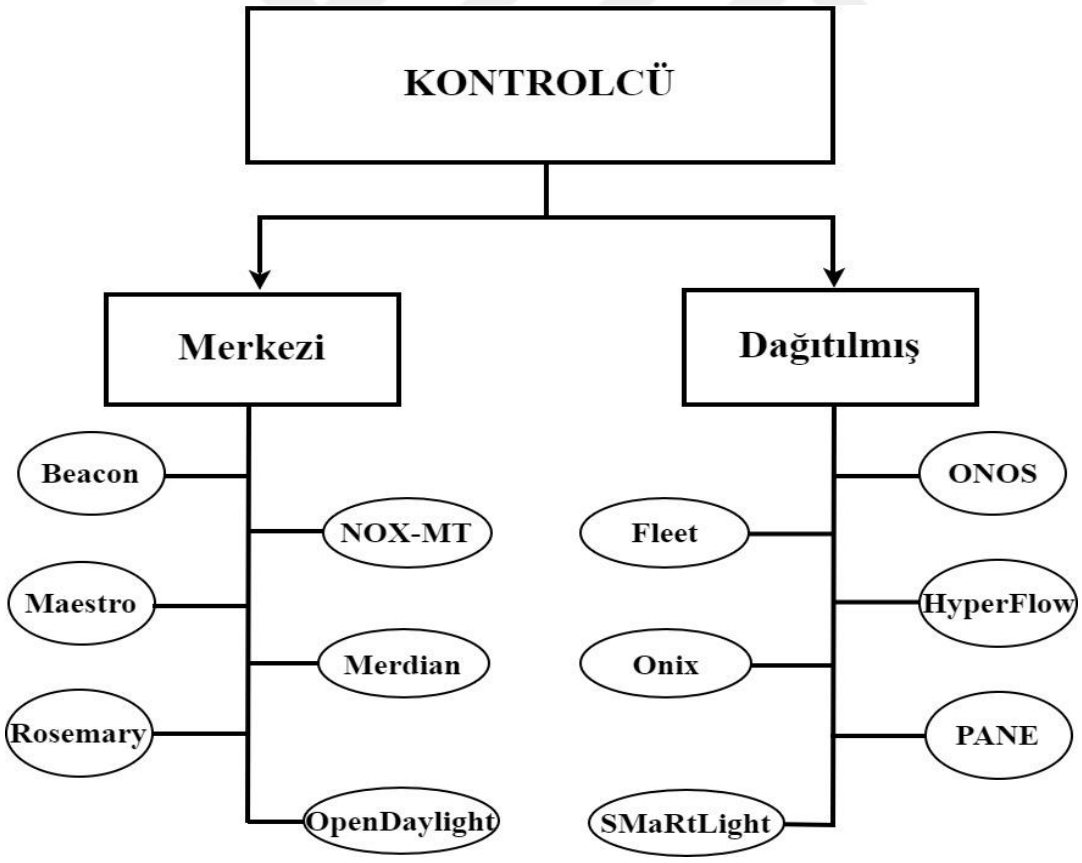
Şekil 2.1. Yazılım Tanımlı Ağ mimarisi

2.1.1. Uygulama düzlemi

Uygulama düzleminde (application layer), ağ yönetimi, kontrolü ve operasyonunu sağlamak için gerekli olan yük dengeleme, güvenlik duvarı, trafik izleme, saldırı tespit sistemi, derin paket inceleme gibi farklı işlevleri olan ağ servisleri yer almaktadır [12]. Bu servisler, kuzey arayüzü (northbound API) üzerinden, kontrolcü ile iletişim kurup ağı izleme ya da ağda değişiklik yapma taleplerini kontrol düzlemine iletebilirler. Sonrasında iletilen bu mesajlar kontrolcü tarafından OpenFlow kurallarına dönüştürülerek veri düzlemindeki anahtarların programlanmasında kullanılmaktadır [10].

2.1.2. Kontrol düzlemi

Kontrol düzleminde (control layer), YTA'nın en önemli bileşeni olan bir veya daha fazla kontrolcü yazılımı yer almaktadır. Tüm ağın küresel görünümüne sahip kontrolcü, ağın beyni gibi çalışmaktadır. Sahip olduğu çeşitli modüller aracılığı ile ağ izleme, ağ denetimi, ağ bilgilerini toplama işlemlerini yerine getirmektedir. Ayrıca Güney arayüzü (Southbound API) aracılığı ile anahtarlama cihazlarına güncel paket iletme kurallarını gönderirken Kuzey arayüzü aracılığı ile çeşitli YTA uygulamalarının isteklerine cevap vermektedir. Kontrolcü mantıksal olarak merkezileştirmiş bir yapıya sahip olmasının yanında aynı kontrolcü uygulamasının bir örneğini çalıştıran farklı sistemler arasında fiziksel olarak da dağıtılabilir. Bu özelliği ile YTA, merkezi, dağıtılmış ve hibrit model (hem merkezi hem de dağıtılmış) kontrolcüleri desteklemektedir. Şekil 2.2' de farklı günümüzde en çok tercih edilen kontrolcülerin kullanım açısından sınıflandırılması verilmiştir [13].



Şekil 2.2. Kontrolcü sınıflandırması [13]

Merkezi kontrolcülerin kullanıldığı YTA mimarisinde tek bir noktadan bütün ağın yönetimi gerçekleştirilmektedir. Tek bir merkezi kontrolcünün kullanılması, ağ yönetimini

basitleştirmenin yanında yeni politikaların hızlıca tüm ağa uygulanmasını da sağlamaktadır. Ancak bu kontrolcülerin kullanıldığı büyük ve karmaşık ağlarda, merkezi kontrolcüler sınırlı kapasiteye sahip oldukları için ölçeklenebilirlik ve güvenlik sorunları ile karşılaşılmaktadır [14]. Kontrolcü, yeni bir akış yönlendirmesi için ilgili kural tanımlamasını ağ üzerindeki tüm anahtarlarla iletişim kurarak gerçekleştirir. Bu iletişimin gerçekleşmesi için tüm ağın taranması ve bağlantı noktalarının keşfedilmesi gerekmektedir. Bu da büyük ölçekli ağlarda yanıt süresinin uzamasına ve ağın aşırı yüklenmesine sebep olmaktadır. Buna ek olarak tek bir noktadan yönetim anlayışı saldırganlar için tek bir hedef noktasını ifade ettiği için ağı saldırılara karşı kolay hedef haline getirmektedir. Merkezi kontrolcüyü ele geçiren saldırgan tüm ağın kontrolünü ele almış olacaktır.

Dağıtılmış kontrolcülerin kullanıldığı YTA modeli, merkezi kontrolcülere göre büyük ölçekli ağlarda ölçeklenebilirlik ve yüksek performans avantajlarına sahiptir [14]. Merkezi kontrolcünün yükü bu modelde dağıtılmış kontrolcüler arasında paylaştırılır. Böylece ağ performansının arttırılması hedeflenir. Ayrıca birden fazla kontrolcünün kullanımı ile tek bir hata noktası ortadan kaldırılarak ağ güvenliği arttırılır. Ancak bu modelde de karşılaşılan birtakım zorluklar vardır. Ağın yönetimi için kullanılacak optimal kontrolcü sayısının belirlenmesi ve kullanılan kontrolcülerin eş zamanlı (senkron) çalışmasının sağlanması karşılaşılan önemli zorluklardandır.

Merkezi ve dağıtılmış kontrolcü YTA modellerinde karşılaşılan zorlukların üstesinden gelebilmek için iki modelin de birlikte kullanıldığı hibrit YTA modeli önerilmektedir. Hibrit YTA modeli, merkezi kontrolcü modelinin basit yönetim özelliğinden yararlanırken dağıtılmış YTA modelinin sağlamış olduğu yüksek ölçeklenebilirlik avantajından yararlanır. Bu model genel sistem yapısının değiştirilmeden mevcut altyapının yükseltilmesine olanak tanıyarak güvenlik sorunlarının çözülmesi ve ağ yönetim optimizasyonunun sağlanması konularında avantaj sağlayabilir. Hibrit YTA modelinin uygulanmasının belirli kısıtlamaları vardır [15]. Kısıtlamalardan biri, heterojen kontrol düzleminin getirmiş olduğu yönetim karmaşıklığıdır. Paralel çalışan kontrolcüler arasındaki görev paylaşımı ve eş zamanlı iletişimin iyi yapılandırılması gerekmektedir. İkinci kısıtlama, veri düzlemi karmaşıklığıdır. Veri düzleminde görev alan protokollerin dönüştürülmesi sırasında yaşanabilecek gecikme ağ performans kayıplarına sebep olabilmektedir.

Kontrolcüler

Kontrolcü (controller), YTA'da en önemli bileşendir. Tüm ağın küresel görünümüne sahip kontrolcü, ağın beyni gibi çalışmaktadır. Bir uçta veri düzlemi cihazları ile diğer uçta yüksek seviyeli uygulamalar arasında yer almaktadır. Sahip olduğu çeşitli modüller aracılığı ile ağ izleme, ağ denetimi ve ağ bilgilerini toplama işlemlerini yerine getirmektedir. Ayrıca güney arayüzüne aracılığı ile anahtarlama cihazlarına güncel akış kurallarını gönderirken kuzey arayüzü aracılığı ile çeşitli YTA uygulamalarının isteklerine cevap vermektedir. Akış kuralları, kontrolcü tarafından reaktif modda veya proaktif modda gönderilebilmektedir [16]. Reaktif modda, veri düzleminde yer alan cihazlardan akış kurulum talebi geldiğinde, kontrolcü önce bu akışı uygulama düzlemindeki protokollere göre kontrol eder ve hangi işlemlerin yapılması gerektiğine karar verir. Sonrasında akış paketlerinin geçeceği yolu belirleyerek yol üzerindeki tüm anahtarlara yeni akış kurallarını yükler. Bu eylemler; ilet (forward), düşür (drop) veya düzenle (set) olabilir. Proaktif modda ise kontrolcü, her pakete tepki vermek yerine, anahtarlara gelebilecek her türlü akış için akış kurallarını önceden kurmaktadır.

Kontrolcüler yazılım tanımlıdır. Günümüzde Java, Python ve C++ gibi programlama dilleri ile yazılmış birbirinden farklı birçok YTA kontrolcü uygulaması vardır. Çizelge 2.1' de OpenFlow destekli kontrolcülerin kısa bir listesi verilmiştir [17].

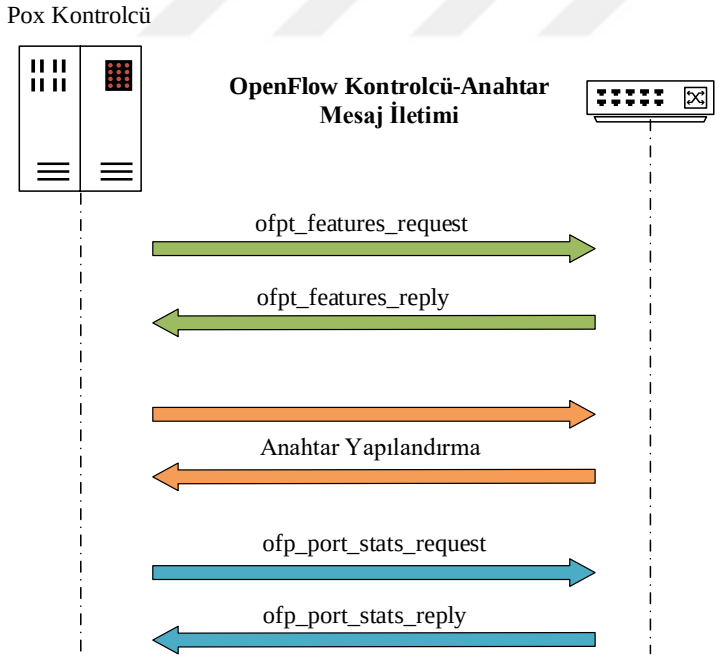
Çizelge 2.1. Kontrolcü listesi [17]

Kontrolcü	Üretici	Programlama Dili	Yapı
Beacon	Stanford	Java	Merkezi
Floodlight	BigSwitch	Java	Merkezi
Maestro	Rice University	Java	Merkezi
NOX	Nicira	Python/C++	Merkezi
POX	Nicira	Python	Merkezi
RouteFlow	CPqD	C++	Dağıtılmış
Maestro	Rice University	Java	Merkezi
Ryu	NTT OSRG ve VA Linux	Python	Merkezi
OpenDayLight	OpenDayLight	Java	Dağıtılmış

POX kontrolcü

OpenFlow ve YTA çalışmalarının erken aşamalarında birçok araştırma ve geliştirme projesinin temelinde aslında C++ ile geliştirilmiş ve Python için de bir uygulama programlama arayüzü (application programming interface-API) sağlayan NOX kontrolcü kullanılmıştır. POX kontrolcü ise, NOX kontrolcüsünün yükseltilmiş Python tabanlı açık kaynak kodlu bir kontrolcü sürümüdür. POX kontrolcüsü, Python dilinin sağlamış olduğu kolay kod yazımı avantajlarının yanı sıra sahip olduğu birçok hazır kütüphane ile büyük veriler üzerinde karmaşık işlemleri hızlı bir şekilde yapma yeteneğine de sahip olmuştur. Sorgulanabilir topoloji için yüksek bir seviyeye ve sanallaştırma desteğine sahiptir. Yeniden kullanılabilir bileşenler ile yol seçimi, topoloji keşfi, yük dengeleme, trafik sınıflandırma gibi ağ fonksiyonlarının gerçekleştirilmesini sağlamaktadır.

Geliştirilen siber güvenlik modeli, anahtarlar ve kontrolcü arasında değiş tokuş edilen OpenFlow mesajlarını işleyen ve bu işlemi daha önceki bölümde açıklandığı gibi yapan çeşitli modüllere sahip bir POX kontrolcü uygulaması olarak geliştirilmiştir.



Şekil 2.3. OpenFlow kontrolcü-anahtar iletişim adımları

Şekil 2.3' de gösterildiği gibi, kontrolcü anahtarın özelliklerini talep edebilir veya anahtarlara mesajlar göndererek bir anahtarı yapılandırabilir. Kontrolcü ile OpenFlow

anahtarı arasında kurulan güvenli bağlantı sırasında kullanılan bazı mesajların açıklamaları şu şekildedir:

ofpt_features_request: Kontrolcü ile OpenFlow anahtar arasında bağlantı kurulduktan sonra, kontrolcünün anahtarın özniteliklerini (akış tablosu, bağlantı noktası bilgileri, grup istatistikleri vb.) istemek için gönderdiği paket.

ofpt_features_reply: OpenFlow anahtarının kontrolcüye gönderdiği, kendi özniteliklerini (bağlantı noktaları, bağlantı noktası hızları, desteklenen tablolar ve eylemler vb.) taşıyan cevap paketi.

ofp_port_stats_request: Kontrolcünün OpenFlow anahtarına göndermiş olduğu bir veya daha çok bağlantı noktası için detaylı istatiksel bilgilerin istendiği paket.

ofp_port_stats_reply: OpenFlow anahtarının kontrolcüye göndermiş olduğu ayrıntılı bağlantı noktası istatiksel bilgilerini (alınan paket sayısı, iletilen paket sayısı, alınan bayt sayısı ve iletilen bayt sayısı vb.) içeren paket.

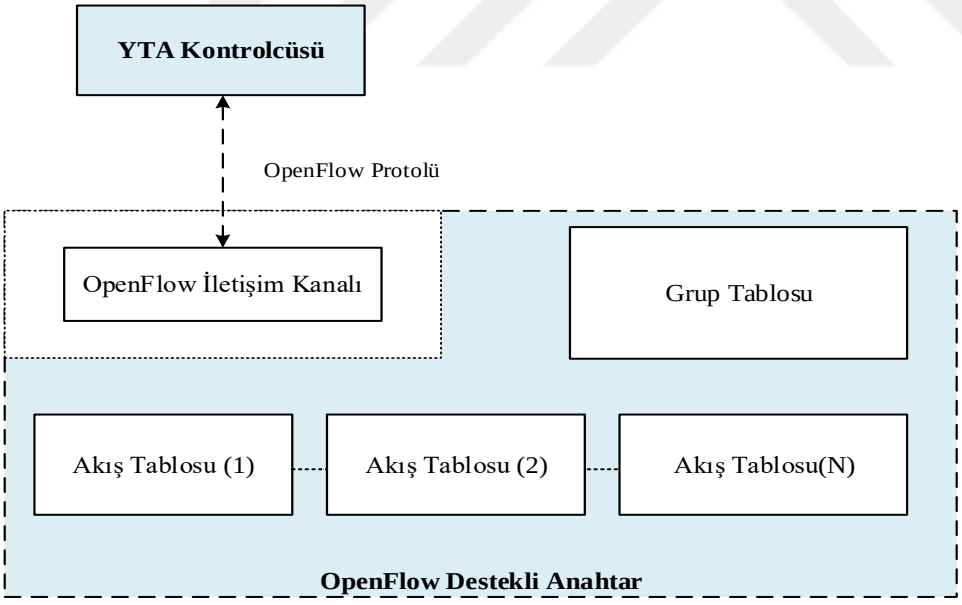
2.1.3. Veri düzlemi

Veri düzlemi (data layer), ağ trafiğinin göndericiden istemci/istemcilere iletiminin yapıldığı düzlemdir. Bu düzlemde paketlerin yönlendirilmesinden sorumlu programlanabilir YTA anahtarları görev yapar. Bu cihazlar geleneksel ağ mimarisindeki ağ cihazlarının yönlendirme özelliklerine sahip değildir. Veri iletimini kontrol düzleminde yer alan kontrolcüden aldıkları kurallara göre gerçekleştirirler. Kurallar YTA anahtarları üzerindeki akış tablolarında tutulmaktadır. Akış tabloları, akış bilgilerini, eşleşen kuralları ve ilişkili eylemleri içerir. Anahtarlara gelen her akış bilgisinin başlık alanı akış giriş kuralları ile eşleştirilir. Eşleştirme gerçekleşirse ilgili işlem gerçekleştirilerek paket iletimi sağlanır. Eşleşme bulunamazsa, paket güney arayüzü üzerinden kontrol düzlemindeki YTA kontrolcüsüne iletilir. YTA kontrolcüsü, anahtarlardan gelen paketler için ne yapılacağına karar verir ve güncel akış kurallarını (ekle, güncelle, ilet vb.) oluşturur. YTA kontrolcüsü, güney arayüzü aracılığı ile gereklilik durumunda anahtarlardan ağın istatiksel bilgilerini toplayıp güncel akış kurallarını iletebilir.

2.1.4. Güney arayüzü

Kontrol ve veri düzlemi arasında iletişim güney arayüzü üzerinden sağlanmaktadır. Güney arayüzü, kontrolcü ile veri düzleminde görev alan ve yönlendirmeden sorumlu YTA anahtarları arasındaki iletişimi sağlayan protokolleri kapsamaktadır. YTA mimarisinde kullanılan birçok protokol olmakla birlikte OpenFlow protokolü bunlar arasında en yaygın kullanılan protokoldür [3].

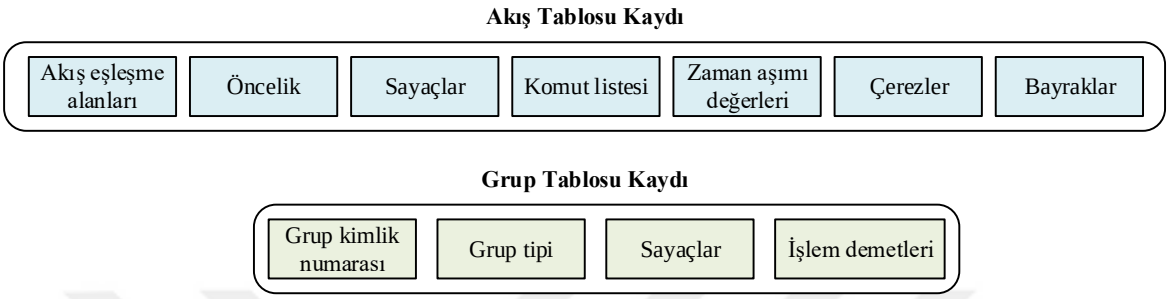
OpenFlow protokolü, YTA kontrolcü ile YTA anahtarları arasındaki iletişim kurallarının tanımlandığı Open Networking Foundation (ONF) tarafından tanımlanan bir standarttır. İlk sürümü 2009 yılında yayımlanan protokol sürekli güncellenmektedir. Anahtar üreticileri de yayımlanan yeni protokol sürümlerine göre güncellemeler yapmaktadır. OpenFlow protokolü, YTA kontrolcüsünün ağ kaynaklarını etkin bir şekilde kullanan YTA anahtarlarına akışları yüklemesine ve bunlardan trafik istatistiklerini toplamasına olanak tanımaktadır. Şekil 2.4, OpenFlow destekli anahtarın bileşenlerini göstermektedir [18].



Şekil 2.4. OpenFlow destekli anahtarların bileşenleri [18]

Bir OpenFlow anahtarı; anahtara akışı nasıl işleyeceğini söyleyen, her akış girişiyle ilişkili bir işlem içeren bir veya birden fazla akış tablosu, bir grup tablosu ve YTA kontrolcüsü ile güvenli iletişim kurulmasını sağlayan OpenFlow iletişim kanallarından oluşmaktadır. OpenFlow protokolü aracılığı ile YTA kontrolcüsü akış ve grup tablolarında, akış kuralı ekle, sil ve güncelle gibi işlemler yapabilmektedir. Şekil 2.5’ de bir akış tablosundaki akış

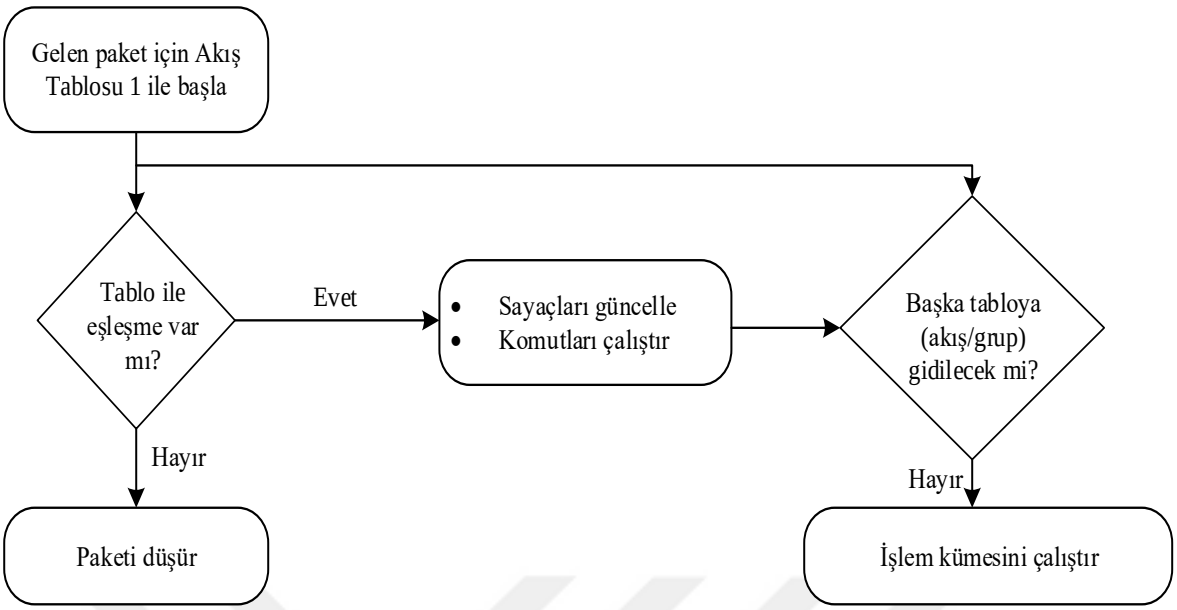
girişinin ve grup tablosunun ana bileşenleri gösterilmiştir. Her bir akış tablosunda; akış eşleştirme alanı, eşleşme öncelik değeri, her akış için bayt, paket ve sürenin kaydını tutan sayaçlar, eşleşme durumunda uygulanacak talimatlar, zaman aşım değerleri, tanımlama değerleri ve bayraklar yer almaktadır.



Şekil 2.5. Bir akış tablosundaki akış girişinin ve grup tablosunun ana bileşenleri [18]

Bir YTA anahtarına gelen bir paket için gerçekleştirilen işlemler Şekil 2.6' da gösterilmiştir. YTA anahtarına giren her paket için ilk olarak akış tablosunda eşleştirme alanı aranmaktadır. Akış tablosunda eşleşen akış var ise akışla ilgili önceden tanımlı eylemler uygulanmaktadır. Birden fazla girdi ile eşleşme olması durumunda yüksek öncelik değerine sahip akışla ilgili eylemler gerçekleştirilmektedir. Bu eylemler; paket başlıklarını güncelleme, paketi düşürme, farklı bağlantı noktalarına yönlendirme, diğer akış tablosunun işletilmesine başlama veya akışın grup tablosunda işlenmesi gibi komutları içermektedir. Paket işlenirken ilgili sayaç değerleri güncellenmektedir. İlk akış tablosu işlendikten sonra süreç devam edecekse diğer tablolar da sırası ile işleme alınmaktadır.

YTA anahtarı, süreç boyunca her bir akışın zaman aşımı değerini kontrol etmekte ve süresi dolan akışları otomatik olarak silmektedir. Grup tablosundaki her bir kayıt için grup kimlik numarası belirlenmektedir. Bu numara ilgili grup tablosu kaydının işletilmesi için kullanılmaktadır. Grup tablosunda, grup tipine göre; hangi eylem ya da eylemlerin nasıl gerçekleştirileceğinin bilgisi yer almaktadır.



Şekil 2.6. OpenFlow destekli anahtarda paket işleme adımları [18]

Kontrolcü ve anahtarlar, OpenFlow güvenli iletişim kanalı üzerinden karşılıklı mesaj gönderme yolu ile haberleşmektedir. İletişim, ilk olarak karşılıklı el sıkışma mesajları ile başlar. Kontrolcü; anahtarların özelliklerini öğrenmek, anahtarlar üzerinde değişiklikler yapmak, akış bilgilerini almak ve güncel kuralları iletmek için anahtara mesajlar yollar. Anahtarlar da mevcut durumlarını iletmek ve güncel kural tanımlamalarını öğrenmek için kontrolcüye mesaj gönder.

2.1.5. Kuzey arayüzü

Uygulama düzlemi ile kontrol düzlemi arasındaki iletişim kuzey arayüzü aracılığı ile sağlanmaktadır. Bu arayüz, uygulama düzleminde yer alan uygulamaların ağı programlamasına olanak tanımaktadır. Ağ operatörleri, servis sağlayıcılar ve araştırmacılar farklı alanlarda özgün ağ uygulamaları geliştirip YTA mimarisi üzerinde uygulayabilirler. Java, C++ ve Python gibi yüksek seviyeli programlama dilleri kullanılarak özelleştirilmiş uygulamalar oluşturmak mümkündür. Kuzey arayüzünde bir dizi protokol uygulanmaktadır. Günümüzde REST API, Frenetic [19] ve Java API gibi farklı kontrolcülere özgü geliştirilmiş protokoller kullanılmaktadır. Bu çeşitliliğin ileride birtakım zorluklara sebep olacağı düşünülerek Open Network Foundation (ONF), kuzeye bağlı arayüz standardizasyonunu sağlamak çalışma gurubu oluşturmuştur.

2.1.6. Doğu-batı arayüzü

Doğu-batı arayüzü (west-east API) dağıtılmış kontrolcü mimarisinde ihtiyaç duyulan bir arayüzdür. Birden fazla kontrolcünün kullanıldığı dağıtılmış mimarilerde iletişim bu hat üzerinden gerçekleştirilmektedir. Doğu-batı arayüzünde kullanılan standart bir protokol olmadığı için her kontrolcü kendi API'sini kullanmaktadır. Bu arayüz aracılığı ile kontrolcü ağ durumu bilgilerini toplayıp yönlendirme kurallarını değiştirebilir. Güney ve kuzey arayüzlerine benzer şekilde, doğu-batı arayüz API'leri de dağıtılmış kontrolcülerin temel bileşenleridir. Farklı kontrolcülerin arasındaki uyumluluğu ve birlikte çalışabilirliği sağlamak için standart bir doğu-batı arayüz API'sine sahip olmak gereklidir [10].

2.1.7. Yazılım tanımlı ağ benzetim ve öykünme platformları

Deneysel ağ uygulama geliştirmelerinin, test edilmesi ve doğrulama işlemleri için gerçek ortama yakın sonuçlar veren sanallaştırma ortamlarına ihtiyaç duyulmaktadır. Benzetim ve öykünme platformları, fiziksel ortam test maliyetlerini düşürerek sağlamış olduğu tekrar edilebilirlik ve ölçeklenebilirlik özellikleri ile araştırmacılara gelişmiş test yatakları sunmaktadır. Farklı işletim sistemlerinin birlikte çalıştırabileceği, açık kaynak kodlu geliştirmelerin yapılabileceği benzetim ve öykünme platformları aşağıda listelenmiştir [20–24].

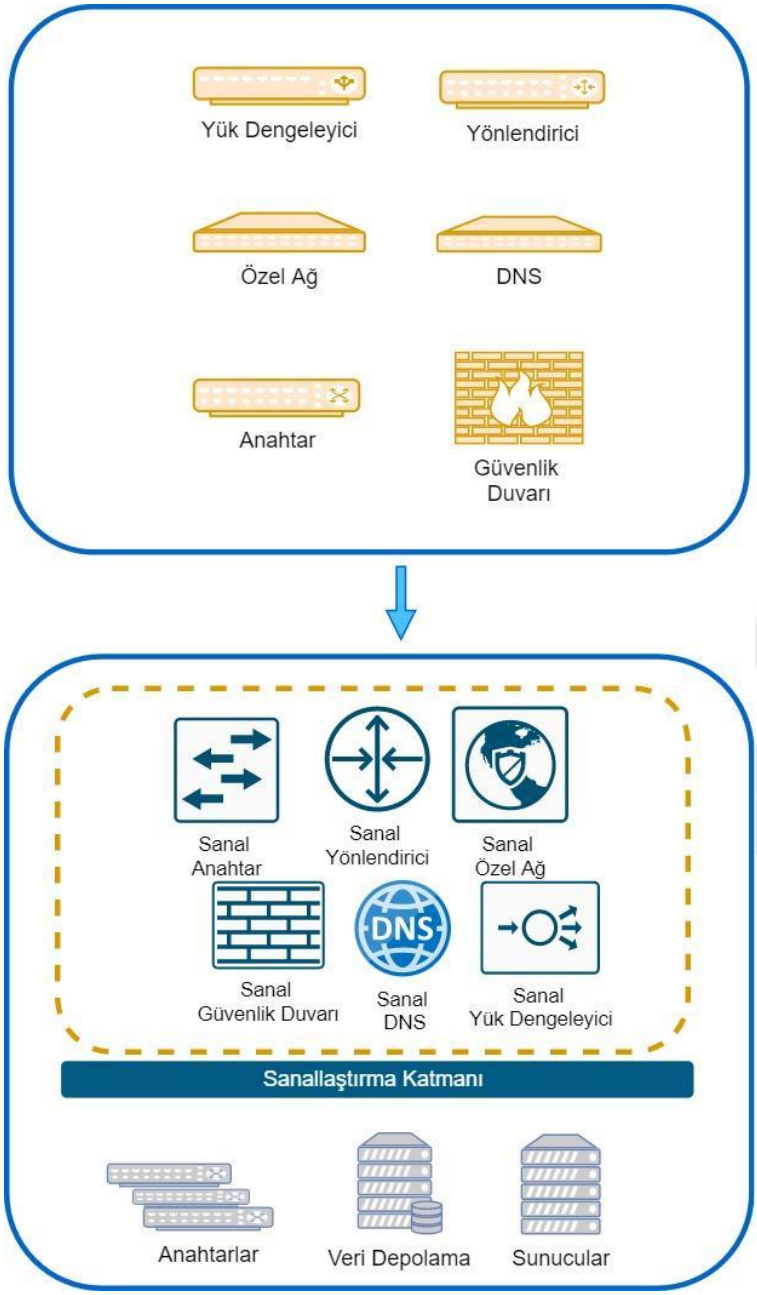
- Mininet, OpenFlow tabanlı YTA ağlarının tek bir sanal makine üzerinden YTA uygulamalarının geliştirildiği, en çok tercih edilen benzetim ve öykünme ortamıdır. Mininet, minimum kaynak kullanımı ile kontrolcü, anahtar ve uygulama kodunu çalıştırabilen gerçek ortama yakın bir ağ ortamı oluşturarak yeni uygulamaların geliştirilmesine imkân vermektedir. Mininet ile geliştirilen YTA uygulamaları herhangi bir güncelleme veya değişikliğe ihtiyaç duyulmadan gerçek donanımlar üzerine aktarılabilmektedir [20] .
- NS-3, araştırma ve eğitim amacı ile kullanılan ağ benzetim platformudur. GNU GPLv2 lisansı altında lisanslanmış ve dünya çapında bir topluluk tarafından sürdürülen ücretsiz, açık kaynaklı bir yazılım ile oluşturulmuştur [21].
- GNS3, Grafik Ağ Simülatörü 3 (Graphical Network Simulator) açık kaynak kodlu bir öykünme ortamıdır. Ağ cihazlarının gerçek ortama yakın bir şekilde yapılandırılıp kullanılabildiği, farklı işletim sistemlerinin koşturulabildiği bir platformdur. VMware

sanallaştırma yazılımları aracılığı ile oluşturulan sanal makineler GNS3 ortamında birbirleri ile haberleştirilebilmektedir [22].

- Vagrant, sanallaştırma yazılımları kullanarak (Virtualbox, VMware, Hyper-V, docker) geliştirme ortamları kurup kullanılabilen bir platformdur. Farklı sanal makineleri bir çatı altında toplayarak konsoldan yönetim imkânı sağlar. Geliştirme ortamı bulut tabanlı olarak paylaşılabilir [23].
- VIRL, Cisco firmasının ağ uygulamaları geliştirme ve test işlemleri için sanal laboratuvar ortamıdır. İşletim sistemleri, sanallaştırma ortamları ve konteyner yapıları destekleyerek geliştiriciler için düşük maliyetli geliştirme ortamı sunmaktadır [24].

