
GAZİ ÜNİVERSİTESİ
Gazi
1926

WEB TABANLI SALDIRI ÖNLEME AMAÇLI YENİ BİR GERÇEK ZAMANLI WEB UYGULAMASI GÜVENLİK DUVARI ALGORİTMASININ GERÇEKLEŞTİRİLMESİ

Adem TEKEREK

DOKTORA TEZİ
ELEKTRONİK VE BİLGİSAYAR EĞİTİMİ ANABİLİM DALI

GAZİ ÜNİVERSİTESİ
BİLİŞİM ENSTİTÜSÜ

KASIM 2016

Adem TEKEREK tarafından hazırlanan "Web Tabanlı Saldırı Önleme Amaçlı Yeni Bir Gerçek Zamanlı Web Uygulaması Güvenlik Duvarı Algoritmasının Gerçekleştirilmesi" adlı tez çalışması aşağıdaki jüri tarafından OY BİRLİĞİ ile Gazi Üniversitesi Elektronik ve Bilgisayar Eğitimi Anabilim Dalında DOKTORA TEZİ olarak kabul edilmiştir.

Danışman: Prof. Dr. Ömer Faruk BAY

Elektrik Elektronik Mühendisliği Anabilim Dalı, Gazi Üniversitesi

Bu tezin, kapsam ve kalite olarak Doktora Tezi olduğunu onaylıyorum ……

Başkan: Doç. Dr. Ertan ONUR

Bilgisayar Mühendisliği Anabilim Dalı, Orta Doğu Teknik Üniversitesi

Bu tezin, kapsam ve kalite olarak Doktora Tezi olduğunu onaylıyorum ……

Üye: Doç Dr. Suat ÖZDEMİR

Bilgisayar Mühendisliği Anabilim Dalı, Gazi Üniversitesi

Bu tezin, kapsam ve kalite olarak Doktora Tezi olduğunu onaylıyorum ……

Üye: Yrd. Doç. Dr. Murat AYDOS

Bilgisayar Mühendisliği Anabilim Dalı, Hacettepe Üniversitesi

Bu tezin, kapsam ve kalite olarak Doktora Tezi olduğunu onaylıyorum ……

Üye: Yrd. Doç. Dr. Cemal KOÇAK

Bilgisayar Mühendisliği Anabilim Dalı, Gazi Üniversitesi

Bu tezin, kapsam ve kalite olarak Doktora Tezi olduğunu onaylıyorum ……

Tez Savunma Tarihi: 10.11.2016

Jüri tarafından kabul edilen bu tezin Doktora Tezi olması için gerekli şartları yerine getirdiğini onaylıyorum.

……

Doç. Dr. Bünyamin CİYLAN

Bilişim Enstitüsü Müdürü

ETİK BEYAN

Gazi Üniversitesi Bilişim 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.

Adem TEKEREK

10.11.2016

WEB TABANLI SALDIRI ÖNLEME AMAÇLI YENİ BİR GERÇEK ZAMANLI WEB UYGULAMASI GÜVENLİK DUVARI ALGORİTMASININ GERÇEKLEŞTİRİLMESİ
(Doktora Tezi)

Adem TEKEREK

GAZİ ÜNİVERSİTESİ
BİLİŞİM ENSTİTÜSÜ
Kasım 2016

ÖZET

İnternet üzerinden verilen hizmetlerin artması, günlük hayatta internetin kullanım oranını artırmıştır. İnternetin kullanım oranının artması internet uygulamalarına yönelik tehditlerin de aynı ölçüde artmasına sebep olmuştur. İnternette verilen hizmetlerin altyapısını büyük oranda web uygulamaları oluşturmaktadır. İnterneti hedef alan tehditler, web uygulamalarının güvenlik ihtiyaçlarını ortaya çıkarmıştır. Web uygulama güvenliğini sağlamak için ağ ortamından yapılan saldırılara karşı güvenlik duvarları, saldırı tespit ve engelleme sistemleri kullanılmaktadır. Web uygulamaları ve web tabanlı servislere karşı yapılan saldırılar, web uygulamalarının iletişim protokolü olarak kullandığı Hyper-Text Transfer Protocol (HTTP) kullanılarak da yapılmaktadır ve HTTP denetimi yapılarak web tabanlı saldırılar engellenebilmektedir. Geliştirilen her bir web uygulamasının yapısı birbirinden farklı mimariye sahip olmaktadır. Web uygulamalarının bu değişik mimari yapısı ve çok fazla değişkene sahip olması HTTP'nin içerik yapısını da değiştirmektedir. Bu durum HTTP denetimi yapılarak web tabanlı saldırıların denetimini ve engellenmesini zorlaştırmaktadır. Bu tez çalışmasında, HTTP denetimi yapılarak web tabanlı saldırıların denetimini gerçekleştirmek için, İmza Tabanlı Denetim (İTD) ve Anormal Tabanlı Denetim (ATD) metodları kullanılarak öğrenme tabanlı, hibrit bir Web Uygulama Güvenlik Duvarı (WUGD) modeli önerilmiştir. İTD ile, bilinen web tabanlı saldırı türlerinden olan SQL (Structured Query Language) Enjeksiyonu, Siteler Arası Kod (XSS) Yazma, Komut Enjeksiyonu (KE) ve Dizin Geçişi Saldırı (DGS) saldırı türlerine karşı imza denetimi gerçekleştirilmiştir. ATD ile ise standart HTTP istek yapısına uymayan HTTP isteklerinin denetimi yapılmıştır. ATD, Alfanümerik Karakter Analizi (AKA), Harf Frekans Analizi (HFA) ve İstek Uzunluk Analizi (İUA) öznitelikleri ile, yapay zekâ tekniklerinden sinir ağları yöntemi kullanılarak öğrenme tabanlı olarak gerçekleştirilmiştir. ATD sonucunda tespit edilen HTTP istekleri tekrar web uygulamasına geldiği zaman ikinci defa ATD gerçekleştirmemek için İTD veritabanı güncellenmektedir. Böylece imza üretimi gerçekleştirilmektedir. Bu durum sistemin yeni saldırı türlerine karşı adaptasyonunu sağlamaktadır. Ayrıca İTD, ATD'ye göre daha hızlı çalıştığı için sistemin hız performansı da artırılmış olmaktadır. Normal olarak tespit edilen HTTP istekleri Normal İsteklerin İmza Tabanlı Denetimi (NİİTD) veritabanına, anormal olarak tespit edilen HTTP istekleri Anormal İsteklerin İmza Tabanlı Denetimi (AİİTD) veritabanına eklenmektedir. Böylece hız performansı yüksek bir modelin geliştirilmesi sağlanmıştır. Bu yüzden web tabanlı saldırıların denetimini en iyi şekilde yapabilmek için hibrit bir denetim modeli önerilmiştir. Önerilen model gerçek zamanlı, web uygulamaları kullanılarak akan HTTP trafik verisiyle gerçek zamanlı olarak ve literatürde üretilmiş farklı veri kümeleri kullanılarak test edilmiştir. Test sonuçları benzer veri kümelerini kullanan çalışmalarla karşılaştırılmıştır. Karşılaştırma sonuçlarına göre, önerilen modelin mevcut çalışmalara göre daha yüksek denetim performansı gösterdiği ve düşük yanlış pozitif oranına sahip olduğu görülmüştür. Ayrıca YSA ile elde edilen sonuçlarla karşılaştırma amacıyla ATD, veri madenciliği yöntemlerinden bayes sınıflandırma (BS) ve yapay zekâ yöntemlerinde bayes tabanlı yapay sinir ağları (BTYSA) yöntemleri ile de denetim yapılmıştır.

Bilim Kodu : 902.3.006
Anahtar Kelimeler : Web uygulama güvenlik duvarı, anormal-tabanlı denetim, imza-tabanlı denetim, yapay zekâ
Sayfa Adedi : 100
Danışman : Prof. Dr. Ömer Faruk BAY

IMPLEMANTATION OF A REAL TIME WEB-BASED INTRUSION PREVENTION AIMED WEB APPLICATION FIREWALL ALGORITM
(Ph.D. Thesis)

Adem TEKEREK

GAZİ UNIVERSITY
INSTITUTE OF INFORMATICS
November 2016

ABSTRACT

The increase of services provided through the internet has increased usage of internet in daily life. The increase in the ratio of the internet usage also causes an increase in the threats to internet applications equally at the same extent. The infrastructure of the services provided on the Internet comprises a major proportion of web applications. Threats targeted at internet reveal the security requirements of web applications. In order to ensure the security of web applications, firewalls, intrusion detection and prevention systems are used against attacks from the network layer. Attacks against to web applications and web-based service are also done by using Hyper-Text Transfer Protocol (HTTP) which the web applications use as a communication protocol, and the web based attacks can be prevented by HTTP detection. The structure of each developed web application has a different architecture from each other. This different architecture of web applications and to have too many variables also changes the structure of HTTP content. This situation complicates detection and prevention of web-based attacks by HTTP detection. In this thesis study, in order to detect web-based attacks by HTTP detection, a hybrid web application firewall (WAF) model that uses Signature Based Detection (SBD) and Anomaly Based Detection (ABD) methods is proposed. SBD carried out provides signature based detection against of known web based attack types, such as Structured Query Language (SQL) Injection, Cross Site Scripting (XSS), Command Injection and Directory traversal. On the other hand, ABD detects HTTP requests that do not match standard HTTP request structure. ABD was performed as learning based using artificial neural network, which is an artificial intelligence technique, by using the features, such as Alphanumeric Character Analysis, Letter Frequency Analysis and Request Length Analysis. When HTTP requests that was identified at the ABD results comes to web application again, in order not to perform ABD second time, SBD dataset is updated. Thus, signature generation was implemented. This situation provides the adaptation of the system against new types of attacks. Besides, because SBD runs faster than ABD, speed performance of the system is also enhanced. HTTP requests that was identified as normal requests and HTTP requests that was identified as anomaly requests are added to the Signature Based Detection of Normal Request (SBDNR) dataset and Anomaly Based Detection Anomaly Request (ABDAR) dataset, respectively. Thus, development of a model that has a high-speed performance is provided. Therefore, in order to detect web-based attacks properly, a hybrid detection model is proposed. The proposed model is tested on both several datasets produced in the literature and streaming HTTP data by using web applications in real time. Test results were compared with the studies that uses the same datasets. According to the comparison results, it is revealed that the proposed model shows a higher detection performance and low false positive rates than the existing studies. Furthermore, in order to compare the results yielded by ANN, ABD is also controlled with a data mining method, Bayesian Classifier, and an artificial intelligence method, Bayesian Neural Networks.

Science Code : 902.3.006
Key Words : Web application firewall, anomaly-based detection, signature-based detection, artificial intelligence
Page Number : 100
Supervisor : Prof. Dr. Ömer Faruk BAY

TEŞEKKÜR

Doktora ve Yüksek Lisans eğitimim boyunca önerileriyle, yönlendirmeleriyle ve yapıcı eleştirileriyle benden yardımlarını hiçbir zaman esirgemeyen danışman hocam sayın Prof. Dr. Ömer Faruk BAY'a teşekkürlerimi sunmayı bir borç bilirim.

Bu çalışmanın hazırlanması surecinde her aşamada deneyimlerini paylaşan, destek ve ilgisiyle çalışmaya yön veren değerli tez izleme komitesi hocalarım Yrd. Doç. Dr. Cemal KOÇAK'a ve Yrd. Doç. Dr. Murat AYDOS'a teşekkürü bir borç bilirim.

0215.STZ.2013-1 kodlu SANTEZ projesi kapsamında tez çalışmasına sağladığı katkılardan dolayı Bilim Sanayi Teknoloji Bakanlığına ve CYM SOFT Bilgi Teknolojilerine teşekkürlerimi sunarım.

Çalışmalarım boyunca tecrübelerinden yararlandığım değerli arkadaşım Dr. Eda Akman AYDIN'a ve mesai arkadaşlarıma, manevi destekleriyle beni hiçbir zaman yalnız bırakmayan aileme sonsuz teşekkürlerimi sunarım.

İÇİNDEKİLER

Sayfa

ÇİZELGELERİN LİSTESİ

Çizelge **Sayfa**

ŞEKİLLERİN LİSTESİ

Şekil **Sayfa**

SİMGELER VE KISALTMALAR

Bu çalışmada kullanılmış bazı kısaltmalar, açıklamaları ile birlikte aşağıda sunulmuştur.

Kısaltmalar	Açıklama
HTTP	Hiper-Metin Transfer Protokolü (Hypertext Transfer Protocol)
İTD	İmza Tabanlı Denetim
ATD	Anormal Tabanlı Denetim
NİİTM	Normal İsteklerin İmza Tabanlı Denetimi
AİİTD	Anormal İsteklerin İmza Tabanlı Denetimi
WUGD	Web Uygulama Güvenlik Duvarı
YSA	Yapay Sinir Ağları
XSS	Siteler Arası Betik Yazma (Cross Site Scripting)
KE	Komut Enjeksiyonu
DGS	Dizin Geçiş Saldırısı
AKA	Alfanümerik Karakter Analizi
HFA	Harf Frekans Analizi
İUA	İstek Uzunluk Analizi
XML	Genişletilebilir İşaretleme Dili (Extensible Markup Language)
OSI	Open Systems Interconnection (Open Systems Interconnection)
HTML	HyperText Markup Language (Hiper Metin İşaretleme Dili)
NIST	Ulusal Standartlar ve Teknoloji Enstitüsü (National Institute of Standards and Technology)
STS	Saldırı Tespit Sistemi
IIS	Internet Information Service (İnternet Bilgi Servisi)
TCP/IP	Geçiş Kontrol Protokolü / İnternet Protokolü (Transmission Control Protocol / Internet Protocol)
WWW	Dünya Çapında Ağ (World Wide Web)
URI	Tekdüzen Kaynak Tanımlayıcı (Uniform Resource Identifier)

SQL	Yapılandırılmış sorgu dili (Structured Query Language)
BTYSA	Bayes Tabanlı Yapay Sinir Ağları
BS	Bayes Sınıflandırma
PCI	Kartı Ödeme Endüstrisi (Payment Card Industry)
ART	Uyarlamalı Rezonans Teorisi (Adaptive Resonance Theory)
YP	Yanlış Pozitif
OWASP	Açık Web Uygulama Güvenliği Projesi **(**Open Web Application Security Project)

1. GİRİŞ

İnternetin kullanımının artmasıyla beraber, web uygulamalarının ve web servislerinin de sayısı her geçen gün artmaktadır. İnternet, statik dokümanların sunumundan, uygulama ve bilgi paylaşımı için kullanılan çok büyük bir platform haline gelmiştir. Eğitim, bankacılık ve ticari iş süreçleri gibi çoğu sektörel süreçler hızla internet kullanılarak gerçekleştirilmeye başlanmıştır. İnternetin kullanımının artmasıyla beraber, internetten sunulan ve bilgi paylaşımı gerçekleştirilen hizmetlerin de güvenlik riskleri de aynı ölçüde artmıştır. İnternette iş süreçlerini yönetmek için web uygulamaları veya web servisleri kullanılmaktadır. Web uygulamaları ile verilen hizmetler arttıkça web uygulamalarına yönelik yapılan saldırılar ve bu saldırıların çeşitleri de artmaktadır. Dolayısıyla web uygulamaları her geçen gün siber saldırıların öncelikli hedefi olmaktadır. Web uygulamalarının sahip olduğu güvenlik zafiyetleri; web uygulamalarının kolay erişilebilir olmaları, kolay erişilebilir olmalarından dolayı en çok güvenlik zafiyetlerine sahip uygulamalar olmaları, web uygulama geliştiricilerinin yeteri düzeyde uygulama güvenlik bilincine sahip olmamaları, web uygulamaları geliştirilirken güvenli yazılım geliştirme yöntemlerinin kullanılmaması ve gelişen teknolojik yöntemlerle web uygulamalarına yapılan saldırı çeşitliliğinin her geçen gün artması olarak sıralanabilir [1-3]. Son derece güvensiz olan internet ortamında bilgi kaybının ve güvenlik zafiyetinin önlenmesi için web uygulamalarının güvenlik ihtiyaçlarının karşılanması gerekmektedir. NIST'in (National Instuitue of Standards and Technology) yayınladığı rapora göre, güvenlik açıklarının %92'ye yakını web uygulamalarından kaynaklanmaktadır [4].

Web uygulamalarının güvenlik ihtiyaçları karşılanmadığı zaman ve web uygulamalarına karşı yapılan saldırılar için önlem alınmadığı takdirde, web uygulamaları kullanılarak sunulan hizmetlerin aksaması ve uygulama içeriklerinin silinmesi veya değiştirilmesi gibi hasarlar meydana gelmektedir. Bu durum ticari faaliyet sürdüren kişi veya kurumlar için ekonomik kayıplara ve itibar kayıplarına sebep olmaktadır. Web uygulamalarının güvenliğini sağlamak için uygulamanın tasarım aşamasında güvenli yazılım geliştirme yöntemleri kullanılarak uygulamaya karşı yapılacak saldırılar bertaraf edilebilir. Fakat uygulama geliştiriciler, uygulamanın fonksiyonelliğini uygulamanın güvenlik ihtiyaçlarından daha ön planda tuttukları için veya güvenlik ihtiyaçları bakımından yeterli

bilgiye sahip olmadıklarından dolayı, geliştirilen web uygulamaları güvenlik zafiyetleri bakımından oldukça yüksek güvenlik riskleri taşımaktadırlar [5].

Güvenlik risklerini kabul edilebilir seviyeye indirebilmek veya bu riskleri tamamen yok edebilmek için web güvenliği konusunda yapılan akademik ve ticari çalışmaların sayısı ve niteliği her geçen gün artmaktadır. Güvenlik zafiyetlerinden faydalanılarak yapılan saldırıları önlemek ve engellemek amacıyla saldırı tespit ve engelleme sistemleri (STS) kullanılmaktadır. Fakat STS'ler imza tabanlı algılama yeteneğine sahip oldukları için tanımlanmayan ya da önceden tespit edilememiş saldırı türleri için yetersiz kalmaktadır [6]. Ayrıca, saldırı tespit sistemleri ağ katmanı üzerinde aktif oldukları için uygulama katmanı üzerinde yapılan saldırılara karşı da etkisiz olmaktadırlar.

Geleneksel güvenlik duvarları da, port kontrolü yaparak ağ katmanı kullanılarak yapılan saldırıları başarılı bir şekilde engellerken, web uygulamalarına karşı yapılan web tabanlı saldırılara karşı etkili değillerdir [7]. Web uygulamalarının güvenlik ihtiyaçları ağ ortamında güvenlik duvarı ve STS gibi araçlar kullanılarak çözülebilmektedir. Fakat web uygulamalarına yapılan saldırılar sadece ağ katmanı kullanılarak yapılmamaktadır. Web uygulamaları iletişim için uygulama katmanı protokolü olan HTTP kullanmaktadır. Dolayısıyla web tabanlı saldırılar da HTTP ortamından gerçekleşmektedir. Ağ ortamında güvenliği sağlamak için kullanılan araçlar ise HTTP kullanılarak yapılan saldırılara karşı etkili değildirler. Web uygulamalarına yapılan saldırılar HTTP isteklerinin ve cevaplarının denetimi yapılarak engellenebilmektedir. HTTP isteklerinde denetleme yapmak için kullanılan araçlar ise Web Uygulama Güvenlik Duvarları (WUGD) olarak adlandırılmaktadır.

Web uygulamalarının güvenliğini sağlamak için artan saldırı çeşidi ve sayısına göre etkin işlevlere sahip WUGD çalışmaları gerçekleştirilmelidir. Bu alanda ticari ve akademik olarak sürdürülen çeşitli çalışmalar bulunmaktadır. Fakat etkin bir yöntem geliştirmek için bu alanda yapılan çalışmaların sayısının ve çeşidinin artması gerekmektedir. Web uygulamaları güvenlik duvarlarının gelişmesi neticesinde web uygulamalarının güvenlik ihtiyaçları azalacaktır. Bu sayede web uygulamalarına karşı yapılan web tabanlı saldırılar etkisiz hale getirilerek itibar kayıpları ve ekonomik kayıplar ortadan kalkacaktır.

WUGD, web uygulamalarını STS'lerin engellemekte yetersiz kaldığı web tabanlı saldırılara karşı korumak için kullanılmaktadır. STS'lerde olduğu gibi, WUGD da ağ tabanlı ya da

sunucu tabanlı olabilmektedir. Temelde STS ile WUGD arasındaki fark, WUGD'nın Open Systems Interconnection (OSI) 7. katmanı olan uygulama katmanında çalışıyor olmasıdır. WUGD uygulama katmanı protokollerinin denetimini yaparak, web tabanlı saldırıların denetimini gerçekleştirmektedir. [7].