2.2. Ağ Fonksiyonlarını Sanallaştırma

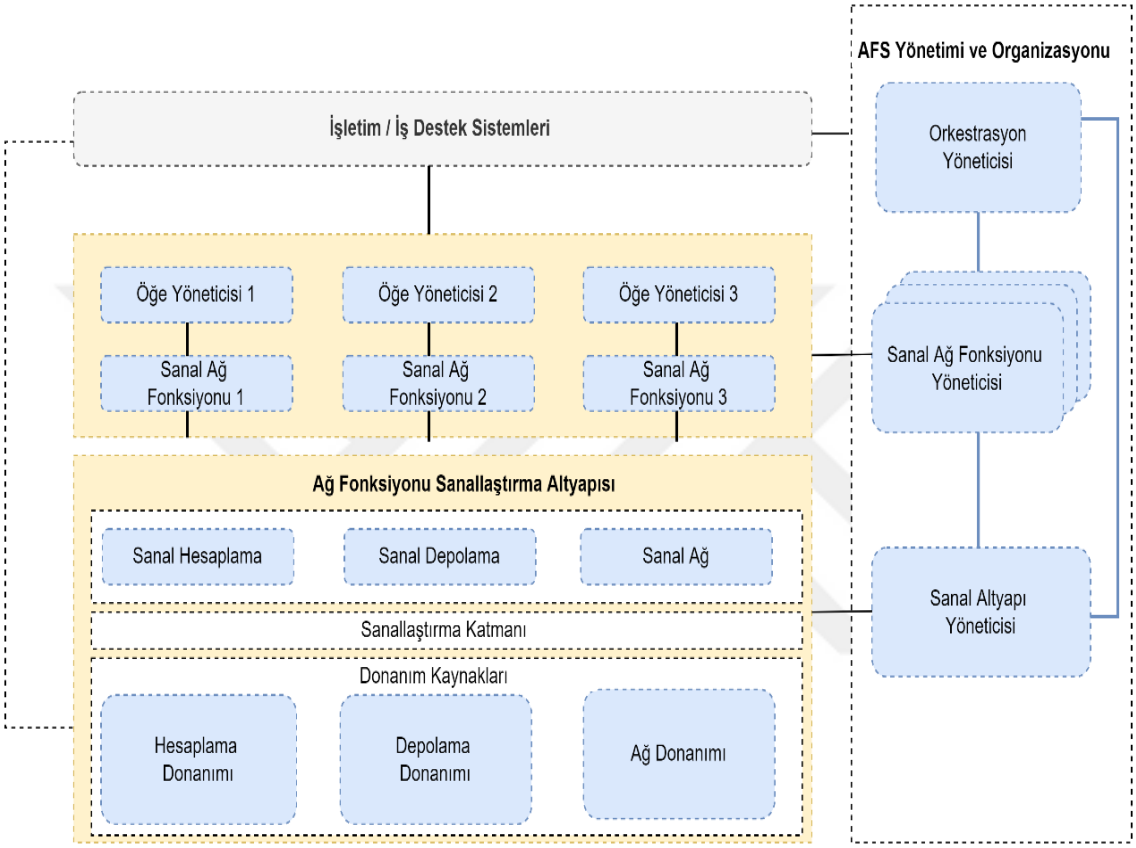
Ağ Fonksiyonları Sanallaştırma (AFS), donanım tabanlı ağ bileşenlerinin ve fonksiyonlarının sanallaştırma altyapılarının içine taşınarak sanallaştırma ortamları üzerinden yönetilmesini sağlayan yeni bir mimari yaklaşımdır. Günümüz ağ mimarisinde temel ağ işlemleri, farklı türde donanım cihazları kullanılarak gerçekleştirilmektedir. Yeni bir ağ servisinin eklenmesi ancak yeni bir donanım cihazının ağ mimarisine eklenmesi ile mümkün olabilmektedir. Bu yeni donanım için de yer ve enerji gibi kaynak gereksinimlerinin sağlanmasının yanı sıra ağdaki diğer cihazlarla entegrasyonun sağlanması önemli bir zorluktur. Gelişimin ve yenilenmenin sürekli yaşandığı ağ teknolojilerinde, donanım cihazlarına yapılan servis yatırımları kısa süreli ve maliyetli olmaktadır [25]. AFS, bu dezavantajlara bir çözüm olarak geliştirilmiş sanallaştırma teknolojisini kullanan yeni bir mimari yaklaşımdır. AFS teknolojisi ile ağ fonksiyonları soyutlaştırılarak özel donanımlara ihtiyaç duyulmadan sanal yazılımlar aracılığı ile genel amaçlı sunucular üzerinde ağ servislerinin çalıştırılması hedeflenir. Şekil 2.7’ de geleneksel ağ fonksiyonlarının, AFS mimarisinde sanallaştırılması gösterilmiştir [26].



Şekil 2.7. Ağ fonksiyonlarının sanallaştırılması

Şekil 2.7' de geleneksel ağ mimarisinde kullanılan fiziksel cihazların AFS mimarisinde sanallaştırılması gösterilmektedir. AFS ile farklı görevdeki ağ cihazları sanallaştırılarak genel amaçlı sunucular üzerine yerleştirilebilir. Böylece YTA mimarisinde olduğu gibi üretici ve donanım bağımlılığı ortadan kaldırılarak ağ maliyetleri düşecektir. Bunlara ek olarak ağ fonksiyonlarının yazılım tabanlı hale getirilmesi ile birlikte dinamik bir yönetim ve etkin ağ kaynakları kullanımı mümkün olmaktadır [5].

AFS teknolojisinin daha net anlaşılabilmesi için AFS altyapı öğelerinin ve çalışma mantığının bilinmesi önemlidir. Şekil 2.8, Avrupa Telekomünikasyon Standartları Enstitüsü (European Telecommunications Standards Institute-ETSI) tarafından standardizasyon süreci başlatılmış olan AFS teknolojisinin modüler mimarisini göstermektedir [26].



Şekil 2.8. AFS modüler mimarisi

ETSI tarafından oluşturulan AFS mimarisi yedi ana bileşenden oluşmaktadır. Bu bileşenler şu şekildedir:

1. Sanal ağ fonksiyonları (virtual network functions - VNF): Ağ fonksiyonlarının yazılım uygulamalarıdır. Güvenlik duvarı, yük dengeleyici ve yönlendirici gibi ağ fonksiyonlarının sanallaştırması ile oluşturulan modüllerden oluşmaktadır.
2. Sanal ağ fonksiyonları altyapısı (virtual network functions infrastructure - NFVI): Sanal ağ fonksiyonlarının dağıtıldığı ortamı oluşturan donanım ve yazılım bileşenlerinin tamamıdır. Hesaplama (hem sanal hem de fiziksel), depolama kaynakları ve sanallaştırma yazılımı sanal ağ fonksiyonları altyapısınının temel parçalarıdır. Altyapı, ağ donanım ve

işlem kaynaklarını yöneterek, sanal ağ fonksiyonları tarafından kullanılabilecek şekilde sanallaştırılmasını sağlamaktadır.

3. Sanallaştırılmış altyapı yöneticisi (virtualized infrastructure manager - VIM): AFS altyapısı üzerinde hesaplama, depolama ve ağ kaynaklarının kontrol edilmesi ve yönetilmesinden sorumludur. Performans ölçümlerinin toplanarak ilgili birimlere iletilmesinden sorumludur.
4. Sanal ağ yöneticisi (virtual network functions manager): Sanal ağ fonksiyonlarının yaşam döngüsünün tüm adımlarını (başlatma, güncelleme, ölçeklendirme, sonlandırma vb.) izlemekten ve yönetmekten sorumludur.
5. Öğe yöneticileri (element management system): Sanal ağ fonksiyonlarını yönetiminden sorumludur.
6. Orkestrasyon yöneticisi (orchestrator manager): AFS altyapısı üzerinde görev sanal ağ fonksiyonlarının alan uyumlu bir şekilde çalışmasından ve yönetiminden sorumludur. Orkestratör kavramı, bulut bilişimde kullanılan hiper yöneticiye benzemektedir.
7. İşletim destek sistemi (operation support system- OSS) ve iş destek sistemi (business support system-BSS): İşletim destek sistemi, ağ yapılandırma, hata ve servis yönetimlerinden sorumlu iken; iş destek sistemi, müşteri yönetimi, hizmet servisleri yönetimi ve ürün yönetiminden sorumludur.

AFS mimarileri, ağ maliyetlerinin düşürülmesi, ağ hizmet kalitesinin arttırılması ve ağ genişletme hızında artış sağlama potansiyeline sahip olmasının yanında, geleneksel yöntemlerle elde edilmesi zor bir seçenek olan hızlı hizmet sunumu için ağ esnekliğini arttırma potansiyeline sahiptir [27]. Organizasyonlar bu mimari ile donanım yenilemeden yeni ağ işlevlerini veya uygulamalarını hızlı ve basit bir şekilde devreye alabilir.

YTA ve AFS ağın programlanabilmesini ve donanım bağımlılığının ortadan kaldırılmasına yönelik oluşturulan yeni yaklaşımlardır. YTA, kontrol ve veri düzlemini birbirinden ayırarak ağların programlanabilir olmasını sağlarken; AFS, ağ fonksiyonlarının sanallaştırılarak donanım maliyetlerini düşürmeyi amaçlar. YTA ve AFS bağımsız çalışabilen teknolojiler olmakla birlikte tamamlayıcı bir ağ çözümü olarak da kullanılır.

2.3. Yazılım Tanımlı Ağ ve Ağ Fonksiyonlarını Sanallaştırma Teknolojileri Arasındaki İlişki

YTA ve AFS teknolojileri farklı ağ çözümleri olarak bağımsız çalışabilseler de yazılım tabanlı bir çözümün, farklı unsurlarını ele almaları açısından birbirlerini tamamlayıcı ve destekleyici özelliktedir. Her iki teknoloji de programlanabilir ağ yapısı ve sanallaştırma yöntemlerini kullanarak etkin bir ağ yönetimi oluşturma gibi ortak hedeflere sahiptir. Çizelge 2.2' de YTA ve AFS teknolojilerinin sahip olduğu özellikler karşılaştırmalı olarak verilmiştir.

Çizelge 2.2. YTA ve AFS teknolojilerinin karşılaştırılması

	AFS	YTA
Kavram	Ağ işlevlerini sanallaştırılarak geleneksel cihazlardan ayırır.	Otomatik ve programlanabilir ağ kontrolünü sağlamak için yönlendirme düzlemini kontrol düzleminden ayırır.
Amaç	Hizmet sağlayıcılar tarafından farklı ağ görevleri olan donanım cihazlarını tümleşik donanım cihazlarıyla değiştirmek için önerilmiştir.	Merkezi yönetim ve kontrolü sağlamak için ağ donanım cihazlarının programlanabilir olmasını sağlar.
Anahtar Noktalar	Standart prosedüre sahiptir. Donanım yönlendirme fonksiyonu, belirli bir donanıma bağlı değildir.	Açık ve programlanabilir kontrol düzlemine sahiptir. Donanım hala yönlendirmeden sorumludur ve kontrol düzlemi karar vermekten sorumludur.
Uygulama	QoS ve akış kontrolü gibi ağ işlevlerini optimize eder.	Anahtarlar, yönlendiriciler ve kablosuz ağlar gibi ağ altyapısı mimarisini optimize eder.
Standart	ETSI	ONF
Avantajlar	Standardizasyon, basitleştirilmiş altyapı bakımı ve esnek genişleme.	Basitleştirilmiş yönetim işlemleri.
İlişki	YTA, AFS ile çakışmaz. Her ikisi de gelişmekte olan ağ mimarileridir. YTA, ağ programlanabilirliğini sağlarken; AFS, ağ işlevlerini donanım aygıtlarından ayırmayı amaçlar. YTA uygulamaları, AFS geliştirme için referans olarak kullanılabilir.	

YTA, fiziksel ağ kaynaklarını (anahtarlar ve yönlendiriciler gibi) soyutlayarak karar vermeyi sanal ağ kontrol düzlemine aktarmaktadır. Kontrol düzlemi, trafiğin nereye gönderileceğini belirler ve donanım, standart donanım aygıtlarına güvenmeden trafiği yönlendirmeye ve işlemeye devam etmektedir. AFS, tüm fiziksel ağ kaynaklarını sanallaştırmayı ve standart donanım cihazlarına bağlı olarak daha fazla cihaz eklemeden ağların genişlemesine izin vermeyi amaçlamaktadır [28]. YTA ve AFS birlikte kullanıldığında büyük ve karmaşık

ağların yönetimi konusunda esnek bir ortam sunmaktadır. Bundan dolayı bulut bilişim ve 5G gibi yeni nesil ağ çözümleri YTA ve AFS mimarilerinin birlikte kullanıldığı çözümler üzerine yapılandırılmaktadır.

2.4. Yazılım Tanımlı Ağ ve Ağ Fonksiyonlarını Sanallaştırma Teknolojileri Kullanım Alanları

YTA ve AFS teknolojileri, vaat ettikleri programlanabilir ağ yapısı, esnek mimari, etkin ağ yönetimi ve donanım bağımsız uygun maliyetli sanallaştırma çözümleri ile günümüzde bulut bilişim, mobil ağ, veri merkezleri ve nesnelerini interneti gibi alanlarda kullanılmaya başlanmıştır.

2.4.1. Bulut bilişim

Bulut bilişim, dağıtık olarak bulunan bilgi işlem altyapı ve hizmetlerinin (sunucu, depolama, uygulama, güvenlik vb.), yerel ağlar veya internet gibi genel ağlar üzerinden kullanıcılara sunulmasıdır. Kullanıcılar bu teknoloji ile herhangi bir zaman diliminde konum bağımsız olarak kritik hizmetlere erişim imkanına sahip olmaktadır. Servis sağlayıcılar, bulut bilişim teknolojisi ile dinamik ve hızlı bir şekilde bilgi işlem altyapı hizmetlerini yönetme ve optimize etme yeteneğine sahip olmaktadırlar. AFS ve YTA, bu dinamik hizmet sağlamanın etkinleştiricisi olarak rol almaktadır. AFS ile sanallaştırılan ağ fonksiyonları, üreticiyi donanım geliştirme veya yenileme maliyetinden kurtarmaktadır. Yeni ağ işlevleri, sanallaştırma yazılımları ile kolayca eklenip güncelleştirebilir. Buna ek olarak, YTA ile entegre olunduğunda kontrolcünün sağlamış olduğu merkezi yönetim anlayışı ile ağ trafik politikaları hızlı bir şekilde tüm ağa uygulanabilir. YTA, AFS ve bulut bilişim teknolojilerinin birlikte kullanılması ile ağ operatörleri verimli ve ölçeklenebilir hizmetler sunar [29, 30].

2.4.2. Mobil ağ

Yeni nesil mobil ağların kullanımının yaygınlaşması ile daha gelişmiş ve ölçeklenebilir teknolojilere olan ihtiyaç artmıştır [31]. Buna ek olarak, nesnelerin interneti (Internet of Things-IoT), bulut bilişim ve 5G gibi yeni nesil teknolojilerden yararlanan mobil kullanıcı

sayısının artışı; veri hızı, kapasite ve kapsama alanın arttırılmasını bir zorunluluk haline getirmiştir.

5G mobil ağları; geniş kapsama alanı, düşük gecikme süresi ve yüksek hızda veri aktarımı gibi üç temel karakteristik özelliğe sahiptir [32]. Bu özelliklerin hizmet sağlayıcılar tarafından karşılanabilmesi için yeni fiziksel altyapıların satın alınması ve işletilmesi gerekmektedir. Yönetim için de yüksek nitelikli mühendislerin çalıştırılması ve ayrıca geniş kapsama alanı elde edebilmek için baz istasyonları gibi ağ sonlandırıcılarının optimum verimle yerleştirilmesi gerekmektedir. Bu gereklilikler hizmet sağlayıcılarını yüksek sermaye harcaması (capital expenditure-CAPEX) ve yüksek operasyonel harcamaya (operational expenditure-OPEX) doğru itmektedir. Bu nedenle mobil ağ operatörleri ve hizmet sağlayıcılar toplam maliyetleri azaltıcı yeni çözüm yöntemlerine ihtiyaç duymaktadır [33].

YTA ve AFS, kontrol ve veri düzlemlerini ayırmak ve ağ işlevlerini sanallaştırmak için çekici çözümler olarak kabul edilmektedir [34]. YTA sahip olduğu merkezi yönetim anlayışı ile ağ kaynaklarının dinamik ve esnek bir şekilde yönetilmesini sağlarken 5G ağlarının sahip olması gereken kesintisiz veri aktarımına olanak sağlamaktadır. AFS, 5G ağlarının sahip olduğu ağ dilimleme (network slicing) özelliğini sağlamış olduğu sanallaştırma teknolojileri ile destekleyerek sistem kaynaklarının verimli kullanılmasını ve kesintisiz hizmet vermeyi garanti etmektedir [35].

2.4.3. Veri merkezleri

Veri merkezleri; şirketler, devlet kurumları, endüstriler ve eğitim kurumları gibi farklı büyüklükteki kuruluşların verilerinin işlenmesi, depolanması ve yedeklenmesi için ölçeklenebilir kaynaklar sunan bir altyapıdır. Bu altyapı üzerinde çok sayıda sunucu, depolama cihazları ve ağ cihazları yer almaktadır. Merkezi ve dağıtık yapıya sahip olabilen veri merkezleri son dönemde artan uygulama çeşitliliği ile bulut tabanlı olarak da hizmet verebilmektedir. Birçok büyük kuruluş yatırımlarını veri merkezleri ve bulut çözümleri üzerine yapmaktadır. Ancak veri merkezleri için kaynaklarının etkin kullanımı, güvenilirlik, yanıt süresi ve ölçeklenebilirlik gibi çözülmesi gereken zorluklar vardır [36].

YTA ve AFS teknolojileri sağlamış olduğu yenilikçi çözümler ile veri merkezlerinde karşılaşılan zorlukların üstesinden gelme potansiyeline sahiptir. YTA ve AFS ile yapılandırılmış veri merkezlerinde, yük dengeleme, ağ güvenliği, trafik izleme ve bant genişliği tahsisi gibi yönetimsel yetenekler sanallaştırma çözümleri ile geliştirilmektedir. Ağ uygulamalarının güncellenmesi veya yenilenmesi gerektiği durumlarda yeni bir donanım almak yerine yazılımsal değişimler ile yapılandırmalar kolayca gerçekleştirilmektedir. Bunlara ek olarak veri merkezlerindeki enerji tüketimi zorluğu da bu teknolojiler ile aşılabilmektedir [37].

2.4.4. Nesnelerin interneti

IoT, akıllı hizmetler sumak için fiziksel ve sanal nesnelerin internet üzerinden birbirleriyle etkileşime girebildiği küresel bir ağ yapısıdır. Çok sayıda cihazın birbirine bağlı olduğu bu heterojen ağ alt yapısında, yüksek verim, yüksek kullanılabilirlik, düşük gecikme ve yüksek performans gereksinimleri ortaya çıkmaktadır. Geleneksel ağ mimarileri ve protokolleri bu gereksinimlerin karşılanmasında mevcut çözüm yöntemleri ile yetersiz kalmaktadır [38].

YTA ve AFS teknolojileri sağlamış olduğu esnek mimari ve sanallaştırma çözümleri ile IoT ağ gereksinimlerinin karşılanmasında için kilit rol oynamaktadır [39]. IoT ağ yapısını oluşturan farklı özellikteki cihazların birlikte çalışabilirlik ve ölçeklenebilirlik gibi zorlukları YTA mimarisinin sağlamış olduğu programlanabilir ağ yönetimi sayesinde çözülebilmektedir. AFS ile IoT heterojen ağını oluşturan cihazlar eş zamanlı çalışma performanslarını sanallaştırma çözümleri ile arttırabilmektedir.

YTA ve AFS tabanlı IoT çözümleri akıllı şehir uygulamalarında önemli rol oynamaktadır [40]. Akıllı şehirleri oluşturan farklı özellikteki IoT cihazlarının kesintisiz haberleşmesinin sağlanması için ortak bir fiziksel altyapıya sahip olması gerekmektedir. IoT ağ iletişimini sağlayan farklı lokasyonlardaki duyarga (sensor) düğümleri, üretici bağımsız birden çok cihazı ve uygulamayı destekleme yeteneğine sahip olmalıdır. YTA ve AFS sağlamış olduğu programlanabilir ağ yönetimi ve sanallaştırma uygulamaları ile uygun maliyetli akıllı şehir çözümleri sağlamaktadır [41], [42].

YTA ve ağ sanallaştırma teknolojileri geniş kullanım alanları ile günümüz ve gelecekteki ağ problemleri için etkili çözümler vaat etmektedir. Günümüzde ağ trafiğinin etkin bir şekilde

sınıflandırılması ve ağ güvenliğinin sağlanması gibi konular, hizmet sağlayıcılar ve ağ yöneticileri için önemli zorluklar olarak görülmektedir.

3. MAKİNE ÖĞRENMESİ

Yapay Zeka için temel bir yaklaşım olarak önerilen makine öğrenmesi, 1959 yılında Arthur Samuel tarafından “bilgisayarlara açıkça programlanmadan öğrenme yeteneği veren çalışma alanı” olarak tanımlanmıştır [43]. 1997 yılında ise Tom Mitchell kitabında makine öğrenmesinin daha modern bir yorumunu yapmıştır: “Bir bilgisayar programının çıktısı, *T* ’deki görevlerle hesaplandığı gibi, deneyimle artar". Buna göre, makine öğrenmesi bir yapay zekâ alanıdır ve bilgisayar sistemlerinin belirli bir görevi yerine getirmesi için açık talimatlar kullanmadan kalıplara ve çıkarsamalara dayanan algoritmaların ve modellerin bilimsel çalışmasıdır. Günümüzde makine öğrenmesi modelleri; kümeleme, sınıflandırma, regresyon gibi farklı problemlerin çözümü için kullanılmaktadır.

Makine öğrenimi, insan yaşamını kolaylaştırmak ve daha iyi hala getirmek için çeşitli alanlarda uygulanmıştır. Doğal dil işleme (Natural Language Processing-NLP), hastalıkların teşhisi, müşteri segmentasyonu, ürün tavsiyesi, güvenlik uygulamaları ve yüz tanıma sistemleri gibi faklı alanlarda makine öğrenimi kullanılmaktadır. Standart bir makine öğrenimi uygulama süreci şunları içerir:

- Verileri toplama ve ön işleme
- Özniteliklerin belirlenmesi ve model seçimi
- Model eğitimi ve test etme
- Eğitilmiş model optimizasyonu
- Yeni örnekleri tahmin etmek için modeli kullanma

Makine öğrenme modelleri temel olarak denetimli, denetimsiz, yarı denetimli ve pekiştirmeli öğrenme olmak üzere dört grupta incelenmektedir [44].

Denetimli öğrenme (supervised learning), girişleri ve bunlara karşılık gelen etiketleri içeren bir takım eğitim verilerini temel alan bir işlev veya model oluşturma fikrine dayanmaktadır. Model, eğitim aşamasında girdiler ve bu girdilerin beklenen çıktıları verilerek eğitilir. Her giriş kümesi için uygun çıkış kümesi algoritmaya sağlanmalıdır. Daha sonra girdi ve çıktılar arasındaki öğrenilen ilişkiye dayalı olarak modele, öngörü işlemini gerçekleştirebilmesi için yeni girdiler sağlanır. Modelin kullandığı algoritma her yeni girdi için öngörüde bulunarak

geri bildirim verir. Zamanla algoritma girdiler ve etiketler arasında ilişki kurarak öngörü başarımını arttırır. Tam olarak eğitim gerçekleştiğinde denetimli öğrenme algoritması daha önce görülmemiş örnekler için etiket öngörüsünde bulunabilir. Denetimli öğrenme algoritmaları sınıflandırma ve regresyon problemlerinde yaygın şekilde kullanılmaktadır. Sınıflandırma problemlerinde çıktı değişkeni, "normal/normal olmayan", "iyi/kötü", "evet/hayır" gibi kategorik değerler alırken regresyon problemlerinde çıktı değişkeni sayısal gerçek değerler alır. Yaygın olarak kullanılan denetimli makine öğrenme algoritmaları şunlardır: k-En Yakın Komşular (k-Nearest Neighbors-kNN), Destek Vektör Makineleri (Support Vector Machine-SVM), Naive Bayes (NB), Karar Ağacı (Decision Tree-DT), Lojistik Regresyon (LR), Yapay Sinir Ağları (Artificial Neural Network-ANN), Rastgele Orman (Random Forest-RF), vb.

Denetimsiz öğrenme (unsupervised learning), model eğitimi etiketlenmemiş veriler kullanılarak gerçekleştirilir. Model veri kümesindeki etiketlenmemiş verilerin benzer ve farklı olan özelliklerini kullanarak kümeler. Denetimsiz öğrenmede kullanılan algoritmalar benzerliğin hesaplanması için farklı matematiksel fonksiyonlardan yararlanmaktadır. Denetimsiz öğrenme modelleri, ilişkilendirme veya kümeleme problemlerinde kullanılmaktadır [45]. Yaygın olarak kullanılan denetimsiz öğrenme algoritmaları: K-Ortalama Kümeleme (K-Means Clustering), Gauss Karışımı Kümeleme (Gaussian Mixture Model-GMM), Kendi Kendini Düzenleyen Harita (Self Organized Map-SOM), Otomatik Kodlayıcı (Auto Encoder), Beklenti-Maksimizasyon algoritması (Expectation-Maximization), Yerel Aykırı Değer Faktörü (Local Outlier Factor-LOF) vb.'dir.

Yarı-denetimli öğrenme (semi-supervised learning), denetimli ve denetimsiz öğrenmenin birlikte kullanılması olarak tanımlanabilir. Bu öğrenme yönteminde, veri kümesi etiketli verilerin küçük bir yüzdesini içerir ve geri kalanı etiketsizdir. Böylece yarı denetimli algoritmalar hem etiketli hem de etiketsiz veriler aracılığıyla verilerin yapısını öğrenir. Yaygın kullanılan yarı denetimli öğrenme algoritmalarından bazıları şunlardır: Etiket Yayılımı (Label Propagation Algorithm- LPA), Gauss Karışım Modeli (Gaussian Mixture Models- GMM), Gizli Markov Modeli (Hidden Marcov Model-HMM), Yarı Denetimli Destek Vektör Makineleri (Semi-Supervised Support Vector Machines-S3VM), Grafik tabanlı algoritmalar, vb.

Takviyeli (Pekiştirici) öğrenme (reinforcement learning), karar verme problemlerini modellemek için kullanılan ajan tabanlı yinelemeli bir süreçtir. Denetimli ve denetimsiz öğrenmede model eğitimi, veri kümesinden alınan örneklerle gerçekleştirilirken; takviyeli öğrenme modelinde öğrenme, dış dünya ile etkileşime giren ajanlar aracılığı ile gerçekleşir. Takviyeli öğrenmede kullanılan ajan, yinelemeli bir şekilde tüm olası durumları keşfedene kadar çevreyle ilgili deneyimlerinden sürekli olarak öğrenir. Bu modelde eğitim verileri; durum-eylem çifti ile ödül ve cezadan oluşur. Algoritma, bir karar verdiğinde kararın doğru olduğu durumlarda ödül, yanlış olduğu durumlarda ise ceza verilir. Bu bilgiler ışığında model kendini en iyi sonuca ulaşmak için yinelemeli olarak eğitir. Yaygın kullanılan pekiştirmeli öğrenme algoritmalarından bazıları, Q-öğrenme (Q-Learning), Geçici Fark (Temporal Difference -TD), Monte Carlo yöntemi, Derin Q-öğrenme (Deep Q-Learning-DQL) vb.'dir.

3.1. Veri Toplama ve Ön İşleme

Veri toplama ve ön işleme aşaması, makine öğrenimi modelleri geliştirmenin en önemli aşamasıdır. Makine öğrenmesi problemi tanımlanarak probleme uygun veriler toplanır. Araştırma problemine uygun olan veri kümeleri belirlenir ya da oluşturulur. Veri kümelerinin seçilmesinde çevrim içi birçok veri kümesi olmakla birlikte genel kabul gören veri kümelerinin seçimi oluşturulan modeli öngörü başarımının belirlenmesi için önemli bir etken olarak görülmektedir. Makine öğrenmesi modelinin, ele alınan problemi doğru modelleyen geçerli verilerle beslendiğinden emin olmak için verilerin temizlenmesi ve ön işleme tabi tutulmuş olması gerekir.

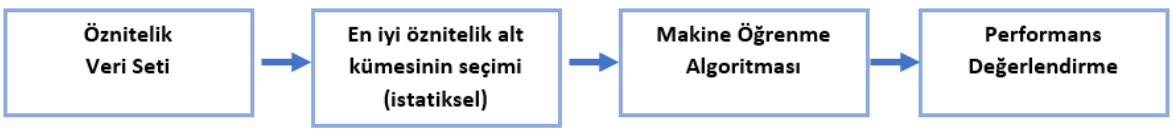
3.2. Öznitelik Seçimi

Makine öğrenmesi sürecinin bir diğer önemli aşamasıdır. Bir öznitelik, bir olay veya olgu hakkında gözlemlenebilen, ölçülebilen, tahminde bulunmaya veya sınıflandırmaya yardımcı olabilecek bilgilerdir. Öznitelik seçimi, makine öğrenmesi algoritmalarının çalışması için gerekli özelliklerin oluşturulmasında verilerin alan bilgilerinin kullanılma sürecidir. Model doğruluğunun arttırılmasında etkili özniteliklerinin seçimi oldukça önemlidir. Öznitelik seçimi kullanılarak var olan veri öznitelikleri birleştirilerek yeni öznitelikler türetilebilmektedir. Öznitelik seçimi, modelin öğrenme doğruluğunu arttırmak için gereksiz öznitelikleri ortadan kaldırmayı amaçlamaktadır [46]. Azaltılmış öznitelik sayısı model

hesaplama yükünü azaltarak modelin daha hızlı ve doğru tahminde bulunma yeteneğini arttıracaktır.

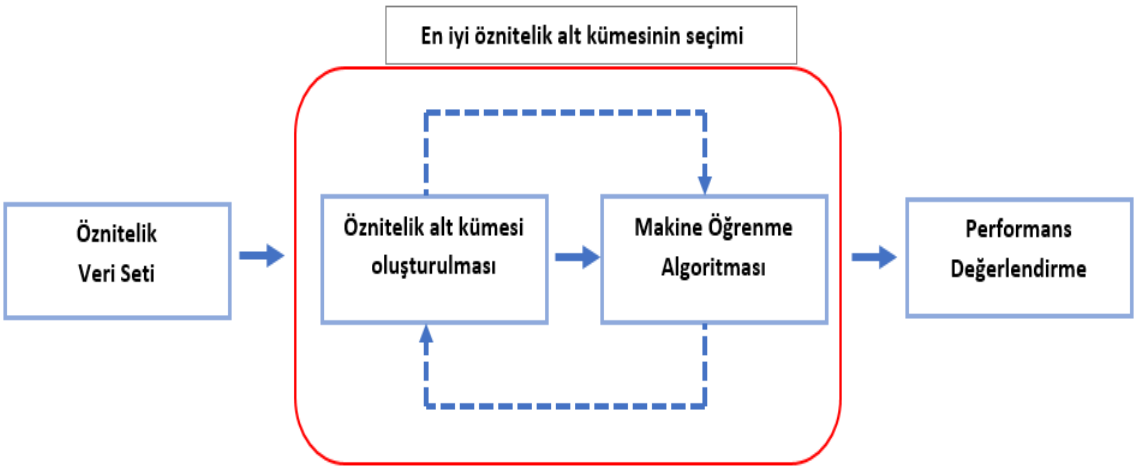
Öznitelik seçiminde kullanılan yöntemler; istatiksel verilere dayanarak filtreleme gerçekleştiren filtreleme yöntemleri (filter methods), öznitelikler üzerinde arama algoritmaları kullanan sarmal yöntemler (wrapper methods) ve en iyi öznitelik alt kümesini bulmaya dayalı gömülü yöntemler (embeded methods) olmak üzere 3 grupta incelenmektedir [47].

Filtreleme yöntemleri, makine öğrenmesi modelini dikkate almadan; uzaklık, bilgi, tutarlılık ve bağımlılık gibi verilerin içsel özelliklerine bakarak uygunluk değerlendirmesi yapan öznitelik sıralama teknikleridir. Şekil 3.1' de filtreleme yöntemi öznitelik seçim aşamaları gösterilmiştir. Buna göre, veri kümesinde bulunan her bir öznitelik değerlendirme fonksiyonları kullanılarak puanlandırılmaktadır. Değerlendirme fonksiyonundan elde edilen puana göre öznitelik ilişki puanı düşük olan öznitelik kaldırılır. Yüksek puana sahip öznitelikler en iyi öznitelik alt kümesi oluşturmak için seçilir. Daha sonra, oluşturulan öznitelik alt kümesi, sınıflandırma algoritmasına girdi olarak sunulur. Çok değişkenli ve tek değişkenli yöntemler olarak kendi içinde gruplandırılmaktadırlar. Çok değişkenli yöntemler, öznitelikler arasındaki bağımlılıkları kullanırken, tek değişkenli yöntemler öznitelikler arasındaki bağımlılıkları göz ardı ederek her özelliği tek tek ele almaktadır. Filtreleme yöntemleri, yüksek boyuttaki verileri kolay ve hızlı bir şekilde ölçeklendirme avantajına sahipken, sınıflandırma algoritmalarından bağımsız çalışmaları düşük doğruluk elde edilmesine neden olabilmektedir [46,47]. Yaygın kullanılan filtreme yöntemleri; Bilgi Kazancı (Information Gain), Kazanç Oranı (Gain Ratio), Korelasyona Dayalı Öznitelik Seçimi (Correlation-based feature selection-CFS), Saklı Markov Filtreleme (Markov Blanket Filtered -MBF) vb.' dir.



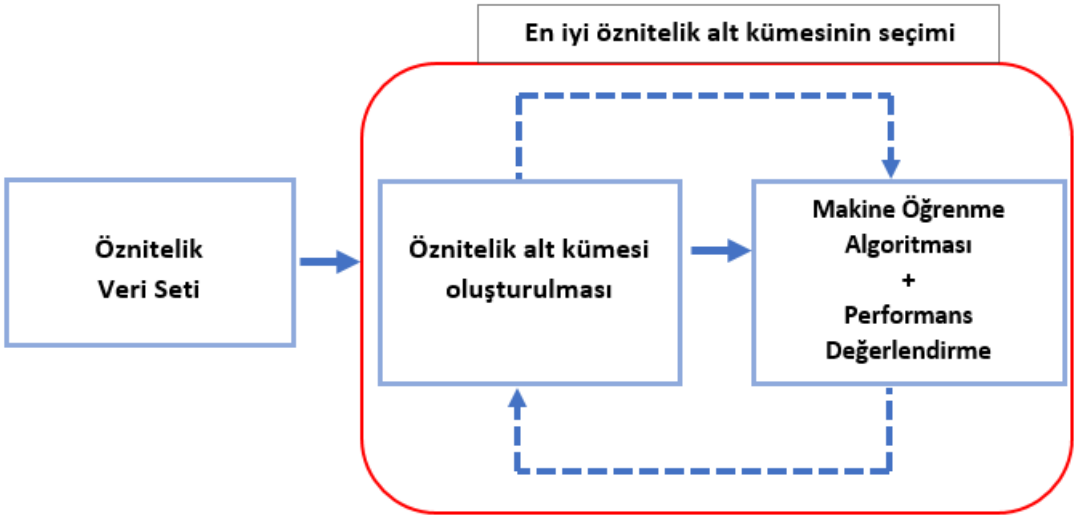
Şekil 3.1. Filtreleme yöntemi öznitelik seçim aşamaları

Sarmal yöntemler, makine öğrenme algoritmasında kullanılan veri kümesindeki tüm öznitelikler denenerek en iyi öngörü performansını gösteren öznitelikler seçilmektedir [48]. Şekil 3.2' de sarmal yöntemlerde kullanılan öznitelik seçim aşamaları gösterilmiştir. Bu yöntemlerde öznitelik alt kümesi uzayında bir arama fonksiyonu tanımlanarak çeşitli öznitelik alt kümeleri oluşturulur. Oluşturulan öznitelik alt kümelerinin değerlendirilmesi, makine öğrenmesi modellerinin eğitilerek test edilmesi ile gerçekleştirilir. Sarmal yöntemlerin, filtreleme yöntemlerine göre doğruluk oranları yüksek olsa da özellikle büyük veri kümelerinde hesaplama maliyetleri yüksek olabilmektedir [49]. Sarmal yöntemlerde, sıralı arama algoritmaları ve buluşsal arama algoritmaları kullanılmaktadır.



Şekil 3.2. Sarmal yöntem özniteliklerinin seçim aşamaları

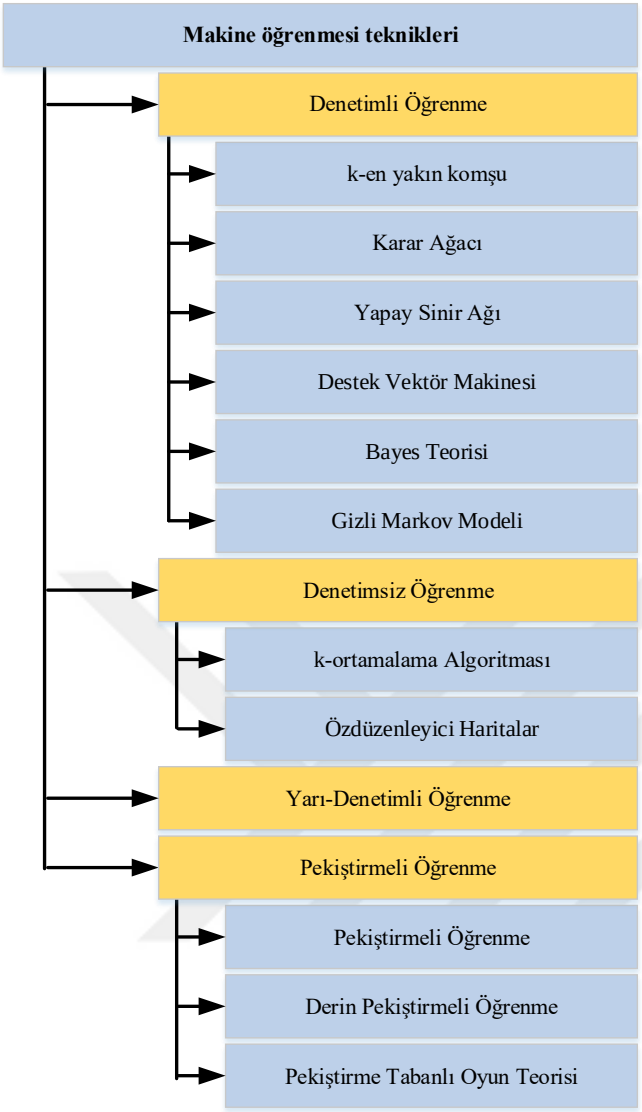
Gömülü yöntemler, makine öğrenme algoritmaları ile beraber kullanılmaktadır. Şekil 3.3' de gömülü yöntem öznitelik seçim aşamaları gösterilmiştir. Bu yöntemlerde, girdi öznitelikleri ile çıktı arasındaki ilişkileri ve öznitelik bağımlılıkları araştırılarak en iyi öznitelik alt kümesini oluşturulur. Öngörü ve öznitelik seçimi süreçlerini eş zamanlı olarak yürütme avantajına sahip olduğu için sarmal yöntemlere göre çok daha az hesaplama maliyetine sahiptir [47].



Şekil 3.3. Gömülü yöntem öznitelik seçim aşamaları

3.3. Probleme Göre Makine Öğrenme Algoritmasını Seçme

Makine öğrenmesinde, çözülecek probleme göre kullanılacak en iyi algoritmayı seçmek önemlidir. Ele alınacak probleme göre makine öğrenmesi algoritmaları farklı başarım sonuçları verebilmektedir. Aynı zamanda kullanılacak veri kümesinin boyutu, özniteliklerin sayısı ve dağılımı gibi parametreler de makine öğrenmesi başarım sonuçlarını etkileyebilmektedir. Bir makine öğrenme algoritması, denetimli, denetimsiz, yarı denetimli veya pekiştirmeli öğrenmeye dayalı olabilir. Şekil 3.4, yaygın olarak kullanılan makine öğrenmesi algoritmalarının bir listesini özetlemektedir [50].



Şekil 3.4. Yaygın kullanılan makine öğrenme algoritmaları [50]

Literatürde, ağ trafiği sınıflandırma ve yönlendirmede yaygın olarak kullanılan makine öğrenmesi algoritmaları; k-En Yakın Komşular, Karar Ağacı Sınıflandırıcı, Yapay Sinir Ağları ve Destek Vektör Makineleri vb.'dir.

3.4. Model Performansını Değerlendirme Ölçütleri

Bir makine öğrenme algoritması ile eğitim gerçekleştirildikten sonra bir model elde edilir, bu modelin performans değerlendirmesi için test edilmesi gerekir. En yaygın ölçüt modelin doğruluk (accuracy) değeridir. Doğruluk değeri modelin test veri kümesinde ne kadar doğru veya kesin öngörüde bulunduğunu gösteren değerdir. Bunun dışında model başarımını gösteren diğer ölçütler; duyarlılık (sensitivity), kesinlik (precision), özgüllük (specificity)

ve F1-puanı (F1-score) gibi parametrelerdir. Bu ölçütlerin hesaplanmasında karmaşıklık matrisinden yararlanılmaktadır. Karmaşıklık matrisi (confusion matrix), öngörü ve gerçek değerleri içeren dört farklı sayısal endeksten oluşmaktadır. Çizelge 3.1' de karmaşıklık matrisi gösterilmiştir. Burada gerçek pozitif (GP) ve gerçek negatif (GN), doğru tahmin edilen değerleri temsil ederken, yanlış pozitif (YP) ve yanlış negatif (YN) yanlış tahmin edilen değerleri temsil eder. Ayrıca, model performansını değerlendirmek için alıcı işletim karakteristiği (receiver operating characteristic-ROC) ve eğrilerin altındaki alan (area under curve-AUC) kullanılmıştır. ROC eğrisinin yatay eksende yanlış pozitif oranı ve dikey eksende gerçek pozitif oranı vardır.

Çizelge 3.1. Karmaşıklık matrisi

	Gerçek Sınıf	
Öngürülen Sınıf	DP	YP
	YN	DN

Karmaşıklık matrisi kullanılarak elde edilen ölçütler Çizelge 3.2' de verilmiştir. Makine öğrenmesi modeli ile ilgili değerlendirmelerin yapılmasında bu ölçütlerden elde edilen sonuçlar kullanılır.

Çizelge 3.2. Karmaşıklık matrisi ölçütler

Ölçütler	Formülasyon	Tanımlama
Doğruluk (Accuracy)	$\frac{DP + DN}{DP + DN + YP + YN}$	Modelin ne sıklıkla doğru tahmin ettiğinin ölçüsü
Duyarlılık (Sensitivity)	$\frac{DP}{DP + YN}$	Modelin ne kadar gerçek pozitif değeri doğru tahmin ettiğinin bir ölçüsü
Özgüllük (Specificity)	$\frac{DN}{DN + YP}$	Modelin ne kadar gerçek negatif değeri tahmin ettiğinin ölçüsü
Kesinlik (Precision)	$\frac{DP}{DP + YP}$	Modelin tüm sınıflardan, doğru olarak ne kadar tahmin edildiğinin bir ölçüsü
F1-puanı (F1-Score)	$\frac{2 * DP}{2 * DP + YP + YN}$	Modelin tüm veri kümesindeki doğruluğu.

Bir makine öğrenmesi problemi için hangi performans ölçütüne göre değerlendirme yapılacağı esas olarak hedeflenen duruma göre değişkenlik göstermektedir. İyi bir öngörü için kullanılan makine öğrenmesi modeli, Yanlış Negatifleri (YN) ve Yanlış Pozitifleri (YP) en aza indirmeyi amaçlamaktadır [51]. Tez kapsamında makine öğrenmesi modellerinin performans değerlendirmeleri Çizelge 3.2.'de verilen ölçütler kullanılarak yapılmıştır.

4. LİTERATÜRDE YER ALAN ÖNCEKİ ÇALIŞMALAR

YTA' nın sahip olduğu merkezi kontrol yeteneği ve programlanabilirlik özellikleri sayesinde ağ trağinin tek bir noktadan kolayca yönetilmesi sağlanmıştır. YTA kontrolcüsü üzerinden özgün ağ yönetim kuralları oluşturularak tüm ağa tek bir noktadan hızlı bir şekilde uygulanabilmektedir. Araştırmacalar çalışmalarında, ağ trafik sınıflandırması ve ağ güvenliği alanlarında YTA' nın sağlamış olduğu avantajlardan yararlanarak farklı uygulamalar geliştirmiş ve açık araştırma konularını belirlemişlerdir.

4.1. Yazılım Tanımlı Ağ ile Trafik Sınıflandırma

Ağ trafiğinin sınıflandırılması bir ağın genel performansının yönetilmesi için çok önemlidir. Sınıflandırılmış trafik ile ağ geneline özelleştirilmiş QoS, trafik dengeleme (load balancing) ve güvenlik gibi politikalar oluşturulabilmektedir. Ağ trafiğinin izlenmesi, hem çevrimiçi (gerçek zamanlı) hem de çevrimdışı (pasif olarak) yapılabilmektedir. Gerçek zamanlı trafik izleme yapılırken ağ üzerinden akan veri paketleri anlık olarak yakalanarak analiz edilerek sınıflandırılır. Çevrimdışı trafik izleme yapılırken ise, paketler ilk önce yakalanıp depolanır, daha sonra analiz edilerek sınıflandırılır [52]. Ağ trafiğini dinlemek ve akış bilgilerini izlemek için birçok yazılım kullanılmaktadır. En çok bilinen ve kullanılan yazılımlar Tcpdump ve Wireshark yazılımlarıdır. Basit ağ yönetim protokolü (simple network management protocol- SNMP) [53] kullanılarak ağ ölçüm verileri çekilmektedir. Ağ ölçüm ve ve izleme için genel olarak çekme ve itme tabanlı mimarilerine dayanan yazılımlar kullanılmaktadır [54]. Çekme tabanlı mimarileri kullanan yazılımlarda merkezi toplama birimi belirli zaman aralıklarında ağ ölçüm değerlerini talep etmektedir. İtme tabanlı mimarileri kullanan yazılımlarda ise belirli zaman aralıklarında ölçüm değerleri merkezi toplama birimine gönderilmektedir. sFlow, NetFlow, Graphite, StatsD ve Ganglia gibi uygulamalar bu mimariyi kullanan yazılımlara örnek olarak verilebilir.

Ağ izleme uygulamaları aracılığı elde edilen trafik verileri protokol bazında sınıflandırılabilir. Geleneksel ağlarda ağ trafiği sınıflandırması için kural tabanlı, yük tabanlı ve korelasyon tabanlı gibi yöntemler kullanılmaktadır. Fakat bu yöntemlerin kendilerine özgü bazı sorunları bulunmaktadır [51].

Kural tabanlı yöntemler, önceden tanımlanmış kurallar ağdaki paketleri sınıflandırmak için kullanılır. Ağ paket başlık bilgisi ve bağlantı port numarası bilgileri sınıflandırmanın temel aldığı özelliklerdir. Ağ trafik akış bilgisinden alınan port numarası bilgisi IANA (Internet Assigned Numbers Authority) tarafından belirlenen protokollere ait port numarası bilgisi ile karşılaştırılarak sınıflandırma gerçekleştirilir [55]. Örneğin DNS trafiği 53 numaralı portu kullanırken FTP trafiği 21 numaralı portu kullanmaktadır. Çizelge 4.1' de sık kullanılan uygulamaların kullandıkları port numaraları listelenmiştir.

Çizelge 4.1. IANA port numaraları [55]

Port Numarası	Protokol	Uygulama
20	TCP	FTP Data
21	TCP	FTP Control
22	TCP	SSH
23	TCP	Telnet
25	TCP	SMTP
53	UDP, TCP	DNS
67,68	UDP	DHCP
69	UDP	TFTP
80	TCP	HTTP
110	UDP	POP3
161	UDP	SNMP
443	TCP	SSL
16,384-32,767	UDP	RDP- based Voice and Video

Bu yöntem bilinen uygulamalar için oldukça yüksek başarım sağlasa da dinamik port numarası kullanan uygulamalar için problem oluşturur. Bu yüzden ağ operatörleri farklı sınıflandırma çözümleri kullanmak istemişlerdir [56].

Yük tabanlı yöntemler, kural tabanlı yöntemlerdeki dinamik port kullanımı ile sınıflandırma başarımının düşmesi araştırmacıları ağ üzerinde akan paketlerin taşımış olduğu yük bilgilerini (payload) kullanarak sınıflandırma yapma çalışmalarına yönlendirmiştir. Derin Paket İncelemesi (Deep Packet Inspection-DPI) olarak da adlandırılan bu yöntem ile bağlantı numarası ve IP adres bilgisi bağımlılığı ortadan kaldırılmıştır. Kullanılan protokol başlık bilgisi değişse de paket yükleri içerisindeki karakteristik imza veya desenlerin eşleştirilmesi yapıldığı için sınıflandırma başarımı yüksek olmaktadır. Ancak derin paket analizi için kullanılan donanımların maliyetlerinin yüksek oluşu ve son dönemde artmakta olan şifreli

paket içeriğinin tespit edilememesi sorunu, bu yöntemlerin sınırlılıklarını oluşturmuştur [57].

Korelasyon tabanlı yöntemler, kural tabanlı ve yük tabanlı yöntemlerde karşılaşılan sınırlılıkları ve zorlukları ortadan kaldırmak için araştırmacılar tarafından kullanılmıştır. Ağ trafiğini oluşturan akışların; paket boyutu, paket varış hızları, akış süresi gibi istatistiksel özelliklerini kullanılır. Ağ trafik sınıflandırma sürecine Karar Ağacı, Destek Vektör Makinesi, k-En Yakın Komşular gibi farklı makine öğrenme teknikleri de dahil edilerek sınıflandırma gerçekleştirilir. Sınıflandırma doğruluğu özellikle ağ paket içeriği ile ilgilenilmediği için şifreli trafik içeren paketlerde nispeten yüksektir. Bunun yanında her akışta yapılan korelasyon incelemesi ek hesaplamalar gerektirdiği için sınıflandırılmış veri kümeleri oluştururken ek tüketim meydana getirmektedir. Geleneksel ağlar birçok farklı protokol tarafından yönetilen çok sayıda yönlendirici ve anahtarlardan oluşmaktadır. Bu dağıtık yapı sebebi ile makine öğrenme yöntemlerinin geleneksel ağlar üzerinde uygulanması önemli bir zorluk oluşturmaktadır [58].

YTA mimarisi geleneksel ağ mimarisinden farklı olarak bütün altyapı bileşenlerini tek bir kontrol noktasında birleştirerek ağ yönetiminin merkezi bir kontrolcü üzerinden yapılmasını sağlamaktadır. Kontrolcü OpenFlow protokolü aracılığı ile akış bilgilerini toplar ve ilgili yönlendirme kurallarını doğrudan tanımlayabilir. Böylece YTA ağ yöneticilerine herhangi bir fiziksel donanıma ihtiyaç duymadan yazılımsal olarak ağ trafik analizi imkânı vermektedir. Makine öğrenimi tabanlı ağ trafik sınıflandırma yöntemleri, ağ trafik verilerinden öğrenilen istatistiksel öznitelikleri kullanır. Paket içeriği sınıflandırma için kullanılmaz bundan dolayı şifrelenmiş verileri sınıflandırabilir. Makine öğrenimi yöntemleri, trafik sınıflandırması için gerekli bilgilere YTA' daki OpenFlow protokolü aracılığı ile ulaşır ve düşük bir hesaplama maliyeti ile sınıflandırma gerçekleştirir.

Makine öğrenmesi kullanılarak yapılan trafik sınıflandırması için çeşitli yaklaşımlar kullanılmıştır. Wang ve arkadaşları çalışmalarında, yarı denetimli öğrenme algoritması ve derin paket incelemesi yönteminden yararlanarak QoS bilinçli bir trafik sınıflandırma sistemi önermişlerdir [59]. Önerilen yöntem ile hem bilinen hem de bilinmeyen uygulamaların trafik akışları farklı QoS sınıflarına ayrılmıştır.

D. Rossi ve S. Valenti çalışmalarında, UDP (User Datagram Protocol) protokolü üzerinden çalışan uygulamaların sınıflandırılmasına odaklanmışlardır [60]. Netflow ağ izleme aracı ile elde edilen UDP ağ trafiği SVM algoritması ile sınıflandırılmaktadır. He ve arkadaşları çalışmalarında, en uygun akış özniteliklerini ve en etkili makine öğrenme sınıflandırıcısını dinamik olarak belirleyen "vTC" adını verdikleri yazılım tanımlı sanal bir trafik sınıflandırma modeli önermişlerdir [61]. Temel ve gelişmiş akış öznitelikleri belirlenmiş ve 6 farklı sınıflandırma algoritması; k-En Yakın Komşular, Destek Vektör Makineleri, Yapay Sinir Ağları, Karar Ağaçları, Naive Bayes ve Adaboost kullanmışlardır.

Amaral ve arkadaşları çalışmalarında, YTA temelli bir trafik sınıflandırma mimarisi önermişlerdir [62]. Youtube, Vimeo, Facebook, Linkedin, Skype, Bittorrent, Web Tarama (HTTP) ve Dropbox uygulamalarının her biri için 500 örnek toplanmış ve belirlenen makine öğrenme algoritmaları (rastgele orman, gradyan arttırma ve aşırı gradyan arttırma) ile bu trafik örnekleri sınıflandırılmaya çalışılmıştır.

Wang ve arkadaşları tarafından akıllı ev ağlarını daha iyi yönetmek için Software Defined Network Home Gateway (SDN-HGW) modeli önerilmiştir [63]. Çalışmada, Çok Katmanlı Algılayıcı (Multi-Layer Perceptron, MLP), Yığın Otomatik Kaydedici (Stacked Autoencoder, SAE) ve Evrişimsel Sinir Ağları (Convolutional Neural Network, CNN) olmak üzere üç derin öğrenme tabanlı yaklaşım kullanarak şifreli veri paketi sınıflandırıcısı geliştirilmiştir. DataNets adı verdikleri bu sınıflandırıcıların geliştirilmesi için 15 uygulamadan alınan 20.000' den fazla şifreli paketleri de içeren veri kümesi kullanılmıştır. Elde edilen sonuçlar, geliştirilen DataNet' lerin akıllı ev ağında gerçek zamanlı işleme için doğru paket sınıflandırması ve yüksek hesaplama verimliliği ile SDN-HGW modeline uygulanabileceğini göstermiştir.

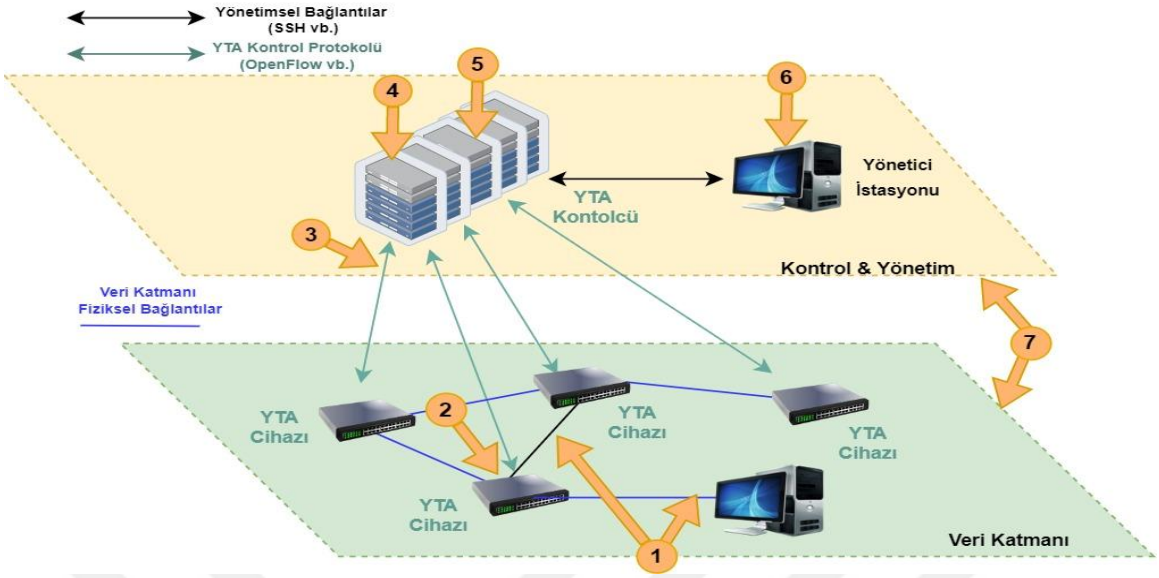
Lim ve arkadaşları çalışmalarında, derin öğrenme yöntemlerini kullanarak trafik sınıflandırması yapabildikleri bir YTA mimarisi önermişlerdir [64]. Uzak Masaüstü Bağlantısı (Remote Desktop Connection, RDP), Skype, Güvenli Kabuk (Secure Shell, SSH), Bittorent ve Hiper-metin Transfer Protokolü (Hyper-Text Transfer Protocol, HTTP) trafiklerinden oluşan veri kümesi üzerinde sınıflandırma yapmak için çok katmanlı Uzun Kısa Süreli Bellek (Long Short Term Memory, LSTM) modeli ve Evrişimsel Sinir Ağları ile tek katmanlı Uzun Kısa Süreli Bellek modellerinin kombinasyonu içeren derin öğrenme modelleri kullanılmıştır.

Meenaxi M Raikar ve arkadaşları çalışmalarında, yoğun veri trafiğinden dolayı geleneksel trafik sınıflandırma yaklaşımlarının sınırlılıklara sahip olduğunu belirtmişlerdir [65]. Bu sınırlıkların üstesinden gelmek için YTA mimarisi ve makine öğrenme metotlarının birlikte kullanıldığı bir model önermişlerdir. HTTP, E-mail ve Görüntü-Ses (Streaming) trafik verileri Destek Vektör Makinesi, Naïve Bayes ve k-En Yakın Komşular algoritmaları ile sınıflandırılmıştır.

Yapılan çalışmalar incelendiğinde ağ trafik sınıflandırmasında makine öğrenmesi metotlarının ve YTA mimarisinin kullanımının yaygınlaştığı görülmektedir. Artan ağ trafiği miktarı ve çeşitliliği ile birlikte geleneksel trafik sınıflandırma yöntemlerinin yerini makine öğrenmesi yöntemleri almaktadır. Bunun yanında YTA mimarisi sahip olduğu mimari yapı ile geleneksel ağlarda kullanılan güvenlik duvarları, trafik dengeleme, ağ ölçüm ve izleme gibi eskiden oldukça zor ve karmaşık olan ağ işlevlerini çok daha kolay ve basit bir hale getirmektedir. Geleneksel ağ ölçüm ve izleme yöntemlerinin hemen hemen hepsi uç cihazlardaki değişkenlerde tutulan Yönetim Bilgi Birimi (Management Informatin Base, MIB) değerlerin biriktirilip işlenmek üzere merkezdeki sunuculara gönderilmesi, daha sonra bu değerlerin analiz edilerek ağın durumuna ilişkin çıkarım yapma esasına dayanmaktadır [53].

4.2. Yazılım Tanımlı Ağ Güvenliği

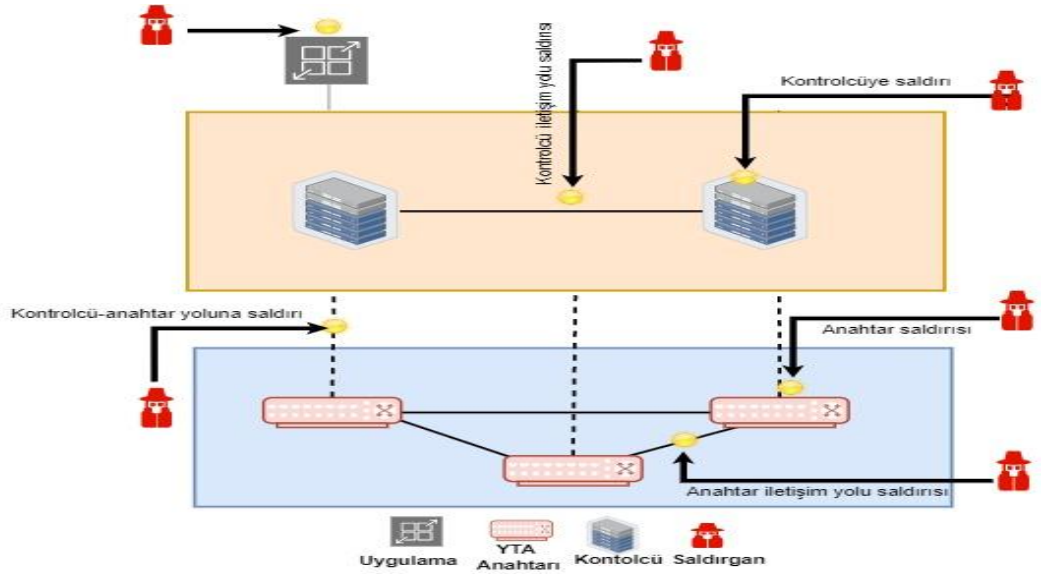
YTA, mimarisi ve tasarımı ile ilgili çeşitli karakteristik özelliklere sahiptir. Bu karakteristik özellikler onu geleneksel ağ mimarilerinden farklı kılmaktadır. YTA, ağları hızlı ve verimli bir şekilde daha fazla esneklikle yönetmeye yardımcı olur. Ayrıca güvenlik bakış açısı ile ağdaki cihazların merkezi olarak kimliğinin doğrulanması ve analiz edilmesini sağlayarak saldırıları tespit etmeyi ve azaltmayı kolaylaştırır [66]. Ancak, mimari yapısından kaynaklanan bazı kusurlar nedeniyle YTA' nın kendisi güvenlik tehditlerine karşı savunmasız hale gelebilmektedir. Genel YTA çerçevesi, her bir düzlemle ilgili güvenlik sorunlarını ayrı ayrı ele almak için birkaç katmana (veya birkaç araştırmacı tarafından önerilen güvenlik vektörlerine) bölünebilir. D. Kreutz ve arkadaşları yaptıkları çalışmada, Şekil 4.1' de gösterildiği gibi YTA mimarisine yönelik, üç tanesi YTA'e özgü olmak üzere, yedi olası saldırı vektörü tanımlamışlardır [1]. Farklı düzlemlerde çalışan cihazların ve iletişim kanallarının hedeflendiği bu saldırılara yönelik alınabilecek önlemler yine bu çalışmada ortaya konulmuştur.



Şekil 4.1. YTA saldırı vektörleri [1]

Zhaogang Shu ve arkadaşları yaptıkları çalışmada; YTA mimarisine yönelik tehditleri ve alınabilecek karşı önlemleri, saldırının meydana geldiği YTA düzlemine (uygulama, kontrol ve veri düzlemi) göre 3 gruba ayırmışlardır [11]. Priyadarsini ve arkadaşları ise çalışmalarında; YTA' ya yönelik saldırıları, düzlemler arasındaki iletişim arayüzlerine yönelik saldırıları da içerecek şekilde kontrol düzlemi saldırıları, veri düzlemi saldırıları, kuzey ve güney arayüz saldırıları olmak üzere 4 grupta incelemişlerdir [67].

Şekil 4.2, YTA mimarisindeki çeşitli saldırı noktalarını göstermektedir;



Şekil 4.2. YTA saldırı noktaları

Bunlar; saldırı noktası 1 ile temsil edilen OpenFlow anahtarlarını, saldırı noktası 2 tarafından temsil edilen iki anahtar arasındaki iletişim yolunu, saldırı noktası 3 tarafından temsil edilen kontrolcüyü, saldırı noktası 4 ile temsil edilen kontrolcü ile anahtar arasındaki iletişim yolunu, saldırı noktası 5 ile temsil edilen iki kontrolcü arasındaki iletişim yolunu, saldırı noktası 6 ile temsil edilen uygulama düzleminde çalışan YTA uygulamalarını ifade etmektedir.

YTA anahtarı: YTA anahtarları, yeni gelen paketlerin iletilmesi için akış tablosunda yer alan kuralları kullanmaktadır. Kontrolcüden gelen kurallar TCAM (ternary content addressable memory) olarak bilinen "üçlü içerik adreslenebilir bellek" ile anahtarda saklanır. Bu bellek türü yüksek hızlı işlemler için kullanılan benzersiz bellek türüdür [68]. YTA anahtarları sınırlı bellek kapasitesine sahip olduğu için yoğun istek talebinde kullanılamaz hale gelmektedir. Akış tablosu belleğinin sınırlı kapasiteye sahip olması, özellikle DDoS gibi saldırılar için YTA anahtarlarını hedef haline getirmektedir [69].

YTA anahtarları arasındaki bağlantılar: Akış paketleri, bir anahtardan başka bir anahtara iletilmek üzere aktarılır. Aktarılan paketlerin çoğu şifrelenmemiştir ve hassas bilgiler içerebilir. Bu paketler, özellikle anahtarlar arasındaki bağlantılar kablosuz olduğunda, saldırganlar tarafından kolayca ele geçirilebilir [11].

YTA kontrolcüsü: Kontrolcü YTA'nın beyni gibi çalışmaktadır. Ağın küresel görünümüne sahip olan kontrolcü bütün fonksiyonel işlemlerin gerçekleştirilmesinden sorumludur. Merkezi kontrolcünün etkisiz kılınması ya da ele geçirilmesi bütün ağın yönetiminin kaybedilmesi anlamına gelmektedir. Ağda tek bir kontrolcünün olması durumunda DDoS gibi hacimsel saldırılara karşı YTA savunmasız kalabilmektedir [70,71].

Kontolcü ve anahtar arasındaki bağlantı: Bir paket anahtara ulaştığında akış tablosunda ilgili iletim kuralı bulunmadığı durumda kontrolcü ile iletişim kurulur. Güncel akış kuralları güneye giden arabirim aracılığı ile anahtara iletilir. Bu kurallar kontrolcü ile anahtar arasındaki bağlantının gizlice dinlenmesi yoluyla saldırgan tarafından tahrif edilebilir, bu da bazı kötü niyetli kuralların eklenmesine veya mevcut kuralların değiştirilmesine neden olabilir [67].

İki kontrolcü arasındaki bağlantılar: Bazı ağ topolojilerinde merkezi kontrolcünün yükünü azaltarak daha etkin ve güvenli bir yönetim sağlamak amacı ile birden çok kontrolcü kullanılabilmektedir [72]. Bu topolojilerde, iletişim doğu-batı bağlantılı API' ler aracılığıyla kontrolcüler arasında paylaşılır. Kontrolcüler arasındaki bu iletişim yolu, saldırganlar tarafından ele geçirilebilir [73]. Bu nedenle, kontolcüler arasındaki iletişim güvenli ve özgün olmalıdır.

Uygulamalar: Yük dengeleme, trafik izleme ve sanallaştırma gibi uygulama yazılımları YTA kontrol düzlemi üzerinde bulunmaktadır. Bu yazılımların çoğu genellikle güvenlik gereksinimlerini göz ardı eden üçüncü şahıslar tarafından kurulmaktadır. Kontrolcü ile iletişim kuzey arayüzü üzerinden sağlandığında, kontrolcüye kötü amaçlı kodlar bu uygulamalar üzerinden yerleştirilebilir [74]. Bundan dolayı YTA tabanlı uygulamalar kontrolcünün hizmetini engellemek için hedef nokta haline gelebilmektedir.

4.2.1. Yazılım tanımlı ağ saldırı çeşitleri

Tüm düzlemler ve arabirimler, düzlemin kendisinde bulunan ağ bileşenlerini tehlikeye atabilecek veya başka bir düzlemdeki öğeleri hedefleyebilecek belirli saldırılara karşı hassastır. Çizelge 4.2' de YTA düzlemlerini ve arabirimlerini hedefleyen yaygın saldırı türleri gösterilmiştir. Ayrıca Şekil 4.2' de gösterilmiş saldırı noktaları detaylandırılarak YTA düzlemlerine etkisi de bu bölümde açıklanmaktadır.

Çizelge 4.2. YTA saldırı çeşitleri

Saldırı	Veri Düzlemi	Güney Arayüzü	Kontrol Düzlemi	Kuzey Arayüzü	Uygulama Düzlemi
Akış Tablosu Saldırısı	X				
Anahtar TCAM tükenmesi	X				
Topoloji Sızdırma Saldırısı	X				
TLS Saldırısı	X	X	X		
Bant Genişliği Tüketme Saldırısı	X	X	X		
LLDP Sızdırma Saldırısı	X				
Kontrolcü Ele Geçirme Saldırısı	X	X	X		
Kontrolcü Kaynak Tüketme Saldırısı			X		
Kötü Amaçlı Uygulamalar				X	X
Politika Çatışma Saldırıları			X	X	X

Veri düzlemi saldırıları

Veri düzleminde yer alan YTA anahtarı, sadece kontrolcü tarafından oluşturulmuş akış kurallarına göre trafiği yönlendirir. Anahtara yapılan saldırılar sonucunda bu yönlendirme görevi kesintiye uğrayabilir ve tüm ağ zarar görebilir. Saldırganlar iki şekilde YTA anahtarına saldırıda bulunabilir. İlk olarak, iki YTA anahtarı arasındaki iletişim sırasında sahte akışlar üretilebilir. İkinci olarak da saldırgan, sel saldırısı oluşturarak anahtarı sürekli kontrolcüye yeni kuralları sormaya zorlayıp, akış tablosunu doldurabilir. Bu da anahtarın sınırlı kaynaklarının tükenmesine ve performans kaybına neden olabilir [75]. Veri düzlemindeki yaygın görülen saldırılar DoS, DDoS [76], aldatma [77] ve izinsiz giriş [78] saldırılarıdır. Aldatma saldırıları, akış kurallarında kötü niyetli değişikliklere neden olurken DoS ve DDoS saldırıları ile saldırgan ilgili servisleri meşru kullanıcılar için kullanılamaz hale getirir. Ağ izinsiz giriş saldırıları ile veri sızıntısı meydana gelebilmektedir.

Güney arayüzü saldırıları

Veri düzlemi ve kontrolcü arasındaki iletişimi sağlayan güney arayüz güvenliği ihlal edilirse her iki uca da kötü amaçlı trafik gönderilebilir. Güney arayüz, OpenFlow protokolünü kullanır. Bu protokol üzerindeki açıklıklar kullanılarak ortadaki adam (POODLE) saldırıları gerçekleştirilmektedir [79]. POODLE saldırısı ile saldırgan kontrolcünün TLS sürümünü SSL 3.0'a düşürür ve saldırı için SSL 3.0 güvenlik açığını kullanır. Böylece saldırgan verileri ele geçirme ve değiştirme, kimlik sahtekarlığı gibi güvenlik ihlallerinde bulunabilmektedir. Bunun dışında DoS ve DDoS saldırıları da güney arayüz protokolüne yönelik gerçekleştirilmektedir [76].

Kontrol düzlemi saldırıları

Kontrol düzlemi, ağın tüm trafiğinin yönetildiği merkezi kontrolcünün bulunduğu düzlemdir. Bütün ağın yönetimini ele geçirmek isteyen saldırganlar için kontrolcü ana hedef olmaktadır. Saldırgan, kontrolcüye bir ve veya daha çok kaynaktan fazla sayıda sahte paket göndererek kontrolcünün sınırlı kaynaklarını tüketebilir. Kontrolcüye yapılan DoS ve DDoS saldırıları, tüm ağın işleyişini olumsuz yönde etkileme potansiyeline sahiptir. Kontrolcüyü ele geçirme veya yetkisiz erişim saldırıları, güvenliğin gizlilik ilkesini hedefler. Kontrolcüdeki güvenlik açıkları, tüm ağı tehlikeye atabilecek sonuçlara yol açabilir [80].

Saldırgan, yanlış yapılandırılmış, savunmasız kontrolcünün yönetimini ele geçirerek tüm ağın yönetimini kontrol edebilir. Ardından, veri düzlemindeki YTA anahtarlarını programlayarak kontrolcüye gelen trafiği düşürebilir ve diğer hedeflere saldırı başlatmak için kullanabilir [1].

Kuzey arayüzü saldırıları

Kuzey arayüzünde kullanılan protokoller uygulama düzlemi ile kontrol düzlemi arasında iletişim kurmak için kullanılmaktadır. İzinsiz giriş saldırısı burada karşılaşılabilecek olası büyük saldırı türlerinden biridir [78]. Bu iletişim için standart bir protokol olmaması güvenlik açısından bir zorluk oluşturmaktadır [81].

Uygulama düzlemi saldırıları

Uygulama düzleminde çalışan birçok YTA uygulaması yer almaktadır. Yetkisiz uygulamalar, güvenliğin gizlilik ve bütünlük ilkesini hedeflemektedir. Kontrolcü, YTA uygulamaları için ağ üzerinde yetkilendirme sağlayarak uygulamaların ağ durumunu okumasını ve yazmasını sağlamaktadır. Bu durum ağ güvenliği için sorun yaratmaktadır. Kötü amaçlı YTA uygulamaları, ağdaki saldırgan etkinliklerini gizlemek, kritik kaynaklara erişmek ve ağın işleyişini manipüle etmek için kullanabilir [82]. Bu düzlemde çalışan uygulamalar da saldırganların hedefi olabilmektedir. Saldırganlar kaynak yoğun istekler göndererek YTA uygulamalarını hizmet veremez hale getirebilmektedir. Web sunucusu bu saldırıdan en çok etkilenen uygulamalardan biridir [76]. Slowrois ve HTTP taşması gibi saldırılar bu düzlemde saldırganlar tarafından kullanılabilmektedir.