WUGD, web uygulamaları sunucusunu web tabalı saldırılara karşı korumak için proxy görevi üstlenen bir güvenlik aracıdır. WUGD tipik ağ katmanı saldırı tespit sisteminin denetleyemediği tehditlere karşı etkilidir. WUGD, HTTP isteklerinin denetimini yaparak web sunucunun güvenliğini sağlamayı amaçlar [8].

Web uygulamalarının güvenliğini sağlamak amacıyla HTTP istek denetiminde genel olarak imza tabanlı denetim (İTD), anormal tabanlı denetim (ATD) ve makine öğrenmesi yöntemleri kullanılarak çeşitli çözümler önerilmiştir. Kullanılan İTD algoritmaları etiketlenmiş verileri belirlemeye yönelik olarak çalıştıkları için yeni geliştirilmiş saldırıların önlenmesinde etkisiz kalmaktadırlar. ATD yöntemleri ise standart veri yapısına uymayan verilerin denetimini gerçekleştirmek için kullanılmaktadır.

Bu tez çalışmasının hedefi öğrenme tabanlı ve hibrit bir WUGD algoritmasının geliştirilmesidir. Çalışmada web tabanlı saldırıları önleme amaçlı olarak İTD ve ATD yöntemlerini bir arada kullanan hibrit bir denetim algoritması geliştirilmiştir. Yapay zekâ tekniklerinden Bayes Sınıflandırma (BS), Yapay Sinir Ağları (YSA) ve Bayes Tabanlı Yapay Sinir Ağları (BTYSA) yöntemleri kullanılarak öğrenme tabanlı WUGD algortiması geliştirilmiştir. İTD ile, bilinen web tabanlı saldırı türlerinden olan SQL (Structured Query Language) Enjeksiyonu, Siteler Arası Kod (XSS) Yazma, Dizin Geçişi (DG) ve Komut Enjeksiyonu (KE) saldırı türlerine karşı imza denetimi gerçekleştirilmiştir. ATD ile, standart HTTP istek yapısına uymayan HTTP isteklerinin denetimi gerçekleştirilmiştir. ATD için ise, Alfanümerik Karakter Analizi (AKA), Harf Frekans Analizi (HFA) ve İstek Uzunluk Analizi (İUA) olmak üzere üç öznitelik kullanılarak gerçekleştirilmiştir.

Özniteliklerden elde edilen sonuçlar YSA giriş parametreleri olarak kullanılarak anormal HTTP isteklerinin denetimi öğrenme tabanlı olarak yapılmıştır. ATD sonucunda tespit edilen normal ve anormal HTTP istekleri tekrar web uygulamasına geldikleri zaman İTD'ye göre daha yavaş çalışan ATD gerçekleştirmeden, tespit edilen HTTP isteklerinin denetimlerinin gerçekleştirilmesi için, normal HTTP istekleri Normal İsteklerin İmza Tabanlı Denetimi (NİİTD) veritabanına, anormal HTTP istekleri Anormal İsteklerin İmza

Tabanlı Denetimi (AİİTD) veritabanına eklenmektedir. İTD hızlı çalışırken sıfır gün saldırılarına karşı etkili değildir. ATD yöntemi ise sıfır gün saldırılarında etkilidir fakat ATD, İTD metodu kadar yüksek hız performansına sahip değildir [9]. Böylece bu çalışmada iki yapının önemli özellikleri ortak bir yapı içerisinde birleştirilerek denetim hızı yüksek model önerilmiştir. Ayrıca ATD, verimadenciliği yöntemlerinden BS ve BTYSA yöntemleri ile de denetim yapılarak YSA ile elde edilen sonuçlarla karşılaştırılmıştır.

Bu çalışma yedi bölümden meydana gelmektedir. Çalışmanın ikinci bölümünde literatürde yapılan benzer çalışmalar açıklanmıştır. Üçüncü bölümde uygulama güvenlik duvarından ve web uygulamaları zafiyet özelliklerinden bahsedilmiştir. Dördüncü bölümde tez çalışmasında kullanılan materyal ve metodlar anlatılmıştır. Beşinci bölümde gerçekleştirilen çalışma açıklanmıştır. Altıncı bölümde test ve değerlendirme sonuçları verilmiştir. Yedinci bölümde ise sonuçlar ve öneriler özetlenmiştir.

2. LİTERATÜR TARAMA

Tez çalışmasında, İTD ve ATD denetim türleri beraber kullanılarak hibrit bir WUGD algoritması tasarlanarak, WUGD yazılımı geliştirilmiştir. İTD ile bilinen web tabanlı saldırı türlerinin denetimi gerçekleştirilirmiştir. ATD ile yapay zeka yöntemlerinde BS, YSA ve BTYSA kullanılarak anormal HTTP isteklerinin denetim süreci gerçekleştirilmiştir. Literatür araştırmasında da BS, YSA ve BTYSA yöntemleriyle ilgili yapılan bazı benzer çalışmalara yer verişmiştir.

Stephan ve diğerleri [10] yeni saldırı türlerini tespit etmek için, imza denetiminde tespit edilemeyen saldırıları tanımlamak için YSA geri yayılım yaklaşımı kullanarak imza güncellemelerine ihtiyaç duymayan web uygulaması güvenlik duvarı prototip geliştirmişlerdir. Geliştirilen prototip, uygulama katmanı seviyesinde web uygulamasına erişimi ve güvenlik duvarının uygulama seviyesini kontrolü esnasındaki kullanıcı davranışlarıyla ilgili bazı ek bilgiler ve bazı parametrelerle denenmiştir. Örnek verilerin daha önceden elde edilmiş örneklerle eşleştirilme ve karşılaştırılmasında sistemin %95 oranında iyi bir performansa sahip olduğu test verilerinden tespit edilmiştir.

Nguyen ve diğerleri [11] yaptıkları çalışmada web saldırı denetiminde öznitelik seçimine yönelik bir çalışma gerçekleştirmişlerdir. Genel öznitelik seçimi (GeFS) için CSIC 2010 ve ECVL/PKDD 2007 veri kümeleri kullanılarak C45, CART, RamdomTree, RandomForest olmak üzere 4 farklı sınıflayıcıyla veri kümelerinde uzman görüşü ile belirlenen 30 farklı öznitelik oluşturularak denetim gerçekleştirilmiştir. Çalışmada siteler arası kod yazma (XSS), SQL enjeksiyonu, LDAP enjeksiyonu, XPATH enjeksiyonu saldırı türleri için sınıflandırma yapılmıştır. Yazarların yaptığı çalışmada HTTP isteklerini denetlemek için kullanılabilecek özniteliklerin performans değerlendirmesi yapılarak, öznitelik seçimi yapılmaktadır. Bizim önerdiğimiz çalışmada da bu çalışmada belirlenen istek uzunluk özniteliği kullanılmıştır.

Palka ve diğerleri [12] yaptıkları çalışmada, web uygulamalarına gelen HTTP isteklerini doğrulayarak ve HTTP isteklerinin parametrelerini, daha önceden kaydedilmiş kullanıcı alışkanlıklarıyla karşılaştırarak, web uygulamalarını koruma altına alan WUGD gerçekleştirmişlerdir. Kullanıcı alışkanlıkları, web uygulamasını ziyaret eden kullanıcıların davranışları sonucunda üretilmiştir. Öğrenme tabanlı WUGD esnek, uygulamaya özel ve devreye alınması kolay bir çözüm sunmasının yanında HTTP isteklerinin denetimi sürecinde

sınıflandırma yapılırken öğrenme sürecinde hatalar gerçekleşmektedir. Yazarlar yaptıkları bu çalışmada öğrenme tabanlı WUGD'nın öğrenme sürecinde gerçekleşen hataları tartışmışlardır. Yazarların yaptığı bu çalışmada kullanıcı davranışlarına göre denetim gerçekleştirilmiştir. Sadece kullanıcı davranışlarının kullanılması web uygulamasına yapılan saldırıların denetiminde yeterli değildir. Çünkü web uygulamalarına yapılan saldırılar sadece kullanıcı davranışlarıyla tespit edilemeyecek kadar çok fazla çeşide ve sayıya sahiptir. Dolayısıyla bizim önerdiğimiz çalışmada web uygulamalarına yapılan bilinen saldırı türlerine karşı ve anormal HTTP isteklerine karşı denetim gerçekleştirebilmek için hibrit model önerilmiştir.

Basile ve diğerleri [13] yaptıkları çalışmada WUGD paket filtrelemek için anormal denetim gerçekleştirmişlerdir. Önerdikleri modelde düzenli ifadeler ile belirtilen metin tabanlı içerik filtreleme yapmışlardır. Modelin etkinliği HTTP proxy olan SQUID'in erişim kontrol özelliğine karşı başarılı bir şekilde test edilmiştir. Yazarların yaptığı çalışmada metin tabanlı anormal denetim yapılmıştır. Metin tabanlı denetim web uygulamalarına yapılan saldırıların denetimi için yeterli değildir. Fakat önerdiğimiz çalışmada web uygulamalarına yapılan bilinen saldırı türlerine ve anormal HTTP isteklerine karşı denetim gerçekleştirebilmek için hibrit model önerilmiştir.

Cho ve diğerleri [14] ise yaptıkları çalışmada kullanıcılar tarafından istenen web sayfalarının benzer özelliklere sahip olduğunu tespit etmişlerdir. Web erişim verilerinden web oturumlarını açığa çıkarmak için bayes parametre tahmin tekniği kullanılarak, oturum bilgilerinden anormallik tahmini gerçekleştirilen oturum anormallik tespiti (Session Anomaly Detection - SAD) ismi verilen bir çalışma geliştirilmiştir. Yazarların yaptığı bu çalışmada web uygulamasına yapılan saldırılar, oturum bilgileri kullanılarak BSyöntemi ile anormal denetim gerçekleştirilerek tespit edilmiştir. Sadece oturum bilgileri kullanılarak ve anormal denetim yapılarak web uygulamalarına yapılan saldırıların denetimi yetersiz kalacaktır. Bizim önerdiğimiz modelin bu çalışmadan farkı hem HTTP isteklerine karşı hibrit bir denetim modelinin önerilmesi, hem de web tabanlı saldırıların en çok yapıldığı HTTP istek satırının denetimini gerçekleştirmesidir.

Razzaq ve diğerleri [15] ise yaptıkları çalışmada BS kullanılarak ontoloji tabanlı uygulama katmanı saldırı tespit sistemi önermiştir. Önerilen sistemde, HTTP istek parametreleri ontoloji değişkenleri olarak belirlenerek değişken değerlerine göre BS işlemi HTTP isteklerinin anormallik tespiti gerçekleştirilmiştir. Yazarların yaptığı anormal tabanlı

denetim bizim önerdiğimiz gibi BS kullanılarak geliştirilmiş olsa da yazarlar ön işleme kısmında ontoloji tabanlı sınıflandırma kullanmışlardır. Bizim yaptığımız çalışma da ise önişleme sürecinde farklı öznitelikler kullanılmıştır.

Bremler-Barr ve diğerleri [16] yaptıkları çalışmada, imza tabanlı denetim kullanan güvenlik araçlarının sıkıştırılmış trafiklere karşı etkili olmadığını belirtmişlerdir. Çalışma ile sıkıştırılmış HTTP trafiği üzerinde durulmaktadır. HTTP, GZIP sıkıştırma yöntemi kullamaktadır ve metin karşılaştırması yapabilmek için bazı farklı açma (sıkıştırılmış veriyi kullanılabilir hale getirme) yöntemlerinin kullanıldığı belirtilmiştir. Yazarlar gerçek HTTP trafiğinin ve gerçek WUGD imzalarının analiz edilmesiyle, verilerin % 84'ünden fazlasının taranmasının atlandığını ortaya çıkarmışlardır. Ayrıca sıkıştırılmış veriyle desen eşleştirerek denetim yapmanın sıkıştırılmamış veriyle yapmaktan daha hızlı olduğunu ortaya koymuşlardır. Bizim önerdiğimiz çalışma ise sıkıştırılmamış HTTP isteklerine karşı denetim gerçekleştiren hibrit bir çalışmadır.

Singh ve diğerleri [17] yaptıkları çalışmada siber saldırı veri kümesi sınıflandırılması için iyileştirilmiş destek vektör makinesi (iSVM) algoritması kullanarak denetim modeli önermişlerdir. Sonuçlar iSVM, Normal ve Hizmet Aksatma (DOS) sınıflarına karşı %100 doğruluk vermiş ve yanlış alarm oranı, eğitim ve test süreleri karşılaştırılabilirliğini göstermişlerdir. Geleneksel SVM performansı, Gauss çekirdek ile bir konformal haritalama marjı etrafında uzaysal çözünürlüğü büyütmek için geliştirilmiştir. Böylece saldırı, sınıflar arasındaki ayrılabilirliliği artacaktır. Bu işlem çekirdek fonksiyonu tarafından uyarılan Riemann geometrik yapısına dayanmaktadır. Bizim önerdiğimiz çalışma ise yazarların önerdiği çalışmadan farklı olarak İTD ve ATD modelleri kullanılarak BS yöntemi ile geliştirilmiş hibrit bir modeldir.

Torrano-Gimenez ve diğerleri [18] yaptıkları çalışmada, anormal tabanlı denetim sonucunda bilinmeyen web tabanlı saldırıları tespit edebilen bir WUGD geliştirmişlerdir. Geliştirilen model bir HTTP isteğinin saldırı olup olmadığına bir XML dosyasının yardımı ile karar vermektedir. XML dosyası web uygulamasını hedef alan istatistiksel olarak normal davranışlara sahip yapay zeka tabanlı olarak üretilmiş normal HTTP isteklerini barındırmaktadır. Normal davranışın dışında sapma gösteren istekler, sistem tarafından anormal olarak nitelendirilmektedir. Sürekli artan eğitim verileri sistemin eğitiminde kullanılmaktadır. Deneyler sonucunda XML dosyasında web uygulamasının karakteristiğini ortaya çırakacak yeteri kadar istek bulunması, çok başarılı denetim oranına ulaşılmasını

sağlarken, yanlış alarm oranını da düşürmektedir. Bizim önerdiğimiz çalışma ise İTD ve ATD modelleri kullanılarak BS yöntem ile geliştirilmiş hibrit bir modeldir. ATD sonucu tespit edilen anormal HTTP istekleri İTD imza listesine eklenerek imza veritabanı güncellenmektedir. Bu özellik sayesinde modelin denetim performansını artırmaktadır.

Peng ve diğerleri [19] yaptıkları çalışmada, ağ tabanlı saldırıların, günümüzde büyük ölçüde birbirine bağlı bilgisayar sistemleri için temel tehdit haline geldiğinden bahsederek, yetkilendirilmemiş aktiviteler ve erişimler için ağların en büyük problemleri olduğunu belirtmişlerdir. Saldırı tespit sistemleri artan güvenlik açıklıklarına karşı kullanılması kaçınılmaz sistemlerdir. Yazarlar çalışma ile imza tabanlı ve anormal tabanlı denetim metotlarının avantajlı taraflarını güçlendiren, hibrit bir saldırı tespit ve görüntüleme sistemi önermişlerdir. Saldırı tespit edildiğinde, sisteme entegre olan bağımsız ajan kötüye kullanım davranışlarına karşı eylem gerçekleştirerek, ağı içerden veya dışarıdan yapılan saldırılara karşı korumaktadır. Yazarlar önerdikleri çalışmayla ağ tabanlı saldırılara karşı hibrit bir model gerçekleştirmişlerdir, ama bizim önerdiğimiz çalışmada web tabanlı saldırılara karşı hibrit bir model geliştirilmiştir.

Locasto ve diğerleri [20] yaptıkları çalışmada, binary kod enjeksiyonu saldırısını önlemek için hibrit bir yaklaşım sunmuşlardır. Model imza tabanlı, anormal tabanlı sınıflandırıcı ve öğrenme aracı olmak üzere üç mekanizmadan oluşmaktadır. Anormal denetim sonucu elde edilen saldırılar ile imza üretimi gerçekleştirilmiştir. Geri besleme mekanizmasına bağlı olarak işleyen öğrenme aracı kod enjeksiyon saldırısını sınıflandırarak engelleyen bir yapıya sahiptir. Tespit edilen saldırı içeren zararlı kod içeren paketler, sıfır gün saldırılarını engellemek için imza üretimi gerçekleştirilmiştir. Yazarların geliştirdiği model binary kod enjeksiyonu için geliştirilmiş hibrit çalışmadır, ama bizim önerdiğimiz model ise yine öğrenme tabanlı olarak hem bilinen saldırı türlerine karşı hem de sıfır gün saldırılarına karşı geliştirilmiş hibrit modeldir.

Hwang ve diğerleri [21] yaptıkları çalışmada yeni bir hibrit saldırı tespit sistemin tasarım gerçekleştirmişlerdir. Hibrit model düşük yanlış pozitif oranına sahip imza tabanlı denetim modelinin ve bilinmeyen yeni saldırıları denetleyen anormal tabanlı denetim modelinin avantajlarını birleştirmektedir. SNORT'da bulunan anormal verilerin belirlenmesini sağlayan anormal denetim modeliyle imza üretimi gerçekleştirilmiştir. Hibrit saldırı tespit sisteminin anormal denetim süreci ile tespit edilen imzalar, daha hızlı denetim gerçekleştirmek için SNORT imza veritabanına eklenmektedir. Deneysel sonuçlara göre

%60 denetim oranı elde edilmiştir. Anormal denetimle üretilen imzalar SNORT performansını %33 oranında artırmıştır. Bizim yaptığımız çalışma mimari olarak yazarların yaptıkları çalışmaya benzer olsa da bizim yaptığımız çalışma web tabanlı saldırıların denetimini gerçekleştirmek için geliştirilmiş hibrit bir modeldir. Fakat yazarların yaptığı çalışma ağ tabanlı saldırıların denetimini yapmak için geliştirilmiştir.

Hendry ve diğerleri [22] yaptıkları çalışmada, ağ tabanlı saldırılara karşı anormal tabanlı denetim algoritması gerçekleştirerek, imza üretimi yapmışlardır. Böylece imza tabanlı denetimin sıfır gün saldırılarına karşı etkili olmama özelliği bertaraf edilmiştir. İmza üretimi için hibrit, danışmanlı ve danışmansız sınıflandırma algoritması önerilmiştir. Gerçek zamanlı olarak oluşturulan imzalar yeni saldırıları denetlemek için gerçek zamanlı olarak devreye girmektedir. Yazarlar farklı yöntemler kullanarak önerilen çalışmaya benzer hibrit bir çalışma gerçekleştirmişlerdir, ama yazarların yaptığı çalışmada ağ tabanlı yapılan saldırılara karşı denetim gerçekleştirilirken, bizim yaptığımız çalışmada web tabanlı saldırılara karşı denetim gerçekleştirilmesi amaçlanmıştır.

Yukarıda önerdiğimiz çalışmayla benzerlik gösteren çalışmalardan bahsedilmiştir. Yapılan çalışmalarda farklı yöntemler kullanılarak web tabanlı saldırılara ve ağ tabanlı saldırılara karşı farklı denetim modelleri geliştirilmeye çalışılmıştır. Kullanılan yöntemler arasında farklı yapay zekâ ve veri madenciliği teknikleri bulunmaktadır. Yukarıda kısaca özetlenen çalışmalarda ATD, İTD, öğrenme tabanlı denetim ve hibrit çalışmalar gerçekleştirilmiştir. ATD modelleri yavaş çalıştıkları için gerçek zamanlı web uygulamalarında tercih edilmemektedirler ama sıfır gün saldırılarına etkili oldukları için tercih edilmektedirler. İTD modelleri ise daha hızlı çalışmaktadırlar, ama sadece imza olarak tanımlanan saldırı türlerine karşı etkilidirler. Bu çalışmada önerilen modelle, İTD yönteminin, bilinen saldırı türlerine karşı etkili olması ve hızlı çalışması özelliklerinin kullanılması ve ATD yönteminin ise sıfır gün saldırılarına karşı etkili olması ve yeni durumlara karşı kendini yenilemesi özellikleri kullanılarak, benzer çalışmalardan farklı olarak İTD ve ATD yöntemlerinin bir arada kullanıldığı hibrit bir model önerilmiştir. Literatürde [19], [21] ve [22] numaralı çalışmalar da hibrit çalışmalardır ancak ağ tabanlı saldırılara yönelik olarak yapılmışlardır. Çalışma [20] hibrit bir çalışmadır ama sadece kod enjeksiyonu saldırıları için yapılmıştır. Önerdiğimiz modelde İTD ile *SQL Enjeksiyonu, Siteler Arası Kod (XSS) Yazma* saldırı, Komut Enjeksiyonu (KE) ve Dizin Geçişi Saldırı (DGS) türlerine karşı denetim yapılmıştır. ATD ile ise şeçilen üç öznitelikle BS, YSA ve BTYSA gerçekleştirilerek anormal HTTP

isteklerinin denetimi karşılaştırmalı olarak gerçekleştirmiştir. İTD hızlı çalışırken sıfır gün saldırılarına karşı etkili değildir. ATD yöntemi ise sıfır gün saldırılarına karşı etkilidir ama İTD kadar hızlı çalışan bir yöntem değildir. Dolayısıyla her iki yöntem beraber kullanılarak, mevcut yöntemlerin avantajları ön plana çıkarılmıştır. Anormal olarak tespit edilen HTTP istekleri ile AİİTD veritabanı güncellenerek, aynı HTTP isteği web uygulamasına tekrar geldiği zaman, AİİTD'ye göre daha yavaş çalışan ATD süreci işletilmeden İTD gerçekleştirilerek anormal HTTP istekleri engellenebilmektedir. Normal olarak tespit edilen HTTP istekleri ile NİİTD veritabanı güncellenerek, aynı HTTP isteği web uygulamasına tekrar geldiği zaman, NİİTD'ye göre daha yavaş çalışan ATD süreci işletilmeden NİİTD gerçekleştirilerek normal HTTP istekleri web uygulamasına yönlendirilerek web uygulaması çalıştırılmaktadır.