4.2.2. Yazılım tanımlı ağ güvenlik çözümleri

YTA, sağlamış olduğu merkezi yönetim ve programlanabilir yapı ile ağ teknolojileri için yeni bir mimari yaklaşım olarak kabul edilmektedir. Literatürde, YTA güvenliği ile ilgili yapılmış birçok kapsamlı çalışma yer almaktadır [10], [67], [83]. Bu çalışmalarda YTA' nın temel mimari özellikleri ve avantajlarının yanında mevcut güvenlik tehditleri ve çözüm önerileri ayrıntılı olarak tartışılmaktadır. Bu bölümde, YTA mimarisinin güvenlik açıklıkları dikkate alınarak çeşitli araştırmacılar tarafından yapılmış güvenlik çalışmaları analiz edilecektir.

Bu çalışmalar şunları içerir:

- YTA saldırılarının önlenmesi ve azaltılması,
- Kimlik doğrulama ve yetkilendirme mekanizmaları,
- YTA güvenliği konusunda Saldırı Tespit Sistemi/Saldırı Engelleme Sistemi ve Güvenlik Duvarı gibi uygulamaların geliştirilmesi.

Mantıksal olarak merkezileştirilmiş kontrolcü, YTA mimarisinin en savunmasız parçasıdır. YTA' ya özgü saldırı vektörlerinin incelendiği çalışmada farklı güvenlik çözümleri de sunulmaktadır [1]. Tek bir kontrolcü üzerindeki saldırılardan korunmak için kontrolcülerin ve uygulamaların yedeklerinin de oluşturulması önerilmektedir. Olası yazılım hatalarına karşı farklı kontrolcülerin kullanılması gerektiği belirtilmektedir. Ayrıca, merkezi kontrolcünün hizmet veremez hale gelmesi durumunda veri düzlemindeki anahtarların başka bir kontrolcü ile iletişim kurarak dinamik yönlendirme mekanizması ile hata toleransını düşürebileceği belirtilmektedir.

Musavi ve arkadaşları kontrolcüye yönelik hacimsel saldırıları engellemek için bir güvenlik sistemi önermişlerdir [84]. Önerilen sistem, YTA yapısını korumak için kontrolcünün işlevlerini kullanmaktadır. Saldırıyı hızlı bir şekilde tespit etmeyi ve kontrolcü hizmet veremez hale gelmeden önce uygun bir hafifletme çözümü sağlamayı amaçlamaktadır. Saldırı tespiti için hedef adrese dayalı bir entropi yöntemi kullanılmaktadır. Entropi, önerilen yöntemde önceden tanımlanan pencere boyutu ve eşik değeri gibi iki parametre kullanılarak hesaplanmaktadır. Kontrolcü üzerine entegre edilmiş modül ile gelen trafiğin entropi değeri hesaplanır. Belirli bir entropi eşiği aşılırsa bu izinsiz giriş olarak tespit edilerek engellenir. Böylece, kötü niyetli etkinlikler hızlı bir şekilde algılanır. Önerilen yöntem DNS ve Web trafiğinde de test edilmiştir. Önerilen yöntemin en önemli özelliği farklı senaryolarda uyarlanabilir olmasıdır. Kullanılan algoritmadaki parametreler, gerçek zamanlı olarak hedeflenen sonuçlara göre değiştirilebilir. Bu çalışma YTA kontrolcüsü üzerindeki entropiye dayalı ilk çözüm olması açısından önemlidir. Çalışmadaki sınırlılık tek bir kontrolcü mimarisi için önerilen bir çözüm olmasıdır. Başka bir entropi tabanlı yaklaşım Sahoo ve arkadaşları tarafından önerilmiştir [85]. Bu çalışmada kontrolcüdeki düşük oranlı saldırıların tespiti için genelleştirilmiş entropi ve farklı olasılık dağılımları arasındaki bilgi mesafesi kullanılmıştır. YTA anahtarları üzerinden elde edilen istatistiksel akış özellikleri kullanılarak erken DDoS saldırı uyarı sistemi önerilmiştir. Önerilen yaklaşım, Mousavi ve

arkadaşlarının yaklaşımıyla karşılaştırıldığında daha hızlı ve doğru tespit oranı sağlamaktadır [84]. Kumar ve arkadaşları entropiye dayalı "SAFETY" ismini verdikleri bir güvenlik çerçevesi önermişlerdir [86]. Algılama ve hafifletme modüllerinden oluşan bu güvenlik çerçevesi sabit zaman penceresinde hedef IP ve TCP başlık bilgilerini kullanarak entropiyi hesaplamaktadır. Algılama aşamasında, entropi değeri ile belirlenen eşik değer karşılaştırılarak saldırı tespiti yapılmaktadır. Saldırının hedefi ve kaynağı da algılama aşamasında tespit edilmektedir. Hafifletme aşamasında ise sınırlı sayıda SYN isteğine izin verilir ve istemci makineden SYN-ACK paketlerinden birinin onayı dönene kadar diğer SYN istekleri engellenir. Böylece kötü niyetli SYN paketleri engellenirken normal trafiğe izin verilmektedir.

Kontrol düzlemi bant genişliği, YTA kontrolcüsünden sonra saldırganların en çok hedef olarak belirlediği ikinci noktadır. Burada kullanılan OpenFlow protokolü bazı güvenlik problemlerine yol açmaktadır. Örneğin bir saldırgan çok fazla sayıda OpenFlow isteği gönderdiğinde kontrol düzlemi ile veri düzlemi arasındaki iletişim yolunda darboğaz meydana gelmektedir. Buradaki darboğaz bütün ağ iletişimini etkileyerek tüm YTA ağını yönetilemez hale getirmektedir. OpenFlow ağlarındaki güvenliği arttırmak amacı ile AVANT-GUARD adlı bir sistem geliştirilmiştir [87]. Geliştirilen sistem ile SYNFlood saldırılarının önlenmesi amaçlanmıştır. Kontrolcü ve anahtar arasında bir katman görevi gören sistem, TCP_SYN mesajlarının hedefe ulaşmadan kontrolcü tarafından kontrol edilmesini sağlamaktadır. Kontrolcü, SNORT benzeri araçlarla saldırı kontrolü yaparak yönlendirme kararı vermektedir. Elde edilen sonuçlar SYNFlood benzeri ağ tarama saldırılarında sistemin başarılı olduğunu göstermektedir. Ancak sistemin uygulama düzeyindeki UDP veya ICMP protokol tabanlı saldırılarda herhangi bir koruma sağlamadığı belirtilmektedir.

Fichera ve arkadaşları "OPERETTA" adı verdikleri çözüm ile SYNFlood saldırılarına karşı Avant-Guard sistemine benzer bir sistem geliştirilmiştir [88]. İstemcilerden gelen SYN istek sayıları kontrolcüde tutulmaktadır. İstemci kendisine gelen SYN_ACK paketi sonrası geriye ACK paketi göndermez ise belli bir eşik değerinden sonra kara listeye eklenmektedir. Kara listeye eklenen bu istemci ip adresine akış tablosunda düşür (drop) kuralı eklenmektedir. ACK cevap mesajı gelir ise beyaz listeye eklenerek anahtardaki ilgili akış tablosuna belirli bir zaman aşımı değeri ile eklenmektedir. OPERETTA' da Avant-Guard' dan farklı olarak, kaynak-hedef arasında güven ilişkisini kurduktan sonra kaynağa RST mesajı yollaması ve

anahtara da ilgili akış kaydını girerek kaynak ile hedefin tekrar haberleşmesini sağlamasıdır. Ağ güvenliğini sağlamak için Saldırı Engelleme Sistemi (Intrusion Prevention System-IPS) ve Saldırı Tespit Sistemi (Intrusion Detection System-IDS) gibi güvenlik çözümleri kullanılmaktadır. Artan saldırı çeşitliliği, bu sistemler üzerinde istatistiksel hesaplamalar yapmayı gerekli kılmıştır. Makine öğrenmesi algoritmaları ile Saldırı Tespit ve Saldırı Engelleme Sistemleri, anlamlı yorumlar ve tahminler yapabilme yeteneği kazanmıştır.

Braga ve arkadaşları tarafından yapılan çalışmada NOX kontrolcüsü ve OpenFlow destekli anahtarlardan oluşan bir ağda, DDoS saldırılarının tespiti için bir yöntem önerilmektedir [89]. Bu yöntemde, genellikle sınıflandırma problemlerinin çözümünde tercih edilen ve danışmansız öğrenme türünde bir Yapay Sinir Ağı olan Öz Düzenleyici Haritalar (Self-Organizing Maps- SOM) kullanılmaktadır. Akış toplayıcı (flow collector), özellik seçim (feature extractor) ve sınıflandırıcı (classifier) olmak üzere üç modülden oluşan bu yöntem NOX kontrolcüsü ile yan yana çalışmaktadır. OpenFlow anahtarlarından kontrolcü tarafından akış toplayıcı modülü aracılığı ile akış bilgileri toplanmaktadır. Akış başına ortalama paket sayısı, byte değeri, süre vb. trafik akışına ait özellikler kullanılarak eğitilen SOM ile ilgili trafik normal ve anormal olarak sınıflandırılmaktadır. Tespit mekanizması yeni saldırılara karşı güncellenebilir ve kullanılan sınıflandırma tekniği gerektiğinde değiştirilebilir. Tespit döngüsüne OpenFlow destekli yeni anahtarların eklenebilmesine veya çıkarılabilmesine olanak sağlamaktadır. Ağ topolojisi değiştiğinde tespit döngüsü de buna göre uyarlanabilir. Ayrıca, DDoS saldırılarının tespitinde yüksek oranda bir başarı sağladığı ve düşük oranda yanlış pozitif (false positive- FP) değerlerine sahip olduğu belirtilmektedir.

YTA' da anomali tespitine yönelik bir başka çalışmada ise kontrolcü üzerine konumlandırılmış veri toplayıcı (data collector), özellik seçimi (feature selection) ve sınıflandırma (classification) modüllerinden oluşan bir sistem önerilmektedir [90]. Öncelikle, ağdaki veriler toplanmakta, sonrasında önemli özellikler seçilmekte ve son olarak trafik normal veya anomali şeklinde sınıflandırılmaktadır. Özellik seçimi, anomalinin doğru bir şekilde tespit edilebilmesinde önemli bir rol oynamaktadır. En uygun özelliklerin seçilebilmesi sınıflandırma algoritmasının performansını arttırmaktadır. Bunun için de bazı optimizasyon teknikleri bulunmaktadır. Bu çalışmada, özellik seçimi için optimizasyon amaçlı üst-sezgisel bir algoritma türü olan İkili Yarasa Algoritması (Binary Bat Algorithm-BBA) ve sınıflandırma için entropi yöntemi kullanılmaktadır [91]. NSL-KDD [92] veri kümesinde normal trafiğe ve dört farklı saldırı çeşidine ait paketler bulunmaktadır. Özellik

seçimi için NSL-KDD veri kümesi BBA algoritmasına girdi olarak verilmiştir. Çıktı olarak istenmeyen özellikler elimine edilmiş ve geriye kalan özellikler sınıflandırma modülüne girdi olarak verilmiştir. DoS, tarama (probe), uzaktan yerel alan ağına oturum açma (remote to local-R2L) ve kullanıcı hesabını tam yetkili yönetici hesabına yükseltme (user to root, U2R) saldırılarının her biri için farklı özellikler seçilmiştir. Sınıflandırma işlemi için J48 Karar Ağacı algoritması kullanılmıştır [93]. Çalışmada, saldırıların tespit oranı ve yanlış pozitif oranı ölçülerek sınıflandırmanın başarısı değerlendirilmektedir. Saldırı tespit oranında en iyi sonucu U2R, yanlış pozitif oranında ise DoS vermiştir. Bu alanda, normal ve anormal trafiği ayırt etmek için makine öğrenmesi algoritmalarını kullanan birçok çalışma bulunmaktadır [9], [94]–[97].

Tuan A Tang ve arkadaşları Ağ Anomali Tespit Sistemi (Network Intrusion Detection System- NIDS) modeli adını verdikleri model ile derin öğrenme (deep learning) yöntemlerini YTA üzerinde kullanarak akış temelli saldırıları tespit etmeye saldırıları hafifletmeye çalışmışlardır [48]. Hacimsel saldırıları merkeze alan çalışmalarında NSL-KDD veri kümesini kullanmış, farklı derin öğrenme algoritma kullanım sonuçları karşılaştırılmıştır. Sonuç olarak akış temelli saldırıların derin öğrenme algoritmaları ile tespit edilip hafifletilebileceği ortaya konulmuştur. Benzer şekilde derin öğrenme yöntemleri kullanılarak YTA ortamlarında saldırı tespiti yapan farklı çalışmalar bulunmaktadır [98], [99].

Son yıllarda yapılan çalışmalarda YTA' ya yönelik saldırıların savunmasında blok zincir kullanımı önerilmektedir. Singh ve arkadaşları literatürdeki blok zinciri tabanlı saldırıları inceleyerek değerlendirmelerde bulunmuştur [100]. YTA' nın, tek bir etki alanındaki saldırıları azaltmak için kullanılabileceğini ancak birden çok etki alanına sahip saldırılarda blok zinciri ve YTA' nın birlikte, daha etkili olacağı belirtilmiştir. El Houda ve diğerleri, çalışmalarında yerel ağda makine öğrenimi tabanlı DDoS algılama ve etki azaltma bileşenlerinden oluşan bir çoklu YTA etki alanı çözümü olan CoChain-Sc ve etki alanları arası DDoS hafifletme için kullanılan Ethereum blok zinciri tabanlı bir modül önermiştir [101]. Benzer şekilde iki çalışma, farklı akıllı sözleşmeler kullanarak bir Ethereum blok zincirinden yasal olmayan IP' ler gönderen ve diğer yasal olmayan IP'lerin bir listesini alan saldırı hafifletme yaklaşımı önermişlerdir [102], [103].

Literatürdeki çalışmalar incelendiğinde YTA' da, geleneksel ağlarda kullanılan savunma yaklaşımlarının (saldırı tespit ve engelleme sistemi, derin paket inceleme, güvenlik duvarı ve balküpü) sanallaştırılarak ilgili sunucular üzerinde çalıştırılması yönteminin yaygın bir şekilde kullanıldığı görülmektedir. Bununla birlikte YTA' nın kendine özgü mimari yapısından kaynaklanan yeni saldırı vektörlerinin de çalışmalara konu olduğu tespit edilmiştir. Özellikle YTA anahtarlarının akış tablolarını ile kontrolcünün kaynaklarını hedef alacak saldırıların ağ iletişiminde ciddi kayıplara yol açabileceği öngörülmektedir. Bu nedenle öncelikle YTA anahtarlarını ve merkezi kontrolcüyü korumaya yönelik yöntemlerin geliştirilmesinin önemli bir konu olduğu değerlendirilmektedir.

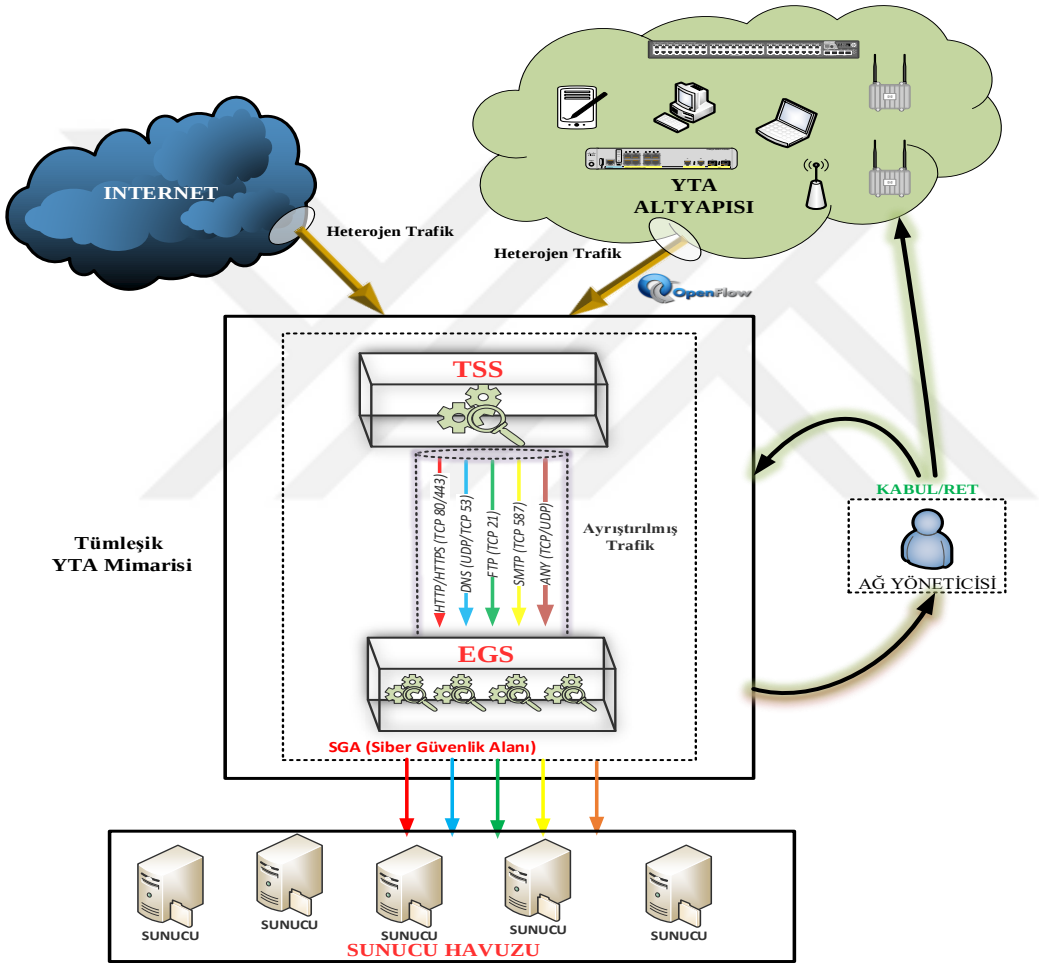
Bu çalışma ile YTA' larda kullanılabilecek bir siber güvenlik modeli önerilmektedir. Bu model YTA ve AFS' nin sağlamış olduğu merkezi kontrol ve sanallaştırma özelliklerini kullanarak sistem kaynaklarının verimli bir şekilde kullanılmasını sağlamaktadır. YTA kontrolcüsü ile bütünleşik olarak çalışan güvenlik sistemi ile daha kısa sürede, ağ trafik türüne özgü farklı güvenlik politikaları tüm ağa hızlı bir şekilde uygulanmaktadır.

Tez kapsamında geliştirilen matematiksel model ile ağ trafiği sürekli izlenip trafik süreklilik ölçümü yapılmaktadır. Belirli şartlar sağlandığında sistem, otonom bir şekilde hareket ederek siber güvenlik alanları oluşturup görevi bittiğinde kaldırma yeteneğine sahiptir.

Farklı trafik türlerine özgü oluşturulan siber güvenlik alanlarında, makine öğrenme modelleri ile ağ trafiği derinlemesine incelenmekte ve anomali özelliği gösterdiği durumda engellenmektedir. Böylece geleneksel saldırı tespit/engelleme sistemlerinde kullanılan "tüm ağ trafiğine tek bir güvenlik kuralının uygulanması" yaklaşımından farklı bir güvenlik yaklaşımı ortaya konulmuştur.

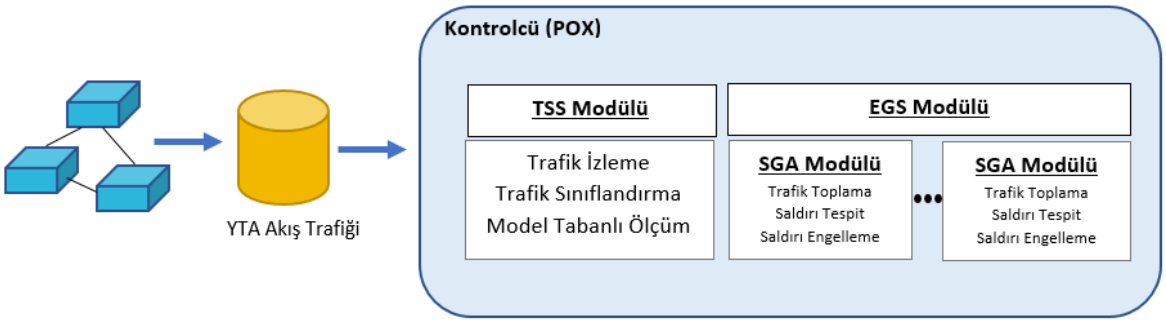
5. YTA'DA AĞ TRAFİĞİNE DUYARLI OTONOM SALDIRI TESPİTİ VE ÖNLEME

Bu çalışmada, YTA mimarisi kullanan ağların güvenliğini sağlamak için ağ trafiğine duyarlı otonom özelliğe sahip bir siber savunma modeli önerilmiştir. Şekil 5.1' de önerilen modelin mimari gösterimi verilmiştir. Önerilen model, YTA kontrolcüsü ile bütünleşik olarak çalışan farklı yeteneklere sahip modüllerden oluşmaktadır.



Şekil 5.1. Tümleşik YTA kontrolcü ve siber güvenlik modeli

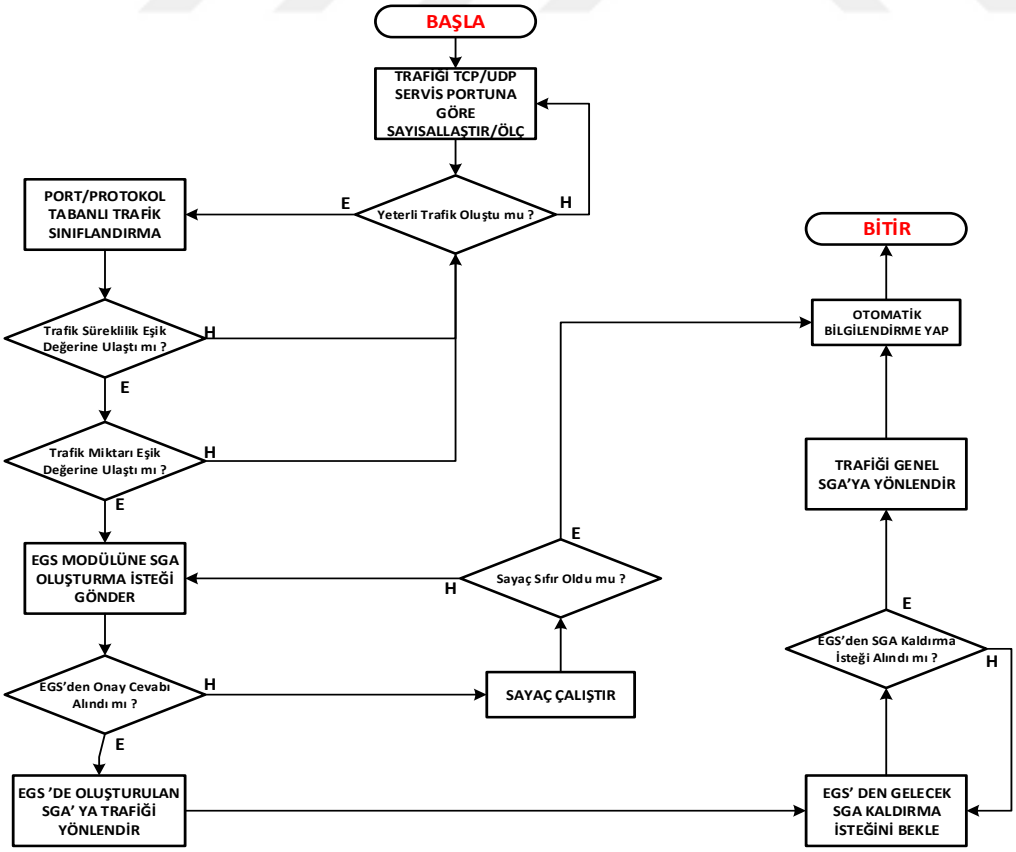
Şekil 5.2' de önerilen siber savunma modelinin gösterimi modüler yapıda verilmiştir. Buna göre; TSS modülü ağdaki akan trafiği kural tabanlı sınıflandırma yöntemini kullanarak ağ trafiğini sınıflandırmaktan ve akış ölçümü yapmaktan sorumludur. EGS modülü ise sınıflandırılmış ağ trafiğinin derinlemesine incelenmesinden ve trafik türüne göre otonom bir şekilde oluşturulacak siber güvenlik alanlarının açılıp kapanmasından sorumludur.



Şekil 5.2. Önerilen modelin modüler gösterimi

5.1. Trafik Sınıflandırma Servisi Modülü

TSS modülü, YTA kontrolcüsü ile eş zamanlı olarak çalışmaktadır. Kontrolcüden gelen ağ trafiği TSS modülüne aktarılır. Şekil 5.3' de TSS modülünün çalışması, akış diyagramı ile gösterilmiştir. TSS modülünde iki temel işlem gerçekleştirilir. Gelen ağ trafiği sınıflandırılır ve ağ trafik sürekliliği ölçülür. Böylece her farklı trafik türü için özgün bir güvenlik yaklaşımı uygulanabilecektir.



Şekil 5.3. TSS modülü akış diyagramı

5.1.1. Trafik sınıflandırma evresi

TSS modülünde ilk olarak kural tabanlı sınıflandırma yöntemini kullanarak merkezi kontrolcüden gelen ağ trafiği izlenip port numarası ve protokol bilgisine göre trafik sınıflandırılmaktadır. Sınıflandırılan trafik süreklilik ölçümü için bir matematiksel modele yönlendirilir. Böylece her farklı trafik türü için benzersiz bir güvenlik yaklaşımı uygulanabilecektir. Matematiksel modelin çalışma adımları şu şekilde özetlenmiştir;

Adım 1: Süreklilik hassasiyetinin hesaplanması

Süreklilik hassasiyeti (S) Eş. 5.1' de gösterildiği gibi toplam trafik süresinin (T_t) ayrık zaman süresine bölünmesi (T_D) ile hesaplanır. S, ağ trafiğinin toplam kaç parçada değerlendireceğinigöstermektedir.

$$S = \frac{T_t}{T_D} \tag{5.1}$$

Adım 2: Elde edilen puanın hesaplanması

Ağ trafiği S değeri kadar bölüme ayrılarak değerlendirilmektedir. İlk bölümden başlayarak tüm bölümler, Eş. 5.2' de gösterilen parçalı fonksiyon içinde değerlendirir. Eş. 5.3 ile $f(\beta)$ fonksiyonundan gelen puanlar, toplam fonksiyonu içinde hesaplanarak bulunur (P_c). Burada elde edilecek en yüksek değer P_{max} , Eş. 5.4'de gösterildiği gibi, tüm parçalarda aynı trafik türünün olması ile mümkün olabilmektedir.

β değişkenine bağlı bir $f(\beta)$ parçalı fonksiyonu olsun;

$$f(\beta) = \begin{cases} \beta \; ve \; \beta + 1 \; arasında \; veri \; varsa \; Puan = S - 1; \text{BİTİR} \\ \beta \; ve \; \beta + 2 \; arasında \; veri \; varsa \; Puan = S - 2; \text{BİTİR} \\ \beta \; ve \; \beta + 3 \; arasında \; veri \; varsa \; Puan = S - 3; \; \text{BİTİR} \\ \vdots \\ \beta \; ve \; S \; arasında \; veri \; varsa \; Puan = 1; \text{BİTİR} \end{cases} \tag{5.2}$$

$$P_c = \sum_{\beta=1}^{S} f(\beta) \tag{5.3}$$

$$P_{max} = (S-1)^2 \tag{5.4}$$

Adım 3: Siber güvenlik alanlarının oluşturulması

Eş. 5.3'de gösterildiği gibi elde edilen toplam puan (P_c) ,"ω" eşik değerinden yüksek ise ağ trafiği incelemeye değer bulunur ve TSS modülü EGS modülünden ilgili trafik için SGA oluşturulmasını talep eder (Eş. 5.5). Eğer YTA birden fazla kontrolcüye sahip ise ω değeri kontrolcü sayısına bölünerek hesaplanır.

$$P_c \geq \omega \tag{5.5}$$

TSS modülü içerisinde ilk evrede sınıflandırılan her farklı trafik, önerilen bir matematiksel model kullanılarak tekrar değerlendirilir. Çizelge 5.1' de önerilen matematiksel modelin hesaplanmasında kullanılan gösterimler ve Çizelge 5.2' de de matematiksel modelin sözde kodu verilmiştir.

Çizelge 5.1. TSS modülü içinde kullanılan gösterimler

Gösterimler	Açıklamalar
TWS	Traffic Window Size (Trafik Pencere Boyutu)
TFS	Traffic Fragment Size (Trafik Parça Boyutu)
SC	Sensibility of Continuity (Süreklilik Duyarlılığı)
K	Threshold Constant (Eşik Sabiti)
EGS	Esnek Güvenlik Servisi
SGA	Siber Güvenlik Alanı
TSS	Trafik Sınıflandırma Servisi
MaxScore	Maksimum Puan
Score	Puan
Window	Pencere
Window_Next	Sonraki Pencere
ObtainScore	Elde Edilen Puan
Counter	Sayıcı
Threshold Score	Eşik Değer Puanı
Application Layer Traffic	Uygulama Düzlemi Trafiği

Bu model ile kontrolcü üzerine gelen trafiğin sürekliliği ve yoğunluğu ölçülmektedir. Geliştirilen matematiksel model ile hesaplanan trafik eşik değeri, *w* eşik değerinden büyük ise gelen trafik şüpheli bulunacak ve EGS Modülü aktif edilecektir. Aksi takdirde ilgili trafik şüpheli bulunmayacak ve ilgili sunucuya yönlendirilecektir. Buradaki *w* eşik değeri trafik türüne göre önceden yapılan ölçümler ile belirlenmiştir. Her trafik türüne göre ayarlanabilir

bir sabit değere sabittir. Ağ topolojisinde birden fazla kontrolcü kullanılması durumunda kontrolcü sayısına bölünerek hesaplanabilir.

Çizelge 5.2. TSS servisi sözde kodu

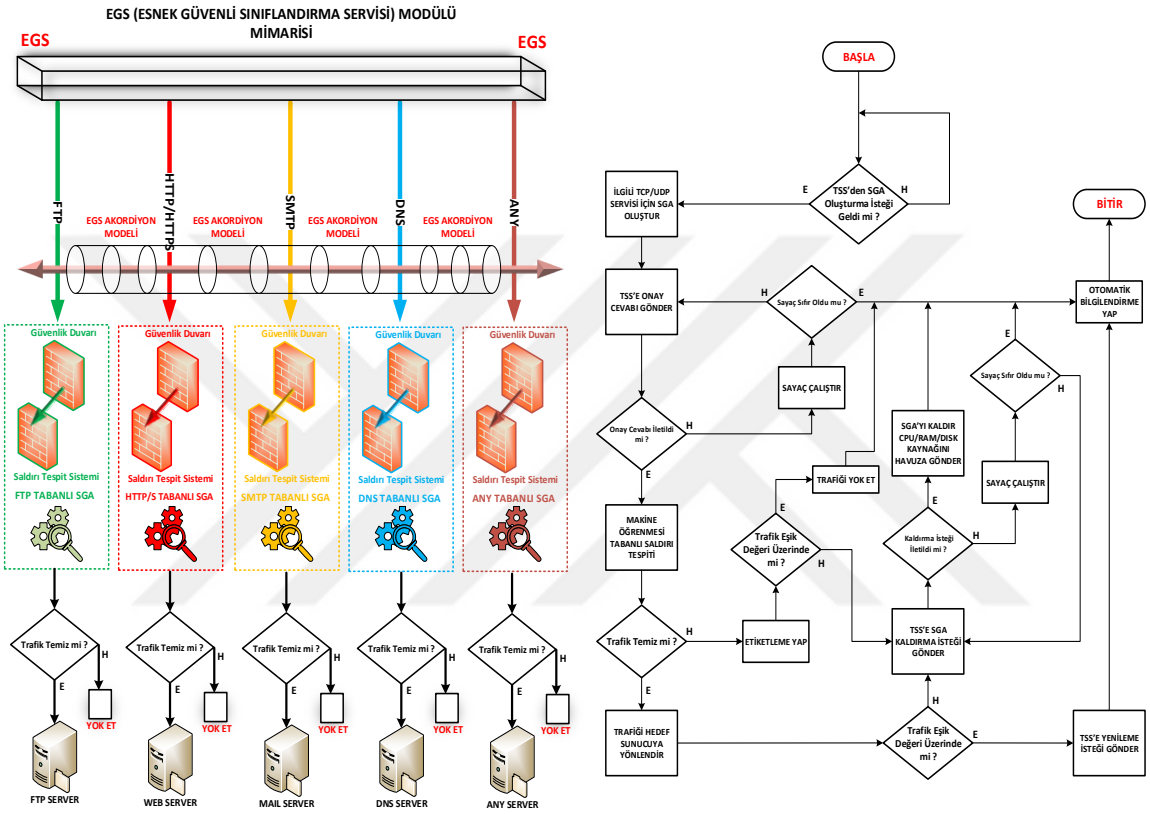
Girdi:	Application Layer Traffic, TWT, TFS, K
Çıktı:	EGS SGA Start & Stop Mesajları
Parametreler:	TWT, TFS, K, MaxScore, Score, Window,Window_Next, SC, ObtainScore, Counter, Threshold Score

```
1.  Set TWS, TFS, K
2.  SC = TWS / TFS
3.  Max Score = (SC-1)^2
4.  Obtain Score = 0
5.  Score = SC
6.  Window = 1
7.  Window_Next = Window
8.  Counter = 1
9.  Threshold Score: Max Score x K
10. Loop (End of the SC)
11.         If (Application Layer Traffic found in Window) {
12.                 Window_Next = Window + 1
13.                 Loop (Wait for Application Layer Traffic such as HTTP, DNS, ICMP)
14.                         If (Application Layer Traffic found in Window_Next)
15.                            Obtain Score += Score – Counter
16.                         Else
17.                                 Counter = Counter + 1
18.                                 Window_Next = Window_Next + 1
19.         Window = Window_Next
20.         Counter = 1
21.         Else
22.                 Window = Window + 1
23.                 SC = SC -1
24. If (Obtain Score more than Threshold Score)
25.         If (EGS SGA is Opened)
26.                 Continue
27.           Else
28.                 Start EGS SGA Virtual machine for http, dns, icmp
29. Else
30.           If (EGS SGA is Closed)
31.                 Continue
32.         Else
33.           Stop EGS SGA Virtual machine for http, dns, icmp
34.                 Forward App Layer traffic to EGS Any
```

5.2. Esnek Güvenlik Servisi Modülü

Bu modül, TSS modülü ile eş zamanlı olarak çalışmaktadır. TSS modülünden gelen şüpheli trafik uyarısı ile ilgili trafik türüne özel bir siber güvenlik alanı oluşturulur. EGS modülü, her benzersiz trafik türü için farklı siber güvenlik bölgelerine sahiptir. Örneğin, TSS modülünden şüpheli web trafiği uyarısı alındığında, yalnızca Web trafiği için siber güvenlik alanı (SGA) oluşturulmaktadır. Şüpheli trafik, ilgili SGA içerisinde incelenmektedir.

TSS modülünden gelen tetikleyici sinyal ile EGS üzerindeki ilişkili SGA kapatılmaktadır. Böylece kullanılan kaynaklar TSS havuzuna aktarılmaktadır. TSS modülü içerisindeki SGA'lar, TSS modülünden gelen tetikleyici sinyaller ile otomatik olarak açılıp kapanmaktadır. Bu özelliği ile EGS modülü akordiyon bir yapıya sahiptir. Şekil 5.4' de EGS modülünün yapısı ve çalışma adımlarını gösteren akış diyagramı verilmiştir.