Önerilen modeli benzer çalışmalardan ayıran en önemli özelliklerden biri denetimi yapılan normal HTTP isteklerinin ve anormal HTTP isteklerinin imza veri tabanının güncellenmesinde kullanılması, geliştirilen WUGD yazılımının yeni saldırı türlerine karşı etkili olmasını sağlayarak, manuel imza güncelleme işlemi gerçekleştirilmeden imza üretimi yapıldığı için sistemin yeni saldırı türlerine karşı adaptasyonunu gerçekleştirilmektedir.

3. WEB UYGULAMA GÜVENLİK DUVARI (WUGD)

WUGD, web uygulamalarına yapılan saldırıları denetlemek ve engellemek için kullanılan güvenlik araçlarıdır. WUGD, tanımlı politikalara göre bir servise veya uygulamaya yapılmak istenen erişimleri denetleyerek, erişim isteklerine izin veren veya erişimi engelleyen, uygulama katmanı seviyesinde denetim yapan, uygulama güvenlik duvarı çeşididir. Web uygulamasına yapılan istek denetimini kötüye kullanım veya anormal denetim türlerine göre gerçekleştirmektedir [23]. Bu bölümde web uygulamalarının çalışma yapısı, zaafiyet özellikleri ve WUGD hakkında bilgi verilmektedir.

WUGD, port izlenmesi yöntemine dayalı olarak, girdi, çıktı denetiminin yanı sıra, WUGD üzerine yapılandırılmış politikalara uymayan HTTP isteklerini engeller. WUGD sistemleri karmaşıklaşan web trafiği üzerinde detaylı denetim yaparak anormal trafiğin engellenmesini sağlayan sistemlerdir. HTTP ve web servisleri trafiğinde detaylı inceleme yaparak saldırı içeren isteklerin veya anormal isteklerin engellenmesi için kullanılan araçlardır.

Klasik güvenlik duvarları HTTP kullanılarak yapılan saldırılara karşı yeterli olmamaktadır. Çünkü önceden yapılan saldırılar genelde ağ seviyesinde yapılmaktaydı ve ağ servislerini hedef alıyorlardı. Günümüzde ise kullanılmayan servisler güvenlik duvarı tarafından kapatılarak, açık olanlar için de güvenliğini kanıtlamış olan sistemler kullanılarak geleneksel güvenlik açıklıkları kapatılmaktadır [24].

Belirli güvenlik politikasına sahip çoğu kapalı ağ yapıları, ihtiyaç duyulmayan servisleri güvenlik duvarı aracılığı ile kapatmaktadır. Fakat internete açılan servisler olan web sunucular zorunlu olarak açık kalmaktadırlar. Bu durumda da saldırılar da sürekli açık olan web sunucu portları ve web sunucuda çalışan uygulamaları hedef almaktadır. Web üzerinden sunulan servislerin çeşitliliği ve bu konuda geçerli bir standart yapı olmaması geliştirilen uygulamaların güvenlik yönünden yeterli olgunluğa erişmesini engellemektedir.

WUGD sistemlerinin kullanımı güvenlik standartları tarafından da tavsiye edilmektedir. Bunlardan en önemlisi WUGD kullanımını zorunlu tutan Payment Card Industry (PCI) Veri Güvenliği standardıdır. Standarda göre 30 Haziran 2008 sonrası PCI uyumlu olma zorunluluğu bulunan tüm kurumlarda ya kod denetimi ya da WUGD sistemini devreye alarak kullanmak zorundadır [25].

Web uygulamalarının kullanım oranlarının artması, web uygulamalarına yapılan saldırı çeşitliliğinin ve sayısının artmasına sebep olmuştur. Web uygulamaları kullanılarak sağlanan hizmet aksamalarının ve bilgi zaafiyetlerinin ortadan kaldırılması için web uygulamalarının güvenlik ihtiyaçlarının karşılanması gerekmektedir. Web uygulamalarının güvenlik ihtiyaçlarını karşılamak için kullanılan en önemli araçlardan biri de WUGD'dır.

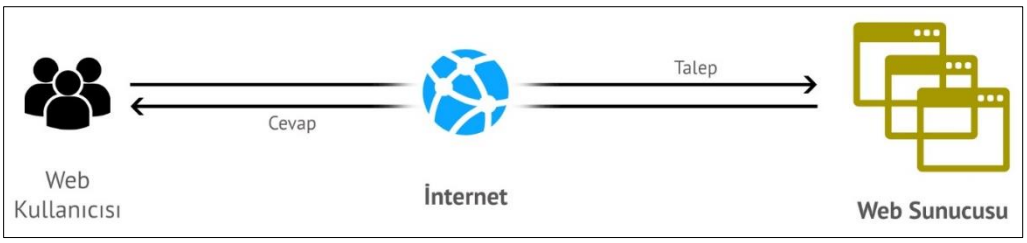
Bu tez çalışmasında web tabanlı saldırıları önlemek ve web uygulamalarını güvenlik ihtiyaçlarını karşılamak amacıyla hibrit bir WUGD algoritması geliştirilmiştir.

3.1. Web Uygulamaları ve Zafiyet Özellikleri

Bu bölümde web tabanlı saldırılarının anlaşılabilmesi için web uygulamalarının çalışma prensipleri, özellikleri, çalışma ortamları ve zaafiyet özellikleri anlatılmıştır.

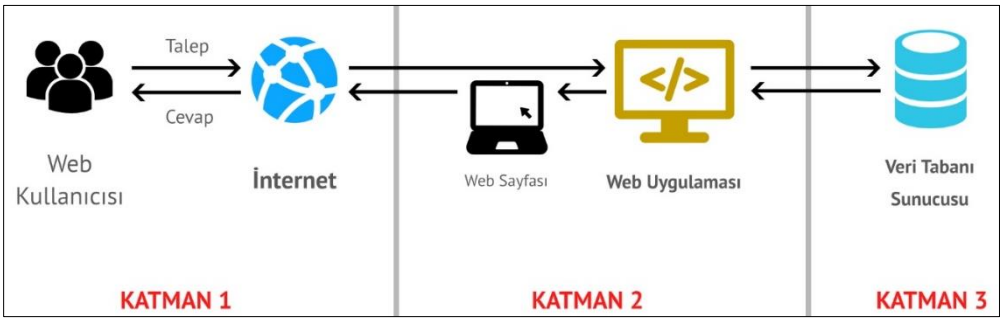
Web uygulamaları kamunun erişimine açık sistemlerdir. Erişim kolaylığından dolayı web uygulamaları her türlü saldırıya ve kötüye kullanıma da açık yapılardır [3]. Web uygulamalarına yapılan saldırıların büyük bir kısmı Open Systems Interconnection (OSI)'nin 7. Katmanı olan uygulama katmanından yapılmaktadır. Ayrıca web uygulamaları HTTP protokolünü kullanarak veri iletişimi gerçekleştirmektedir. WUGD OSI 7 uygulama katmanı ve HTTP protokolünün denetimini yaparak web uygulamalarına yapılan saldırıları engellemeye çalışmaktadır.

Web uygulamaları dinamik veya statik yapıda çalışan HyperText Markup Language (HTML) içerikleri bulunan uygulamalardır. Şekil 3.1'de gösterilen ve statik yapıda çalışan web uygulamaları, kullanıcıdan gelen talepler doğrultusunda ilgili web sayfalarının gösterilmesini sağlayan HTML kodlarını içermektedirler.



Şekil 3.1. Statik web sitesi çalışma yapısı şematik gösterimi [26]

Dinamik web siteleri ise, kullanıcı istekleri doğrultusunda çalışan web uygulamalarıdır. Dinamik web siteleri Şekil 3.2'de şematik olarak gösterildiği gibi, üç katmanlı bir yapı içerisinde çalışmaktadır [27].



Şekil 3.2. Dinamik web sitesi çalışma yapısı şematik gösterimi [26]

Şekil 3.2'de Katman 1 web siteleri için taleplerin başladığı, son kullanıcı girdilerinin oluştuğu web tarayıcılarıdır. Web tarayıcıları kullanılarak, web sunucusuna içerikle ilgili talepler iletilir. Katman 2 dinamik sayfaların üretildiği uygulama katmanıdır. Katman 3 ise web uygulamaları tarafından kullanılan verilerin depolandığı veri tabanlarıdır.

Dinamik içerikli web uygulamalarında, web tarayıcıları taleplerini web uygulamalarına ilettikten sonra, bu talepler doğrultusunda veri tabanı sorgulaması yapılır ve talep edilen isteklere ait sonuçların yer aldığı sayfalar üretilerek, tarayıcılar üzerinde gösterilir. Dinamik içerikli web sayfalarının bu esnek çalışma yapısı birçok güvenlik tehdidini ve ihlâllerini beraberinde getirmektedir.

Web uygulamaları korunmasızlık ve kolay erişilebilirlik seviyesinin yüksek olması nedeniyle bilgi güvenliği zafiyetinin en fazla gerçekleştiği sistemlerdir. Uygulama katmanı seviyesi zafiyetleri genel olarak web uygulamaları geliştirilirken kullanılan programlama dilindeki kontrol eksikliğinden ve son kullanıcıdan alınan girdilerin güvenlik kontrollerinden yeterince geçirilmemesinden kaynaklanmaktadır. Web uygulamaları devreye alınmadan önce web tabanlı saldırılara karşı güvenlik testlerinden geçirilmesi, uygulamanın geliştirilmesi sürecinden kaynaklanacak zafiyetlerin tespit edilerek, önlem alınmasını sağlayacaktır. Çünkü web uygulamalarında bulunan güvenlik zafiyetleri sadece web uygulamalarını etkilememektedir. Web uygulamasının üzerinde koştuğu işletim sisteminin, hatta tüm ağın ele geçirilmesi için ilk adımı oluşturulabilmektedir.

Web uygulamalarının geliştirilme sürecinde son kullanıcıdan alınan girdilerin güvenlik kontrolü yapılmadığı zaman, acil önem arzeden bazı mantıksal hatalar ve yüksek seviyeli teknik zafiyetler meydana gelebilmektedir. Web uygulama güvenliğinin sağlanması için bir takım öneriler şu şekilde sıralanabilir;

- Web uygulama güvenlik taraması ile her devreye alım öncesi güvenlik taramasından geçirilmesi,
- Devredeki tüm sistemlerin detaylı kod analizinden geçirilmesi,
- WUGD kullanılmasıdır.

National Instuitue of Standards and Technology (NIST) yayınladığı rapora göre, güvenlik açıklarının %92'ye yakını internet/web uygulamalarından kaynaklanmaktadır [28]. Web uygulamalarına yapılan saldırıların %70'i uygulama seviyesindeki ataklardan kaynaklanmaktadır, ilaveten ticari içerikli web uygulamalarının %75'i ise korunmasız durumdadır [29]. Web uygulamalarında oluşabilecek zafiyet, güvenlik önlemlerini (güvenlik duvarı, saldırı tespit ve önleme sistemleri, vb.) devre dışı bırakarak güvenilir bölgede yer alan sistemleri tehdit etmektedir.

3.2. WUGD Yerleşim Modelleri

WUGD için dört yerleşim senaryosu vardır. Bunlar köprü, pasif, tümleşik ve ters proxy modelleridir.

Köprü yerleşim modeli

Web sunucularından önce köprü olarak yerleştirilerek HTTP trafiğinin üzerinden geçirilerek denetim gerçekleştirilir. Bu yöntemin kullanılması durumunda, ağ topolojisinde herhangi bir değişiklik yapılmasına gerek yoktur. Üzerinden geçen tüm HTTP trafiği alınır, denetlenir ve saldırı içeriyorsa engellenir. Tek dezavantajı ise sistemde yaşanacak herhangi bir arıza sonrası HTTP trafiğinin kesilmesinden dolayı hizmet aksamasına sebep olmaktadır. WUGD arıza durumunda web trafiğinin geçişini engellemeyecek özellikte olmalıdır.

Pasif yerleşim modeli

Bu yerleşim modelinde WUGD web sunucuların olduğu anahtarın trafiğini aynalayarak veya hub aracılığı ile bağlanır ve pasif olarak web sunuculara giden trafiğin bir kopyasını alır. Alınan kopya trafik üzerinde detaylı denetim yaparak saldırı içeren veya anormal olan

istekler için bir cevap üretilir. Saldırganın gönderdiği isteğin cevabı WUGD tarafından gönderilecek cevaplardan önce ulaşabilir ya da saldırgan HTTP kullanarak geriye yönelik UDP bağlantısı açmaya çalıştığında WUGD işlevsiz kalabilir.

Bütünleşik yerleşim modeli

Basit WUGD yerleşimidir ve tamamen çalışan işletim sistemine, web sunucusuna bağlı olarak çalıştırılır. Örneğin Microsoft Internet Information Service (IIS) için kullanabilecek WUGD, linux işletim sistemi için kullanılamaz, benzer şekilde Apache için kullanılan WUGD yazılımı IIS için kullanılamaz. Bütünleşik WUGD yazılımları özellikle güvenliği sağlanacak sistemlerin sayısı fazla değilse tercih edilebilir.

Ters proxy yerleşim modeli

Ters proxy yerleşim modelinde web uygulamasına erişmeye çalışan tüm istemciler web sunucu yerine ters proxy modundaki WUGD sistemine ulaşır, WUGD isteği aldıktan ve gerekli denetimlerden sonra, web sunucuya iletir veya doğrudan isteği engeller. Bu model, diğerlerine göre ağ topolojisinde en fazla değişiklik gerektiren modeldir. WUGD tarafından korunmaya alınacak tüm web uygulamalarının trafiği WUGD cihazına ulaşacak şekilde ağ topolojisi değiştirilmelidir [30].

3.3. WUGD Denetim Modelleri

Kara liste ve beyaz liste olmak üzere iki tür denetimi modeli bulunmaktadır. Kara liste denetim modeli, bilinen saldırıların denetimlerinin gerçekleştirildiği imza tabanlı denetim türüdür. Kara listede bulunan herhangi bir HTTP isteği tespit edildiği zaman kara liste devreye girerek HTTP isteği engellenir. Çoğu WUGD araçları kara liste özelliğine sahiptir. Diğer taraftan, beyaz liste bilinmeyen saldırı türlerini tespit etmek için kullanılan denetim modelidir [31]. Beyaz liste denetim modelinden sadece system tarafından izin verilen HTTP isteklerinin geçişine izin verilir. İzin verilen HTTP isteklerinin dışındaki bütün istekler engellenir.

Kara liste modeli

Kara liste güvenlik modeli, yasaklanmış belirli işlemler hariç her şeye izin verilmesidir. Ayrıca kötüye kullanım olarak da tanımlanan kara liste modeli, saldırı içeren bir trafiği

puanlayarak, istemci IP adreslerinin, uygulama çerezlerinin veya kullanıcı hesaplarının denetlenerek, gerekli durumlarda engellenmesidir [32]. Sadece yasaklanan saldırı türlerinin denetlenmesi ve bunların dışındakilere izin verilmesini öngören bir yaklaşımdır. Kara listede imza olarak tanımlanmamış saldırı türlerine karşı etkili olmayan statik bir denetim yaklaşımdır.

Beyaz liste modeli

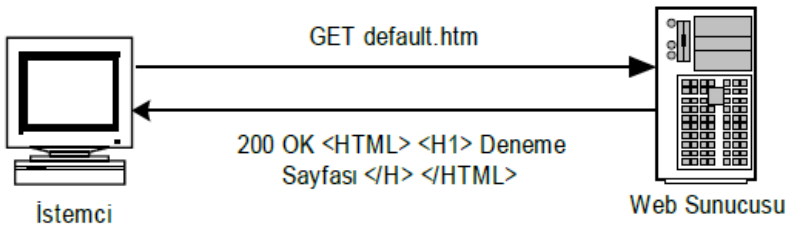
İzin verilen belirli işlemler hariç, her şeyin engellenmesi yaklaşımıdır. Sadece izin verilen isteklerin sürdürülmesi ve diğerlerinin engellenmesini yöntemidir. Beyaz liste modelinde genel olarak makine öğrenme tabanlı algoritmalar kullanılmaktadır. HTTP istekleri incelenerek web uygulamalarının haritası çıkarılmaya çalışılır. Gelen isteğin davranışının belirlenmesi ilkesine bağlı olarak bu işlem sonrasında web uygulamalarına yapılacak yeni tip benzer saldırılar devre dışı bırakılabilmektedir. Makine öğrenme tabanlı model özelliği olmayan araçlarda WUGD arkasında koruma altına alınmak istenen bütün web uygulamaların tanımlanması gerekmektedir. Sistem devreye alınmadan önce bu model devre dışı bırakılarak uygulamanın yapısına göre veri toplaması gerekmektedir.

Öğrenme tabanlı model

Öğrenme tabanlı model HTTP isteklerini inceleyerek web sayfalarının haritasını çıkarmaya çalışır. Öğrenme işlemi sonucunda web sayfalarının yapısına uymayan HTTP istekleri veya yeni tip saldırı içeren istekler engellenebilmektedir. Öğrenme özelliği olmayan WUGD üzerinde denetlenmek istenen bütün saldırı türlerine, karşı imza listesi oluşturulmalıdır.

3.4. HTTP (Hypertext Transmission Protokol)

HTTP, World Wide Web (WWW) üzerinde veri göndermek için kullanılan yöntemdir. Günümüzde çoğunlukla kullanılan HTTP 1.1 sürümüdür ve RFC 2616'da ayrıntılı olarak tanıtılmıştır. HTTP, istemci isteği ve sunucu cevabından oluşan iletişim kurallarının bütününü ifade eder. İstemciler, sunucunun HTTP protokol ile TCP bağlantısı kurar. İstemci, "GET/ HTTP/1.1" ve onu izleyen ve isteği tanımlayan bilgileri içeren MIME iletisi içeren bir TCP paketi ile web uygulamasını talep eder. Sunucu isteğe göre bir yanıt dizgisini cevap olarak döndürür. Bu yanıt, olumlu olduğunda istenilen sayfayı içeren ileti veya yanıt olumsuz olduğunda bir hata iletisi veya başka bir bilgi olabilir [33]. Tipik bir HTTP iletişimi Şekil 3.3'de gösterilmiştir.



Şekil 3.3. İstemci, sunucu arasındaki HTTP trafiği [30],

HTTP, RFC 2616 internet standardı tarafından tanımlanan bir protokoldür. HTTP günümüz internetinde yaygın olarak kullanılmakta olan bir protokoldür. HTTP protokolü uygulama katmanında çalışmaktadır. HTTP protokolü 1990 yılında sunucu ve istemci arasında veri iletişimini sağlamak için kullanılmaya başlanmıştır. Protokol TCP/IP protokolünü kullanmaktadır, varsayılan port olarak 80 veya 443 numaralı portu kullanır ancak bu port gerekli olması halinde değiştirilebilir.

Günümüzde HTTP 1.1 versiyonu kullanılmaktadır. Bu versiyonda aynı port, istek ve cevaplarda kullanılabilmektedir. Daha önceki versiyon olan (HTTP 1.0) versiyonunda istek ve cevaplar için farklı portlar kullanılmaktaydı. HTTP protokolünde iletişim istek ve cevap şeklinde gerçekleşmektedir. İstek mesajı kullanıcı tarafından gönderilen ve sunucudan belirli bir bilgiyi isteyen mesajlar olup sunucu bu mesajları cevap paketleri ile cevaplamaktadır.

HTTP iletişimi Uniform Resource Identifier (URI) komutları ile sağlanır, URI, WWW gibi değişik isimlerle de anılmaktadır. URI, kaynağı isimle veya yer bilgisiyle bulmaya yarayan biçimlendirilmiş bir cümledir.