Şekil 5.4. EGS modülünün yapısı ve akış diyagramı

5.2.1. Trafiğe duyarlı siber güvenlik alanı

TSS tarafından sınıflandırılan ağ trafiği kullanılan matematiksel model ile değerlendirilerek ve şüpheli bulunması durumunda EGS modülü içerisinde bulunan SGA içerisinde incelenmektedir. EGS modülü içerisinde farklı ağ trafikleri için oluşturulmuş SGA' lar yer almaktadır.

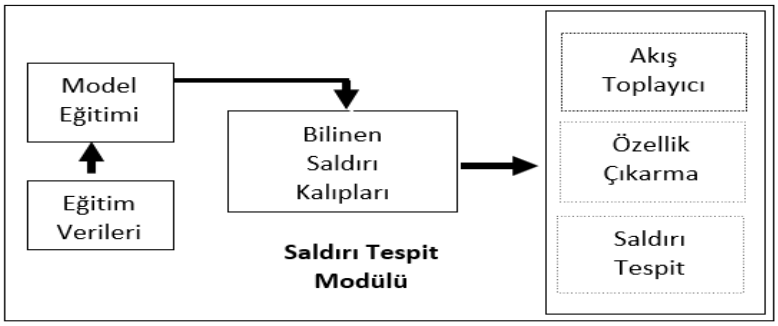
Gelen trafik türüne göre aktif olacak SGA' lar, şüpheli ağ trafiğini değerlendirerek; engellemekte ya da izin vermektedir. SGA' lara gelen trafiğin saldırı trafiği olup olmadığı makine öğrenme yöntemleri ile tespit edilmiştir. EGS modülü içerisinde; web trafiği için

Web_SGA, DNS trafiği için DNS_SGA ve ICMP trafiği için ICMP_SGA servisleri yer almaktadır. Bu SGA' lar TSS' den gelen tetikleyici sinyal ile aktif olmaktadır. SGA'lar görevleri bittiğinde otomatik olarak kapanmaktadır. Görevi biten SGA kullandığı sistem kaynaklarını (RAM, CPU, disk gibi kaynaklar) EGS kaynak havuzuna aktarmaktadır. Böylece sistem kaynakları etkin bir şekilde kullanılmaktadır.

TSS tarafından şüpheli bir trafik uyarısı gelmediği durumda ayrıştırılmış trafik ilgili sunucuya aktarılmaktadır. Böylece diğer sunucularla olan iletişim kesinti olmaksızın saldırı durumunda bile devam edebilmektedir. EGS modülü içerisinde bulunan siber güvenlik alanları TSS modülünden gelen şüpheli trafik akış tetikleyici sinyali ile oluşturulur. EGS modülünde sınıflandırılan trafiğe göre ilgili SGA otonom olarak çalışır. SGA içerisinde üç alt modül yer almaktadır. Bu modüller; akış özellikleri toplama modülü, saldırı tespit modülü ve saldırı engelleme modülüdür.

Akış öznitelikleri toplama modülü: Bu modül, protokol, kaynak IP, hedef IP, kaynak ve hedef port numarası gibi tüm akış istatistiklerini toplamak için OpenFlow mesajındaki bir paket veya bir zamanlayıcı işlevi tarafından tetiklenir.

Saldırı tespit modülü: Toplanan akış özellikleri bu modül üzerinden alınır ve bir akışın anormal olup olmadığına karar verilir. Bu modül, optimize edilmiş bir makine öğrenme modeli içerir.



Şekil 5.5. Saldırı tespit modülü çalışma adımları

Şekil 5.5' de saldırı tespit modülünde gerçekleşen iş adımlarını gösterilmektedir. Saldırı tespit modülünden gelen tetikleyici sinyal ile saldırı engelleme modülü devreye alınmaktadır.

Saldırı engelleme modülü: Saldırı tespit modülündeki sonuçlara göre saldırı engelleme modülü akışla ilgili kararlar (akışı düşürme veya iletme) almaktadır.

Bu çalışmada geliştirilen model ile kontrolcü önceden belirlenmiş zaman aralıklarında veri düzleminde bulunan OpenFlow anahtarları izler. Her seferinde, bir *flow_stats* mesajı aracılığıyla anahtarların akışları hakkında istatistik raporu talep eden merkezi kontrolcü üzerinde bir iş parçacığı (thread) çalışır. Talep alındığında, akış tablosundaki her akış girişi hakkında istatistik raporunu içeren bir yanıt mesajı iletilir. İletinin bu eşleşme alanı, hedef IP, kaynak IP, TCP port numarası gibi bir akışı eşleştirmek için kullanılır. Sırasıyla, kontrolcü akış istatistiklerini alır ve ardından istatistikler, trafik izleme modülü aracılığı ile izlenir. Akış üzerindeki paket ve protokol bilgisi verileri kullanılarak sınıflandırılan trafik üzerinde, geliştirilen matematiksel model kullanılarak süreklilik ölçümü yapılır. Burada yapılan ölçüm sonuçlarına göre trafik şüpheli bulunursa EGS modülü bir mesaj ile tetiklenir. Şüpheli bir trafik bulunmaz ise kontrolcü ilgili trafiği hedefe yönlendirir.

İlk aşamadan sonra, ikinci aşamada şüpheli trafik mesajı ile tetiklenen EGS modülü üzerinde bulunan siber güvenlik alanları gelen trafik türüne göre aktif olur. EGS üzerinde her farklı trafik türüne özgü SGA' lar yer almaktadır. Bu SGA' lar üzerinde; akış özelliklerini toplama, saldırı tespit ve saldırı engelleme modülleri yer almaktadır. Saldırı tespit süreçlerinde Şekil 5.6' da gösterilen Algoritma I, saldırı engelleme sürecinde Şekil 5.7' de gösterilen Algoritma II kullanılmaktadır.

Algoritma I: Saldırı Sınıflandırma (Attack_Classification)

Girdi: *Eğitilmiş Makine Öğrenmesi sınıflandırıcısı, Atak Uyarısı, Akış istatistikleri*
Çıktı: *Sınıflandırılmış Trafik*

```
1: if Atak Uyarısı then
2:     Akış özniteliklerini oluştur
3:     for Akış öznitelikleri do
4:         Öznitelik setini test et
5:         Trafiği sınıflandır
6:         Atak Engelleme fonksiyonunu çağır
7:     end for
8: end if
9: return Sınıflandırılmış trafik
```

Şekil 5.6. Saldırı tespit modülü algoritması

Algoritma II: Saldırı Engelleme (Attack_Mitigation)

Girdi: *Anahtar* s_i, *Hedef* $dest_{ip}$, *protokol*

Çıktı: *Akışları düşürmek için kural ekle*

1: **for all** *Switch* $(s_i) \in$ *Kontrolcü* (C) **do**

2: Akış kurallarını oluştur, FR($dest_{ip}$, *protokol, drop*)

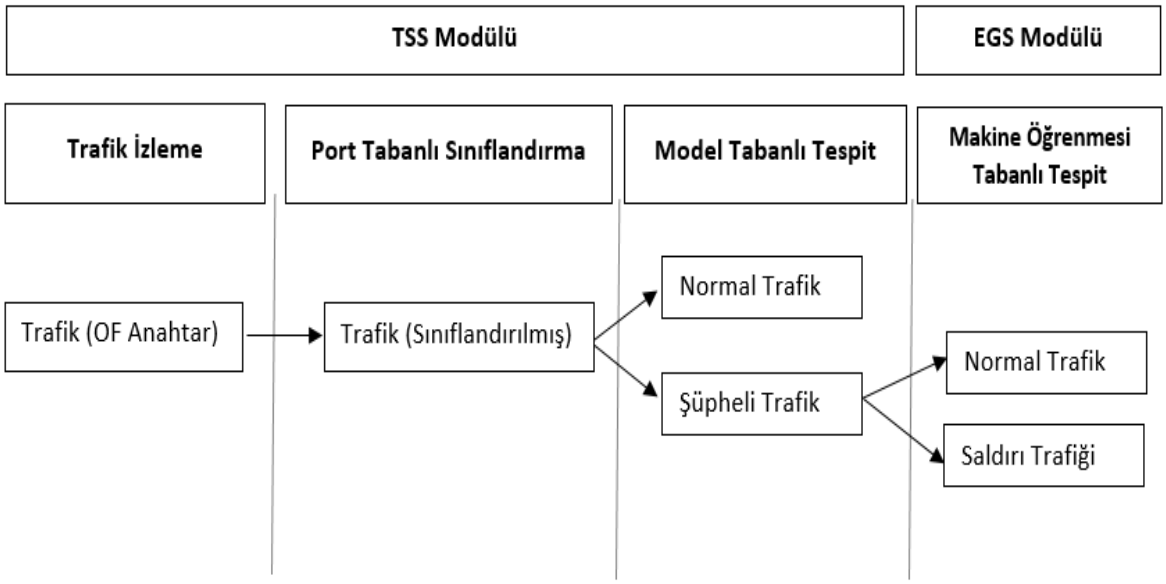
3: Akış kurallarını güncelle, $(s_i) \leftarrow C$

4: **end for**

Çıktı: *Akışları düşürmek için akış tablosuna kural ekle*

Şekil 5.7. Saldırı engelleme modülü algoritması

Algoritma I' de gösterildiği gibi toplayıcı üzerinde toplanan akış özellikleri eğitilmiş makine öğrenme modeli kullanılarak değerlendirilir. Burada tespit edilen saldırı trafiği ile Saldırı Engelleme süreci başlamış olur. Algoritma II' de gösterildiği gibi saldırı trafiğini oluşturan makine için kontrolcü üzerinde kural tanımlanır ve saldırgandan gelen istek paketleri engellenir. Önerilen yöntemin çalışma düzeni Şekil 5.8' da verilmiştir.



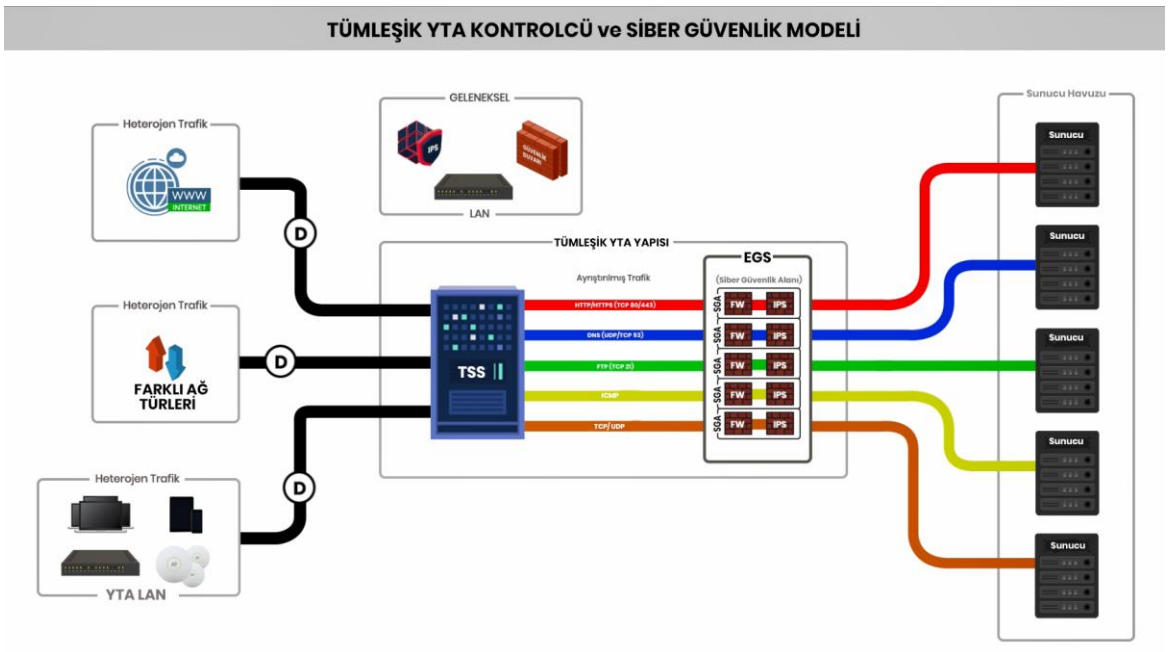
Şekil 5.8. Önerilen yöntemin çalışma düzeni

6. ÖNERİLEN GÜVENLİK MODELİNİN BENZETİMİ VE TESTİ

Çalışma kapsamında önerilen siber güvenlik modelinin benzetimi ve testi için gerçek ortama benzer özellikler gösteren bir topoloji oluşturularak tüm makineler sanal ortam üzerine kurulmuştur. Topoloji üzerindeki makineler kullanılarak Web, DNS ve ICMP ağ trafik türleri için veri kümeleri ve makine öğrenme modelleri oluşturulmuştur. Kurgulanan farklı saldırı senaryoları için önerilen yöntem devreye alınmış ve elde edilen bulgular bu bölümde tartışılmıştır.

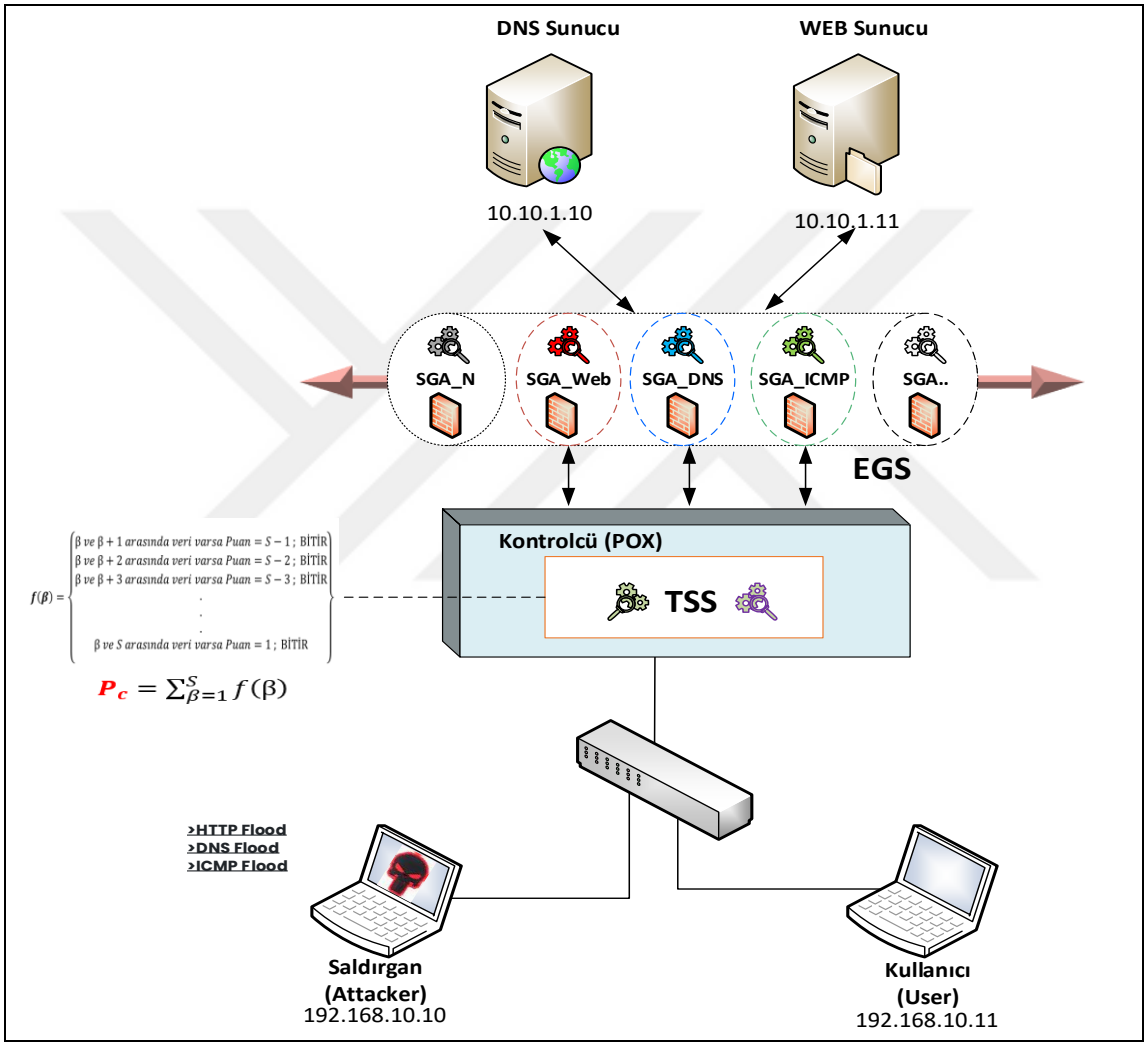
6.1. Mimari ve Benzetim Ortamı

Bu çalışmada, YTA mimarisi üzerinde çalışan ağ trafik sınıflandırmasına ve makine öğrenmesine dayalı otonom bir siber güvenlik mimari modeli sunulmuştur. Bu model ile ağdaki akan trafik üzerinde TCP/UDP gibi uygulama servisi portlarına göre trafik sınıflandırması yapabilen, sınıflandırılmış her bir trafiği birbirinden bağımsız olarak işleyebilecek sanal güvenlik duvarlarını ve saldırı tespit sistemlerini otonom olarak oluşturup kaldırabilen, YTA kontrolcüsü ile bütünleşik olarak çalışıp ağdaki trafik hakkında karar verebilecek bir yöntem geliştirilmesine odaklanılmıştır. Bu kapsamda oluşturulan mantıksal mimari Şekil 6.1' de, fiziksel mimari Şekil 6.2' de verilmiştir.



Şekil 6.1. Mantıksal mimari

Şekil 6.2' de fiziksel topoloji ayrıntılı olarak gösterilmiştir. Ağ istemcileri OpenFlow anahtarı yardımıyla kontrolcüye bağlanırken ağa hizmeti verecek DNS ve HTTP sunucuları da kontrolcüye bağlanmıştır. Kontrolcü ile eş zamanlı bir şekilde çalışacak olan trafik sınıflandırmasından sorumlu TSS ve içerisinde siber güvenlik alanları barındıran EGS modülü kontrolcü üzerine konumlandırılmış ve sanal makine olarak mimariye dahil edilmiştir.



Şekil 6.2. Tümleşik YTA kontrolcü ve siber güvenlik modeli fiziksel mimarisi

Geliştirilen uygulamanın fiziksel topolojisi, VMware sanal ortamında kurulmuştur. VMware herhangi bir makineye birden fazla işletim sistemi kurulmasına yardımcı olan bir sanallaştırma yazılımıdır. Topolojide yer alan bütün makineler bu sanal ortam üzerine konumlandırılmıştır. TSS ve EGS modülleri kontrolcü üzerinde modüler olarak çalışmaktadır. Saldırgan makine üzerinde; saldırıların gerçekleştirileceği bütün araçların yer aldığı Kali Linux işletim sistemi kullanılmıştır. Sunucular ve siber güvenlik alanları için

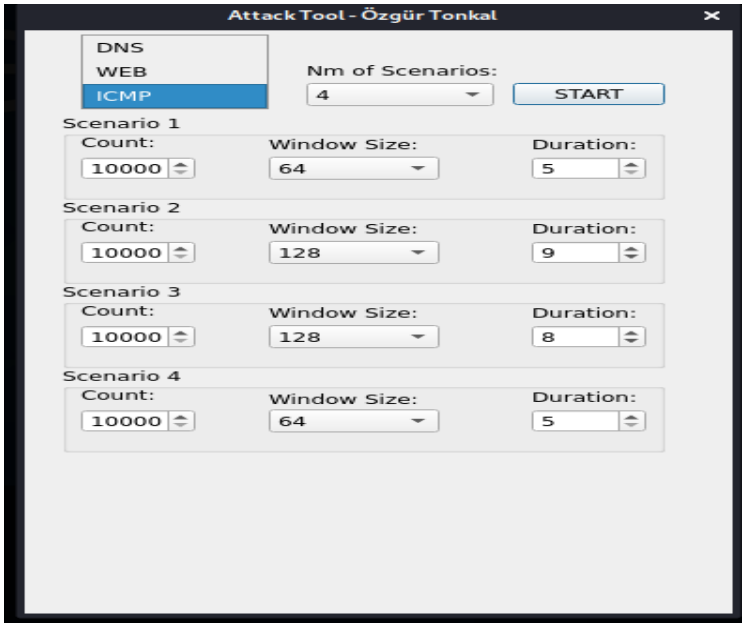
OpenBSD işletim sistemi kullanılmıştır. İstemci sanal makinesi üzerinde ise Windows 10 işletim sistemi kurulmuştur. Kontrolcü olarak POX kontrolcü seçilmiştir. Çizelge 6.1' de sanal ortam üzerine kurulan makinelerin ayrıntılı sistem özellikleri gösterilmiştir.

Çizelge 6.1. Sanal ortam üzerine kurulan makine özellikleri

Sanal Makine	İşletim sistemi	İşlemci	Hard Disk	Bellek
İstemci (User)	Windows 10	1 vCPU	30 GB	512 MB
Saldırgan (Attacker)	Kali Linux	2 vCPU	40 GB	2 GB
Sunucu (Web)	OpenBSD	1 vCPU	30 GB	512 MB
Sunucu (DNS)	OpenBSD	1 vCPU	30 GB	512 MB
Kontrolcü (Controller)	Ubuntu-POX	8 vCPU	30 GB	2 GB
Siber Savunma Alanları (SGA)	OpenBSD	3 vCPU	90 GB	3 GB

6.2. Veri Kümesinin ve Makine Öğrenme Modelinin Oluşturulması

Önerilen siber güvenlik modelinde makine öğrenmesi aracılığı ile saldırı tespitinin gerçekleştirilmesi için bir veri kümesine ihtiyaç duyulur. Bu yüzden saldırı ve normal ağ trafiklerini içeren bir veri kümesi oluşturulan topoloji üzerinden elde edilmiştir. Bu çalışmada uygulama senaryosuna uygun bir anonim veri kümesi olmadığı için bu çalışmaya özel olarak geliştirilen ağ trafik üretici yazılımı kullanılarak bir veri kümesi oluşturulmuştur. Şekil 6.3' de geliştirilen ağ trafik üretici yazılımının ekran görüntüsü verilmiştir.



Şekil 6.3. Trafik üretici yazılım arayüzü

Şekil 6.2’ de gösterilen uygulama topolojisi üzerine konumlandırılan saldırgan (attacker) ve kullanıcı (user) makineleri üzerinden “Pycharm” geliştirme ortamı kullanılarak geliştirilmiş, ağ trafik üretici yazılımı ile normal ve saldırı trafikleri oluşturulmuştur. Trafik üretici yazılımı ile uygulama topolojine yerleştirilmiş olan sunuculara yönelik farklı hacimsel ve süreklilikte ağ trafikleri oluşturulmuştur. Hedef sunuculara yönelik saldırı, saldırgan istemci makinesi üzerinden yoğun trafik ve sahte IP istekleri oluşturarak gerçekleştirilmiştir. Veri kümesi, normal trafik verileri dahil olmak üzere HTTP flood, DNS flood ve ICMP flood saldırı trafiklerinden oluşmaktadır.

Şekil 6.4’ de, veri kümesi oluşturma aşamasında kullanılan Algoritma III verilmiştir. Algoritma III kullanılarak akış istatistikleri topolojide yer alan OpenFlow anahtarı üzerinden belirli aralıklarla toplanmaktadır. Anahtar üzerinde her olay için, bir olay isteği (event request) ve yanıt işleyicisi (reply handler) çalışmaktadır. İzleme süresince anahtardan her olay için akış istatistikleri (*OFPFlowStatsRequest)* çağrılarak istenir. Yanıt olarak, akış istatistiklerini döndüren olay yanıt işleyicisi (*OFPFlowStatsReply)* çağrılır. Benzer şekilde, port durum isteği (*OFPPortStatsRequest*) çağrılarak port istatistikleri de alınır. Bu istatistikler belirlenen izleme aralığında CSV formatında bir dosyaya yazılır ve bu sayede veri kümesi oluşturulur.

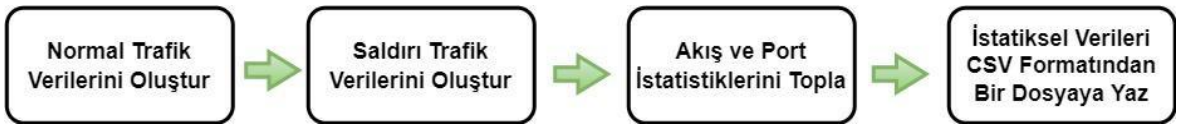
Algoritma III: YTA Trafik Veri Kümesi oluşturma algoritması

Girdi: *Anahtardaki trafik istatistikleri. 30 s aralıklarla anahtardan alınmış*
Çıktı: *YTA trafiği veri kümesi*
1: Her flow (akış) için akış ve bağlantı noktası istatistiklerini topla.
2: **for** $i = flow1\ to\ flow_n$ ***do***
3: Veri kümesini oluşturmak için anahtardan tüm akış ve bağlantı noktası istatistiklerini bir dosyada topla
4: Saldırı Trafiği Veri Kümesini oluştur ve 1 olarak etiketle.
5: Normal Trafik Veri Kümesini oluştur ve 0 olarak etiketle.
6: **end for**
7: **return** *Trafik Veri Kümesi*

Şekil 6.4. YTA veri kümesi oluşturma algoritması

Şekil 6.5’ de veri kümesi oluşturma adımları gösterilmektedir. Veri kümesi oluşturulduktan sonra toplanan ağ trafiği, port ve protokol bilgileri kullanılarak DNS, HTTP ve ICMP veri kümeleri olarak ayrıştırılır.

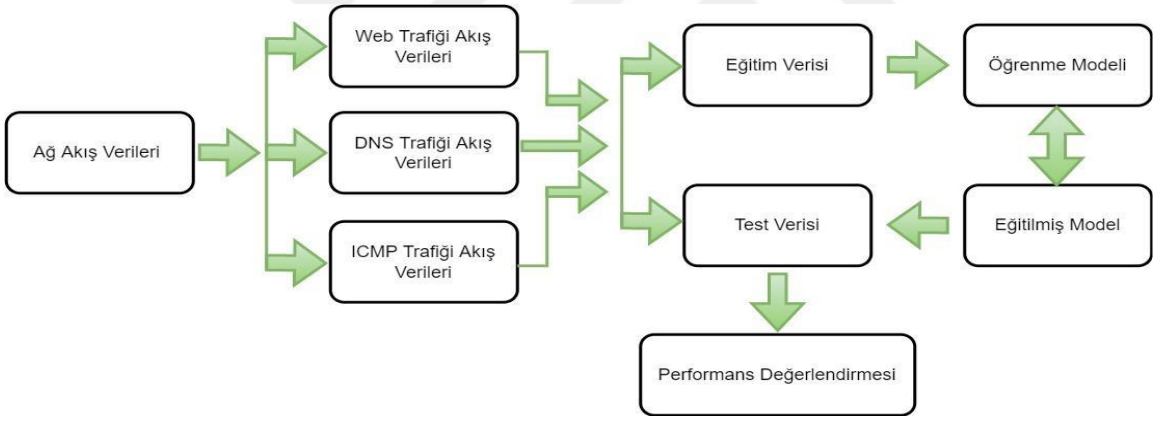
Üç ayrı veri kümesi oluşturmadaki amaç farklı saldırı türlerinin makine öğrenme modelinin genelleştirme yapabilmesi açısından gereklidir. Her farklı veri kümesi için makine öğrenmesi algoritmaları kullanılarak eğitim ve test aşamaları gerçekleştirilmiştir.



Şekil 6.5. Veri kümesi oluşturma adımları

Şekil 6.6, veri kümesi üzerinde makine öğrenimi modellerini eğitme ve test etme sürecini göstermektedir. Veri kümesi eğitim ve test olarak bölünmeden önce, veri kümesi üzerinde bazı ön işlemler uygulanmıştır.

Uygulanan bu ön işlemler; eksik değerlerin ele alınması, boş değerlerin kaldırılması, yenilenen kayıtların silinmesi ve kategorik değerlerin kodlanması gibi işlemlerdir.



Şekil 6.6. Makine öğrenme yöntemi ile sınıflandırma

Ön işleme sonucunda oluşturulan etiketli veri kümesi; 7130 ağ trafik verisi ve 22 farklı öznitelik içermektedir. Çizelge 6.2, her bir saldırı türündeki ağ trafik veri sayısını ve Çizelge 6.3, dikkate alınan 19 farklı ağ trafik özniteliklerini göstermektedir.

Çizelge 6.2. Trafik paketlerinin dağılımı

	Saldırı Ağ Trafik Verileri	Normal Ağ Trafik Verileri	Toplam Ağ Trafik Verileri
Web Trafik	1378	1592	2970
DNS Trafik	1140	842	1982
ICMP Trafik	1087	1091	2178

Çizelge 6.3. Sınıflandırmada kullanılan öznitelikler

No	Öznitelikler (Features)	Açıklamalar
1	Packet Count	Tek bir akış sırasındaki paket sayısı
2	Byte count	Tek bir akış sırasındaki bayt sayısı
3	Total number of flows in a switch	Anahtardaki toplam akış sayısı
4	Packet Count per-flow	Akış başına paket sayısı
5	Byte Count per-flow	Akış başına gönderilen bayt sayısı
6	Duration_n_secs	Akışın anahtarda kaldığı nanosaniye cinsinden süre
7	Duration_ secs	Akışın anahtarda kaldığı süre
8	Total_duration	Akışın anahtarın akış tablosunda kaldığı toplam süre (Duration_n_secs, Duration_ secs toplamı)
9	Protocol	Trafiğin kullandığı protokol
10	Source IP	Kaynak IP
11	Destination IP	Hedef IP
12	Number of Packet_in messages	Packet_in mesajlarının sayısı
13	Packet Rate	Saniyede gönderilen paket sayısı
14	Port number	Port numarası
15	txbytes	Veri aktarımı hızı
16	rxbytes	Veri alma hızı
17	dt	Sayıya dönüştürülmüş tarih ve saati bilgisi
18	Port bandwidth	Bağlantı noktası bant genişliği (tx_kbps ve rx_kbps'nin toplamı)
19	Label	Trafik türünü gösteren sınıf etiketi (Normal:0, saldırı:1)

Veri kümesi, kaynak ve hedef makineleri tanımlayan öznitelikler dışında, bayt_sayısı, süre_sn, paket hızı ve akış başına paket sayısı gibi istatistiksel özniteliklere de sahiptir. Makine öğrenmesinin eğitimine başlamadan önce veriler ön işleme tabi tutulmuştur. Bu aşamada, OpenFlow anahtar bilgisi, akış başına bayt ve akış başına paket özellikleri, yinelenen değerler içerdiğinden veri kümesinden çıkarılmıştır. Kategorik değişkenler kaynak-hedef IP ve sayısal değerleri olmayan protokol gibi öznitelikler tek-sıcak kodlama (one-hot encoding) kullanılarak kodlanmıştır [30]. “dt” özniteliği ile gösterilen ve zaman bilgisini içeren öznitelik gereksiz bulunmuş ve veri kümesinden çıkarılmıştır. Bütün sayısal verilere normalizasyon uygulanarak veri ön işleme aşaması tamamlanmıştır.

Daha sonra oluşturulan 3 ayrı veri kümesinin herbirinin %70 oranındaki kısmı eğitim veri kümesi olarak bölünürken, %30 oranındaki kısmı test veri kümesi olarak ayrılmıştır. Veri kümesi oluşturma, veri ön işleme ve eğitim-test verilerinin ayrılması aşamalarından sonra normal ve saldırı trafiklerini sınıflandırmada kullanılacak makine öğrenme yöntemleri belirlenmiştir. Makine öğrenmesinin temel fikirlerinden biri mevcut eğitim veri kümesini kullanarak bir model geliştirmektir. Makine öğrenmesinde kullanılabilecek pek çok model mevcuttur. Herhangi bir makine öğrenme modeli bir veri kümesinde çok iyi sonuçlar üretebilirken başka bir veri kümesi üzerinde çok kötü sonuçlar üretebilir. Aynı veri kümesi

üzerinde çalışsalar bile farklı modellerin de sonuçları birbirlerinden farklı çıkabilir. Bu yüzden, belirli bir veri kümesi üzerinde en iyi performansı veren modeli bulabilmek önemli bir konudur. Bu yüzden farklı makine öğrenme modelleri aynı veri kümesine uygulanarak performansları karşılaştırmalı ve en iyi performansı gösteren model uygulamaya alınmalıdır.

Web, DNS ve ICMP eğitim veri kümeleri üzerinde; k-En Yakın Komşu, Karar Ağacı, Destek Vektör Makinesi ve Yapay Siniri Ağı makine öğrenme algoritmaları kullanılarak eğitim gerçekleştirilmiştir. Ardından eğitilmiş makine öğrenme modelleri test veri kümeleri kullanılarak performans testine tabi tutulmuştur. Performans testlerinden sonra Web, DNS ve ICMP ağ trafiklerinden oluşan veri kümeleri üzerinde en yüksek doğruluk oranını veren makine öğrenme modeli belirlenmiştir.

Makine öğrenme modellerinin sınıflandırma test sonuçları

Bir makine öğrenme modelinin performansını belirleyebilmek için modelin eğitim veri kümesine dayanarak elde edilmesi ve bu modeli kullanarak test edilmesi gerekir. Bu yüzden makine öğrenme modellerinin performanslarını karşılaştırmak için model performans değerlendirme yöntemlerinin ve performans ölçütlerinin kullanılması gerekir. Bu çalışmada, farklı makine öğrenme modellerinin sınıflandırma performansını değerlendirmek için, karmaşıklık matrisi kullanılmıştır. Karmaşıklık matrisi önceki bölümlerde verilmişti (Bkz. Çizelge 3.1). Karmaşıklık matrisinde yer alan kavramların anlamları şu şekildedir;

•DP (doğru pozitif) / TP (true positive): Gerçek durumu pozitifken sınıflandırma sonucu da pozitif çıkan örneklerin sayısı. (Model doğru sınıflandırma yapmıştır.)
•YN (yanlış negatif) / FN (false negative): Gerçek durumu pozitifken sınıflandırma sonucu negatif çıkan örneklerin sayısı. (Model yanlış sınıflandırma yapmıştır.)
•DN (doğru negatif) / TN (true negative): Gerçek durumu negatifken sınıflandırma sonucu da negatif çıkan örneklerin sayısı. (Model Doğru sınıflandırma yapmıştır)
•YP (yanlış pozitif) / FP (false positive): Gerçek durumu negatifken sınıflandırma sonucu pozitif çıkan örneklerin sayısı. (Model Yanlış sınıflandırma yapmıştır)

Bir makine öğrenme modelinin performansını belirlemek üzere bazı ölçütlerden yararlanılır. Bu ölçütler test veri kümesi kullanılarak hesaplanır. Ölçütlerden en çok tercih edilenleri şunlardır:

•Doğruluk (Accuracy)

•Duyarlılık (Sensitivity-Recall)

•Özgünlük-Belirleyicilik (Specificity)

•Kesinlik (Precision)

•F1-Puan (F1-Score)

Bu ölçütlerin eşitlikleri önceki bölümlerde verilmişti (Bkz. Çizelge 3.2). Ayrıca, model performansını değerlendirmek için ROC ve AUC kullanılmıştır. ROC eğrisinin yatay eksende yanlış pozitif oranı ve dikey eksende doğru pozitif oranı vardır. Bir makine öğrenme modelinin genel performansını karakterize etmek için eğri altında kalan alan AUC ölçeği kullanılır.

•AUC' un tanım aralığı 0,5 <= AUC <= 1 olup, 0,5 ve 1 sırasıyla alt ve üst sınırları oluşturmaktadır. AUC, DP ve DN yakalama oranını gösterir. AUC yükseldikçe, model sınıflandırmasının daha iyileştiği söylenebilir. Eğrinin altında kalan alanın 1' e yaklaşması istenir.