HTTP protokolü 80 numaralı portu varsayılan port olarak kullanarak sunucu bağlantısını gerçekleştirir. HTTP sunucu erişim gerçekleştikten sonra sunucudan herhangi bir işlemin gerçekleştirilmesi veya bir verinin bir kısmının getirilmesi istenilir. Sunucuda bir işlemin gerçekleştirmesi için yazılan sorgu parametre olarak isimlendirilir. Gerekli olan işleme göre parametre değişik şekillerde sunucuya gönderilir [34].

HTTP iletişimi istek ve cevap formatında sağlanmaktadır. HTTP protokolü Message Header, Field-name, Field-value ve Field-Content gibi değişik başlıklar içerir. Bu başlıklar son kullanıcı ve sunucu arasında doğru bir iletişimin olmasını sağlar.

HTTP İstek (HTTP Request)

Kullanıcı tarafından sunucudan veri istemek için oluşturulan istek paketi içinde istek satırı, başlık, istek başlığı içerir. İstek satırı, istek metodu ve tam URI bilgisini içerir. Eğer istek başlığında URI tam olarak belirtilmezse, host bilgisi, host-header kısmında belirtilir. Eğer istek mesajındaki URI cümlesi doğru şekilde tanımlanmazsa sunucu bu istekleri Bad Request hata mesajı ile cevaplar [35].

HTTP Cevap (HTTP Response)

HTTP Cevap mesajı, kullanıcı tarafından oluşturulan istek mesajlarına cevap vermek için oluşturulur. HTTP Cevap mesajları durum başlığı içerir, bu kısımda isteklerin durumları bildirilir, istek mesajlarındaki başlıklar, cevap mesajlarında da bulunmaktadır General Header, Response-Header ve Message Header gibi. HTTP mesajları işlemin nasıl yapılacağını veya daha fazla bilgi gerekip gerekmediğini belirtmek için durum kodlarını kullanmaktadır [35].

4. MATERYAL VE METOT

Web uygulamalarına yapılan saldırıların çeşidi ve sayısı her geçen gün artmaktadır. Bu saldırı türleri bilinen web tabanlı saldırı türleri olabileceği gibi sıfır gün saldırıları da olabilmektedir. Bilinen web tabanlı saldırı türlerinin denetimi kötüye kullanım olarak sınıflandırılarak İTD ile yapılabilirken, sıfır gün saldırılarının denetimi ATD ile gerçekleştirilebilmektedir. Dolayısıyla web uygulamalarına yapılan saldırıların denetimini daha etkin bir şekilde gerçekleştirmek için her iki denetim yönteminin kullanıldığı hibrit bir sistemin geliştirilmesi, web uygulamalarının güvenlik ihtiyacının karşılanması açısından önemli olacaktır.

Tez kapsamında, hibrit bir WUGD algoritmasını geliştirmek için iki farklı metot kullanılmıştır. Bilinen web tabanlı saldırıları denetlemek için İTD, normal HTTP istek yapısına uymayan HTTP isteklerinin denetimini yapmak ve sıfır gün saldırılarını denetmek için ATD kullanılmıştır. ATD BS, YSA ve BTYSA yöntemleri kullanılarak ayrı ayrı geliştirilerek, elde edilen sonuçlar karşılaştırılmıştır. Bu bölümde kullanılan materyaller ve metotlar anlatılmıştır.

4.1. İmza Tabanlı Denetim (İTD)

Kötüye kullanım olarak da tanımlanan imza denetimidir. Uzman tarafından incelenen ağ trafiğinin karakteristiği, davranışı ve içeriği incelenerek ortaya çıkarılır. Karakteristiği incelenen trafik eğer saldırı niteliği taşıyorsa imza olarak belirlenerek, İTD için imza veritabanına eklenir [36].

İmza tabanlı sistemler genelde hızlı çalışırlar, sadece imza veritabanında bulunan saldırı tiplerine karşı etkilidirler. Saldırı tespit sistemleri ve antivirüs programları imza tabanlı olarak çalışırlar. Yeni bir saldırı tekniği geliştirildiğinde belirtilen sistemin etkili olabilmesi için imza veritabanının saldırı tekniğine göre güncellenmesi gereklidir. Aksi takdirde yapılan saldırılara karşı etkisiz olmaktadır [37].

İTD bilinen web tabanlı saldırıların denetim sürecidir. İTD imza veri tabanına bağlı olarak imzaların sürekli güncellenmesi sonucunda etkilidir. En önemli dezavantajı imza olarak tanımlanmamış saldırı türlerine karşı etkili olmamasıdır. Ama ATD türüne göre daha hızlı çalıştığı için denetim hızı daha yüksektir [38].

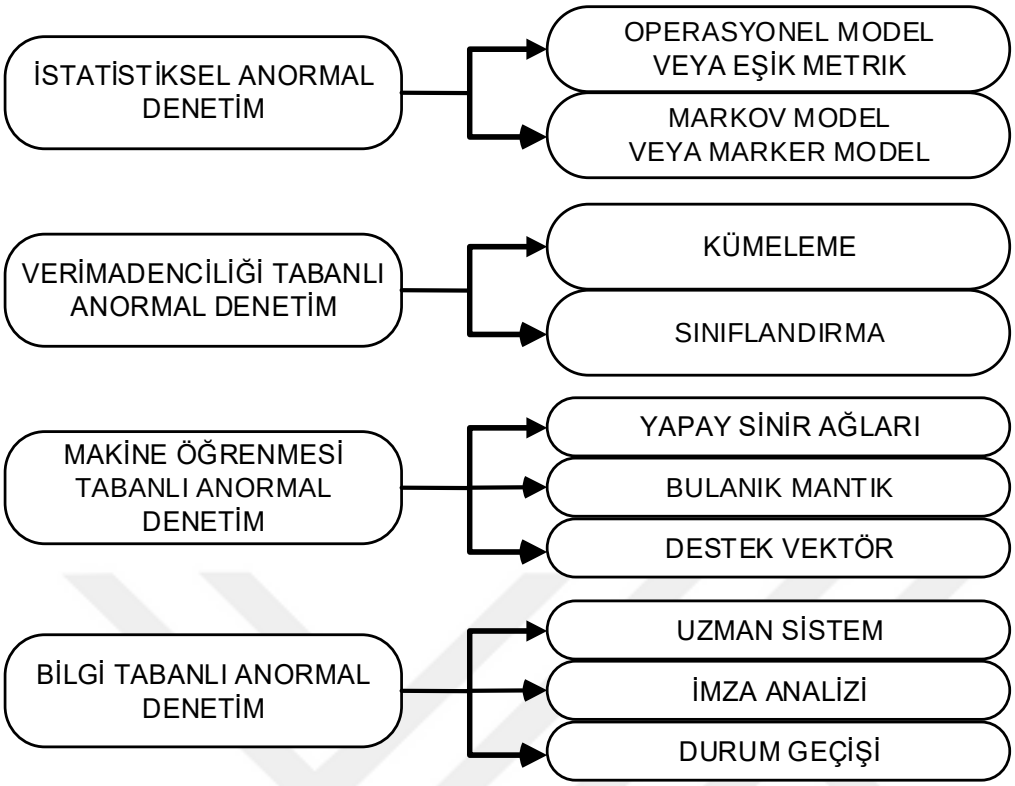
4.2. Anormal Tabanlı Denetim (ATD)

ATD, saldırı tespit yaklaşımlarından bir diğeridir. Kötüye kullanım tespitine göre bazı avantajlarının olmasının yanında en önemli avantajı bilinmeyen saldırıların tespitinde etkili olmasıdır. Bu yüzden son zamanlarda birçok anormallik tespit yöntemi geliştirilmiştir. Anormallik tespit sistemleri beklenen normal kullanım profillerinden sapma gösteren etkinlikleri anormallik olarak işaretlerler. Anormallik tespitinde, sistemin normal davranışı genellikle istatistiksel yöntemler yardımıyla elde edilir [39, 40].

Normal veriden farklı davranış gösteren HTTP istekleri anormal olarak nitelendirilmektedir. Öğrenilmiş veriyle eşleşmeyen istek tespit edildiği zaman sistem devreye girerek isteği engelleyerek düşürür. Veriler için doğru şekilde temsil bulma ve veri ile kullanmak için bir öğrenme algoritması belirleme sapma tespiti sistemleri için zor olabilir.

Anormallik tespitinde anormallik değeri kullanılır, bu özellik anormal tabanlı tespit yöntemlerinde önemli bir bilgidir. Eğer anormallik değeri hesabının parametreleri iyi şekilde seçilebilirse o zaman sistemin başarı oranı artar. Hesaplanan anormallik değerlerine göre web erişimleri normal ve anormal şeklinde sınıflanır ve anormal sınıfında yer alanlar saldırı olarak kabul edilir [41].

Bu bölümde ATD tekniklerden bahsedilmektedir. Şekil 4.1'de gösterildiği gibi anormal tabanlı denetim teknikleri; İstatistiksel anormal denetim, veri madenciliğine dayalı anormal denetim, makine öğrenmesi tabanlı anormal denetim, bilgi tabanlı anormal denetimdir.



Şekil 4.1. Anormal tabanlı denetim teknikleri [42].

4.2.1. İstatistiksel anormal denetim

İstatistiksel anormal denetim, saldırı algılama için kullanılan tekniklerden birisidir. İstatistiksel anormal denetim teknikleri, test edilen verilerde önemli ölçüde sapma olup olmadığını tespit etmek için kullanılır [43]. İstatistiksel anormal denetimi saldırı tespiti için, normal verilerin ortalama ve varyans gibi istatistiksel özellikleri kullanılır. Sistem her bir anormal değerlere skor atar, skor değeri belirli eşiği aştıktan sonra anormal değerler belirlenmiş olur. Istatistiksel anormal analizin iki aşaması bulunmaktadır. Birinci aşamada normal değerler için davranış profilleri oluşturulur. Daha sonra normal verilerden sapan verileri tespit etmek için birçok teknik kullanılır.

Güvenlik zaafiyetleriyle ilgili olarak ön bilgi gerektirmemesi, sıfır gün saldırılarına karşı etkili olması ve kötü amaçlı eylemler için doğru bildirim sunması istatistiksel anormal denetimin avantantajlarındandır.

4.2.2. Veri madenciliğine dayalı anormal denetim

Saldırı tespit sistemleri imza tabanında var olan bilinen saldırıları denetler. Veri madenciliği saldırı tespit sistemlerinin desen oluşturma süreçlerinde iyi bir çözüm olabilir ve büyük veri depoları veya önceki faydalı ve yok sayılmış modellerin ayrıştırılma işlemi olarak kullanılabilir. Veri madenciliği süreci veriyi azaltarak anormal denetim için daha anlamlı hale getirir [44, 45].

Veri madenciliğinin en önemli avantajı; normal veriyi, anormal veriden ayırarak gerçek saldırı verisini ortaya çıkarmasıdır. Veri madenciliğinde kümeleme ve sınıflandırma teknikleri kullanılabilmektedir.

4.2.3. Makine öğrenmesi tabanlı anormal denetim

Makine öğrenmesi belirli görev için, bir sistemin veya programın kendi performansını öğrenme ve geliştirme kabiliyeti olarak tanımlanabilir. Makine öğrenmesi önceki sonuçlara dayanılarak yeni edinilen bilgilere göre strateji yürüten bir sistemin performansının geliştirilmesine yoğunlaşmaktadır [46].

Bu özelliği makine öğrenmesi tekniğinin birçok alanda kullanımını sağlasa da, bazı eksik tarafları da bulunmaktadır. Birçok durumda, makine öğrenmesi tekniği istatistiksel teknikler ve veri madenciliği teknikleri ile örtüşmektedir [47]. Makine öğrenmesi tabanlı anormal denetimlerde YSA, bulanık mantık ve destek vektörü yaklaşımları kullanılmaktadır.

4.2.4. Bilgi tabanlı anormal denetim

Bilgi tabanlı anormal denetim, bilinen saldırı türleri ve sistem güvenlik açıklıkları ile ilgili olarak tanımlanan bilgiler doğrultusunda saldırıları engellemek ve güvenlik zaafiyetleri ile ilgili olarak denetim gerçekleştirme yöntemidir. Ancak en önemli gereksinimi saldırı türlerine göre bilgi bankasının düzenli olarak güncellenmesi gereklidir [48]. Bilgi tabanlı denetim tekniğinin bazı avantajları, bu tekniğin doğruluğu iyidir ve çok düşük yanlış alarm oranına sahiptir. Bilgi tabanlı anormal denetimlerde durum geçiş analizi, uzman sistemler ve imza tabanlı denetim yöntemleri kullanılmaktadır.

4.3. Bayes Sınıflandırma Teoremi

Bayes sınıflandırma teoremi, bir veri madenciliği yöntemidir. Bayes teoremi istatistiksel sınıfladırma yöntemidir. İstatistiksel sınıflandırma yöntemleri örnek veriler kullanılarak oluşturulan matematiksel model kullanılarak yeni verilerin önceden oluşturulan sınıflara eklenmesi şeklinde çalışırlar. Bayes teoreminde verilen hangi sınıfa hangi olasılıkl ait olduğu matematiksel olarak hesaplanarak tespit edilir.

4.3.1. İstatistiksel sınıflandırma

Sınıflandırma işlemini veriyi özelliklerine göre farklı sınıflara ayırma işlemidir. Sınıflandırma karşılaşılan verilerin hangi sınıfa ait olduklarını tahmin etmek için yapılır ve bu işlem eğitici yardımı ve sınıflandırma modeli oluşturularak gerçekleştirilir.

Sınıflandırma işleminde en önemli adım sınıflandırma modelinin oluşturulmasıdır. Model oluşturma süreci sırasıyla; önceden bilinen sınıfların tanımlanması, verilen her bir örneğin tanımlanan sınıflara eklenmesi, örnek öğrenme kümelerinin tanımlanması ve karar ağacı, sınıflandırma kuralı veya matematiksel formül oluşturulması şeklindedir.

Oluşturulan modelin kullanılması; sınıfları bilinmeyen örnek verilerin sınıflarının tahmini, oluşturulan modelin doğruluk derecesinin kestirimi, test kümesinin bilinen sınıf etiketlerinin tahmin sonuçlarının elde edilen sınıflandırma sonuçları ile karşılaştırılması, doğruluk derecesi test kümesindeki tahmin başarı oranının belirlenmesi şeklinde sıralanabilir.

Test verikümesi, öğrenme veri kümesinden bağımsızdır, fakat aynı dağılımlara sahip olmalıdır. Eğer modelin doğruluk derecesi kabul edilebilir ise model kabul başarı oranı kabul edilir ve yeni verileri sınıflandırmak için uygulama gerçekleştirilebilir. Günümüzde istatistiksel sınıflandırma yöntemlerinin uygulama alanlarına; kredi kartı başvuru değerlendirme, kredi başvurusu değerlendirme, pazarlama, hastalık teşhisi ve eğitim başvuruları değerlendirme alanları örnek verilebilir.

4.3.2. Diskriminant analizi

Diskriminant analizi, iki veya daha fazla sınıfın ayrımı ile ilgilenen çok değişkenli sınıflandırma analizllerinden biridir. Diskriminant analizin amacı, sınıfların hangi

değişkenler açısından birbirinden farklılaştığının ortaya çıkarmaktır. Diğer bir ifadeyle, sınıfların ayırıcı özelliklerinin belirlenmesidir [49].

Diskriminant fonksiyonları sınıflar arası farklılığa etki eden tahmin değişkenlerinin hangileri olduğunu ortaya çıkarır. Sınıflar arası farklılığa etki eden bu değişkenlere de diskriminant değişkenler adı verilir. Diskriminant analizinin bir diğer işlevi ise, sınıflardan herhangi birisine ait olan fakat hangi sınıftan geldiği bilinmeyen bir birimin ait olduğu sınıfı en az hata ile saptamaktır.

Diskriminant analizinin amacı ikiye ayrılabilir, bunlar;

- Diskriminant fonksiyonlarını saptayarak ve bu fonksiyonlar aracılığıyla sınıflar arası ayırıma en fazla etki eden ayırıcı değişkenleri belirlemek,
- Hangi sınıftan geldiği bilinmeyen bir birimin hangi gruba dâhil edileceğini belirlemektir.

4.3.3. Bayes sınıflandırma teoremi

İstatistiksel karar yöntemlerinden biri olan bayes teoremi, olasılıklı bir bilginin incelenmesine objektif bir bakış açısını ve bilimsel gerçekten ziyade bilginin aşamalarını esas almaktadır [50].

Olasılıksal bir sınıflandırma olan BS, bir sınıfın olasılığını tahmin etmek için koşullu önsel değerleri kullanmaktır. BS, bazı öğrenme problemlerinde yaygın olarak kullanılan en pratik yaklaşımdır. Olasılık ve istatistik biliminde, klasik görüş olan sıklıkçı görüşün karşısındaki alternatif görüş yaklaşım bayes teoremidir. Bayes teoremi bir olayın öngörülen sonucuyla ilgili inanışların bir ölçüsü olarak görülmektedir. İki taraf arasında yıllardır süregelen tartışmada sıklıkçılar inanışın matematiksel teorilerde yer alamayacak subjektif bir kavram olduğunu savunurken; Bayes teoremi bu eleştiriyi kabul etmekle beraber, objektif bir yolla inanışların güncellenmesinin bu durumu ortadan kaldıracağını savunmaktadır [51].

Bayes teoremi, bir olayın meydana gelmesinde birbirinden bağımsız birden fazla etkenin olması koşulunda, olayın hangi etkenin etkinliği ile ortaya çıktığını gösteren istatistiksel bir sınıflandırma teoremidir [52, 84]. Eşitlik 4,1'de BS kuralı verilmiştir.

A ve B rasgele olaylar olmak üzere;

$$P(A|B)\frac{P(B|A)P(A)}{P(B)} \quad (4.1)$$

P(A) : A olayının bağımsız olasılığı,
P(B) : B olayının bağımsız olasılığı,
P(B|A) : A olayının olduğu bilindiğinde B olayının olasılığı,
P(A|B) : B olayının olduğu bilindiğinde A olayının olasılığı,

Bayes kuralına dayanarak P(A|B)'yi maksimum yapan durumlar hesaplanabilir. "E" A olayının bütün durumlarının kümesi ifade etmek üzere;

A_{MAX} = argmax A∈E P(A | B)

= argmax A∈E $\frac{P(B|A)P(A)}{P(B)}$

= argmax A∈E $P(B|A)P(A)$

Burada P(B) sabit olarak göz ardı edilebilir. Sınıflandırıcının görevi yeni bir örnek için doğru sınıfı tahmin etmektir. Sınıflandırma problemi öncül olasılıklar kullanılarak şu şekilde formüle edilebilir:

P(v |X) = örnek durum için olasılık,
X = <a_1, a_2, …, a_i> v sınıfının örnekleri,
X örneğine P(v|X) olasılığını maksimum yapan sınıf etiketi atanarak.

BS Teoremi:

P(v|X) = P(X|v) P(v)/P(X)

P(X) bütün sınıflar için sabittir,
P(v) = sınıfın bütün örneklere karşı göreli frekansı,
P(v|X) maksimum = P(X|v) P(v) maksimum,
P(X|v) = P(a_1, a_2, …, a_i |v)'yı hesaplamak çok zordur, çok fazla ihtimal gerektirmektedir.

Özniteliklerin bağımsızlığı varsayımına göre; P(a_1, a_2, … a_k |v) = P(a_1| v)·…·P(a_k |v)

Eğer k. öznitelik kesikli değerde ise P(a_k | v), v sınıfında k. öznitelik olarak a_k değerine sahip örneklerin göreli frekansı olarak tahmin edilmektedir.

BS'yi örneklendirmek için tenis maçının hava şartlarına göre oynanma olasılığının tahmini yapılmaktadır. Çizelge 4.1'de örnek veri kümesi verilmiştir.

Çizelge 4.1. Örnek veri kümesi

Sıra	Görünüm	Sıcaklık	Nem	Rüzgâr	Oyun
1	Güneşli	Sıcak	Yüksek	Yok	Evet
2	Güneşli	Sıcak	Normal	Var	Hayır
3	Bulutlu	Sıcak	Yüksek	Yok	Evet
4	Yağmurlu	Ilık	Yüksek	Yok	Evet
5	Yağmurlu	Serin	Normal	Var	Hayır
6	Yağmurlu	Serin	Normal	Var	Hayır
7	Bulutlu	Serin	Normal	Var	Evet
8	Güneşli	Ilık	Yüksek	Yok	Evet
9	Güneşli	Serin	Normal	Yok	Evet
10	Yağmurlu	Ilık	Yüksek	Var	Hayır

Çizelge 4.2'de, çizelge 4.1'de bulunan verilerin frekans ve olasılık değerleri verilmiştir.

Çizelge 4.2. Örnek veri kümesi frekans değerleri

		Frekanslar		Olasılıklar	
Durum	**Detay**	**Oyun Evet**	**Oyun Hayır**	**Oyun Evet**	**Oyun Hayır**
Görünüm	Güneşli	3	1	3/6	1 /4
	Bulutlu	2	0	2/6	0 /4
	Yağmurlu	1	3	1/6	3 /4
Sıcaklık	Sıcak	2	1	2/6	1 /4
	Ilık	2	1	2/6	1 /4
	Serin	2	2	2/6	2 /4
Nem	Yüksek	4	1	4/6	1 /4
	Normal	2	3	2/6	3 /4
Rüzgâr	Yok	5	0	5/6	0 /4
	Var	1	4	1/6	4 /4
Oyun		6	4	6 /10	4 /10

Çizelge 4.2'deki frekans ve olasılık değerleri kullanılarak yeni bir durumla karşılaşıldığında tenis oynama olasılığının sınıflandırma tahmini gerçekleştirilmektedir. Çizelge 4.3'te yeni bir durum örnek olarak verilmiştir.

Çizelge 4.3. Yeni durum örnek tahmin verisi

Görünüm	Sıcaklık	Nem	Rüzgar	Oyun
Güneşli	Serin	Yüksek	Var	?

P(EVET) = Görünüm | Güneşli * Sıcaklık | Serin * Nem | Yüksek * Rüzgar | Var
P(HAYIR) = Görünüm | Güneşli * Sıcaklık | Serin * Nem | Yüksek * Rüzgar | Var

P(EVET) = 3/6 * 2/6 * 4/6 * 1/6 = 0,5 * 0,33 * 0,66 * 0,16 = 0.017424
P(HAYIR) = 1/4 * 2/4 * 1/4 * 4/4 = 0,25 * 0.5 * 0,25 * 1 = 0,03125

Veri kümesindeki olasılık dağılımına dayanarak her bir sınıf için olasılıklar hesaplanmaktadır. Her bir özniteliğin olasılığı hesaba katılarak, bütün öznitelikler eşdeğer

önemde ele alınır ve olasılıklar çarpılır. Bir sınıfın toplam olasılığını hesaba katılarak özniteliklerin olasılıklarıyla çarpılarak, yeni durumun tahimini gerçekleştirilmektedir.

P(EVET) = 0.017424 * 6/10 = 0,0104544
P(HAYIR) = 0,03125 * 4/10 = 0,0125

P(HAYIR) > P(EVET) olduğu için, yeni durum olasılıüı "HAYIR" olarak sınıflandırma anlamına gelmektedir. Her sınıf için olasılıklar hesaplanarak, örnek olasılığı yüksek olan sınıfa atanmaktadır.

4.4. Yapay Sinir Ağları (YSA)

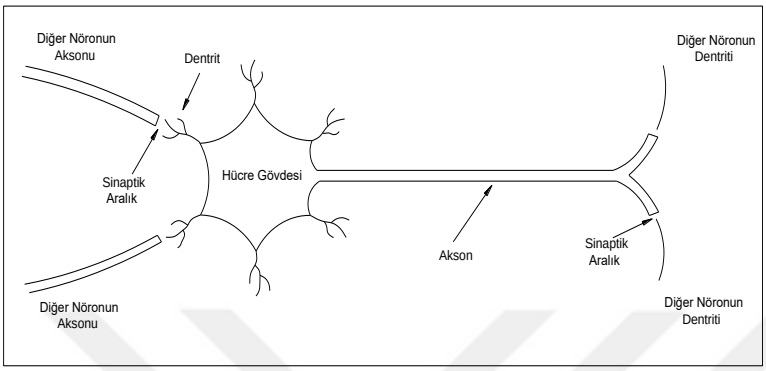
Bilimsel olarak karşılaşılan problemler doğrusal ve doğrusal olmayan olarak nitelendirilmekte ve modellenebilmektedir. Ancak gerçekte pek çok problem doğrusal değildir. Doğrusal olmayan problemler özellikle değişken sayısının artışına bağlı olarak zorluk dereceleri artmaktadır. Bu tip problemlerin belirleyici yöntemlerle çözümünde sonuca ulaşılamama ya da çözüm süresinin çok uzun olması gibi sorunlarla karşılaşılmaktadır. Bu nedenle doğrusal olmayan problemlerin çözümünde sezgisel yöntemler problemin çözümünde kullanılan yaklaşımlardandır.

Sezgisel yöntemlerden biri olan YSA, canlılarda bulunan sinir sistemi yapısının elektronik ortamda modellenmesi sonucu elde edilen temel zekâ niteliklerine sahip elektronik yapılardır. YSA'lar, canlılarda olduğu gibi belirli koşullar altında nasıl tepki vereceğini öğrenerek, benzer koşullarda karar alma yeteneğine sahiptirler. YSA'lar veri kümeleri sayesinde öğrenerek; ilişkilendirme, tahmin, sınıflandırma, genelleme, iyileştirme ve nitelik belirleme yetenekleri kazanabilmektedirler.

Haykin YSA'nın tanımını "bir sinir ağı, basit işlem birimlerinden oluşan, deneyimsel bilgileri biriktirmeye yönelik doğal bir eğilimi olan ve bunların kullanılmasını sağlayan yoğun bir şekilde paralel dağıtılmış bir işlemcidir. Bu işlemci iki şekilde beyin ile benzerlik göstermektedir" şeklinde yapmıştır [53-56].

Beynin bilgi işleme yöntemine uygun olarak YSA, bir öğrenme sürecinden sonra bilgiyi toplama, hücreler arasındaki bağlantı ağırlıkları ile bu bilgiyi saklama ve genelleme yeteneğine sahip paralel dağılmış bir işlemcidir.

Biyolojik öğrenme kuralına göre; bir nörondan dentrit yoluyla gelen ve bir akson yoluyla alınan giriş, onun bir darbe üretmesine sebep olur. Sonraki aksonal girişlerin darbe üretmesi olasılığı artar. Böylelikle yapılan davranışın sonucu ortaya çıkar. Şekil 4.1'de biyolojik sinir hücresinin yapısı gösterilmiştir [56].



Şekil 4.2. Biyolojik sinir hücresi

YSA çalışmaları 1950'li yıllarda başlamıştır. 1950'lerde Frank Rosenblatt tarafından basit sinir modellerine dayalı bir hesaplama modeli önerilmiş ve ardından perceptron diye bilinen tek katmanlı ilk YSA modeli ortaya çıkmıştır [58,58].

1954 yılında B.G. Farley ve W.A. Clark tarafından bir ağ içerisinde uyarılara tepki veren, uyarılara adapte olabilen model oluşturmuşlardır. 1960'lı yıllarda Widrow ve Hoff, bu basit nöron modellerini kullanarak öğrenebilen ve ilk uyarlanabilen sistemler üzerinde çalışmış ve delta kuralı olarak bilinen; gerçek çıkış ile istenen çıkış arasındaki farka eşit bir hata terimi kullanarak ağ ağırlıklarının değiştirildiği bir öğrenme kuralı geliştirilmiştir [59].

1982'de J.J. Hopfield tarafından yapılan çalışmada, nöronların karşılıklı etkileşimlerine dayanan bir sinir hesaplama modeli önerilmiştir. Bu model, bir enerji fonksiyonunu, alabileceği en az değerine indiren 1. mertebe doğrusal olmayan diferansiyel denklemlerden oluşmuştur. Hopfield; ağ seviyesinde, tek tek nöron seviyesinde var olmayan hesaplama kapasitesinin bulunduğunu öne sürmüştür [60].

1986'da Grossberg, Adaptive Resonance Theory (ART) yani Uyarlanabilir Rezonans Teorisi adında bir YSA yapısını geliştirmiştir. O sıralarda Kohenen'de "kendi kendini düzenleyen nitelik haritasını (self-organizing maps) geliştirmiştir [61]. 1986'da Rumelhart ve arkadaşları paralel dağılımlı işleme adlı kitaplarında, ileri beslemeli (feed-forward)

ağlarda yeni öğrenme modeli olan hatanın geriye yayılım algoritmasını (backpropagation algorithm) geliştirmişlerdir [62].

Günümüzde endüstride birçok YSA uygulamasında ileri beslemeli geri yayılımlı öğrenme yöntemi ile bunun değişik türleri kullanılmaktadır. Geriye yayılma algoritması, kullanımı çok yaygın olan ve öğrenilmesi kolay bir ağdır.

YSA'lar, doğrusal problemlerin çözümlerinde de kullanılabilmesine rağmen doğrusal olmayan problemlerde daha yaygın kullanıma sahiptirler. Bunun sebebi doğrusal olmayan problemlerin sezgisel olmayan yöntemlerle çözümünde karşılaşılan problemlerdir. YSA'nın problemleri çözümde sağladığı hızın arkasında paralellik özelliği yatmaktadır. YSA'ların yapı taşı olan sinir hücrelerinin eş zamanlı olarak hesap yapabilmesi hesaplama süresini düşürmektedir. YSA'nın problem çözümünün daha kolay olmasının sebebi uyarlanabilirlik ve tasarım kolaylığıdır. Uyarlanabilirlik aynı YSA mimarisinin, farklı problemler için kullanılabilmesidir. Sonuçları farklılaştıran ise YSA'nın yapısı değil; problemin girdileri ve eğitim sürecidir. Tasarım kolaylığı ise yapı taşı olan sinir hücrelerinin tüm YSA'larda aynı olmasından kaynaklanmaktadır.

YSA'nın tercih edilmesindeki diğer bir sebep ise öğrenebilirlik özelliğidir. YSA'lar eğitim esnasında karşılaştıkları durumları ve sonuçlarını öğrenip aynı durumlarla tekrar karşılaştıklarında kararlı olarak aynı sonucu üretebilmektedirler. Hatta YSA sayesinde çözüm yolu tam olarak bilinmeyen sınıflandırma v.b. problemlerde dahi örnek veriler kullanılarak çözülebilmektedir. YSA'lar genelleme adı verilen yetenekleri ile eğitim esnasında karşılaşmadıkları durumlar için de isabetli sonuçlar üretebilmektedir.

YSA'nın tercih sebebi olmasındaki diğer bir özelliği ise ağın yapısında karşılaşılabilecek hataların hoş görülmesidir. Yapay sinir hücrelerinin paralel ve bağımsız çalışan yapısı sayesinde, ağın bir bölümü işlevini yerine getiremezse dahi kalan kısım işlevini yerine getirmektedir. YSA öğrenebilirlik özelliği sayesinde yeni yapısıyla öğrenme süreçlerinden geçerek yine isabetli sonuçlar üretebilmektedir.

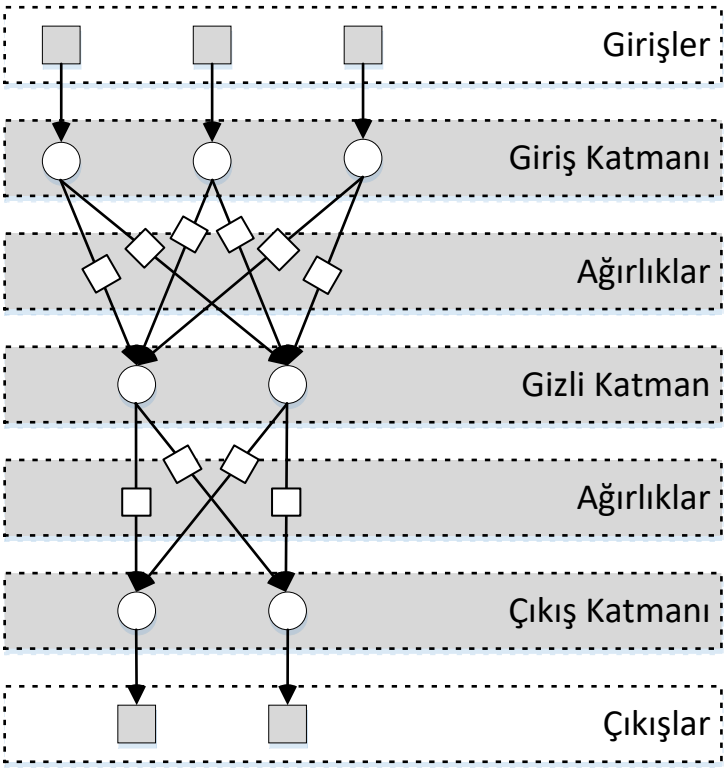
YSA'ların problem çözümüne yönelik olumlu özellikleri olduğu gibi olumsuz özellikleri de vardır. Bunlardan en önemlisi YSA'nın kullanılmadan önce bir eğitim sürecinden geçmesi gerekliliğidir. Eğitim süreci için ise probleme ait girdi ve bu girdiler neticesinde elde edilmesi beklenen sonuçları içeren veri kümelerine ihtiyaç vardır. Veri kümelerinin temini

problemin yapısına göre son derece zor ölçümleri gerektirebilmektedir. Diğer bir olumsuz özelliği karar verme esnasında, eğitim sürecinde karşılaştığı durumlara bağımlı kalmasıdır. Bu durumlardan bağımsız sonuç üretmeleri mümkün değildir.

YSA günümüz teknolojisinde çok farklı alanlarda kullanılmaktadır. Özellikle son 20-25 yılda hızlı bir şekilde gelişen YSA birçok araştırma ve çalışmadan sonra her geçen gün değişik uygulamalarda kullanılmaya başlanmıştır. YSA'nın uygulama alanları örüntü tanıma, ses tanıma, karakter, imza, parmak izi tanımada, damar tanıma, veri iletişimi, robotik sistemler, üretim sistemlerinin programlanması, ürün analizi gibi çok geniş alanda kullanım alanları mevcuttur.

4.4.1. Yapay sinir ağlarının yapısı

Sinir hücreleri genellikle birkaç katman halinde dizilerek bir yapay sinir ağını meydana getirirler. Başka bir anlatımla, genellikle bir yapay sinir ağı birden fazla katmandan ve birden fazla yapay sinir hücresinden meydana gelir. İlk katman genellikle giriş katmanıdır. Çıkış katmanı ise son katmandır. Aradaki diğer katmanlar ise gizli katman ya da ara katman olarak adlandırılırlar. Bir ağda birden fazla gizli katman olabilir. Şekil 4.3'de örnek bir yapay sinir ağı gösterilmiştir.



Şekil 4.3. Genel yapay sinir ağı

Giriş katmanı

Giriş katmanı, dışarıdan giriş bilgilerini alarak bir sonraki katman olan ara katmanlara iletirler. Girişler (x1, x2, x3, …. xn) kendinden önceki sinir hücresinden veya dış dünyadan sinir ağına gelebilir. Bir sinir hücresi genelde gelişi güzel birçok girdileri alabilir.

YSA'da giriş sayısı çözüm üretilecek problemin özelliğine göre değişir. Problemin çözümü için gerekli optimum sayıda giriş kullanılmalıdır. Giriş sayısının gerektiğinden az olması problemin gerektiği gibi çözümünde başarısız olur. Giriş sayısının gerektiğinden fazla olması ise YSA eğitiminin süresini artırarak, eğitim performansını olumsuz olarak etkiler.

Ara katman (Gizli katman)

Giriş katmanından gelen bilgiler işlenerek çıkış katmanına gönderilirler. Ara katman sayısı ağdan ağa değişebilir. Bazı yapay sinir ağlarında ara katman bulunmadığı gibi bazı yapay sinir ağlarında ise birden fazla ara katman bulunmaktadır. Ara katmanlardaki nöron sayıları giriş ve çıkış sayısından bağımsızdır. Birden fazla ara katman olan ağlarda ara katmanların kendi aralarındaki nöron sayıları da farklı olabilir. Ara katmanların ve bu katmanlardaki nöronların sayısının artması hesaplama karmaşıklığını ve süresini arttırmasına rağmen yapay sinir ağının daha karmaşık problemlerin çözümünde de kullanılabilmesini sağlar.

Çıkış katmanı

Çıkış katmandaki işlem elemanları ara katmandan gelen bilgileri isleyerek ağın girdi katmanından sunulan girdi seti için üretilmesi gereken çıktıyı üreten katmandır ve üretilen çıktı dışarıya gönderilir. Geri beslemeli ağlarda bu katmanda üretilen çıktı kullanılarak ağın yeni ağırlık değerleri hesaplanır.

4.4.2. YSA'ların genel özellikleri

YSA'nın hesaplama ve bilgi işleme özelliği, paralel dağılmış yapısından, öğrenebilme ve genelleme yeteneğinden aldığı söylenebilir. Genelleme; eğitim ya da öğrenme sürecinde

karşılaşılmayan girişler için de YSA'nın uygun tepkileri üretmesi olarak tanımlanır. Bu özellikler, YSA'nın karmaşık problemleri çözebilme yeteneğini göstermektedir.

Doğrusal Olmama

YSA'larının en önemli özelliklerinden birisidir. YSA' nın temel iş süreci elemanı olan hücre doğrusal değildir. Dolayısıyla hücrelerin birleşmesinden meydana gelen YSA da doğrusal değildir ve bu özellik bütün ağa yayılmış durumdadır. Bu özelliği ile doğrusal olmayan karmaşık problemlerin çözümünde en önemli araç olmuştur.

Öğrenebilirlik

YSA'nın arzu edilen davranışı gösterebilmesi için amaca uygun olarak ayarlanması gerekir. Bu, hücreler arasında doğru bağlantıların yapılması ve bağlantıların uygun ağırlıklara sahip olması gerektiğini ifade eder. YSA' nın karmaşık yapısı nedeniyle bağlantılar ve ağırlıklar önceden ayarlı olarak verilemez ya da tasarlanamaz. Bu nedenle YSA, istenen davranışı gösterecek şekilde ilgilendiği problemden aldığı eğitim örneklerini kullanarak problemi öğrenmelidir.

Genelleme

YSA, ilgilendiği problemi öğrendikten sonra eğitim sırasında karşılaşmadığı test örnekleri için de arzu edilen tepkiyi üretebilir. Örneğin, karakter tanıma amacıyla eğitilmiş bir YSA, bozuk karakter girişlerinde de doğru karakterleri verebilir ya da bir sistemin eğitilmiş YSA modeli, eğitim sürecinde verilmeyen giriş sinyalleri için de sistemle aynı davranışı gösterebilir.

Uyarlanabilirlik

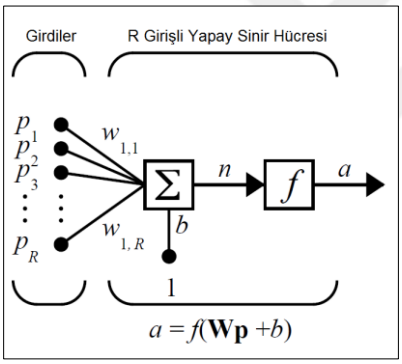
YSA, ilgilendiği problemdeki değişikliklere göre işlem elemanlarının ağırlıklarını ayarlar. Yani, belirli bir problemi çözmek amacıyla eğitilen YSA, problemdeki değişimlere göre tekrar eğitilebilir, değişimler devamlı ise gerçek zamanda da eğitime devam edilebilir. Bu özelliği ile YSA, uyarlamalı örnek tanıma, sinyal işleme ve denetim gibi alanlarda etkin olarak kullanılır.

Hata Toleransı

YSA, çok sayıda paralel dağılmış bir yapıya sahiptir ve ağın sahip olduğu bilgi, ağdaki bütün bağlantıların üzerine dağılmış durumdadır. Bu nedenle, eğitilmiş bir YSA'nın bazı bağlantılarının hatta bazı hücrelerinin etkisiz hale gelmesi, ağın doğru bilgi üretmesini etkilemez. Bu nedenle, geleneksel yöntemlere göre hatayı tolere etme yetenekleri son derece yüksektir.

4.4.3. Yapay sinir ağı hücresi

Biyolojik sinir sistemlerinde olduğu gibi YSA'lar da nöron adı verilen yapay sinir hücrelerinden oluşmaktadır. Yapay sinir hücrelerinde de sinyali alıp, toplayıp, işleyip sonuçları ilettikleri kısımlar vardır. Bu kısımlar Şekil 4,4'de girdiler (p) , ağırlıklar (w), birleştirme işlevi (Σ), etkinleştirme işlevi(f) ve çıktılar (a) olarak gösterilmektedir.



Şekil 4.4. R adet girişi tek çıkışı olan yapay sinir hücresi modeli [63].

Girdiler, nöronların giriş birimleridir. Girdilerin kaynağı YSA'nın girişinde verilen değerler olabileceği gibi başka bir nöronun çıkışı da olabilmektedir. Her bir girdinin sonuca olan etkisini belirlemek için ağırlıklar kullanılmaktadır. Girdilere verilen değerler bu ağırlıklarla çarpma işlemine tabi tutularak çekirdeğe iletilmektedirler. Çekirdeğe ulaşan ağırlıklarla çarpılmış girdi değerlerini birleştirme işlevi tek bir değere dönüştürmektedir. Bu fonksiyon genellikle toplama işlemi olmakla birlikte bazı durumlarda Sigma-Pi gibi daha karmaşık kurallar da belirlenebilmektedir. Birleştirme fonksiyonlarında meyil (bias) ve eşik (threshold) değerlerini kullanmaktadır.