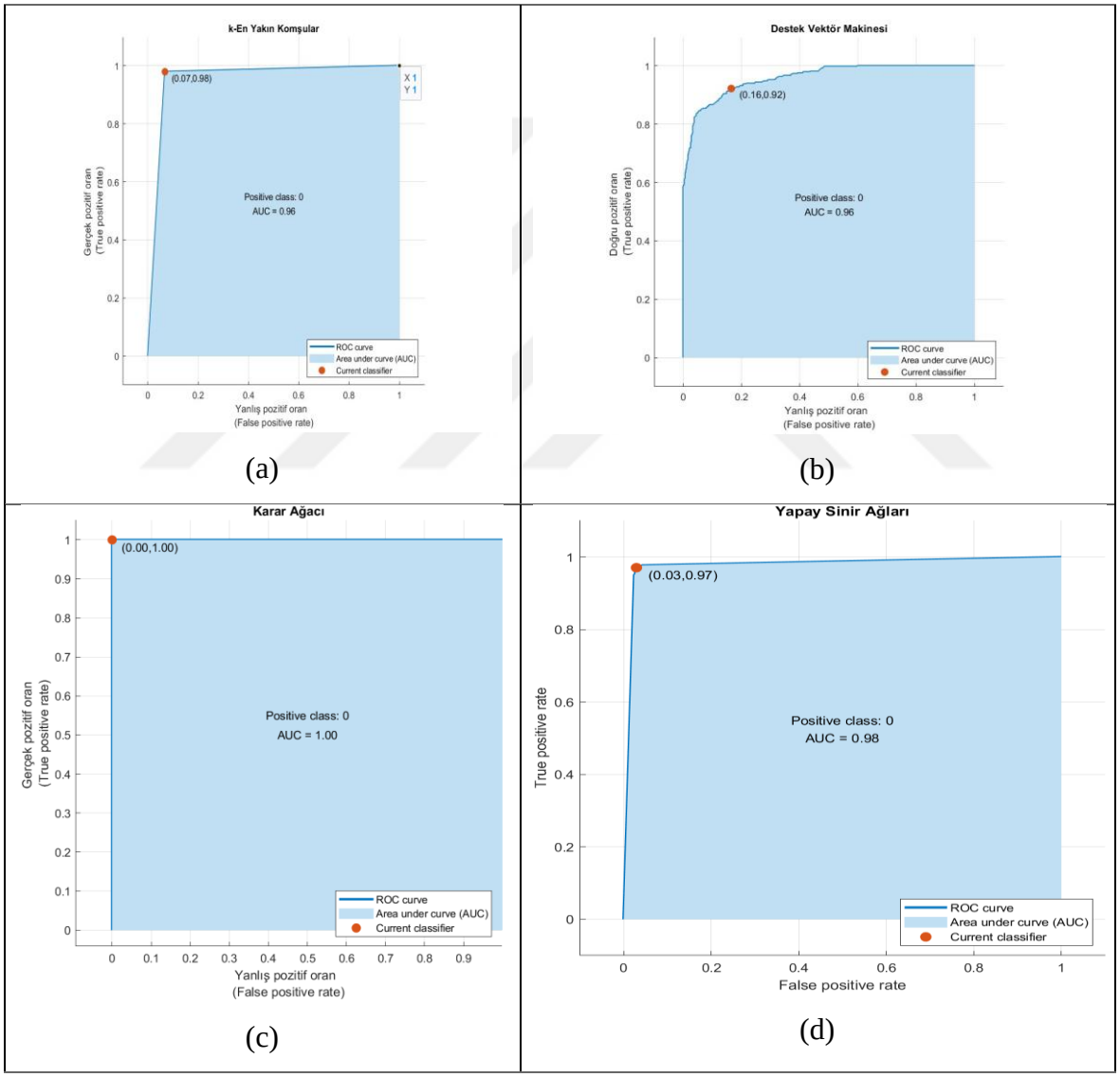
•Modelin öngörü etkinliği, Çizelge 6.4' de verilen AUC'a ilişkin değer aralıkları dikkate alınarak değerlendirilmektedir.

Çizelge 6.4. AUC değer aralıkları

AUC değer aralıkları	Açıklaması
AUC =0,5	Model sınıflandırma etkinliğine sahip değildir
0,5 <AUC < 0,7	Model zayıf sınıflandırma etkinliğine sahiptir.
0,7 ≤AUC < 0,8	Model kabul edilebilir bir sınıflandırma etkinliğine sahiptir.
0,8 ≤AUC < 0,9	Model mükemmel bir sınıflandırma etkinliğine sahiptir.
0,9 ≤AUC	Model üstün bir sınıflandırma etkinliğine sahiptir.

Çizelge 6.5. Web trafiği için karmaşıklık matrisi ve sınıflandırma sonuçları

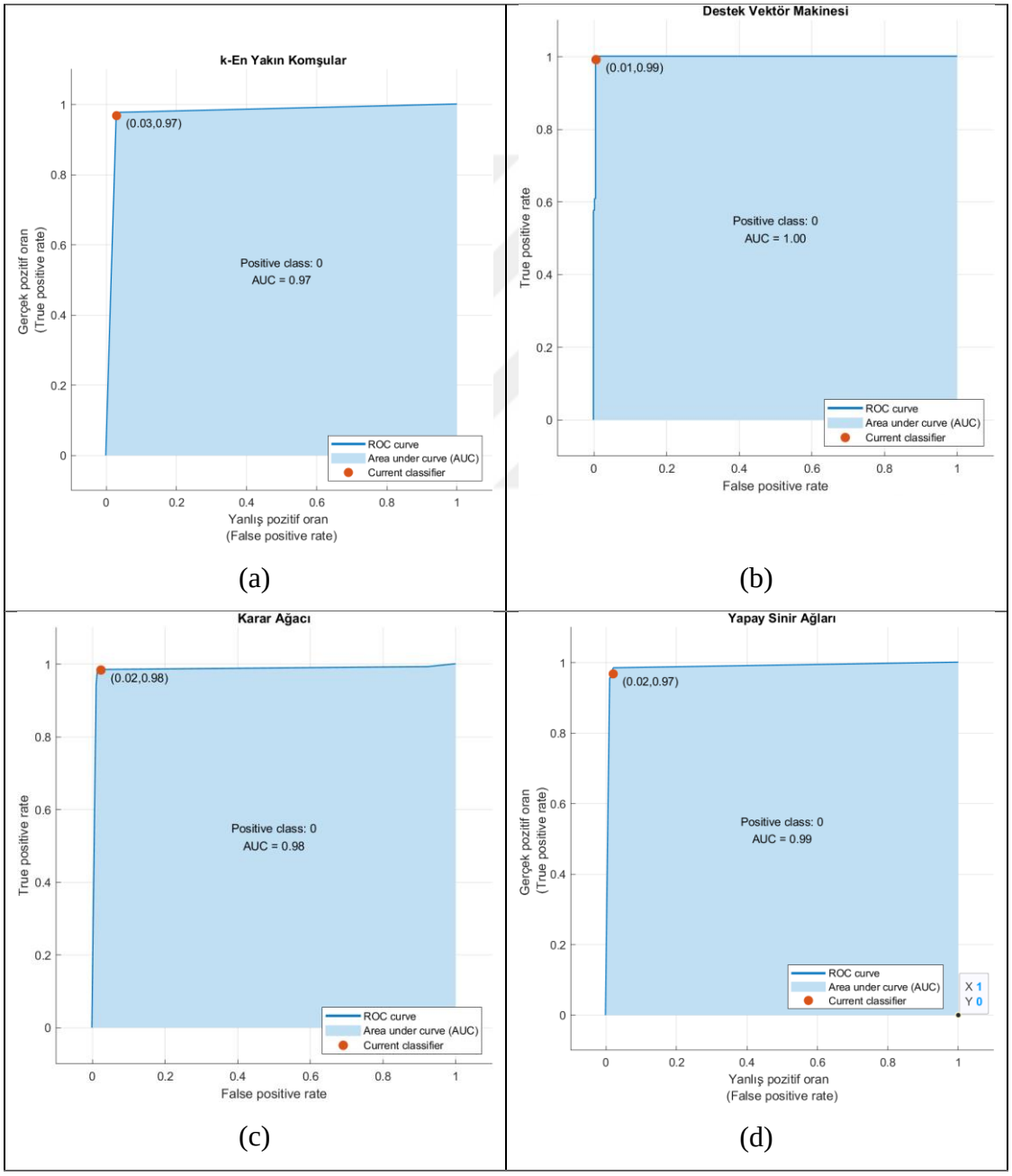
Makine Öğrenme Modeli	DP	YN	YP	DN	Doğruluk (%)	Duyarlılık (%)	Özgüllük (%)	Kesinlik (%)	F1-Puanı (%)
k-En Yakın Komşular	468	10	28	272	95,74	97,91	93,22	94,35	96,1
Destek Vektör Makinesi	441	37	68	341	88,22	92,26	83,54	86,64	89,36
Karar Ağacı	478	0	0	413	100	100	100	100	100
Yapay Sinir Ağları	464	14	12	401	97,08	97,07	97,09	97,48	97,27



Şekil 6.7. Web trafiği için ROC eğrileri (a) k-En Yakın Komşular, (b) Destek Vektör Makinesi, (c) Karar Ağacı, (d) Yapay Sinir Ağları

Çizelge 6.6. DNS trafiği için karmaşıklık matrisi ve sınıflandırma sonuçları

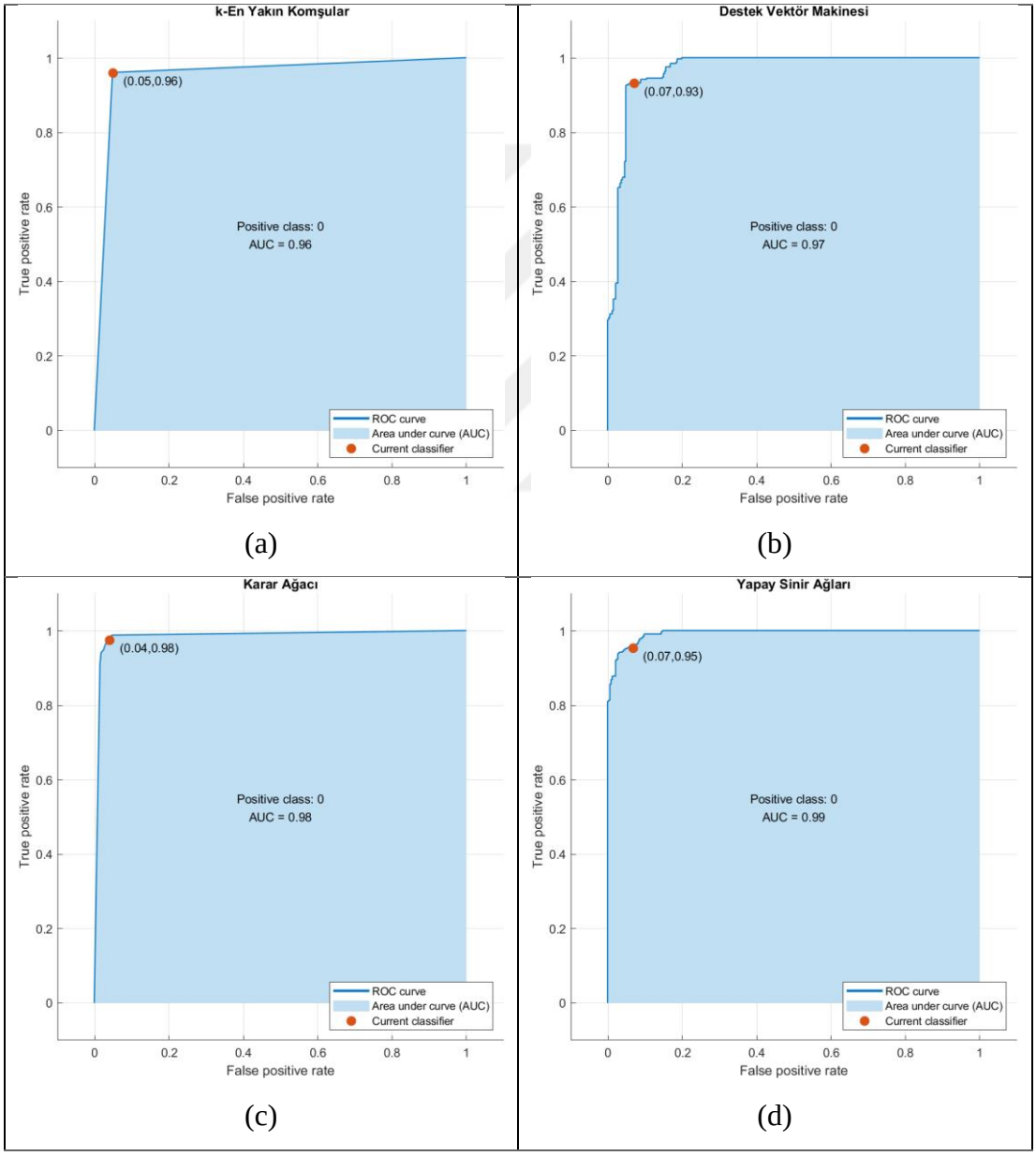
Makine Öğrenme Modeli	DP	YN	YP	DN	Doğruluk (%)	Duyarlılık (%)	Özgüllük (%)	Kesinlik (%)	F1-Puanı (%)
k-En Yakın Komşular	244	8	10	332	96,97	96,83	97,08	96,06	96,44
Destek Vektör Makinesi	250	2	2	340	99,33	99,21	99,42	99,21	99,21
Karar Ağacı	248	4	8	334	97,98	98,41	97,66	96,88	97,64
Yapay Sinir Ağları	244	8	7	335	97,47	96,83	97,95	97,21	97,02



Şekil 6.8. DNS trafiği ROC eğrileri (a) k-En Yakın Komşular, (b) Destek Vektör Makinesi, (c) Karar Ağacı, (d) Yapay Sinir Ağları

Çizelge 6.7. ICMP trafiği için karmaşıklık matrisi ve sınıflandırma sonuçları

Makine Öğrenme Modeli	DP	YN	YP	DN	Doğruluk (%)	Duyarlılık (%)	Özgüllük (%)	Kesinlik (%)	F1-Puanı (%)
k-En Yakın Komşular	314	13	16	310	95,56	96,02	95,09	95,15	95,59
Destek Vektör Makinesi	305	22	23	303	93,11	93,27	92,94	92,99	93,13
Karar Ağacı	319	8	13	313	96,78	97,55	96,01	96,08	96,81
Yapay Sinir Ağları	312	15	22	304	94,33	95,41	93,25	93,41	94,40



Şekil 6.9. ICMP trafiği için ROC eğrileri (a) k-En Yakın Komşular, (b) Destek Vektör Makinesi, (c) Karar Ağacı, (d) Yapay Sinir Ağları

Sınıflandırma işlemini k-En Yakın Komşular algoritması ile gerçekleştirmek için bakılacak komşu sayısı olan k değeri 10 olarak belirlenmiş ve uzaklık fonksiyonu olarak Öklid seçilmiştir. Karar Ağacı yönteminde bölme kriteri olarak Gini algoritması belirlenmiştir. Yapay Sinir Ağı yöntemi ile sınıflandırma için giriş katmanı ile çıkış katmanı arasında sadece tek bir gizli katman kullanılmıştır. Giriş katmanında 18 giriş düğümü bulunmaktadır. Çıkış katmanında 1 adet nöron bulunmaktadır. Gizli katmandaki nöron sayısı ise 10 olup, eğitim algoritması olarak Levenberg-Marquardt algoritması kullanılmıştır. Destek Vektör Makinesi yöntemi ile sınıflandırmada ise kernel olarak Radial Basis Function seçilmiş, kutu kısıtlama düzeyi 1, kernel ölçek değeri 0,9 olarak belirlenmiştir.

Deneysel çalışmalar sonucunda, üç farklı veri kümesi için de tüm sınıflandırma modelleri ile çok iyi sonuçlar elde edilmiştir. Çizelge 6.6’ da karmaşıklık matrisi ve Şekil 6.7’de ROC eğrileri verilen; DNS trafiklerinden oluşan veri kümesi üzerinde, en iyi doğruluk oranı Destek Vektör Makinesi yöntemi ile %99,21 olarak elde edilirken, Karar Ağacı, Yapay Sinir Ağları ve k-En Yakın Komşular yöntemleri ile sırasıyla %97,64, %97,02 ve %96,44 olarak belirlenmiştir. Çizelge 6.5’ de karmaşıklık matrisi ve Şekil 6.6’de ROC eğrileri verilen; Web trafiklerinden oluşan veri kümesi üzerinde ise en iyi doğruluk oranı Karar Ağacı yöntemi ile %100 olarak elde edilirken, Yapay Sinir Ağları, k-En Yakın Komşular ve Destek Vektör Makinesi yöntemleri ile sırasıyla %97,27, %96,1 ve %89,36 olarak belirlenmiştir. Son olarak, Çizelge 6.7’ de karmaşıklık matrisi ve Şekil 6.8’de ROC eğrileri verilen; ICMP trafiklerinden oluşan veri kümesi üzerinde, en iyi doğruluk oranı Karar Ağacı yöntemi ile %96,81 olarak elde edilirken, k-En Yakın Komşular, Yapay Sinir Ağları ve Destek Vektör Makinesi yöntemleri ile sırasıyla %95,59, %93,4 ve %93,13 olarak belirlenmiştir. Deneysel çalışma sonucunda makine öğrenmesi yöntemleri ile elde edilen sınıflandırma performans sonuçları özet listesi Çizelge 6.8’de verilmiştir.

Çizelge 6.8. Makine öğrenme modellerinin sınıflandırma sonuçları

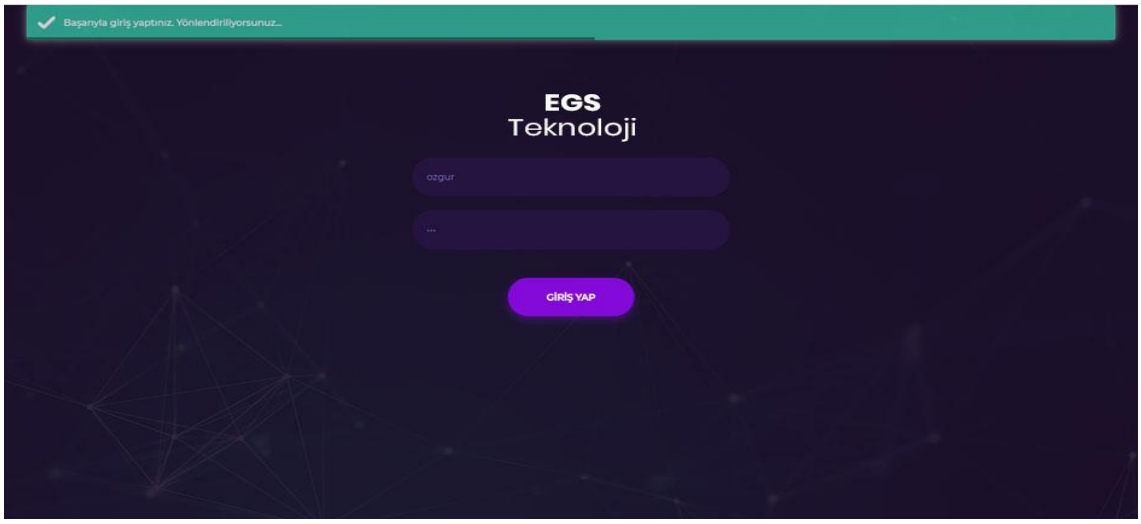
Saldırı Türü	Makine Öğrenme Modeli	Doğruluk (%)	Duyarlılık (%)	Özgüllük (%)	Kesinlik (%)	F1-Puanı (%)	Süre (sn)
HTTP Flood	k-En Yakın Komşular	95,74	97,91	93,22	94,35	96,1	3,0283
	Karar Ağacı	100	100	100	100	100	3,4141
	Yapay Sinir Ağları	97,08	97,07	97,09	97,48	97,27	7,212
	Destek Vektör Makinesi	88,22	92,26	83,54	86,64	89,36	4,6593
DNS Flood	k-En Yakın Komşular	96,97	96,83	97,08	96,06	96,44	3,793
	Karar Ağacı	97,98	98,41	97,66	96,88	97,64	4,1379
	Yapay Sinir Ağları	97,47	96,83	97,95	97,21	97,02	7,0242
	Destek Vektör Makinesi	99,33	99,21	99,42	99,21	99,21	8,1567
ICMP Flood	k-En Yakın Komşular	95,56	96,02	95,09	95,15	95,59	3,7376
	Karar Ağacı	96,78	97,55	96,01	96,08	96,81	3,4174
	Yapay Sinir Ağları	94,33	95,41	93,25	93,41	94,4	8,7326
	Destek Vektör Makinesi	93,11	93,27	92,94	92,99	93,13	7,1715

Elde edilen sınıflandırma sonuçlarına göre en başarılı sonuçlar; DNS trafiklerinden oluşan veri kümesi için Destek Vektör Makinesi modelinin kullanımı ile gerçekleşirken, Web ve ICMP trafiklerinden oluşan veri kümelerinde Karar Ağacı modelinin kullanımı ile gerçekleşmiştir. Bundan sonraki aşamada çevrimdışı ortamda eğitilmiş modeller gerçek zamanlı olarak saldırı tespitinde kullanılmıştır. Şekil 6.10’ da yapılan testlerin doğruluk ve eğitim sürelerinin grafikleri verilmiştir.

Şekil 6.10. Test süresi ve doğruluk grafikleri (a) HTTP saldırısı, (b) DNS saldırısı, (c) ICMP saldırısı

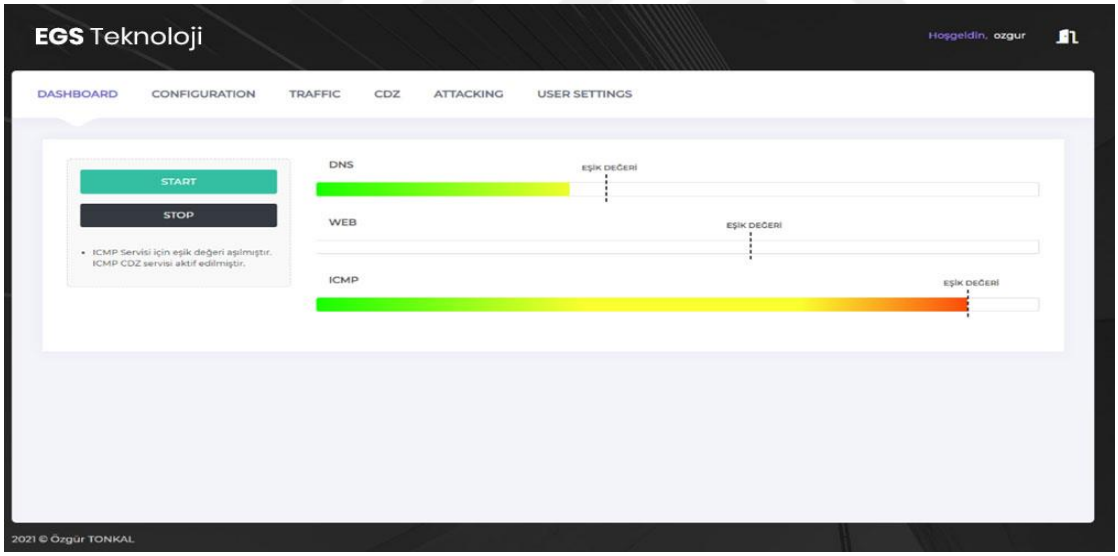
6.3. Önerilen Yöntemin Uygulamaya Alınması ve Test Edilmesi

Geliştirilen trafik sınıflandırmaya dayalı otonom güvenlik modelinin testi için bir arayüz geliştirilmiştir. "EGS Teknoloji" olarak isimlendirilen bu uygulama üzerinden izleme, ölçüm ve analiz işlemleri gerçekleştirilmektedir. Oluşturulan arayüz 7 farklı alt arayüzden oluşmaktadır.



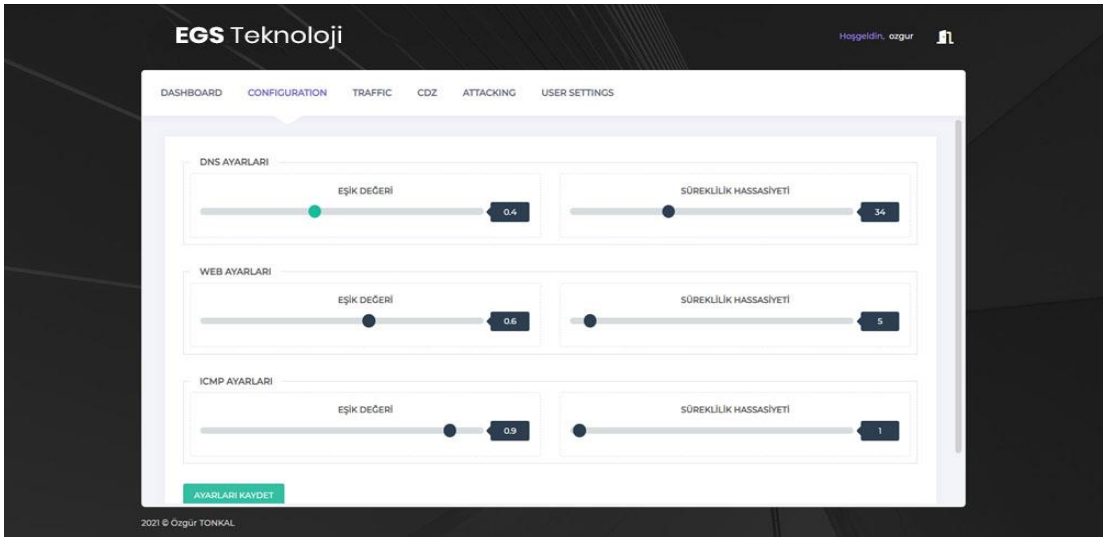
Şekil 6.11. Uygulama giriş arayüzü

Şekil 6.11, giriş arayüzünü göstermektedir. Uygulamaya kullanıcı adı ve şifresi ile giriş yapılmaktadır.



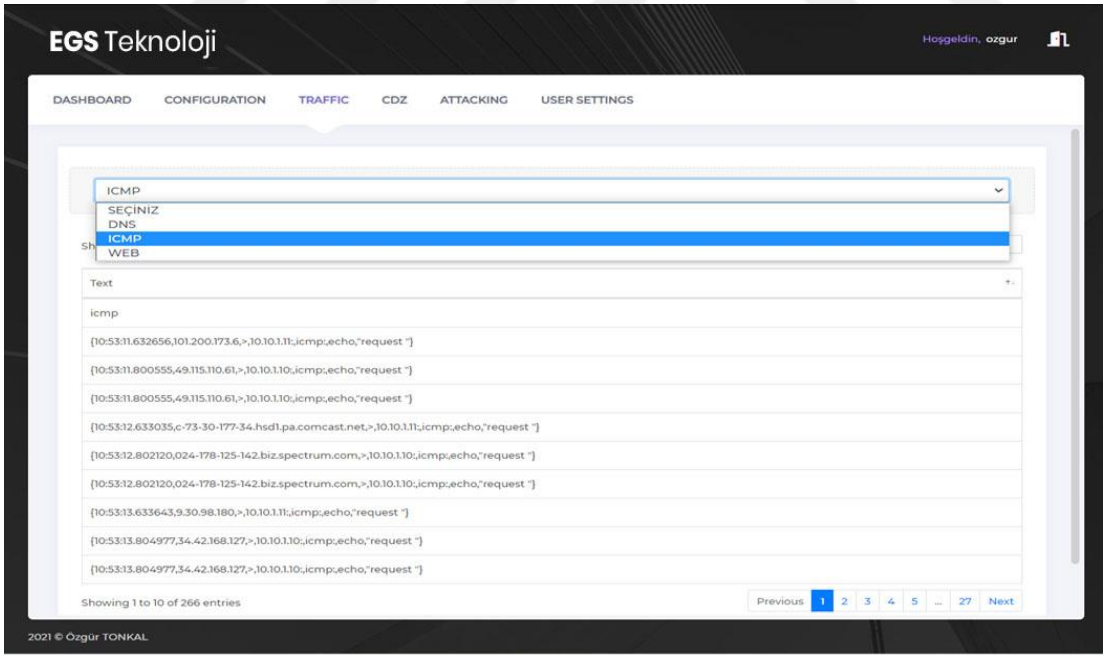
Şekil 6.12. Koruma başlatma/durdurma arayüzü

Şekil 6.12’ de koruma sisteminin aktif/pasif edildiği kontrol arayüzü gösterilmektedir. Bu arayüz aracılığı ile trafik sürekli dinlenir ve matematiksel model kullanılarak trafik sürekliği ve yoğunluğu hesaplanarak belirlenen eşik değerine ulaşılıp ulaşılmadığı (her trafik değerin için farklı) ölçülür. Kritik değerlere ulaşıldığında ilgili SGA’nın açıldığı ya da görevi biten SGA’ nın kapandığı bilgisi burada verilir.



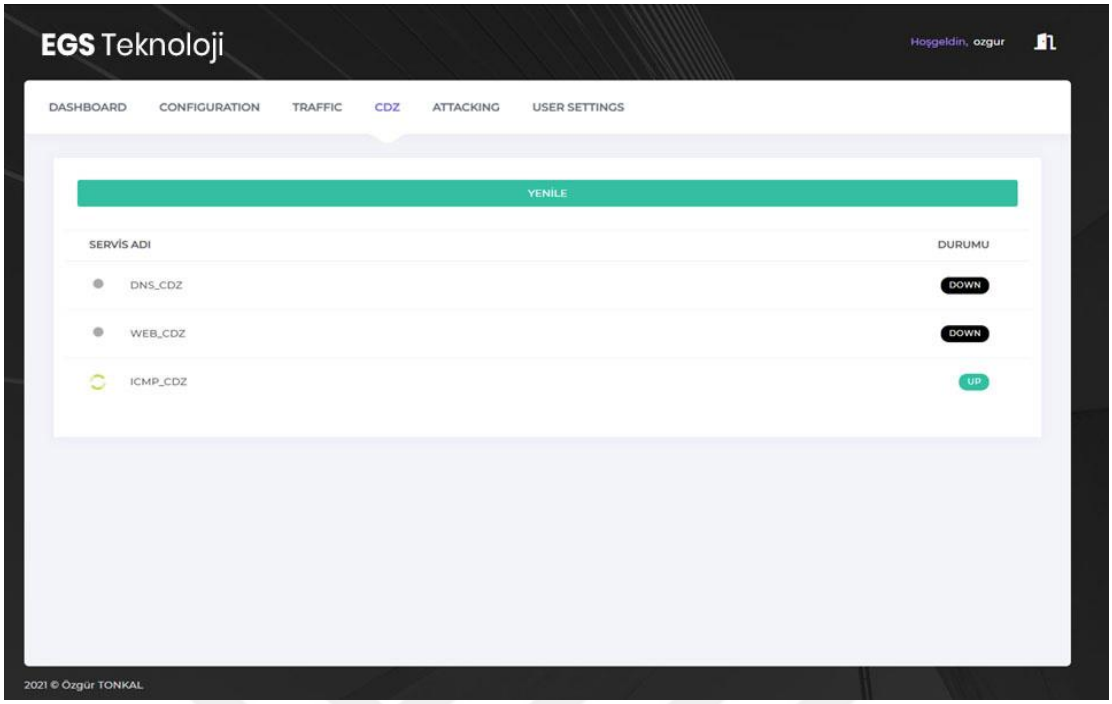
Şekil 6.13. Matematiksel model değişkenlerinin ayarlandığı arayüz

Şekil 6.13' de yapılandırma ayarlarının gerçekleştirildiği arayüzün görüntüsü verilmiştir. Burada farklı trafik türleri için matematiksel model aracılığı yapılacak hesaplamalarda kullanılacak eşik değer ve süreklilik hassasiyeti değerleri ayarlanır.



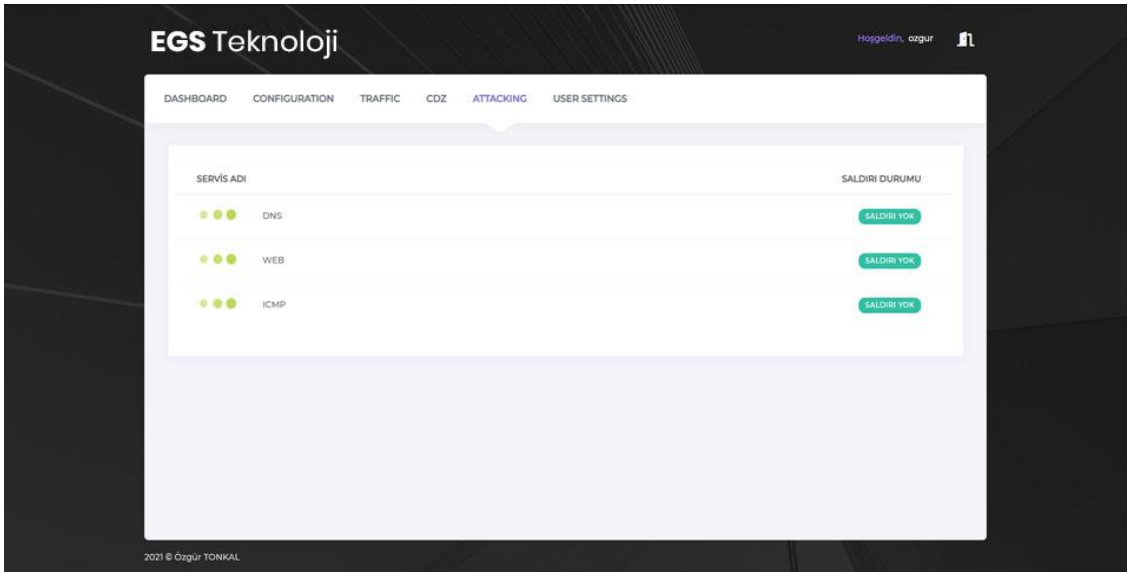
Şekil 6.14. Trafik Paketlerinin izlendiği arayüz

Şekil 6.14' de ekran görüntüsü verilen dördüncü arayüz, ağ trafik paketlerinin liste şeklinde görüntülebilmesine imkân verir. Bu arayüz ile ağ trafik paketlerinin yapısı ayrıntılı olarak gözlemlenebilir.



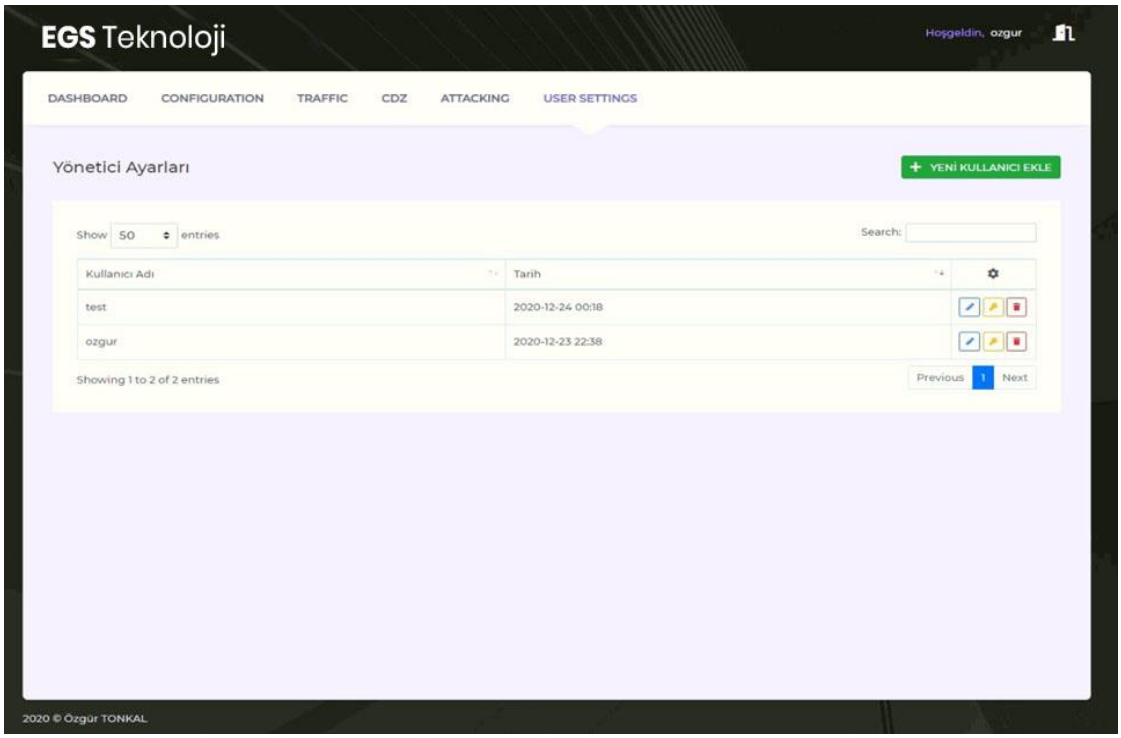
Şekil 6.15. SGA izleme arayüzü

Şekil 6.15’ de ekran görüntüsü verilen beşinci arayüz, SGA durum bilgilerini içerir. TSS modülünde gerçekleşen sınıflandırma ve ölçüm işlemleri sonucunda şüpheli trafik uyarısı geldiği anda EGS üzerindeki ilgili siber güvenlik alanı aktif olarak trafik değerlendirme süreci başlatılır. Bu ekran aracılığı ile aktif/pasif olan SGA durum bilgisi kullanıcıya verilir.