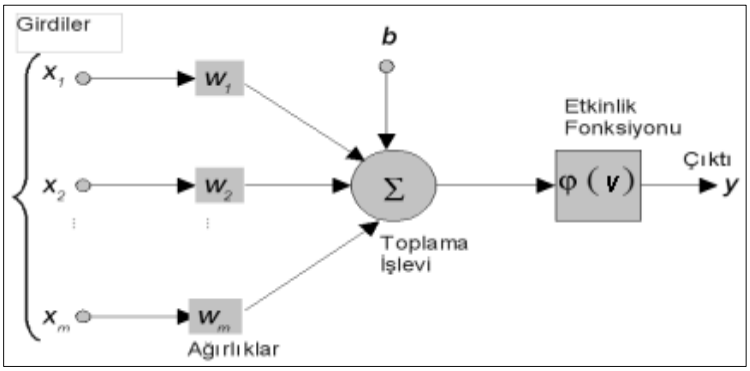
Birleştirme işlevinden elde edilen değer, etkileşim işlevine iletilmektedir. Etkileşim işlevleri nöronun çıkışını belirli bir değer aralığında tutmaktadırlar. YSA'nın doğrusal olup olmayacağını seçilen etkileşim işlevi belirlemektedir. Çözümü aranan sorun doğrusal ise

çözüm için kullanılacak YSA'da doğrusal etkileşim işlevleri seçilmelidir. Çözümü aranan sorun doğrusal değil ise çözüm için kullanılacak YSA'da doğrusal olmayan etkileşim işlevleri seçilmelidir. En yaygın kullanılan etkileşim işlevleri sigmoid ve tanjant hiperbolik fonksiyonlarıdır. Geri yayılımlı ağlarda etkileşim işlevlerinin türevleri kullanılmaktadır. Bu sebeple etkileşim işlevinin türevinin, kolay hesaplanabilir olması tercih edilmektedir. Etkileşim İşlevlerinin sonucu aynı zamanda nöronun çıktısıdır. Çıktılar YSA'nın sonucu olabileceği gibi diğer nöronların girdisi de olabilmektedir.

En basit yapay sinir hücresi Şekil 4,5'de görüleceği üzere dış ortamdan ya da diğer nöronlardan alınan veriler yani girişler, ağırlıklar, birleştirme fonksiyonu, aktivasyon fonksiyonu, sapma ve çıkış olmak üzere 6 ana bileşenden oluşmaktadır. Dış ortamdan alınan veriler ağırlıklar aracılığıyla nörona bağlanır ve bu ağırlıklar ilgili girişin etkisini belirler. Toplama fonksiyonu ise net girişi hesaplar.

Net giriş; dış ortamdan alınan verilerle ve bu verilerin ağırlıkların çarpımının bir sonucudur. Aktivasyon fonksiyonu işlem süresince net çıkısı hesaplar ve bu işlem aynı zamanda nöron çıkısını verir.

Genelde aktivasyon fonksiyonu doğrusal olmayan (nonlineer) bir fonksiyondur. Şekil 4.5'de görülen b bir sabittir, bias veya aktivasyon fonksiyonunun eşik değeri olarak adlandırılır.



Şekil 4.5. Temel yapay sinir ağı hücresi.

Girdiler (x1 , x2 ....xm), diğer hücrelerden ya da dış ortamlardan hücreye giren bilgilerdir. Bunlar ağın öğrenmesi istenen örnekler tarafından belirlenir. Ağırlıklar (W_1, W_2, ... W_m), girdi kümesi veya kendinden önceki bir tabakadaki başka bir işlem elemanının bu işlem elemanı üzerindeki etkisini ifade eden değerlerdir. Her bir girdi, o girdiyi işlem elemanına

bağlayan ağırlık değeriyle çarpılarak, toplam fonksiyonu aracılığıyla birleştirilir. Toplam fonksiyonu eşitlik 4.2'de verilen eşitlikle hesaplanmaktadır.

$$\text{net} = \sum_{i=1}^{n} W_i X_i + b \quad (4.2)$$

Toplam fonksiyonundan elde edilen değer doğrusal ya da doğrusal olmayan türevlenebilir bir transfer fonksiyonundan geçirilerek işlem elemanının çıktısı olarak eşitlik 4.3'de verilen eşitlikle hesaplanmaktadır.

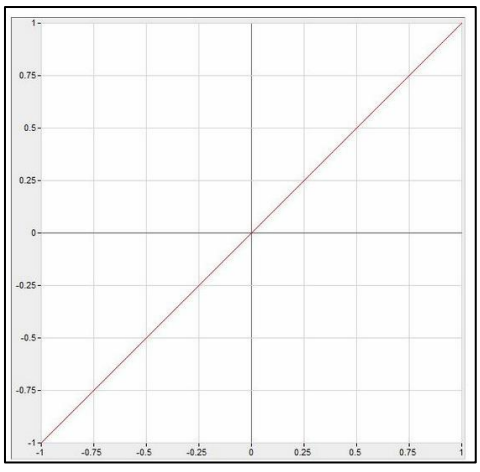
$$Y = (\text{net}) = f(\sum_{i=1}^{n} WiXi + b) \quad (4.3)$$

4.4.4. Aktivasyon fonksiyonu

Matematiksel olarak modellenmiş bir yapay sinir hücresinin birleştirme fonksiyonundan elde edilen net girdiyi bir işlemden geçirerek hücre çıktısını belirleyen ve genellikle doğrusal olmayan bir fonksiyondur. Sıkıştırma, transfer, işlemci veya eşik fonksiyonu olarak da isimlendirilebilir. En çok kullanılan aktivasyon fonksiyonları aşağıda gösterilmiştir.

Doğrusal aktivasyon Fonksiyonu

Doğrusal bir problemi çözmek amacıyla kullanılan bu fonksiyon, hücrenin net girdisini doğrudan hücre çıkışı olarak vermektedir. Doğrusal aktivasyon fonksiyonu matematiksel olarak Y=A*v şeklinde tanımlanabilir. A sabit katsayıdır. Şekil 4.6'de doğrusal aktivasyon fonksiyonunun grafiği gösterilmektedir.



Şekil 4.6. Doğrusal aktivasyon fonksiyonu

Sigmoid aktivasyon fonksiyonu

Yapay sinir ağlarında kullanılan bu fonksiyonun türevi alınabilir, sürekli ve Doğrusal olmayan bir fonksiyon olması nedeniyle doğrusal olmayan problemlerin Çözümünde kullanılan en yaygın aktivasyon fonksiyonudur. Eşitlik 4.4'da sigmoid aktivasyon fonksiyonu matematiksel olarak gösterilmiştir.

$$\hat{y} = f(v_i) = \frac{1}{1+e^{-\beta v_i}} \tag{4.4}$$

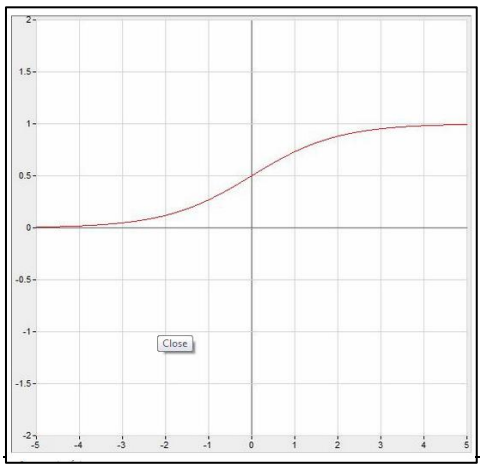
Eşitlik 4.4' da verilen sigmoid tipli aktivasyon fonksiyonunun alabileceği değerlerin aralığı (değer kümesi) eşitlik 4.5'de gösterilmiştir.

$$0 < \hat{y} < 1 \quad \hat{y} \epsilon R \tag{4.5}$$

Eşitlik 4.6' da verilen sigmoid aktivasyon fonksiyonunda, i, v değişkeninin değeri yerine yazıldığında eşitlik 4.4' de gösterilen ifadeye ulaşılmaktadır.

$$\hat{y} = f(v_i) = \frac{1}{1+e^{-\beta[\sum_{i=1}^{n} w_i x_i - Q]}} = \frac{1}{1+e^{-\beta v_i}} \tag{4.6}$$

Şekil 4.7'da Sigmoid aktivasyon fonksiyonunun grafiği gösterilmektedir.



Sekil 4.7. Sigmoid aktivasyon fonksiyonu

Sigmoid aktivasyon fonksiyonu gibi lojistik dağılım gösteren fonksiyonlar, özellikle olasılık modellerinde (logit, probit, vs.) mükemmel bir seçim olarak kullanılabilir.

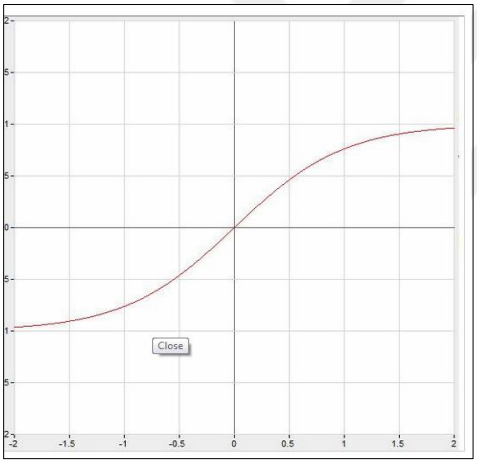
Hiperbolik Tanjant Aktivasyon Fonksiyonu

YSA'larda yoğun olarak kullanılan bir diğer aktivasyon fonksiyonu olmasıyla birlikte bu fonksiyon, sigmoid fonksiyonuna benzer bir fonksiyondur. Ayrıca bu fonksiyonun, sigmoid tipli fonksiyona göre avantajı, negatif değişkenler içeren zaman serilerini de modelleyebilmesidir. Kısaca özetlersek sigmoid fonksiyonunda çıkış değerleri 0 ile 1 arasında değişirken, hiperbolik tanjant fonksiyonunun çıkış değerleri -1 ile 1 arasında değişmektedir. Eşitlik 4.7'da hiperbolik tanjant tipli aktivasyon fonksiyonu verilmiştir.

Eşitlik 4.6'da verilen, hiperbolik tanjant tipli aktivasyon fonksiyonunda, i, v değeri matematiksel olarak yerine yazıldığında eşitlik 4.7'a ulaşılmaktadır.

$$\hat{y} = f(v_i) = \frac{e^{\beta[\sum_{i=1}^{n} w_i x_i - Q]} - e^{-\beta[\sum_{i=1}^{n} w_i x_i - Q]}}{e^{\beta[\sum_{i=1}^{n} w_i x_i - Q]} + e^{-\beta[\sum_{i=1}^{n} w_i x_i - Q]}} = \tanh(\beta v_i) \quad (4.7)$$

Şekil 4.8'de hiperbolik tanjant tipli aktivasyon fonksiyonun grafiği gösterilmektedir.



Şekil 4.8. Hiperbolik tanjant aktivasyon fonksiyonu

4.4.5. Yapay sinir ağlarının sınıflandırılması

Çözüm bekleyen problemin yapısına göre geliştirilmiş pek çok YSA türü vardır. YSA'lar genel olarak yapılarına göre ve öğrenme yöntemlerine göre sınıflandırılmaktadırlar. Yapılarına göre sınıflandırmada nöronlar arası bağlantılar esas alınırken, öğrenme yöntemine göre sınıflandırmada ağırlıkların en iyi sonucu vermesi için nasıl iyileştirildiği esas alınmaktadır.

YSA'ların yapılarına göre sınıflandırılması: YSA'ların yapılarına göre sınıflandırılmasında ileri beslemeli ve geri yayılımlı olmak üzere verinin akış yönünü esas alan iki sınıf söz konusudur. İleri beslemeli YSA'larda veri giriş katmanından başlayarak çıkış katmanına

doğru tek yönlü bir akış izler. Her bir sinir hücresi ancak bir önceki katmandan girdileri alır işler ve çıktıları ancak bir sonraki katmana iletir. Öğrenme oranları nispeten daha düşüktür. Geriye yayılımlı ağlarda ise ileri beslemeli ağ yapısına ek olarak katmanların çıkışından önceki katmanlara doğru bağlantılar kurulmaktadır. Bu bağlantılar sayesinde katman çıkışlarının türevleri alınarak önceki katmanlara yeniden giriş olarak verilmektedir. Bu durum her bir katmandaki ağırlıkların çıkışa göre yenilenmesini sağlamaktadır. Her bir katmanda hem mevcut girişlerin hem de önceki girişlerin bulunması nedeniyle tahmin uygulamalarda yüksek başarım elde edilmektedir. İleri beslemeli ağlara daha yavaş olmakla birlikte, daha isabetli sonuçlar üretmektedirler [64].

Yapay sinir ağları isleyiş olarak benzer olmalarına rağmen herhangi bir tasarım ve isleyiş standardı bulunmamaktadır. YSA'lar, birbirleri ile bağlantılı sinir hücrelerinden oluşurlar. Her bir sinir hücresi arasındaki bağlantıların yapısı ağın yapısını belirler. Hedeflenen değere ulaşmak için bağlantıların nasıl değiştirileceği öğrenme algoritması tarafından belirlenir. Kullanılan öğrenme algoritmasına göre, hatayı sıfıra indirecek şekilde, ağın ağırlıkları değiştirilir. YSA'lar yapılarına göre sınıflandırılırlar.

İleri beslemeli yapay sinir ağları

İleri beslemeli yapay sinir ağları tek katmanlı ve çok katmalı olmak üzere iki gruba ayrılır; tek katmanlı ileri beslemeli yapay sinir ağları en basit ve temel ağ yapısıdır. Bir giriş katmanı ve bir çıkış katmanı vardır. Bu tip bir ağda bilgi girişten çıkışa doğru ilerler yani ağ ileri beslemelidir. Tek katmanlı olarak isimlendirilmesinin sebebi, giriş katmanının veri üzerinde hiçbir işlem yapmadan veriyi çıkış katmanına iletmesidir.

Çok katmanlı ileri beslemeli yapay sinir ağında ise gizli katmanların birden fazladır. Bu sebeple giriş katmanından gelen Verilere bir takım işlemler uygulanabilmektedir. İleri beslemeli yapay sinir ağlarında gecikme yoktur. Bunun sebebi işlemin girişlerden çıkışlara doğru ilerlemesidir. Çıkış değerleri, öğreticiden alınan istenen çıkış değeriyle karşılaştırılarak bir hata sinyali elde edilerek ağ ağırlıkları güncellenir [64].

Geri beslemeli yapay sinir ağları

Geri beslemeli yapay sinir ağı, çıkış ve ara katlardaki çıkışların, giriş birimlerine veya önceki ara katmanlara geri beslendiği bir ağ yapısıdır. İleri beslemeli yapay sinir ağlarının tersine; bu ağda kontrol uygulamalarında olduğu gibi gecikmeler söz konusudur.

Bu çeşit yapay sinir ağlarının dinamik hafızaları bulunmaktadır ve her hangi bir andaki çıkış hem o andaki hem de önceki girişleri yansıtır. Bundan dolayı, özellikle önceden tahmin uygulamaları için uygundurlar. Geri beslemeli nöral ağ, çıkışları girişlerine bağlanan ileri beslemeli bir ağdan elde edilir. Ağın t anındaki çıkışı o(t) ise, t + Δ anındaki çıkışı o(t+Δ)'dır. Δ sabiti sembolik anlamda gecikme süresidir. Geri beslemeli YSA eşitlik 4.8'deki gibi ifade edilebilir.

$$o(t + \Delta) = f[W.\, o(t)] \qquad (4.8)$$

Şekil 4.9'da, eşitlik 4.7 ifade edilmiştir. Dikkat edilmesi gereken nokta başlangıç anında x(t)'ye ihtiyaç duyulmasıdır. Başlangıç anında o(0) = x(0)'dır.

Şekil 4.9. Geri beslemeli yapay sinir ağı

Geri yayılımlı yapay sinir ağları

Geri yayılımlı yapay sinir ağlarının en büyük özelliği hem ileri beslemeli hem de geri beslemeli çevriminin her ikisini de kendi içinde bulundurmasıdır. Geri yayılımlı yapay sinir ağına Hopfield, Counterpropogation, Cognitron, Kendini Ayarlayan Haritalı Ağlar (SOM - Self Organizing Maps), Boltzman Makinesi örnek olarak verilebilir.

Geri yayılımlı YSA'lar özellikle birinci dereceden doğrusal sistemleri modellemekte oldukça başarılıdırlar. Zamana bağlı olayları izlemede, daha önce elde edilen sonuçları değerlendirmedeki başarılı çıktıları ile özellikle ses ve karakter tanıma problemlerinde etkin olarak kullanılmaktadır.

4.4.6. Yapay sinir ağlarının eğitimi

Canlılar, doğumdan sonraki gelişme sürecinde çevresinden duyu organlarıyla algıladığı davranışları yorumlar ve bu bilgileri diğer davranışlarında kullanır.

Artık meydana gelen herhangi bir olay karşısında nasıl tepki göstereceğini çoğu zaman bilmektedir. Ama hiç karşılaşmadığı bir olay karşısında yine tecrübesiz kalabilir. Yapay sinir

ağlarının öğrenme sürecinde de, tıpkı dış ortamdan gözle veya vücudun diğer organlarıyla uyarıların alınması gibi dış ortamdan girişler alınır, bu girişlerin beyin merkezine iletilerek burada değerlendirilip tepki verilmesi gibi yapay sinir ağında da aktivasyon fonksiyonundan geçirilerek bir tepki çıkışı üretilir. Bu çıkış yine tecrübeyle verilen çıkışla karşılaştırılarak hata bulunur. Çeşitli öğrenme algoritmalarıyla hata azaltılıp gerçek çıkışa yaklaşılmaya çalışılır.

Yapay sinir ağlarının eğitiminde yenilenen, yapay sinir ağının ağırlıklarıdır. Ağırlıklar her bir çevrimde yenilenerek amaca ulaşılmaya çalışılır. Amaca ulaşmanın veya yaklaşmanın ölçüsü de yine dışarıdan verilen bir değerdir. Eğer yapay sinir ağı verilen giriş-çıkış çiftleriyle amaca ulaşmış ise ağırlık değerleri saklanır. Ağırlıkların sürekli yenilenip istenilen sonuca ulaşılana kadar geçen zamana öğrenme adı verilir.

YSA öğrendikten sonra daha önce verilmeyen girişler verilip, sinir ağı çıkışıyla gerçek çıkışı yaklaşımı incelenir. Eğer yeni verilen örneklere de doğru yaklaşıyorsa sinir ağı işi öğrenmiş demektir. Sinir ağına verilen örnek sayısı uygun değer değerden fazla ise sinir ağı işi öğrenmemiş ezberlemiştir. Genelde eldeki örneklerin yüzde sekseni ağa verilip ağ eğitilir, daha sonra geri kalan yüzde yirmilik kısım verilip ağın davranışı incelenir diğer bir deyişle ağ böylece test edilir.

Yapay sinir ağlarının verilen girdilere göre çıktı üretebilmesinin yolu ağın öğrenebilmesidir. Bu öğrenme işleminin de birden fazla yöntemi vardır. Yapay sinir ağları öğrenme algoritmalarına göre danışmanlı, danışmansız ve takviyeli öğrenme olarak üçe ayrılır.

YSA'ların öğrenme yöntemlerine göre sınıflandırılması: YSA'nın kullanım amacı sayısal bir hesaplama olabileceği gibi yığın halindeki öğelerin uygun öbeklere ayrımı da olabilmektedir. İlk durumdaki kullanım amacı regresyon problemleri ikincisi ise sınıflandırma problemlerini tanımlamaktadır. Her iki durumda da öğrenmeden kasıt, YSA'ya verilen girdiler neticesinde beklenilene en yakın sonucu elde etmektir. Bu amaca ulaşmak için çeşitli yöntemler kullanılarak ağırlıklara atanacak en uygun değerler tespit edilmektedir. Bütün öğrenme yöntemlerinde, bilinen verilerden hazırlanmış bir veri kümesinden yararlanılmaktadır. Bu veri kümeleri sayesinde öğeler arasındaki bağlantı tespit edilerek öğrenme sürecinde tanımlanmayan durumlar içinde isabetli sonuçlar elde edilmektedir.

Öğrenme; gözlem, eğitim ve hareketin doğal yapıda meydana getirdiği davranış değişikliği olarak tanımlanmaktadır. Bir takım metot ve kurallar, gözlem ve eğitim ile ağdaki ağırlıkların değiştirilmesi sağlanmalıdır. Bunun için genel olarak üç öğrenme metodundan ve bunların uygulandığı değişik öğrenme kurallarından söz edilebilir. Bu öğrenme kuralları aşağıda açıklanmaktadır.

Danışmanlı öğrenme

Danışmanlı öğrenmede, yapay sinir ağı kullanılmadan önce eğitilmesi gerekir. Eğitme işlemi için giriş bilgilerinin yanında çıkış bilgileri de verilmelidir. Çoğu uygulama için ağa gerçek örnek kümesi verilme zorunluluğu vardır. Bu örnek kümesi ile ağ eğitilip istenen istatistiksel doğruluk elde edildiğinde eğitme işlemi tamamlanmış olur. Ağ kullanılmaya başladığında eğitim sonucunda elde edilen ağırlık değerleri çoğunlukla sabit kalır, bir daha değiştirilmez.

Danışmanlı öğrenme (supervised learning): Danışmanlı öğrenme yönteminde, YSA'nın uygulanacağı probleme ait bir dizi girdi ve bu girdilere karşılık elde edilmesi beklenen çıktılardan oluşan iki ayrı veri kümesi kullanılmaktadır. Veri kümesindeki girdiler için YSA'nın çıkışında beklenen değerlerin bilinmesi nedeniyle bu yönteme danışmanlı öğrenme denilmektedir. YSA çıkışında elde edilen ile beklenilen arasındaki fark hata olarak belirlenmektedir.

Danışmanlı öğrenme sürece hata değeri istenilen seviyeye düşünceye kadar tekrar eden ve her bir tekrarda ağırlık değerlerinin değiştiği bir süreçtir. Geriye yayılımlı YSA'lar danışmanlı öğrenme yöntemlerini kullanılmaktadır.

Bu tip öğrenmede, YSA'ya örnek olarak bir doğru çıkış verilir. Bu öğrenmede ağın ürettiği çıktılar ile hedef çıktılar arasındaki fark hata olarak ele alınır ve bu hata minimize edilmeye çalışılır. Bunun için de bağlantıların ağırlıkları en uygun çıkışı verecek şekilde değiştirilir. Bu sebeple danışmanlı öğrenme algoritmasının bir "öğretmene" veya "danışmana" ihtiyacı vardır. Widrow-Hoff tarafından geliştirilen delta kuralı ve Rumelhart ve McClelland tarafından geliştirilen genelleştirilmiş delta kuralı veya geri besleme (back propagation) algoritması danışmanlı öğrenme algoritmalarına örnek olarak verilebilir [65].

Danışmansız öğrenme

Danışmansız öğrenmede ağa öğrenme sırasında sadece örnek girdiler verilmektedir. Herhangi bir beklenen çıktı bilgisi verilmez. Girişte verilen bilgilere göre ağ her bir örneği kendi arasında sınıflandıracak şekilde kendi kurallarını oluşturur. Ağ bağlantı ağırlıklarını aynı özellikte olan dokuları ayırabilecek şekilde düzenleyerek öğrenme işlemini tamamlar [65].

Danışmansız Öğrenme (unsupervised learning) : YSA'nın uygulanacağı probleme girdilere karşılık elde edilmesi beklenen çıktılardan bilinmediği durumlarda danışmansız öğrenme kullanılmaktadır. Öğrenme için yalnızca girdilerden oluşan bir veri kümesi kullanılmaktadır. Her bir öğeye ait girdi verileri kullanılarak bu öğeler arasındaki bağıntıya göre uygun bağlantı değerleri atanmaktadır. Ağırlıklı olarak sınıflandırma problemlerinde kullanılmaktadır [65].

Takviyeli öğrenme

Bu öğrenme kuralı danışmanlı öğrenme kuralının özel bir şeklidir. Her girdi seti için olması (üretilmesi) gereken çıktı setini sisteme göstermek yerine, sistemin kendisine gösterilen girdilere karşılık çıktısını üretmesini bekler ve üretilen çıktının doğru veya yanlış olduğunu gösteren bir sinyal üretir. Bu sinyal dikkate alınarak, eğitim süreci devam ettirilir.

Takviyeli Öğrenme (reinforcement training): Takviyeli öğrenme algoritması, istenilen çıkışın bilinmesine gerek duymaz. Takviyeli öğrenme (reinforcement training) yöntemi öğreticili öğrenme yöntemine benzemekle birlikte, ağa hedef çıktılar yerine, ağın çıktılarının ne ölçüde doğru olduğunu belirten bir skor veya derece bildirilir. Web uygulamaları sürekli ulaşılabilen sistemler olduğu için saldırılara da açık uygulamalardır. Web uygulamalarının güvenliğini sağlamak amacıyla öğrenme özelliğine sahip ve gerçek zamanlı bir web uygulaması güvenlik duvarı algoritması geliştirilmesi amaçlanmıştır. YSA öğrenebilme özelliğine sahip bir yöntem olduğu için, öğrenme tabanlı anormal denetimi gerçekleştirmek için tercih edilmiştir. YSA'larda yapısına göre ileri beslemeli geri yayılımlı (feed-forward backprop) mimarisi, öğrenme yönetimine göre danışmanlı öğrenme yöntemi kullanılmaktadır [65].

4.5. Bayes Tabanlı Yapay Sinir Ağları (BTYSA)

YSA eğitimi rasgele, farklı eğitim algoritmaları, değişik sayıdaki iterasyon, değişik sayıda gizli katman kullanılarak gerçekleştirilmektedir. YSA geçmişte oluşmuş eğilimlere dayalı

bir eğitim modelidir. YSA ağının çıktılarını tahmin edebilmek için YSA girişleri, gerçek dünyadaki çıktıları etkileyen değişkenler olarak seçilir ve bu girişlerin eğilimleri çıktı değişkenlerini tahmin etmek için kullanılır [66].

YSA ile yapılan tahminlerde daima bir hata oranı vardır ve tahminler iyileştirilerek hata oranı azaltılır.

$$y = f(x;w) + e \tag{4.9}$$

Eşitlik (4.9)'a göre, y istenilen gerçek çıkış değeri, f ağ tarafından tahmin edilen çıkış değeri, e hata, w ağırlıklar, x ise giriş değerleridir. Hata değeri olan e bilinse bile aynı ağ aynı parametrelerle birden fazla çalıştırıldığında, her defasında da farklı ağırlıklar (w) elde edilebilmektedir. Çünkü ağın eğitimi sırasında oluşan belirsizlikten dolayı ağırlıklar rastgele atanmaktadır. YSA eğitiminde farklı türde eğitim algortimaları bulunmaktadır. Geri yayılım algoritması, danışmanlı öğrenme ve hataların azaltılması sürecine dayanan ve en çok kullanılan YSA eğitim algoritmasıdır.

BTYSA, eğitim ağırlıkları ve sapma değerlerini (hata) güncelleyerek optimize ederek YSA eğitiminin iyileştirilmesi için kullanılmaktadır. Aşağıda YSA ağırlık değerlerinin optimize edilmesi anlatılmıştır.

Geri yayılım sinir ağları giriş katmanı, bir veya daha fazla gizli katman ve bir çıkış katmanı yapısına sahiptir. Ağın yapısında giriş katmanından alınan giriş vektörleri gizli katmanlar aracılıyla çıkış katmanına aktarılır. Katmanlar arasındaki geçişler aşağıdaki (4.10) ve (4.11) eşitlikler kullanılarak gerçekleştirilmektedir.

y_i gizli katman çıkışını ifade etmektedir:

$$y_i = \phi\left(\sum_{j=1}^{d} v_{ji} x_j + a_i\right) \tag{4.10}$$

o_i çıkış katmanı çıkışını ifade etmektedir:

$$o_i = \varphi\left(\sum_{j=1}^{h} w_{ji} y_j + b_i\right) \tag{4.11}$$

$\phi(x)$ ve $\varphi(x)$ aktivasyon fonksiyonlarıdır ve genellikle sigmoid ve veya hiperbolik tanjant fonksiyonları kullanılmaktadır.

Ağ eğitimi yukarıda verilen (4.10) ve (4.11) eşitlikleri kullanılarak doğrusal olmayan bir şekilde yapılmaktadır. Ağın eğitiminin amacı E_D hata oranının azaltılmasıdır. Ağın çıkış değeri ile istenen çıkış d_l arasındaki fark eşitlik (4.12)'de verilen hata oranını oluşturmaktadır.

$$E_D = \frac{1}{2}\sum_{l=1}^{n}\{d_l - \varphi[\sum_{j=1}^{h} w_{jl}\phi(\sum_{i=1}^{d} v_{ij}x_i + a_j) + b_l]\}^2 \qquad (4.12)$$

YSA ile yapılan karmaşık modellemelerde, belirsiz girişler fazla eğitim problemi ile tahmin yapmaya çalışırken, tahminlerdeki hata oranını iyileştirmeye çalışır ve bu durum tahminlerde daha fazla hataya sebep olabilmektedir.

P(D) bayes sınıflandırmada, istatistiksel olarak olasılık fonksiyonunu göstermek için kullanılır. Bayes yaklaşımında, ağ eğitilirken tahmin edilen parametrelerdeki belirsizliklerin, belli bir dağılımı takip ettiği varsayılır. YSA'da ağ eğitmek için girişler kullanılarak çıkışlar tahmin edilmektedir, bu aşamada girişlerin ağırlıkları rastgele atanmaktadır. BYYSA yöntemi ile ağırlıkların istatistiksel olarak seçimi gerçekleştirilerek, YSA ağırlıklarının bulunduğu çok geniş bir veri kümesi içerisinde iyileştirme yapılmaktadır.

Veri kümesindeki ağırlık parametrelerinin olasılık yoğunluklarını p(w|D) yardımıyla bularak bu dağılım eşitlik (4.13) daraltılarak daha iyi sonuç alabilmek için YSA'da rastgele oluşturulan ağırlık değerleri BTYSA ile iyileştirilerek YSA'nın tahmin değerleri iyileştirilmektedir.

$$P(w|D) = \frac{P(D|w)p(w)}{P(D)} \qquad (4.13)$$

Buradaki P(D|w) veri kümesinin komşuluğudur. P(D) sonraki olasılık yoğunluğunu 1'e eşitlemeyi garanti eden ve parametre uzayı üzerinden bir integral tarafından hesaplanan eşitliktir.

Bayes tekniğinin kullanımının esas amacı ağırlıklardaki (w) belirsizliği azaltmaktır. Böylece aşırı iyileştirme problemi de azaltılmış olur. Bir ağ belirsiz girişlerle, eğitim verisi ve tahminler kullanılarak çok fazla eğitim yapıldığında ve kötü tahmin çıktığı zaman aşırı iyileştirme oluşur.

Nabney tarafından BTYSA tekniği ile, YSA'nın aşırı iyileştirme probleminin nasıl çözüleceğini ispatlanmıştır. Sinir ağlarında ağın eğitilmesi için optimize edilmesi gereklidir ve bu yüzden hata fonksiyonunun her iterasyonda bir önceki hata oranına göre düşürülmesi sağlanmalıdır. Bayes teoremi ile, YSA ağırlıkları için bir sonraki dağılım elde edilerek sadece belli bir dağılım içerisinde bulunabilmesi sağlanır. Böylece en uygun ağırlık değerleri belirlenmiş olur [67].

Açıkçası, E_D hata değerini azaltmak için ağırlık değerleri değiştirilmektedir. Böylece, ağırlık değerlerinin ayarlamasını hata değerlerinin azaltılması oranıyla yapılmaktadır.

$$\Delta w_{jk} = -\eta \frac{\partial E}{\partial w_{jk}} \tag{4.14}$$

$$\Delta v_{jk} = -\eta \frac{\partial E}{\partial v_{ij}} \tag{4.14}$$

Eşitlik (4.14) ve (4.15)'da $\eta \in (0,1)$ eğitim ifadesin öğrenme oranını tanımlayan oran parametresine karşılık gelmektedir. Bu bağlam geleneksel geri yayılım sinir ağıdır. Daha önce de tartışıldığı gibi geri yayılım sinir ağının büyük problemi aşırı öğrenme ve ağı genelleştirememesidir. Aşırı öğrenme probleminden kaçınmak ve genelleştirme kapasitesini geliştirmek ve eğitim algoritması olarak bayes regülasyonu kullanan BTYSA önerilmektedir.

Geri kalan kısım MacKay tarafından ortaya atılan bayes teorisi ile özetlemektedir [68]. Tahmin fonksiyonu eşitlik (4.16) ile ifade edilmektedir.

$$F(w) = \alpha E_w(w) + \beta E_D(w) \tag{4.16}$$

Ew eşitlik (4.17)'de m ağ ağırlık parametrelerinin toplam sayısı, Wi ise ağırlık parametrelerini ifade etmektedir.

$$E_W(w) = \frac{1}{2}\sum_{i=1}^{m} w_i^2 \tag{4.17}$$

E_W ağ ağırlıklarının karelerinin toplamın, α ve β ise hiper parametrelerdir. Bayes teoremine göre P(w|α) nın ön olasılığı ve P(D|w,β) benzerlik fonksiyonu biliniyorsa, bayes teoremini kullanarak, eşitlik (4.18)'da P(w|D,α ,β) sonsal olasılığı elde edebilir. D, eğitim için kullanılan n adet giriş ve çıkıştan oluşan veri kümesini ifade etmektedir.

$$P(w|D,\alpha,\beta) = \frac{P(D|w,\beta)P(w|\alpha)}{P(D|\alpha,\beta)} \quad (4.18)$$

P(D | α ,β)'nın logaritma farkını kullanarak eşitlik (4.19) ve eşitlik (4.20) elde edilmektedir [69].

$$\alpha = \frac{\gamma}{2E_w} \quad (4.19)$$

$$\beta = \frac{n-\gamma}{2E_D} \quad (4.20)$$

λ miktarı eşitlik (4.21) tarafından verilen iyi tanımlanmış bir parametredir.

$$\gamma = \sum_{i=1}^{m}\left(\frac{\lambda_i}{\lambda_i+\alpha}\right) \quad (4.21)$$

Bu eşitlikte Δi, βE_D in Hessian matrisinin öz değeridir. $H= \beta\nabla\nabla E_D$ bu teknikteki yenilik yerel minimuma elde etmek için olasılığı azaltması ve ceza parametresi ekleyerek sinir ağının genelleştirilebilmesinin arttırabilmesidir.

4.6. Kullanılan Veri Kümeleri

Saldırı önleme ve belirleme sistemlerini değerlendirmek için saldırı içeren verilerin yanı sıra normal kabul edilen verileri de içeren test veri kümeleri kullanılmaktadır. Fakat HTTP trafiği ele alındığında, yaygın olarak kullanılabilir veri kümeleri çok azdır. Çünkü var olan çoğu veri kümesi ihtiyaç duyulan HTTP istek ve cevap verilerini içermemektedirler [70]. Web tabanlı saldırı denetimdeki en önemli problem, WUGD test etmek için web tabanlı saldırıları içeren kullanılabilir veri kümelerinin bulunmamasıdır.

Bu çalışmada, WUGD 2015, CSIC 2010 ve ECML-PKDD 2007 veri kümeleri kullanılmıştır. Kullanılan veri kümelerinde bulunan HTTP istekleri bazı ön işleme sürecinden geçirilmişlerdir. HTTP istekleri farklı parametrelerden oluşmaktadır. Değişkenlerin tamamı, normal veya anormal istekleri ayırt etmek için herhangi bir bilgi sağlamadığından, sadece ayırt edici özelliğe sahip değişkenler kullanılmıştır. Yinelenen HTTP istekleri kaldırılmıştır.

CSIC 2010

CSIC 2010, Bilgi Güvenliği Enstitüsü (Spanish Research National Council) tarafından geliştirilmiş, web tabanlı saldırı önleme sistemlerini test etmek için otomatik olarak

üretilmiş, binlerce web isteği içeren HTTP veri kümesidir. Bir e-ticaret web uygulaması kullanılarak üretilen veri kümesinin üretiminde, web uygulamasının bütün parametrelerini toplayacak şekilde üretim gerçekleştirilmiştir.

Otomatik olarak üretilen CSIC 2010 (http://www.isi.csic.es/dataset/) verikümesi 36.000 normal HTTP isteği ve 25.000'den fazla anormal HTTP isteği içermektedir. HTTP istekleri normal ve anormal olarak etiketlenmiştir ve verikümesi bilinen saldırı türlerinden SQL enjeksiyonu, bellek taşması, CRLF enjeksiyonu, Siteler arası kod yazma (XSS) gibi saldırı türlerini de içermektedir [71,72]. CSIC 2010 veri kümesinin test amaçlı olarak kullanılabilmesi için normal olarak belirlenen trafikten oluşan eğitim kümesi, yine normal trafikten oluşan test kümesi ve anormal trafikten oluşan test kümeleri ayrı ayrı üretilerek yayınlanmıştır.

ECML-PKDD 2007

ECML-PKDD 2007 (http://www.lirmm.fr/pkdd2007-challenge) veri kümesi ECML-PKDD konferansları kapsamında üretilmiştir. Veri kümesi, % 20'si saldırı içeren 50000 veriden oluşmaktadır. Saldırı olarak belirlenen verilerin % 10'u bilinen saldırı türlerinin dışında, tanımlanamayan veya anormal nitelikte verilerdir. Veri kümesi sadece web tabanlı saldırıları değil aynı zamanda saldırı tespit sistemlerinin test edilmesi sürecinde de kullanılabilmektedir [73]. Veri kümesinde bulunan anormal ve sıradışı veriler gerçek saldırılara benziyor olsalar da rastgele üretildikleri için ve gerçek zaafiyetleri hedeflemedikleri için başarılı olma ihtimalleri düşüktür ama yinede web uygulamalarının çalışmasını aksatacak düzeyde tahribata sebep olabilirler. Saldırı içeren bir veri türü, SQL enjeksiyonu ve Komut enjeksiyonu gibi birçok sınıfını içerebilir.

WUGD 2015

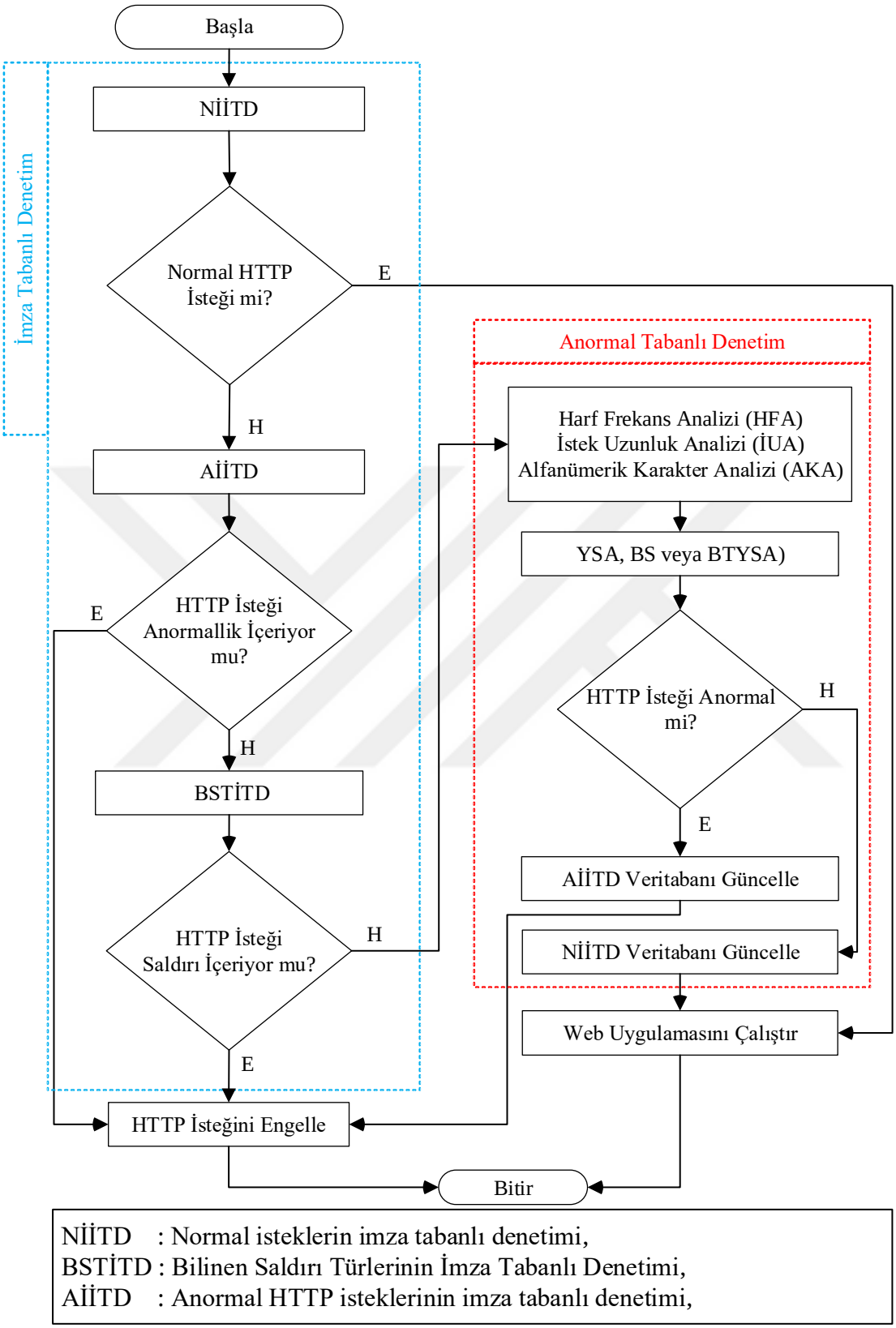
WUGD 2015 verikümesi, tez çalışması kapsamında üretilmiştir. Tez kapsamında geliştirilen web uygulaması kullanılarak web uygulamasına gelen HTTP istekleri ile otomatik olarak oluşturulmuştur. WUGD 2015 veri kümesi HTTP istek yapısı 38 farklı parametreden oluşmaktadır. Veri kümesi 35208 HTTP veriden oluşmaktadır. Bu veri kümesi % 30’u anormal olan ve SQL enjeksiyonu, bellek taşması ve XSS saldırı türlerini içeren, gerçek zamanlı çalışan web uygulamasından üretilmiş veri kümesi kullanılmıştır. HTTP parametrelerinin tamamı istek denetiminde kullanılacak veri yapısına sahip olmadığı için

özellikle URL verisini içeren QUERY_STRING, REQUEST_URI alanları İTD ve ATD denetimleri yapmak için kullanılmıştır.