Şekil 6.16. Saldırı izleme arayüzü

Şekil 6.16’ da ekran görüntüsü verilen altıncı arayüz, hangi saldırıların aktif olduğunun bilgisini vermektedir. Saldırı aktif olduğunda geliştirilen güvenlik modeli kullanılarak saldırı trafiği engellenmekte, normal trafikler hedeflerine iletilmektedir.



Şekil 6.17. Kullanıcı tanımlama arayüzü

Şekil 6.17’ de ekran görüntüsü verilen yedinci arayüz, geliştirilen uygulamayı kullanacak kişiler için kullanıcı giriş bilgilerinin tanımlanması veya değiştirilmesine imkân verir.

Testler esnasında aşağıdaki senaryo ve durumlar incelenmiştir:

- Normal ağ kullanıcıları ile aynı anda farklı tipte yoğun ağ trafikleri oluşturulmuş, geliştirilen sistemin yanlış uyarı verip vermediği, normal ağ trafiklerini engelleyip engellemediği incelenmiştir.
- Saldırı senaryoları uygulanmaya başladıktan ne kadar süre sonra geliştirilen sistemin saldırıları tespit edebildiği incelenmiştir. Tüm testler bittiğinde bu sürelerin ortalaması hesaplanmıştır.
- Saldırı senaryoları uygulanmaya başladıktan hemen sonra, sistem kaynaklarının ne kadar kullanıldığı hesaplanmış ve koruma sisteminin sisteme getirdiği ek yük hesaplanmıştır.

- Saldırı önlendikten ne kadar süre sonra normal ağ kullanıcılarının ağ kaynaklarına kesintisiz erişebildiği incelenmiştir.
- Koruma sistemi öğrenme modundayken farklı özelliklere sahip trafikler uygulanmış, sistem koruma moduna geçtiğinde benzer ve farklı özellikteki ağ trafikleri uygulanarak koruma sisteminin davranışı incelenmiştir.
- Hangi türdeki atakların geliştirilen koruma sistemi tarafından tespit edilebildiği incelenmiştir.
- Geliştirilen matematiksel modelin sistem başarısına etkisi incelenmiştir.
- Geliştirilen siber güvenlik modeli test edilerek maliyet, gerçek zamanlılık yönlerinden incelenmiştir.
- Farklı makine öğrenme yöntemlerinin saldırı tespit başarımları karşılaştırmalı olarak incelenmiştir.

6.4. Deneysel Bulgular ve Tartışma

Önerilen güvenlik modelinin test aşaması geliştirilen uygulama arayüzü (EGS Teknoloji) üzerinden gerçekleştirilmiştir. Saldırı trafiklerini oluşturmak için bu çalışma için özel olarak geliştirilen ve hacimsel trafik üretebilen "Trafik Üretici" yazılımı kullanılmıştır. Bu yazılım tek bir arayüz aracılığı ile hping3 komutunu kullanarak farklı türde ve özellikte saldırı trafikleri oluşturabilme imkânı vermektedir. Çizelge 6.9, test senaryolarında kullanılan hping3 parametrelerini göstermektedir.

Çizelge 6.9. Testlerde kullanılan hping3 parametreleri ve açıklamaları

Parametre	Açıklama
-c	Paket sayısı
-w	Pencere paket boyutu
-d	Veri boyutu
-p	Port Numarası (hedef)
-a	Sahte ip adresi
-flood	Paketleri en hızlı şekilde yanıt göstermeksizin gönderir
-S	TCP paketleri
-1	ICMP paketleri
-2	UDP paketleri

Testlerde Gerçekleştirilen Senaryolar

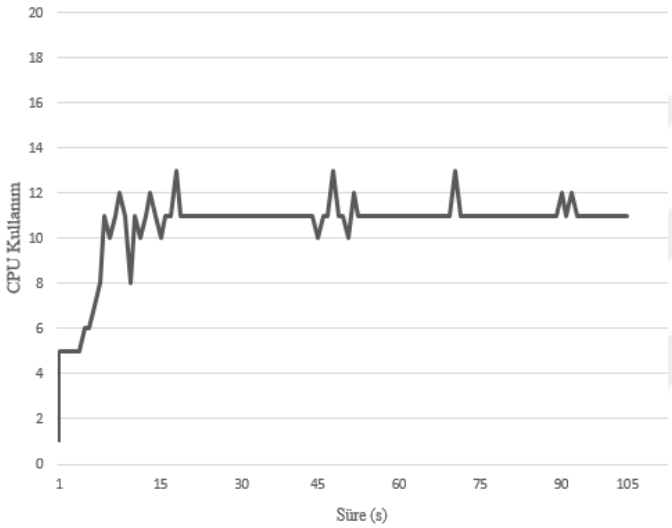
Önerilen güvenlik modelinin uygulama testleri farklı senaryolar oluşturularak gerçekleştirilmiştir. Çizelge 6.10, bu senaryoları özetlemektedir. Gerçekleştirilen hacimsel saldırılar, hedef sunuculara ulaşmadan önce otonom olarak çalışan sanal siber güvenlik alanlarında değerlendirilip engelleme/izin verme yaklaşımları uygulanmıştır. Oluşturulan topoloji, test düzleminin kullanabildiği sistem kaynakları ölçüsünde çalışmaktadır (Bkz. Çizelge 6.1). Bu kapsamda Çizelge 6.10' da listelenen senaryo testleri, önerilen modelin en alakalı işlevsel yönlerini doğrulamak amacıyla yapılmıştır.

Çizelge 6.10. Test senaryo özetleri

Senaryo	Açıklama	Değerlendirilen Parametreler	Hping3 parametreler
I	Web Flood saldırısı, istemci bilgisayarın sunucuya erişimi var iken saldırgan bilgisayar üzerinden Web Sunucusuna yönelik başlatılmıştır.	Engelleme Süresi; paket ortalama gidiş-dönüş süresi (RTT); paket kaybı (packet loss); sistem kaynak kullanımı (CPU-RAM)	-flood; -S; -p
II	DNS Flood saldırısı, istemci bilgisayarın sunucuya erişimi var iken saldırgan bilgisayar üzerinden Web Sunucusuna yönelik başlatılmıştır.	Engelleme Süresi; paket ortalama gidiş-dönüş süresi (RTT); paket kaybı (packet loss); sistem kaynak kullanımı (CPU-RAM)	-flood; -2; -p
III	ICMP Flood saldırısı, istemci bilgisayarın sunuculara erişimi var iken, saldırgan bilgisayar üzerinden Web ve DNS sunucularına yönelik başlatılmıştır.	Engelleme Süresi; paket ortalama gidiş-dönüş süresi (RTT); paket kaybı (packet loss); sistem kaynak kullanımı (CPU-RAM)	-flood; -1; -p
IV	Web, DNS ve ICMP flood saldırıları, istemci bilgisayarın sunuculara erişimi var iken, saldırgan bilgisayar üzerinden Web ve DNS sunucularına yönelik başlatılmıştır.	Engelleme Süresi; paket ortalama gidiş-dönüş süresi (RTT); paket kaybı (packet loss); sistem kaynak kullanımı (CPU-RAM)	-flood; -S; -1; -2; -p

Uygulama topolojisi oluşturulduktan sonra istemci bilgisayar üzerinden sunucu bilgisayarlara (Web Sunucu, DNS sunucu) erişim testleri gerçekleştirilmiştir. Anahtar iletişim başladığı anda takas işlemlerini gerçekleştirerek, akış kurallarını günceller ve kendisine bağlı olan bütün cihazların MAC adresleri, IP adresleri gibi bilgilerinin akış tablosuna kayıt işlemlerini tamamlar.

Şekil 6.18, ağdaki tüm bilgisayarlar normal iletişim durumunda iken sistem kaynak kullanım eğilimini göstermektedir.



Şekil 6.18. Saldırı öncesi CPU kaynak kullanımı

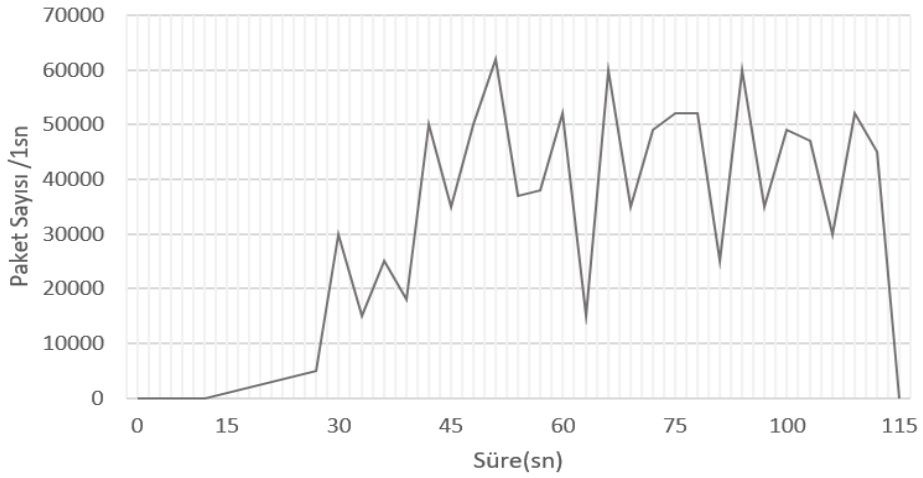
Çizelge 6.11, saldırı olmadığında bazı QoS ölçümlerini göstermektedir. Bu sonuçlar, istemci bilgisayardan alınan ICMP testlerinden elde edilmiştir.

Çizelge 6.11. Saldırı olmadan sistem performansı

Test No	WEB SUNUCU		DNS Sunucu	
	Ortalama RTT (ms)	Paket Kaybı (%)	Ortalama RTT (ms)	Paket Kaybı (%)
1	0,983	0	1,309	0
2	0,988	0	1,263	0
3	1,231	0	1,293	0
4	1,291	0	1,215	0
5	1,180	0	1,309	0
Ortalama	1,134	0	1,277	0

Şekil 6.19, önerilen Siber Savunma Sistemi devre dışı olduğunda ICMP flood saldırısı tarafından üretilen trafik hacmini göstermektedir. 50 saniye sonra, ağın bu hacimsel saldırıdan dolayı önemli miktarda trafiğe maruz kaldığı görülebilir.

Bu saldırı 62.000 paket/sn (3.720.000 paket/dak)' de tepe noktasına ulaşmıştır. Bu noktadan sonra ağ verimli bir şekilde çalışamaz hale gelmektedir. Gönderilen paketlerin sunuculara erişimi kesintiye uğramıştır.

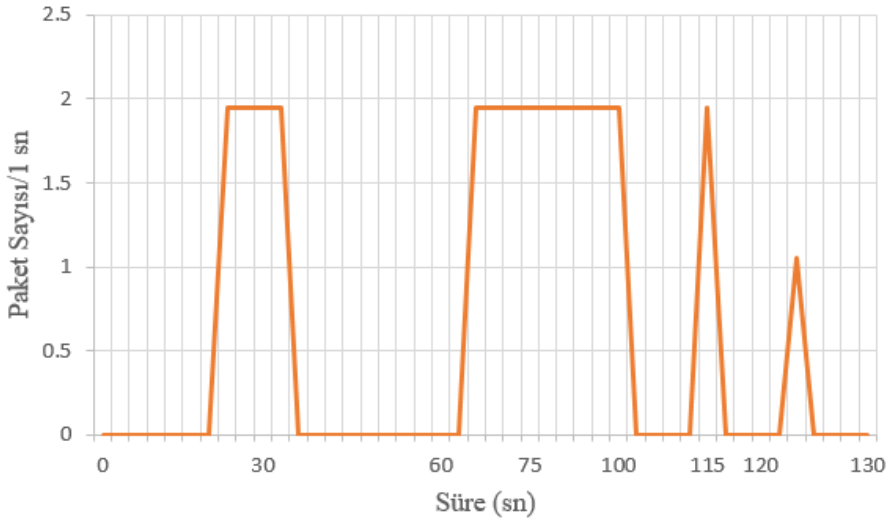


Şekil 6.19. Savunma sistemi kapalıyken ICMP flood saldırısı ağ trafiği

Senaryo I

Bu senaryo kapsamında gerçekleştirilen test, istemci bilgisayar, web sunucusuna "normal" erişimini korurken, saldırgan bilgisayar üzerinden, "hping3 http flood" parametreleri kullanılarak gerçekleştirildi. Saldırgan bilgisayar, saldırı paketlerini 512 bayt sabit yük boyutuna sahip saldırı paketlerini, saniyede 2000 sabit paket hızı (pakets per second-pps) ile sunucuya iletmiştir. Web sunucusu saldırı altındayken istemci bilgisayar üzerindeki veri akış hızı Şekil 6.20 'de gösterilmektedir.

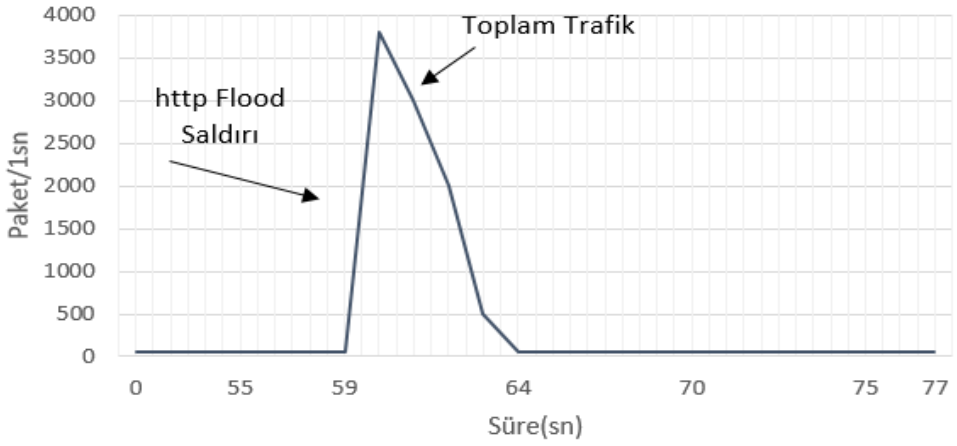
İstemci bilgisayardan sunucuya gönderilen paketlerde, ilk etapta iletim gecikmelerinin yaşandığı ve bir süre sonra sunucuya paketlerin ulaşmadığı gözlemlenmektedir. Bu oran değişimleri, saldırıya uğrayan sunucunun kaynak yetersizliğinden veya ağ tıkanıklığından kaynaklanmaktadır.



Şekil 6.20. Senaryo I- saldırı süresince istemci paket iletim oranı değişimi

Saldırgan makine üzerinden saldırı trafiği başlatıldıktan sonra geliştirilen uygulama arayüzü aracılığı ile koruma devreye alınır. Kontrolcü üzerindeki TSS modülü aktif olarak gelen trafik sınıflandırılır ve matematiksel model aracılığı ile süreklilik ölçümü yapılır. Belirlenen eşik değeri üzerinde bir hesaplama sonucu ortaya çıkarsa TSS modülü devreye alınarak ilgili trafik siber güvenlik alanlarında (SGA) değerlendirilir. Eğitilmiş makine öğrenme modeli kullanılarak yapılan bu değerlendirme sonucunda gelen trafik saldırı ise düşürülür (drop). Normal trafik ise ilgili sunucuya iletilir (forward). Koruma devredeyken bütün ağ trafik türleri için aynı süreçler gerçekleştirilmektedir.

Web sunucusuna yönelik gerçekleştirilen “HTTP flood” saldırısının grafiksel gösterimi Şekil 6.21’ de verilmiştir. Burada üretilen trafik, zamanın bir fonksiyonu olarak görebilir. Grafikte sistemin ilk 60 saniyedeki normal kullanımı gözlemlenebilir. Ardından, saldırgan makine üzerinden bir HTTP flood saldırısı başlatılmıştır. Koruma devreye girdikten sonra (64. saniyeden itibaren) engelleme gerçekleşerek ağ “normal” çalışma durumuna geçmiştir.



Şekil 6.21. Senaryo I- saldırı tespiti ve engellenmesi

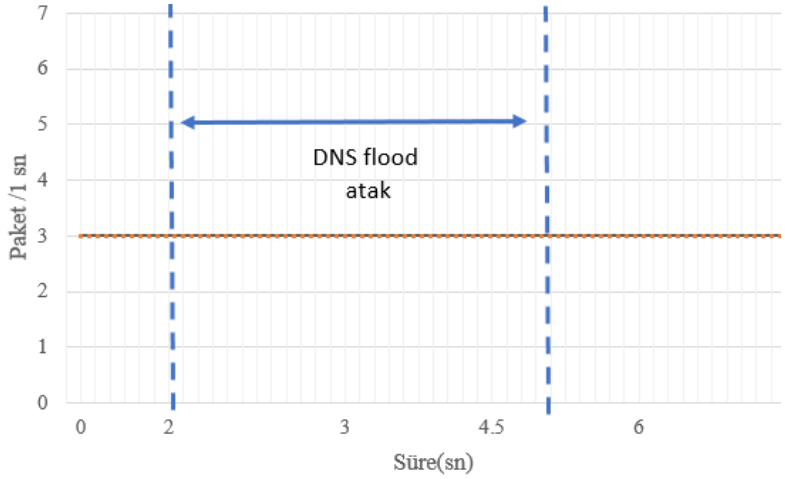
Çizelge 6.12; saldırı engelleme süresi, ortalama paket gidiş-dönüş süresi (RTT) ve paket kaybının yüzdesi değişkenlerine ilişkin alınan sonuçları göstermektedir. Ortalama RTT sonuçları ve paket kaybı yüzdesi, istemci bilgisayarından oluşturulan ICMP protokolü vasıtası ile elde edilirken saldırı engelleme süresi, Wireshark G/Ç grafiğinden elde edilmiştir. Önerilen güvenlik sistemimiz, HTTP flood saldırısını tespit etmek ve engellemek için ortalama olarak yaklaşık 7 saniyeye ihtiyaç duymaktadır.

Çizelge 6.12. Senaryo I-Koruma devredeyken saldırı süresince sistem performansı

Test No	WEB Sunucu		
	Engelleme Süresi (sn)	Ortalama RTT (ms)	Paket Kaybı (%)
1	5	1,619	0
2	6	1,547	0
3	6	2,541	0
4	7	1,848	0
5	10	2,767	0
Ortalama	6,8	2,058	0

Senaryo II

Bu senaryoda, saldırgan makine üzerinden sahte IP adresleri kullanılarak “hping3 DNS flood” saldırısı gerçekleştirilmiştir. Saldırgan bilgisayar, 512 bayt sabit yük boyutuna sahip saldırı paketlerini saniyede 2000 sabit paket hızı (pakets per second-pps) ile DNS sunucuya iletmiştir. DNS sunucu saldırı altındayken istemci bilgisayar üzerindeki veri akışı Şekil 6.22’ de gösterilmektedir.

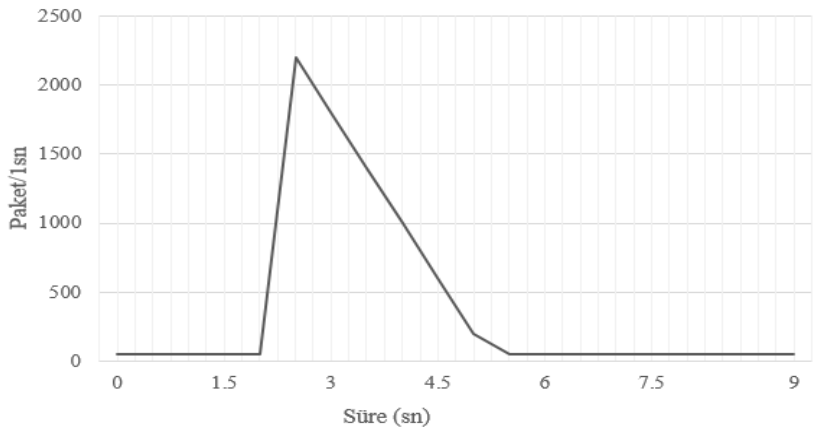


Şekil 6.22. Senaryo II- saldırı süresince istemci paket iletim oranı

Koruma devreye alındıktan sonra saldırgan tarafından DNS sunucuya gönderilen saldırı paketleri düşülmüştür. İstemci bilgisayardan DNS sunucuya gönderilen paketlerin iletim oranının saldırı sırasında sabit kaldığı gözlemlenmiştir.

Saldırı algılama ve engelleme süreçleri grafiksel olarak Şekil 6.23' de verilmiştir. Burada üretilen trafiği, zamanın bir fonksiyonu olarak görebiliriz.

İlk iki saniye boyunca ağ normal durumunda çalışmıştır. Ardından, saldırgan makine üzerinden sunucuya yönelik "DNS flood" saldırısı gerçekleştirilmiştir. Bu saldırı, başlangıcından 3 saniye sonra engelleme gerçekleşmiştir. 5 saniyeden sonra, trafik dengelenmiş ve "olağan" trafik akışına geri dönülmüştür.



Şekil 6.23. Senaryo II -saldırı tespiti ve engellenmesi

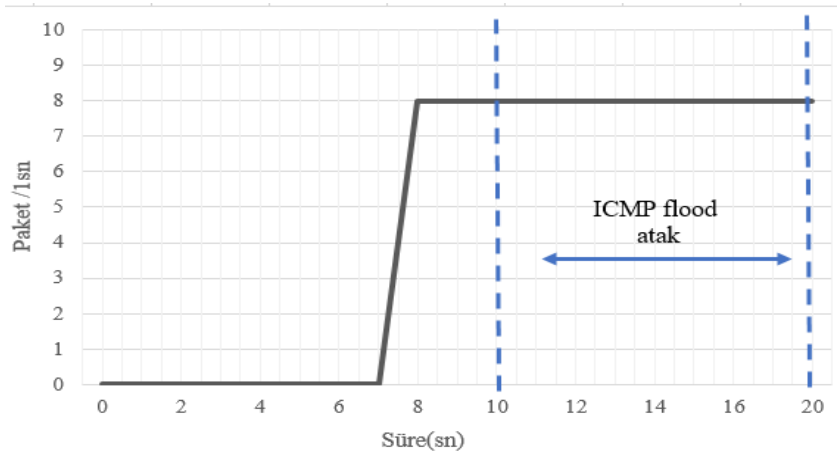
Çizelge 6.13; saldırı engelleme süresi, ortalama paket gidiş-dönüş süresi (RTT) ve paket kaybının yüzdesi değişkenlerine ilişkin alınan sonuçları göstermektedir. Ortalama RTT sonuçları ve paket kaybı yüzdesi, istemci bilgisayarından oluşturulan ping komutu ile alınırken saldırı engelleme süresi, Wireshark G/Ç grafiğinden alınmıştır. Çözümümüz, DNS flood saldırısını tespit etmek ve engellemek için ortalama olarak yaklaşık 4 saniyeye ihtiyaç duymaktadır.

Çizelge 6.13. Senaryo II-Koruma devredeyken saldırı süresince sistem performansı

Test No	DNS Sunucu		
	Engelleme Süresi (sn)	Ortalama RTT (ms)	Paket Kaybı (%)
1	3	1,557	0
2	3	2,523	0
3	4	2,560	0
4	4	2,640	0
5	3	1,897	0
Ortalama	3,4	2,235	0

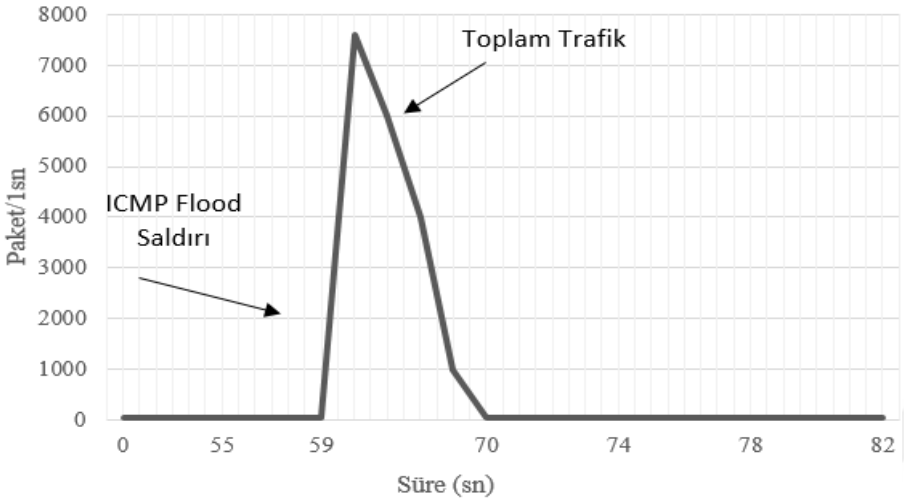
Senaryo III

Bu senaryoda; istemci bilgisayar, Web ve DNS sunucularına "normal" erişimini korurken, saldırgan bilgisayar üzerinden, hping3 parametreleri kullanılarak ICMP flood saldırısı gerçekleştirilmiştir. Saldırgan bilgisayar, 512 bayt sabit yük boyutuna sahip saldırı paketlerini saniyede 2000 sabit paket hızı ile Web ve DNS sunucularına göndermiştir. Sunucular saldırı altındayken istemci bilgisayar üzerindeki veri akışı Şekil 6.24' deki gibidir.



Şekil 6.24. Senaryo III- saldırı süresince istemci paket iletim oranı

Saldırı algılama ve engelleme süreçlerinin grafiksel olarak görselleştirmesi Şekil 6.25' de sunulmaktadır. Burada üretilen trafiği, zamanın bir fonksiyonu olarak görebiliriz.



Şekil 6.25. Senaryo III -saldırı tespiti ve engellenmesi

Grafikte sistemin ilk 60 saniyedeki normal kullanımı gözlemlenebilir. Ardından, saldırgan makine üzerinden bir ICMP flood saldırısı başlatılır. Koruma devreye girdikten sonra (70. saniyeden itibaren) engelleme gerçekleşerek ağ "normal" çalışma durumuna geçmiştir.

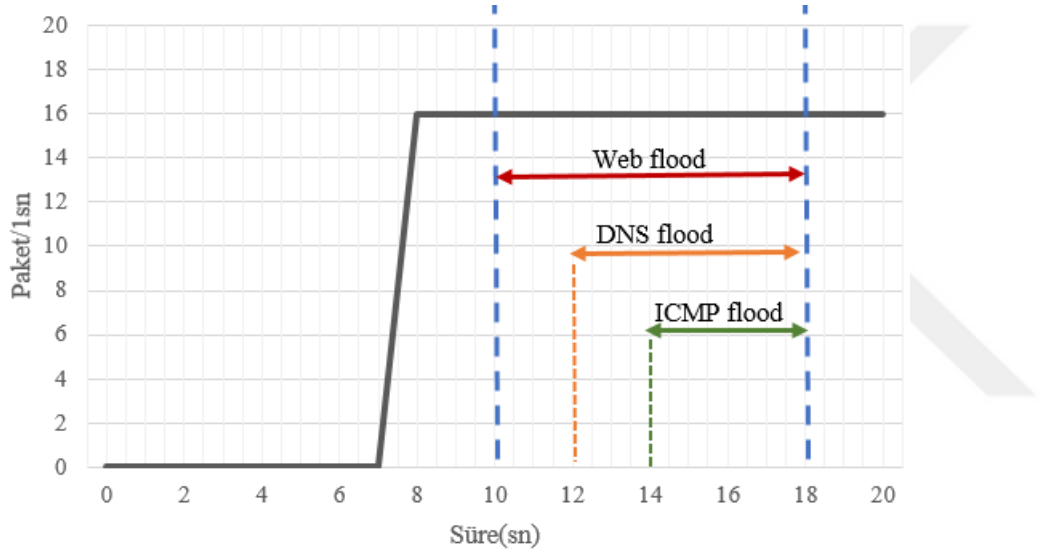
Çizelge 6.14; saldırı engelleme süresi, ortalama paket gidiş-dönüş süresi (RTT) ve paket kaybının yüzdesi değişkenlerine ilişkin alınan sonuçları göstermektedir. Ortalama RTT sonuçları ve paket kaybı yüzdesi, istemci bilgisayarından oluşturulan ping komutu ile alınırken saldırı engelleme süresi, Wireshark G/Ç grafiğinden alınmıştır. Çözümümüz, DNS ve Web Sunuculara yönelik gerçekleştirilen ICMP flood saldırısını tespit etmek ve engellemek için ortalama olarak yaklaşık 13 saniyeye ihtiyaç duymaktadır.

Çizelge 6.14. Senaryo III-Koruma devredeyken saldırı süresince sistem performansı

	DNS Sunucu			WEB Sunucu		
Test No	Engelleme Süresi (sn)	Ortalama RTT(ms)	Paket Kaybı (%)	Engelleme Süresi (sn)	Ortalama RTT(ms)	Paket Kaybı (%)
1	5	1,634	0	6	1,748	0
2	5	1,645	0	6	1,870	0
3	6	1,586	0	7	1,982	0
4	7	2,185	0	7	2,010	0
5	6	1,865	0	8	2,620	0
Ortalama	5,8	1,783	0	6,8	2,046	0

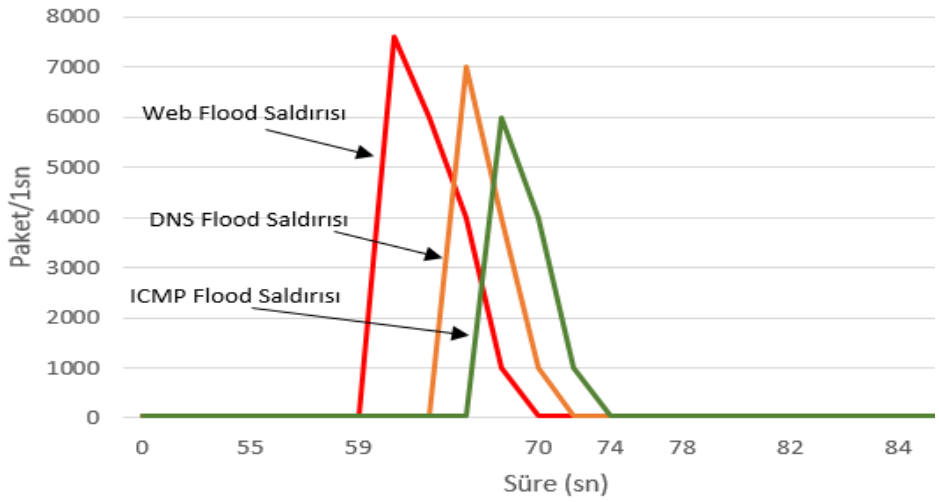
Senaryo IV

Bu senaryoda; istemci bilgisayar, Web ve DNS sunucularına "normal" erişimini korurken, saldırgan bilgisayar üzerinden, hping3 parametreleri kullanılarak sırasıyla Web flood, DNS flood ve ICMP flood saldırıları gerçekleştirilmiştir. Saldırgan bilgisayar üzerinden 512 bayt sabit yük boyutuna sahip saldırı paketleri Web ve DNS sunucularına saniyede 2000 sabit paket hızı ile iletilmiştir. Buradaki amaç 3 saldırının birlikte gerçekleşmesi durumunda sistemin nasıl çalışacağını ölçmektir. Sunucular saldırı altındayken istemci bilgisayar üzerindeki veri akışı Şekil 6.26' deki gibidir.



Şekil 6.26. Senaryo IV- saldırı süresince istemci paket iletim oranı

Saldırı algılama ve engelleme süreçlerinin grafiksel olarak görselleştirmesi Şekil 6.27' de sunulmaktadır. Burada üretilen trafiği, zamanın bir fonksiyonu olarak görebiliriz.



Şekil 6.27. Senaryo IV -saldırı tespiti ve engellenmesi

Grafikte sistemin ilk 60 saniyedeki normal kullanımı gözlemlenebilir. Ardından, iki saniye aralıklarla saldırgan makine üzerinden sırasıyla Web, DNS ve ICMP flood saldırıları başlatılır. Koruma devreye girdikten sonra (74. saniyeden itibaren) tüm saldırı türleri için engelleme gerçekleşerek ağ "normal" çalışma durumuna geçmiştir.

Çizelge 6.15; saldırı engelleme süresi, ortalama paket gidiş-dönüş süresi (RTT) ve paket kaybının yüzdesi değişkenlerine ilişkin alınan sonuçları göstermektedir. Ortalama RTT sonuçları ve paket kaybı yüzdesi, istemci bilgisayarından oluşturulan ping komutu ile alınırken saldırı engelleme süresi, Wireshark G/Ç grafiğinden alınmıştır. Çözümümüz, DNS ve Web Sunuculara yönelik gerçekleştirilen Web, DNS ve ICMP flood saldırılarını tespit etmek ve engellemek için ortalama olarak yaklaşık 14 saniyeye ihtiyaç duymaktadır.

Çizelge 6.15. Senaryo IV-Koruma devredeyken saldırı süresince sistem performansı

Test No	DNS Sunucu			WEB Sunucu		
	Engelleme Süresi (sn)	Ortalama RTT(ms)	Paket Kaybı (%)	Engelleme Süresi (sn)	Ortalama RTT(ms)	Paket Kaybı (%)
1	7	1,782	0	7	1,748	0
2	7	1,805	0	6	1,860	0
3	6	1,765	0	8	1,920	0
4	7	2,160	0	8	2,010	0
5	6	1,875	0	6	2,640	0
Ortalama	6,6	1,877	0	7	1,835	0

Farklı senaryolar kurgulanarak yapılan testlerde elde edilen sonuçların özet bilgileri Çizelge 6.16' da verilmiştir. Bu senaryolar kapsamında, saldırgan bilgisayar üzerinden oluşturulan ve uygulama sunucularını hedef alan farklı türdeki saldırıların (Web, DNS ve ICMP flood saldırıları) YTA iletişimine etkisi araştırılmıştır. Önerilen modele uygun olarak geliştirilen "EGS Teknoloji" yazılım arayüzleri kullanılarak otonom özelliklere sahip koruma sistemi saldırı süresince devreye alınmıştır. Ağ trafik türüne özgü çalışan ve içerisinde eğitilmiş makine öğrenme modellerinin çalıştığı siber güvenlik alanlarında değerlendirilen trafik, saldırı ise düşürülmüş normal trafik ise ilgili sunuculara yönlendirilmiştir.

Çizelge 6.16. Değerlendirme senaryolarından edilen sonuçların karşılaştırılması

Senaryo	Engelleme Süresi(sn)	Ortalama RTT(ms)	Paket Kaybı (%)
Normal Kullanım	-	1,205	0
Senaryo I	6,8	2,058	0
Senaryo II	3,4	2,235	0
Senaryo III	12,6	1,914	0
Senaryo IV	13,6	1,856	0
Ortalama	9,1	2,015	0

7. SONUÇ VE ÖNERİLER

Mevcut YTA kontrolcüleri; güvenlik duvarı ve saldırı tespit sistemleri gibi siber savunma sistemlerini içinde barındıran, ağ trafiği ile ilgili karar verirken aktif olarak bu bileşenleri sürece dahil ederek sonuca ulaşabilecek bir mimari yapıya sahip değildir. YTA kontrolcüsü ağda bulunan diğer güvenlik ürünleri ile entegre olarak saldırıların engellenmesini amaçlamaktadır. Bu çözüm, ağdaki trafik ile ilgili doğru karar verme oranını azaltmanın yanı sıra sonucu ulaşma süresini artırmaktadır. Çünkü kontrolcü ilgili trafiği ağdaki güvenlik duvarı ve saldırı tespit sistemlerine göndererek, gelecek sonuca göre trafik hakkında bir karar verecektir. YTA kontrolcü ve siber savunma sistemleri arasında sürekli trafik akışının olması, gereksiz yere bu bileşenler arasında yüksek boyutlarda trafiğin sürekli olarak dolaşmasına neden olacaktır. Bu durum, ağdaki paket teslim oranının düşürmekte ve uçtan uca gecikme süresinin artırmaktadır. Sonuç olarak YTA kontrolcüsünün kendi içinde tümleşik bir siber savunma sistemini barındırmaması, ağın toplam performansı üzerine olumsuz etki oluşturmaktadır.

Mevcut saldırı tespit ve engelleme sistemleri, imza tabanlı geleneksel yöntemler kullanarak saldırıları engellemeye çalışmaktadır. İmza tabanlı siber savunma sistemleri sıfır/sıfırıncı gün saldırılarılarına karşı savunmasız kalmaktadır. İmza güncellemelerinin sürekli olarak takip edilmesi ve imza veritabanının güncel halde tutulması gerekmektedir. Aksi halde yeni çıkmış saldırılara karşı savunmasız kalınmaktadır. Ayrıca, imza tabanlı yöntemlerle şifrelenmiş saldırılar tespit edilememektedir. Tüm bu olumsuzluklara ek olarak saldırı türü ve sayısının sürekli olarak artması da göz önünde bulundurulduğunda, makine öğrenmesi ve derin öğrenme gibi kendi kendine öğrenen ve karar verebilen teknolojilerin saldırı tespiti için kullanılması gerekmektedir. Özet olarak hem günümüzdeki ağ altyapılarında hem de gelecekte kullanılacak YTA tabanlı ağ altyapılarında geleneksel siber güvenlik yaklaşımlarının kullanılıyor olması çözülmesi gereken önemli bir problemdir.

Saldırı çeşitleri ve yöntemleri sürekli olarak artmaktadır. Bu dinamikliğe ayak uyduracak, ağ ve güvenlik altyapısını içinde bulunduğu şartlara göre otonom olarak güncelleyecek teknolojilere ihtiyaç duyulmaktadır. Saldırı tespiti ve engelleme bakış açısı, statik yaklaşımlardan dinamik yaklaşımlara geçirilmelidir. Hâlihazırda sunulan siber savunma sistemlerini oluşturan donanım ve yazılımlar statik bir yapı ile siber koruma oluşturmaya

çalışmaktadır. Tüm trafik türleri için tek bir cihaz/ürün ile siber savunma yapılmaktadır. Hâlbuki saldırının oluşturduğu trafik türlerine göre ayrı cihaz/ürünler ile çok daha başarılı siber savunma yapılabilir. Örneğin, DNS sunuculara yönelik birçok saldırı türü bulunmaktadır. Sadece DNS saldırılarını engellemeye yönelik ürünlere ihtiyaç duyulmaktadır. Her bir uygulama düzlemi servisine yönelik birbirinden bağımsız olarak çalışabilen otonom olarak oluşturulup kaldırılabilen savunma yaklaşımı ile bu problemler çözülebilir. Özet olarak, trafik farkındalığı olmadan tüm saldırı tiplerine tek bir siber savunma sistemi ile durağan olarak koruma sağlanması günümüzde bile yetersiz kalmaktadır. Saldırı türleri ve sayısının sürekli artmasının yanı sıra YTA altyapısının güvenlik ihtiyaçları da göz önünde bulundurulduğunda, mevcut yaklaşımlar gelecekte yetersiz kalacaktır.

Günümüz siber savunma sistemleri; DDoS gibi hacimsel şekilde yüksek trafikler oluşturarak servislerin kesintiye uğramasına neden olan saldırılara karşı yetersiz kalmaktadır. DDoS saldırılarının doğası gereği gerçek trafiklerden ayırt edilmesi oldukça zordur ve %100 bir başarı elde edilememektedir. Ancak, ağın dayanıklılık ve güvenilirlik seviyesi arttırılarak yüksek trafik yükü altında çalışabilme süresi artırılabilir. Günümüzde DDoS saldırılarını engellemeye yönelik yaklaşımlar, trafik eşik değerine göre ilgili trafiği engelleyebilecek çözümler sunmaktadır. Örneğin, belirli süre içerisinde belirlenen eşik değerinden fazla trafik gelmesi durumunda trafik engellenmektedir. Bu yaklaşım gerçekten yoğun trafiğin oluştuğu şartlarda yanlış alarm üreterek gerçek trafiği de engellemektedir. Özet olarak halihazırda sunulan çözümler, üretilen sahte trafikler ya da gerçekten oluşan yoğun ağ trafiği altında hizmet veremez hale gelmektedir. Günümüzdeki siber savunma sistemlerinin ağ dayanıklılığı ve ağ güvenilirliği yeterli seviyede değildir.

Mevcut YTA kontrolcüsünün yönetimi; bir ağ yöneticisi tarafından el ile yapılmaktadır. Ağ yöneticisinin girdiği direktif ve konfigürasyonlara göre YTA kontrolcü cihazları çalışmaktadır. Bu yönetim anlayışından dolayı ağ yöneticisinin gözden kaçırdığı ya da teknik bilgi yetersizliğinden kaynaklı problemler ortaya çıkmaktadır. Bulunduğu şartlara göre kendi kendine konfigürasyon yapabilen, ağdaki diğer cihazlara direktif gönderebilen insan kaynaklı hataları en aza indiren yönetim anlayışına gerek duyulmaktadır. Hem geleneksel ağ yönetiminde hem de YTA tabanlı ağ yönetiminde böyle bir yaklaşım bulunmamaktadır.

Bu çalışmada, Yazılım Tanımlı Ağlarda Ağ Trafiğine Duyarlı Bir Yaklaşım ile Otonom Saldırı Tespit ve Önleme Modeli önerilmiştir. Önerilen model, YTA kontrolcüsü ile eş zamanlı çalışan TSS ve EGS modüllerini içermektedir. TSS modülü ile trafik sınıflandırma ve süreklilik ölçümü işlemleri gerçekleştirilirken, EGS modülü ile saldırı tespit ve engelleme işlemleri gerçekleştirilir. EGS modülü içerisinde otonom olarak açılıp kapanma özelliğine sahip farklı siber güvenlik alanları yer almaktadır. Siber güvenlik alanları içerisinde; saldırı tespit ve engelleme işlemleri için eğitilmiş makine öğrenme modelleri kullanılmaktadır. Geliştirilen uygulama kullanılarak farklı senaryoların testleri gerçekleştirilmiştir. Senaryo kapsamında, YTA uygulama katmanında çalışan Web ve DNS sunucularını hedef alan hacimsel saldırılar (HTTP, DNS ve ICMP flood saldırısı) gerçekleştirilmiştir. Önerilen modelin kullanıldığı güvenlik sistemi devreye alındıktan sonra ağ trafik gecikmeleri ve sistem kaynak kullanımları ölçülmüştür. Elde edilen test sonuçları anormallik tespitinde, önerilen modelin başarılı sonuçlar verdiğini göstermektedir.

Bu tez kapsamında geliştirilen çözümler şu şekilde özetlenebilir:

- Hâlihazırda sunulan siber savunma sistemlerini oluşturan donanım ve yazılımlar statik bir yapı ile siber koruma oluşturmaya çalışmaktadır. Tüm trafik türleri için tek bir cihaz/ürün ile siber savunma yapılmaktadır. Bu tez çalışmasında; trafik farkında ve otonom çalışma yeteneklerine sahip bir model önerisinde bulunularak, EGS Teknoloji olarak adlandırılan bir arayüz uygulaması geliştirilmiştir. TSS modülü ile ağ trafik miktarı ve süreklilik süresi belirlenen eşik değerine ulaşan her bir trafik türü için otonom olarak bir Güvenlik Duvarı ve Saldırı Engelleme Sistemi oluşturulmaktadır. Bu sayede, şartları sağlayan her bir trafik türü için birbirinden bağımsız olarak çalışabilen sanal siber savunma sistemleri oluşturulmaktadır. Her bir trafik türü kendisi için özelleştirilmiş siber savunma sistemleri tarafından detaylı olarak incelenmektedir. TSS modülü sayesinde, saldırılar daha kolay ve hızlı şekilde, yüksek başarı oranlarıyla engellenebilmektedir. Saldırı türlerinin ve saldırı sayısının sürekli artmasının yanı sıra YTA altyapısının güvenlik ihtiyaçları da göz önünde bulundurulduğunda, TSS teknolojisi ihtiyaç duyulan dinamikliği sağlayarak derinlemesine trafik analizi yapabilmeyi mümkün hale getirmektedir.

- TSS modülü ile ağda heterojen olarak akan trafik, port ve protokol bilgileri kullanılarak sınıflandırılmaktadır. Sınıflandırılmış trafik, EGS modülüne yönlendirilerek belirli şartları sağlaması durumunda otonom olarak ilgili trafik için siber savuma sistemi oluşturulmakta ve trafik analiz edilmeye başlanmaktadır. Böylece, yoğun trafik altında ağın daha uzun süre sorunsuzca çalışabilmesi mümkün olmaktadır. Dolayısıyla ağın dayanıklılık seviyesi arttırılmaktadır.
- TSS modülü ile ağdaki bir trafik türünün yoğun olması sadece ilgili trafik için oluşturulan sanal savunma sistemini etkileyecektir. Diğer trafik türleri için oluşturulan sanal savunma sistemleri bu yoğunluktan etkilenmeyecektir. Sadece yoğun trafiğe maruz kalan savunma sistemi çalışamaz hale geldiğinde, sistemde genel bir hizmet kesintisi olmayacaktır. Dolayısıyla, ağın güvenilirliği yüksek oranlarda arttırılmaktadır.
- EGS modülü üzerinde farklı ağ trafik türleri için oluşturulmuş siber güvenlik alanları yer almaktadır. TSS modülünden gelen tetikleyici sinyal ile açılan siber güvenlik alanları görevi bittiğinde otomatik olarak kapanmaktadır. Böylece sistem kaynakları etkin ve verimli bir şekilde kullanılmaktadır.
- YTA' ların yönetimi ağ yöneticisinin yaptığı yapılandırma ayarlarına göre çalışmaktadır. Dolayısıyla, insan kaynaklı yapılan hatalardan dolayı zaman zaman problemler oluşmaktadır. Bu çalışmada, ağ yöneticisine sadece onay amaçlı ihtiyaç duyulmaktadır. Ağ yöneticisi geliştirilen uygulama arayüzü üzerinden sistem korumasını aktif ettiğinde YTA kontrolcüsü güvenlik ile ilgili kararları otonom olarak alıp tüm YTA bileşenlerine uygulayabilmektedir.

Tez çalışması süresince karşılaşılan zorluklar şu şekilde özetlenmiştir;

- YTA kontrolcüsü ile birlikte çalışacak farklı yeteneklere sahip modül ve fonksiyonların nasıl geliştirilebileceğine yönelik yeterli sayıda detaylı teknik dokümanın mevcut olmaması uygulama geliştirme sürecinin uzamasına neden olmuştur.
- YTA güvenlik uygulamaları test süreçleri için yeterli sayıda veri kümesi ve test ortamı mevcut değildir. Bu nedenle geliştirilen uygulamanın test işlemleri için uygun veri kümesi ve test ortamı oluşturabilmek için ek çalışmalar yapılması gerekmiştir.
- Uygulama geliştirme sürecinde, sistem test aşamaları gerçekleştirilirken sistem kaynakları yetersiz kalmış ve sistem bellek kapasitesinin yükseltilmesi gerekmiştir.

Bu çalışmada, YTA uygulama düzleminde çalışan sunucuları hedef alan hacimsel saldırılar dikkate alınmıştır. Sonraki çalışmalarda, hedef sunucu sayısı arttırılarak saldırı türlerinin çeşitlendirilmesi sağlanabilir. Bunlara ek olarak; kontrolcü sayısının arttırılmasının koruma yeteneği üzerine etkisi araştırılabilir.

KAYNAKLAR

1. Kreutz, D., Ramos, F., and Verissimo, P. (2013). Towards secure and dependable software-defined network. *In HotSDN' 13: Proceedings of the second ACM SIGCOMM workshop on Hot topics in software defined networking*, 13(2), 55-60.

2. İnternet: Grand View Research. (2017). Software Defined Networking Market Size, Share & Trends Analysis Report By End-Use, By Service, By Solution, By Application, By Region, And Segment Forecasts, 2018 - 2024. *GVR*. San Francisco, 35-36. Web: https://www.grandviewresearch.com/industry-analysis/software-defined-networking-sdn-market-analysis, Son Erişim Tarihi: 09.06.2020.

3. Cui, Y., Yan, L., Li, S., Xing, H., Pan, W., Zhu, J., and Zheng, X. (2016). SD-Anti-DDoS: Fast and efficient DDoS defense in software-defined networks. *Journal Of Network And Computer Applications*, 68, 65-79.

4. Ng, B., Hayes, M., & Seah, W. (2015). *Developing a traffic classification platform for enterprise networks with SDN: Experiences and lessons learned.* 2015 IFIP Networking Conference (IFIP Networking), Toulouse, 1-9

5. Hawilo, H., Shami, A., Mirahmadi, M., and Asal, R. (2014). NFV: state of the art, challenges, and implementation in next generation mobile networks (vEPC). *IEEE Network*, 28(6), 18-26.

6. Mijumbi, R., Serrat, J., Gorricho, J., Bouten, N., De Turck, F., and Boutaba, R. (2016). Network Function Virtualization: State-of-the-Art and Research Challenges. *IEEE Communications Surveys & Tutorials*, 18(1), 236-262.

7. Yong Li, and Min Chen. (2015). Software-Defined Network Function Virtualization: A Survey. *IEEE Access*, *3*, 2542-2553.

8. Bonfim, M., Dias, K., & Fernandes, S. (2019). Integrated NFV/SDN Architectures. *ACM Computing Surveys*, 51(6), 1-39.

9. Tonkal, Ö., Polat, H., Başaran, E., Cömert, Z., and Kocaoğlu, R. (2021). Machine Learning Approach Equipped with Neighbourhood Component Analysis for DDoS Attack Detection in Software-Defined Networking. *Electronics*, 10 (11), 1227.

10. Kreutz, D., Ramos, F., Esteves Verissimo, P., Esteve Rothenberg, C., Azodolmolky, S., and Uhlig, S. (2015). Software-Defined Networking: A Comprehensive Survey. *Proceedings Of The IEEE*, 103(1), 14-76.

11. Shu, Z., Wan, J., Li, D., Lin, J., Vasilakos, A., and Imran, M. (2016). Security in Software-Defined Networking: Threats and Countermeasures. *Mobile Networks And Applications*, 21(5), 764-776.

12. Ahmad, I., Namal, S., Ylianttila, M., and Gurtov, A. (2015). Security in Software Defined Networks: A Survey. *IEEE Communications Surveys & Tutorials*, 17(4), 2317-2346.

13. Paliwal, M., Shrimankar, D., and Tembhurne, O. (2018). Controllers in SDN: A Review Report. *IEEE Access*, 6, 36256-36270.

14. Ahmad, S., and Mir, A. (2020). Scalability, Consistency, Reliability and Security in SDN Controllers: A Survey of Diverse SDN Controllers. *Journal Of Network And Systems Management*, 29(1),1-59

15. Sandhya, Sinha, Y., and Haribabu, K. (2017). A survey: Hybrid SDN. *Journal Of Network And Computer Applications*, 100, 35-55.

16. İnternet: *OpenFlow: Proactive vs Reactive Flows.* Web: http://networkstatic.net/openflow-proactive-vs-reactive-flows/, Son Erişim Tarihi: 09.02.2022.

17. Jarraya, Y., Madi, T., and Debbabi, M. (2014). A Survey and a Layered Taxonomy of Software-Defined Networking. *IEEE Communications Surveys; Tutorials*, 16(4), 1955-1980.

18. İnternet: ONF. *OpenFlow Switch Specification*, v1.5.1. Web: https://www.opennetworking.org/wp-content/uploads/2014/10/openflow-switchv1.5.1.pdf, Son Erişim Tarihi: 28.02.2022

19. Foster, N., Harrison, R., Freedman, M., Monsanto, C., Rexford, J., Story, A., and Walker, D. (2011). Frenetic. *ACM SIGPLAN Notices*, 46(9), 279-291.

20. İnternet*: Mininet: An Instant Virtual Network on your Laptop (or other PC).* Web: http://mininet.org/, Son Erişim Tarihi: 28.02.2022.

21. İnternet: Network Simulator, NS3. Web:https://www.nsnam.org, Son Erişim Tarihi: 28.02.2022.

22. İnternet: Graphical Network Simulator, GNS3. Web:https://www.gns3.com/, Son Erişim Tarihi: 28.02.2022.

23. İnternet: Vagrant. Web: https://www.vagrantup.com/, Son Erişim Tarihi: 28.02.2022.

24. İnternet: VIRL. Web: https://learningnetwork.cisco.com/s/virl, Son Erişim Tarihi: 28.02.2022.

25. Chiosi, M., Clarke, D., Willis, P., Reid, A., Feger, J., Bugenhagen, M., Khan, W., Fargano, M., Cui, C., Deng, H., Telekom, D. and Michel, U. (2012). *Network Functions Virtualisation: An introduction, benefits, enablers, challenges and call for action.* SDN and OpenFlow World Congress, Darmstadt, Germany.

26. İnternet: Network Functions Virtualisation (NFV) architectural framework. Web: https://www.etsi.org/deliver/etsi_gs/NFV/001_099/002/01.02.01_60/gs_NFV002v010%0A201p.pdf , Son Erişim Tarihi: 28.02.2022.

27. Szabo, R., Kind, M., Westphal, F., Woesner, H., Jocha, D., and Csaszar, A. (2015). Elastic network functions: opportunities and challenges. *IEEE Network*, 29(3), 15-21.

28. Yi, B., Wang, X., Li, K., Das, S., and Huang, M. (2018). A comprehensive survey of Network Function Virtualization. *Computer Networks*, 133, 212-262.

29. Vilalta, R., Mayoral, A., Munoz, R., Casellas, R., and Martinez, R. (2015). *The SDN/NFV Cloud Computing platform and transport network of the ADRENALINE testbed.* Proceedings Of The 2015 1St IEEE Conference On Network Softwarization (Netsoft), Londra, 1-5.

30. Lv, Z., and Xiu, W. (2020). Interaction of Edge-Cloud Computing Based on SDN and NFV for Next Generation IoT. *IEEE Internet Of Things Journal*, 7(7), 5706-5712.

31. Shafi, M., Molisch, A., Smith, P., Haustein, T., Zhu, P., and De Silva, P. et al. (2017). 5G: A Tutorial Overview of Standards, Trials, Challenges, Deployment, and Practice. *IEEE Journal On Selected Areas In Communications*, 35(6), 1201-1221.

32. Panwar, N., Sharma, S., and Singh, A. (2016). A survey on 5G: The next generation of mobile communication. *Physical Communication*, 18, 64-84.

33. Nguyen, V., Brunstrom, A., Grinnemo, K., and Taheri, J. (2017). SDN/NFV-Based Mobile Packet Core Network Architectures: A Survey. *IEEE Communications Surveys &Amp; Tutorials*, 19(3), 1567-1602.

34. Bouras, C., Ntarzanos, P., & Papazois, A. (2016). *Cost modeling for SDN/NFV based mobile 5G networks.* 2016 8Th International Congress On Ultra Modern Telecommunications And Control Systems And Workshops (ICUMT), Lisbon ,56-61

35. Ordonez-Lucena, J., Ameigeiras, P., Lopez, D., Ramos-Munoz, J., Lorca, J., and Folgueira, J. (2017). Network Slicing for 5G with SDN/NFV: Concepts, Architectures, and Challenges. *IEEE Communications Magazine*, 55(5), 80-87.

36. Souza, R., Dias, K., and Fernandes, S. (2020). NFV Data Centers: A Systematic Review. *IEEE Access*, 8, 51713-51735.

37. Heller, B., Seetharaman, S., Mahadevan, P., Yiakoumis, Y., Sharma, P., Banerjee, S. and McKeown, N. (2010). *ElasticTree: Saving Energy in Data Center Networks. USENIX Symposium on Networkes Systems Design and Implementation*, California,249-264

38. Hakiri, A., Gokhale, A., Berthou, P., Schmidt, D., and Gayraud, T. (2014). Software-Defined Networking: Challenges and research opportunities for Future Internet. *Computer Networks*, 75, 453-471.

39. Alam, I., Sharif, K., Li, F., Latif, Z., Karim, M., and Biswas, S. et al. (2021). A Survey of Network Virtualization Techniques for Internet of Things Using SDN and NFV. *ACM Computing Surveys*, 53(2), 1-40.

40. Bizanis, N., and Kuipers, F. (2016). SDN and Virtualization Solutions for the Internet of Things: A Survey. *IEEE Access*, 4, 5591-5606.

41. Islam, M. J., Rahman, A., Kabir, S., Khatun, A., Pritom, A. I., and Zaman, M. (2020). SDoT-NFV: Enhancing a distributed SDN-IoT architecture security with NFV implementation for smart city. *Dept. Comput. Sci. Eng., Green Univ. Bangladesh, Dhaka, Bangladesh, Tech. Rep. 2020A3321*.

42. Rahman, A., Islam, M., Sunny, F., and Nasir, M. (2019). *DistBlockSDN: A Distributed Secure Blockchain Based SDN-IoT Architecture with NFV Implementation for Smart Cities.* 2019 2Nd International Conference On Innovation In Engineering And Technology (ICIET), Bangladesh,1-6

43. Samuel, A. (1959). Some Studies in Machine Learning Using the Game of Checkers. *IBM Journal Of Research And Development*, 3(3), 210-229.

44. Khan, M., Bashier, E., and Mohammed, M. (2016). *Machine Learning: Algorithms and Applications* (First edition). Boca Raton: CRC Press, 7-11.

45. Lundberg, S., and Lee, S. (2017). A Unified Approach to Interpreting Model Predictions. In *NIPS'17: Proceedings of the 31st International Conference on Neural Information Processing Systems*. Red Hook, NY, United States: Curran Associates Inc., 1-10.

46. Chandrashekar, G., and Sahin, F. (2014). A survey on feature selection methods. *Computers &Amp; Electrical Engineering*, 40(1), 16-28.

47. Saeys, Y., Inza, I., and Larranaga, P. (2007). A review of feature selection techniques in bioinformatics. *Bioinformatics*, 23(19), 2507-2517.

48. Guyon, I., and Elisseeff, A. (2003). An Introduction to Variable and Feature Selection. *Journal Of Machine Learning Research 3*, 3, 1157-1182.

49. Hall, M., & Holmes, G. (2003). Benchmarking attribute selection techniques for discrete class data mining. *IEEE Transactions On Knowledge And Data Engineering*, 15(6), 1437-1447.

50. Xie, J., Yu, F., Huang, T., Xie, R., Liu, J., Wang, C., & Liu, Y. (2019). A Survey of Machine Learning Techniques Applied to Software Defined Networking (SDN): Research Issues and Challenges. *IEEE Communications Surveys &Amp; Tutorials*, 21(1), 393-430.

51. Nguyen, T., and Armitage, G. (2008). A survey of techniques for internet traffic classification using machine learning. *IEEE Communications Surveys &Amp; Tutorials*, 10(4), 56-76.

52. Callado, A., Kamienski, C., Szabo, G., Gero, B., Kelner, J., Fernandes, S., and Sadok, D. (2009). A Survey on Internet Traffic Identification. *IEEE Communications Surveys & Tutorials*, 11(3), 37-52.

53. Steinke, S. (2003). Simple Network Management Protocol-SNMP. In *Network Tutorial*. CRC Press, 495-498.

54. Kurt Ç. (2020). *Yazılım Tanımlı Ağlarda Telemetrik Yöntemle Gerçek Zamanlı Saldırı Tespiti ve Önleme*, Doktora Tezi, Gazi Üniversitesi Fen Bilimleri Enstitüsü, Ankara, 1-10.

55. İnternet: IANA, *Internet Assigned Numbers Authority*. Web: https://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xhtml, Son Erişim Tarihi: 28.02.2022.

56. Dehghani, F., Movahhedinia, N., Khayyambashi, M., and Kianian, S. (2010). *Real-Time Traffic Classification Based on Statistical and Payload Content Features.* 2010 2Nd International Workshop On Intelligent Systems And Applications, Wuhan, 1-4

57. Carela Español, V. (2014). *Network traffic classification: from theory to practice*. Doktora Tezi, Universitat Polit`ecnica de Catalunya BarcelonaTech Department d'Arquitectura de Computadors, Barcelona, 1-170.

58. Mestres, A., Rodriguez-Natal, A., Carner, J., Barlet-Ros, P., Alarcón, E., and Solé, M. et al. (2017). Knowledge-Defined Networking. *ACM SIGCOMM Computer Communication Review*, 47(3), 2-10.

59. Wang, P., Lin, S., and Luo, M. (2016). *A Framework for QoS-aware Traffic Classification Using Semi-supervised Machine Learning in SDNs. 2*016 IEEE International Conference On Services Computing (SCC), San Francisco,760-765

60. Rossi, D., and Valenti, S. (2010). *Fine-grained traffic classification with netflow data.* Proceedings Of The 6Th International Wireless Communications And Mobile Computing Conference On ZZZ - IWCMC'10, New York, 479-483

61. He, L., Xu, C., & Luo, Y. (2016). *vTC.* Proceedings Of The 2016 ACM International Workshop On Security In Software Defined Networks & Network Function Virtualization, New York ,53-56

62. Amaral, P., Dinis, J., Pinto, P., Bernardo, L., Tavares, J., and Mamede, H. (2016). *Machine Learning in Software Defined Networks: Data collection and traffic classification. 2*016 IEEE 24Th International Conference On Network Protocols (ICNP), Singapore ,1-5

63. Wang, P., Ye, F., Chen, X., and Qian, Y. (2018). Datanet: Deep Learning Based Encrypted Network Traffic Classification in SDN Home Gateway. *IEEE Access*, 6, 55380-55391.

64. Lim, H., Kim, J., Kim, K., Hong, Y., and Han, Y. (2019). Payload-Based Traffic Classification Using Multi-Layer LSTM in Software Defined Networks. *Applied Sciences*, 9(12), 2550.

65. Raikar, M., S M, M., Mulla, M., Shetti, N., and Karanandi, M. (2020). Data Traffic Classification in Software Defined Networks (SDN) using supervised-learning. *Procedia Computer Science*, 171, 2750-2759.

66. Schehlmann, L., Abt, S., and Baier, H. (2014). *Blessing or curse? Revisiting security aspects of Software-Defined Networking.* 10Th International Conference On Network And Service Management (CNSM) And Workshop, Rio de Janerio, 382-387

67. Priyadarsini, M., and Bera, P. (2021). Software defined networking architecture, traffic management, security, and placement: A survey. *Computer Networks*, 192, 108047.

68. Li, X., and Xie, W. (2017). *CRAFT: A Cache Reduction Architecture for Flow Tables in Software-Defined Networks.* 2017 IEEE Symposium On Computers And Communications (ISCC), Heraklion, 967-972

69. Correa Chica, J., Imbachi, J., and Botero Vega, J. (2020). Security in SDN: A comprehensive survey. *Journal Of Network And Computer Applications*, 159, 102595.

70. Han, T., Jan, S., Tan, Z., Usman, M., Jan, M., Khan, R., & Xu, Y. (2019). A comprehensive survey of security threats and their mitigation techniques for next-generation SDN controllers. *Concurrency And Computation: Practice And Experience*, 32(16).

71. Wang, S., Chavez, K., and Kandeepan, S. (2017). SECO: SDN sEcure COntroller algorithm for detecting and defending denial of service attacks. *2017 5Th International Conference On Information And Communication Technology (Icoic7)*, Melaka, 1-6

72. Li, H., Li, P., Guo, S., and Nayak, A. (2014). Byzantine-Resilient Secure Software-Defined Networks with Multiple Controllers in Cloud. *IEEE Transactions On Cloud Computing*, 2(4), 436-447.

73. Li, W., Meng, W., and Kwok, L. (2016). A survey on OpenFlow-based Software Defined Networks: Security challenges and countermeasures. *Journal Of Network And Computer Applications*, 68, 126-139.

74. Kaur, S., Kumar, K., Aggarwal, N., and Singh, G. (2021). A comprehensive survey of DDoS defense solutions in SDN: Taxonomy, research challenges, and future directions. *Computers & Security*, 110, 102423.

75. Yoon, C., Lee, S., Kang, H., Park, T., Shin, S., and Yegneswaran, V. et al. (2017). Flow Wars: Systemizing the Attack Surface and Defenses in Software-Defined Networks. *IEEE/ACM Transactions On Networking*, 25(6), 3514-3530.

76. Dayal, N., Maity, P., Srivastava, S., and Khondoker, R. (2016). Research Trends in Security and DDoS in SDN. *Security And Communication Networks*, 9(18), 6386-6411.

77. Ruffy, F., Hommel, W., and Eye, F. (2016). *A STRIDE-based Security Architecture for Software-Defined Networking.* In The International Symposium on Advances in Software Defined Networks. Lisbon, Portugal: IARIA, 95-101.

78. Rawat, D., and Reddy, S. (2017). Software Defined Networking Architecture, Security and Energy Efficiency: A Survey. *IEEE Communications Surveys & Tutorials*, 19(1), 325-346.

79. Ranjbar, A., Komu, M., Salmela, P., & Aura, T. (2016). *An SDN-based approach to enhance the end-to-end security: SSL/TLS case study*. NOMS 2016 - 2016 IEEE/IFIP Network Operations And Management Symposium, İstanbul, 281-288

80. Akhunzada, A., Gani, A., Anuar, N., Abdelaziz, A., Khan, M., Hayat, A., and Khan, S. (2016). Secure and dependable software defined networks. *Journal Of Network And Computer Applications*, 61, 199-221.

81. Kreutz, D., Ramos, F., Esteves Verissimo, P., Esteve Rothenberg, C., Azodolmolky, S., and Uhlig, S. (2015). Software-Defined Networking: A Comprehensive Survey. *Proceedings Of The IEEE*, 103(1), 14-76.

82. Scott-Hayward, S., Natarajan, S., and Sezer, S. (2016). A Survey of Security in Software Defined Networks. *IEEE Communications Surveys &Amp; Tutorials*, 18(1), 623-654.

83. Yurekten, O., and Demirci, M. (2021). SDN-based cyber defense: A survey. *Future Generation Computer Systems*, 115, 126-149.

84. Mousavi, S., and St-Hilaire, M. (2015). E*arly detection of DDoS attacks against SDN controllers*. 2015 International Conference On Computing, Networking And Communications (ICNC), Garden Grove, 77-81

85. Sahoo, K., Puthal, D., Tiwary, M., Rodrigues, J., Sahoo, B., and Dash, R. (2018). An early detection of low rate DDoS attack to SDN based data center networks using information distance metrics. *Future Generation Computer Systems*, 89, 685-697.

86. Kumar, P., Tripathi, M., Nehra, A., Conti, M., and Lal, C. (2018). SAFETY: Early Detection and Mitigation of TCP SYN Flood Utilizing Entropy in SDN. *IEEE Transactions On Network And Service Management*, 15(4), 1545-1559.

87. Shin, S., Yegneswaran, V., Porras, P., and Gu, G. (2013). AVANT-GUARD: Scalable and vigilant switch flow management in software-defined networks. *ACM SIGSAC Conference on Computer and Communications Security (CCS 2013)*, 413–424.

88. Fichera, S., Galluccio, L., Grancagnolo, S. C., Morabito, G., and Palazzo, S. (2015). OPERETTA: An Openflow-based remedy to mitigate TCP SYNFLOOD attacks against web servers. *Computer Networks*, 92, 89–100.

89. Braga, R., Mota, E., and Passito, A. (2010). L*ightweight DDoS flooding attack detection using NOX/OpenFlow*. IEEE Local Computer Network Conference, Denver, 408-415.

90. Sathya, R., and Thangarajan, R. (2015). Efficient anomaly detection and mitigation in software defined networking environment. *2015 2Nd International Conference On Electronics And Communication Systems (ICECS)*, 2, 479–484.

91. Mirjalili, S., and Yang, X. (2013). Binary bat algorithm. *Neural Computing And Applications*, 25(3-4), 663-681.

92. Tavallaee, M., Bagheri, E., Lu, W., and Ghorbani, A. (2009). *A detailed analysis of the KDD CUP 99 data set*. 2009 IEEE Symposium On Computational Intelligence For Security And Defense Applications, Ottowa, 1-6

93. Bhargava, N., Sharma, G., Bhargava, R., and Mathuria, M. (2013). Decision Tree Analysis on J48 Algorithm for Data Mining. *International Journal Of Advanced Research In Computer Science And Software Engineering*, 3(6), 1114-1119.

94. Perez-Diaz, J., Valdovinos, I., Choo, K., & Zhu, D. (2020). A Flexible SDN-Based Architecture for Identifying and Mitigating Low-Rate DDoS Attacks Using Machine Learning. *IEEE Access*, 8, 155859-155872.

95. Kyaw, A., Zin Oo, M., & Khin, C. (2020). Machine-Learning Based DDOS Attack Classifier in Software Defined Network. *2020 17Th International Conference On Electrical Engineering/Electronics, Computer, Telecommunications And Information Technology (ECTI-CON)*, 431-434.

96. Janarthanam, S., Prakash, N., & Shanthakumar, M. (2020). Adaptive Learning Method for DDoS Attacks on Software Defined Network Function Virtualization. *EAI Endorsed Transactions On Cloud Systems*, 6(18), 166286.

97. Sahoo, K., Tripathy, B., Naik, K., Ramasubbareddy, S., Balusamy, B., Khari, M., & Burgos, D. (2020). An Evolutionary SVM Model for DDOS Attack Detection in Software Defined Networks. *IEEE Access*, 8, 132502-132513.

98. Cui, J., He, J., Xu, Y., and Zhong, H. (2018). TDDAD: Time-based detection and defense scheme against DDoS attack on SDN controller. *In Australasian Conference on Information Security and Privacy, Springer, Cham*. 10946, 649–665.

99. Li, C., Wu, Y., Yuan, X., Sun, Z., Wang, W., Li, X., and Gong, L. (2018). Detection and defense of DDoS attack-based on deep learning in OpenFlow-based SDN. *International Journal Of Communication Systems*, 31(5), e3497.

100. Singh, R., Tanwar, S., and Sharma, T. (2019). Utilization of blockchain for mitigating the distributed denial of service attacks. *Security And Privacy*, 3(3).

101. Abou El Houda, Z., Hafid, A., and Khoukhi, L. (2019). Cochain-SC: An Intra- and Inter-Domain Ddos Mitigation Scheme Based on Blockchain Using SDN and Smart Contract. *IEEE Access*, 7, 98893-98907.

102. Rodrigues, B., Bocek, T., Lareida, A., Hausheer, D., Rafati, S., and Stiller, B. (2017). *A Blockchain-Based Architecture for Collaborative DDoS Mitigation with Smart Contracts*. Lecture Notes In Computer Science, Zurih, 16-29.

103. El Houda, Z., Hafid, A., and Khoukhi, L. (2019). *Co-IoT: A Collaborative DDoS Mitigation Scheme in IoT Environment Based on Blockchain Using SDN*. 2019 IEEE Global Communications Conference (GLOBECOM), Waikoloa, 1-6.

GAZİ GELECEKTİR...