5. WUGD'NİN TASARIMI VE GERÇEKLEŞTİRİLMESİ

Bu tez çalışmasında web tabanlı saldırıları önleme amaçlı yeni bir gerçek zamanlı hibrit WUGD algortirması önerilmiştir. Önerilen algoritmada; İTD ve ATD kullanılarak hibrit bir denetim aşamalı olarak gerçekleştirilmiştir. İTD Normal İsteklerin İmza Tabanlı Denetimi (NİİTD), Anormal İsteklerin İmza Tabanlı Denetimi (AİİTD) ve Bilinen Saldırı Türlerinin İmza Tabanlı Denetimi (BSTİTD) olmak üzere üç aşamadan oluşmaktadır. İTD sürecinde tespit edilemeyen HTTP isteği ATD sürecine aktarılmaktadır. ATD Alfanümerik Karakter Analizi (AKA), Harf Frekans Analizi (HFA) ve İstek Uzunluk Analizi (İUA) olmak üzere 3 farklı öznitelik kullanılarak, sırasıyla veri madeciliği yötemlerinden BS, yapay zekâ yöntemlerinde YSA ve BTYSA olmak üzere 3 farklı sınıflandırıcı ayrı ayrı kullanılarak, geliştirilen WUGD testi gerçekleştirilmiştir. Önerilen algoritmanın akış diyagramı Şekil 5.1'de verilmiştir.

Bu bölümde, Şekil 5.1'de akış diyagramı verilen WUGD algoritmasının, her bir denetim türü ve sınıflandırma araçlarının tasarım ve gerçekleştirme aşamaları anlatılmıştır. İlk önce İTD kısımları olan NİİTD, AİİTD ve BSTİTD anlatılmıştır, daha sonra ATD gerçekleştirmek için kullanılan BS, YSA ve BTYSA yöntemleri anlatılarak elde edilen sonuçlar verilmiştir.



Şekil 5.1. Önerilen WUGD modeli akış diyagramı

5.1. İmza Tabanlı Denetim (İTD)

İTD, bilinen saldırı türlerinin denetimini gerçekleştirmek için kullanılan ve kötüye kullanım olarak da ifade edilen imza denetim türüdür. Daha önceden tespit edilen kötüye kullanım hareketleri, sistemlere imza olarak tanımlanır. Saldırı tespit ve önleme sistemleri belirlenen kötüye kullanım hareketlerine karşı etkili hale getirilmiş olur. Bu yapının en önemli dezavantaji imza veritabanı güncellemeye ihtiyaç duymasıdır. Dolayısıyla imza veritabanı güncellenmeyen sistemler, yeni saldırı türlerine karşı etkisiz olmaktadırlar.

Uzman tarafından ağ trafiği incelenerek saldırının karakteristiği, davranışı ve içeriği ortaya çıkarılır. Her bir saldırının karateristiği ortaya çıkarılarak, tespit edilen etiketlenmiş veriler imza olarak tanımlanır. İTD sistemleri genelde hızlı çalışırlar, ancak sadece imza veri tabanında bulunan saldırı türlerine karşı etkilidirler. Genelde saldırı tespit sistemleri ve anti virüs programları imza tabanlı olarak çalışırlar [74]. Yeni bir saldırı tekniği geliştirildiğinde belirtilen sistemin etkili olabilmesi için imza veri tabanının saldırı tekniğine göre güncellenmesi gereklidir. Aksi takdirde yapılan saldırılara karşı etkisiz olacaktır. Geliştirilen sistem ile SQL Enjeksiyonu, Siteler Arası Kod (XSS) Yazma, Dizin Geçişi Saldırı (Directory Traversal Attack), Komut Enjeksiyonu (Command Injection) imza tabanlı denetimi gerçekleştirilmiştir.

Bu tez çalışmasında önerilen algoritmada İTD 3 aşamadan oluşmaktadır. Birinci aşama NİİTD olarak isimlendirilmiştir ve ATD sonucu tespit edilen normal HTTP istekleri, tekrar web uygulamasına geldiğinde, ikinci defa ATD gerçekleştirilmeden web uygulamasının çalıştırılaması sağlanmaktadır. Bu durum önerilen algoritmanın hız performansını artırmaktadır. İkinci aşama AİİTD olarak isimlendirilmiştir ve ATD sonucu tespit edilen anormal HTTP istekleri, tekrar web uygulamasına geldiğinde, ikinci defa ATD gerçekleştirilmeden anormal HTTP isteği engellenmektedir. Üçüncü aşama bilinen web tabanlı saldırı türlerinin denetiminin gerçekleştirildiği BSTİTD olarak isimlendirilen aşamadır. Bu aşamada imza veritabanının manuel olarak güncellenmesi gerekmektedir ve kötüye kullanım olarak bilinen saldırı türlerinin denetim mekanizmasını oluşturmaktadır.

Önerilen algoritmada ATD sonucu imza üretimi gerçekleştirilmektedir. NİİTD ve AİİTD aşamaları ATD sonucu üretilen imzalar kullanılarak güncellenmektedir. Bu durum önerilen algoritma kullanılarak geliştirilen WUGD sisteminin sıfır gün saldırılarına karşı adaptasyonunu sağlamaktadır ve hız bakımından denetim perfosrmansını artırmaktadır.

5.1.1. Normal isteklerin imza tabanlı denetimi (NİİTD)

İTD'nin birinci aşaması Normal İsteklerin İmza Tabanlı Denetimi (NİİTD) olarak isimlendirilmiştir. ATD sonucunda normal olarak tespit edilen normal HTTP istekleri NİİTD veritabanına eklenmektedir. Normal HTTP istekleri web uygulamasından ikinci defa istemci tarafından talep edildiği zaman İTD'ye göre daha yavaş çalışan ATD süresine dâhil edilmeden normal HTTP isteklerinin denetimi NİİTD veritabanı kullanılarak gerçekleştirilmektedir. Normal olarak tespit edilen HTTP isteği NİİTD veritabanına cümle olarak eklenmektedir. Karşılaştırma işleminde, NİİTD veritabanında HTTP isteği yok ise karşılaştırma YANLIŞ olarak cevap vererek HTTP isteğinin denetimini AİİTD sürecine aktarmaktadır. Eğer karşılaştırma işlemi DOĞRU olarak cevap verirse HTTP isteği doğrudan web uygulamasına yönlendirilerek web uygulaması çalıştırılmaktadır.

5.1.2. Anormal isteklerin imza tabanlı denetimi (AİİTD)

İTD'nin ikinci aşamasında ise yine normal HTTP isteklerinin denetiminde olduğu gibi, ATD gerçekleştirilerek, anormal olarak tespit edilen anormal HTTP isteklerinin denetimi gerçekleştirilmektedir. ATD sonucunda tespit edilen anormal HTTP istekleri web uygulamasına ikinci defa istemci tarafından talep edildiği zaman İTD'ye göre daha yavaş çalışan ATD sürecine dâhil edilmeden anormal HTTP isteklerinin denetiminin gerçekleştirilmesi amaçlanmıştır. Böylece anormal olarak tespit edilen HTTP istekleri ikinci defa web uygulamasına gönderildiklerinde daha hızlı cevap verebilmek için sistemde AİİTD gerçekleştilmiştir. Bu sayede modelin denetim hız performansı artmıştır. Karşılaştırma işleminde, AİİTD veritabanında HTTP isteği yok ise karşılaştırma YANLIŞ olarak cevap vererek HTTP isteğinin denetimini BSTİTD sürecine aktarmaktadır. Eğer karşılaştırma işlemi DOĞRU olarak cevap verirse HTTP isteği engellenmektedir.

5.1.3. Bilinen saldırı türlerinin imza tabanlı denetimi (BSTİTD)

İTD'nin üçüncü aşamasında bilinen web tabanlı saldırı türlerine karşı denetim gerçekleştirilmektedir. Bilinen Saldırı Türlerinin İmza Tabanlı Denetimi (BSTİTD), web uygulamalarını hedef alan ve web uygulamalarının güvenlik zaafiyetlerini kullanarak riski ortaya çıkaran, saldırı niteliği taşıyan web tabanlı saldırı türlerinin denetlenmesidir. Her yıl, en çok yapılan 10 bilinen web tabanlı saldırı türü Open Web Application Security Project (OWASP) açık web uygulama güvenliği topluluğu tarafında belirlenerek yayınlamaktadır.

OWASP, web uygulamalarının güvenlik zaafiyetlerinin oluşturduğu problemlere karşı çözüm üretmek için kurulmuş bir topluluktur. OWASP'ın tüm araçları ve yayınları ücretsiz olarak sunulmaktadır. OWASP'ın her yıl yayınladığı en çok yapılan 10 saldırı türleri değerlendirildiğinde, SQL enjeksiyonu, XSS enjeksiyonu, Dizin Geçişi Saldırısı ve Komut Enjeksiyonu saldırı türleri bütün listelerde yer almaktadır ve geliştirilen sistem ile de bu saldırı türlerine karşı imza tabanlı denetimi gerçekleştirilmiştir. Ayrıca QWASP tarafından yayınlanan diğer saldırı türlerine karşı da geliştirlen sistem kullanılarak İ gerçekleştirilebilir. Belirtilen dört saldırı türüne karşı denetim gerçekleştirilmesinin sebebi, geliştirilen sistem kullanılarak İTD'nin gerçekleştirilebileceğini göstermektedir.

SQL enjeksiyonu

SQL enjeksiyonu web uygulamlarını tehdit eden en önemli saldırı türlerinden birisidir. SQL, veritabanı yönetim sistemlerinde sorgu yapmak üzerine özelleşmiş hem ANSI (American National Standards Institute) hem de ISO (International Organization for Standardization) standardı olan yapısal bir dilidir. Değişen büyüklükteki ilişkisel veritabanı uygulamalarına SQL sorguları aracılığıyla ulaşılabilir. SQL'i destekleyen birçok veritabanı yönetim sistemi (MS SQL Server, Oracle, MySQL, gibi) standart dile özel eklentiler getirir [75].

SQL enjeksiyonu, uygulama parametreleri aracılığı ile gönderilen bilgilerin gerektiği gibi kontrol edilmemesi sebebi ile arka planda çalışan veritabanına gönderilen sorgulara, saldırganın kendi sorgularını eklmesine yardımcı olan bir güvenlik açığıdır.

Web uygulamaları kullanıcı kaynaklı girdileri ve talepleri için değişik SQL cümleleri oluşturmada kullanabilir. SQL enjeksiyonu yöntemi, kullanıcı girdilerine göre SQL cümleleri oluşturan web uygulamalarında, kullanıcı girdilerinin doğrulanmaması veya yetersiz doğrulanmasından kaynaklanan zafiyetler kullanılarak, SQL cümlelerinin kötüye kullanılmasını sağlayan sızma testleridir [76]. SQL enjeksiyonu sızma yöntemiyle yapılabilecek işlemler aşağıda sıralanmıştır.

- Veri tabanları üzerinde işlemler (sorgulama, ekleme, silme, değiştirme, vb.) yapılabilir.
- Kimlik doğrulama mekanizmaları atlatılabilir.
- İşletim sistemi seviyesinde komutlar çalıştırılabilir.
- Etki alanında yeni kullanıcılar veya gruplar oluşturulabilir.

Eğer bir web uygulaması, kullanıcı kaynaklı girdiyi etkin bir biçimde denetlemezse, SQL enjeksiyon yöntemiyle SQL cümlesi oluşumu değiştirilerek güvenlik ihlalleri oluşturulabilir.

SQL enjeksiyon yöntemiyle SQL cümlesi değiştirilerek bilgisayar sistemlerine sızılması durumunda, SQL servisini çalıştıran kullanıcı haklarına sahip olunacaktır. SQL enjeksiyonu yapılırken SQL'e özel anlam içeren karakterlerin kullanılması gereklidir. Bu karakterler ve kısa açıklamaları Çizelge 5.1.'de verilmiştir.

Çizelge 5.1. SQL enjeksiyonu karakterleri

Karakter	**Açıklama**
' veya "	String ayracı
-- veya #	Tek satırlık açıklama
/*...*/	Çok satırlı açıklama
+	Ekleme, birleştirme (URL adresi içerisinde boşluk)
II	Birleştirmek
%	Genel (wildcard) özellik ayracı
?Param 1=foo&Param2=bar	URL Parametresi
PRINT	Cevap gerektirmeyen komutlar
@variable	Yerel değişken
@@variable	Bölgesel değişken
wait for delay '0:0:10'	Zaman geciktirme
'	Virgül
;	Noktalı virgül
!	Ünlem
SELECT * FROM --	SQL Enjeksiyonu Cümlesi
#	SQL Enjeksiyonu Cümlesi
–';	SQL Enjeksiyonu Cümlesi
or 'x' = 'x	SQL Enjeksiyonu Cümlesi
' or '= '#	SQL Enjeksiyonu Cümlesi
' or a = a–	SQL Enjeksiyonu Cümlesi
%20or%201 =1	SQL Enjeksiyonu Cümlesi

Siteler arası kod (XSS) yazma

HTML kodlarının arasına istemci tabanlı kod gömülmesi yoluyla kullanıcının tarayıcısında istenen istemci tabanlı kodun çalıştırılabilmesi olarak tanımlanır. XSS genellikle HTML/JavaScript dilinde yazılmaktadır, ancak VBScript, ActiveX, Java, Flash veya web tarayıcılar tarafından desteklenen diğer dillerde de kodlama yapılabilmektedir [77]. XSS yöntemiyle zararlı kodun kullanıcı web tarayıcısında çalıştığında, zararlı kod sunucu web sitesinin tarayıcı için tanımlı olduğu güvenlik ayarları kapsamında çalışacaktır. Çizelge 5,2'de eğer web tarayıcısı üzerinde herhangi bir kısıtlamaya gidilmemişse zararlı kod vasıtasıyla tarayıcı tarafından erişilen her türlü hassas veri okunabilir, değiştirilebilir ve e-posta aracılığıyla farklı yerlere iletilebilir.

Siteler arası script calıştırma zafiyeti olarak bilinen XSS, kötü niyetli kişilerin web uygulaması üzerinden diğer kullanıcılara istemci tarafında calışmak üzere kod (genellikle JavaScript ve HTML) gönderip kötü amaçlarla çalıştırmalarına imkân tanımaktadır.

XSS zafiyetleri uygulamalarda dışarıdan alınan bilgiler için yeterli girdi ve çıktı denetimi yapılmadığı durumlarda ortaya çıkar ve art niyetli bir kullanıcı istediği gibi javascript kodu calıştırarak hedef aldığı kişilere ait oturum bilgilerini çalabilir, hedef aldığı kişilerin browserini istediği gibi yönlendirebilir. Ele geçirdiği kurban browseri kullanılarak iç ağda port tarama, ortamda ses kaydı ve görüntü kaydı gerçekleştirebilir. Reflected (yansıtılan) XSS açıklığı en sık karşılaşılan XSS acıklığı türüdür. İlgili açıklık türünde, hedef sisteme gönderilen kod parçacığı (payload) kalıcı olarak veritabanında tutulmamaktadır. Bu sebeple ilgili açıklığın istismarı için, öncesinde kullanıcı tarafında bir bağlantı ziyaret ettirme şeklinde bir sosyal mühendislik saldırısı gerçekleştirilmelidir. Reflected XSS acıklığı HTTP GET ve POST taleplerinin her ikisinde iletilen parametrelerde de bulunabilir. Reflected XSS açıklığı, temelde hedef sisteme gönderilen payloadun, dönen sunucu cevabı içerisinde encode edilmeden dondurulmesi durumunda açığa çıkmaktadır. Bu durumda isteği yapan istemci tarafında enjekte edilen kod parçacığı eylemini gerçekleştirecektir. Bu açıklık türü istismar edilerek istemci tarafında HTML, javascript, action script benzeri kod parçacıkları sayfaya enjekte edilebilir. Kulanıcı kandırma veya çerez hırsızlığı gercekleştirilebilir.

Çizelge 5.2. Siteler arası kod (XSS) yazma karakterleri

Karakter	**Açıklama**
/*...*/	Çok satırlı açıklama
+	Ekleme, birleştirme
<script></script>	Script ayracı
'	Virgül
"	Çift tırnak
:	Noktalı virgül
()	Ünlem
/	Slaş
<>	Büyük ve küçük işareti
&	Ampersand
<script> alert(document.cookie) </script>	XSS Cümlesi
<img src =’javascript:alert(document.cookie)’>	XSS Cümlesi
&ltscript&gtalert(document.cookie);&ltscript&gtalert	XSS Cümlesi

Dizin geçişi saldırı (Directory traversal attack)

Dizin geçişi, kullanıcı tarafından sağlanan, yetersiz güvenlik doğrulamalı dosya adlarının kötüye kullanılmasından oluşur. Bu saldırının amacı, bir uygulamayı, erişilebilir olması amaçlanmayan dosyalara erişilebilir kılmaktır. Bu saldırı, kodtaki bir hatayı kötüye kullanmanın aksine, yazılım tam da istenildiği gibi çalıştığı halde, güvenlik eksikliğini kötüye kullanır. Dizin geçişi; dizin tırmanma, geri izleme veya ../ (nokta nokta slash) olarak da bilinir. Bu saldırının bazı yöntemleri kurallı saldırılardır [78]. Çizelge 5.3.te dizin geçişi için kullanılan karakterler verilmiştir.

Çizelge 5.3. Dizin geçişi saldırı karakterleri

Karakter	**Açıklama**
%	Yüzde
/	Slaş
../	İki nokta
/etc/passwd	Dizin Geçişi Cümlesi
/etc/shadow	Dizin Geçişi Cümlesi
%00../../../../../../etc/passwd	Dizin Geçişi Cümlesi
/../../../../../../../../../../../boot.ini	Dizin Geçişi Cümlesi
..%c0%af../..%c0%af../..%c0%af../..%c0%af../boot.ini	Dizin Geçişi Cümlesi

Komut enjeksiyonu (Command injection)

Komut enjeksiyonu, savunmasız bir uygulama aracılığıyla, uygulamayı barındıran işletim sistemi ve web uygulaması üzerinde rastgele komutların işletilmesini hedefleyen bir saldırı yöntemidir. Komut enjeksiyonu saldırıları, bir uygulama sistem kabuğuna kullanıcı kaynaklı güvenli olmayan veriler geçirdiğinde mümkündür. Komut enjeksiyonunda, saldırgan tarafından sağlanan uygulama komutları, genellikle savunmasız uygulamanın yetkileri ile işletilir. Komut enjeksiyonu saldırıları, büyük ölçüde, yetersiz kimlik doğrulama nedeniyle mümkün olmaktadır. Saldırgana, sonrasında uygulama tarafından işletilecek kendi kodunu eklemeye izin verdiği için bu saldırı yöntemi kod enjeksiyonundan farklılık göstermektedir. Kod enjeksiyonunda saldırgan, uygulamanın varsayılan işlevlerini, sistem komutlarının işletilmesine gerek duyulmadan genişletmektedir [79]. Çizelge 5.4.'te komut enjeksiyonu için kullanılan karakterler verilmiştir.

Çizelge 5.4. Komut enkeksiyonu karakterleri

Karakter	**Açıklama**
%	Yüzde
|	Ayraç
< >	Büyük ve küçük işareti
/	Slaş
()	Parantez
;id;	Komut Enkeksiyonu Cümlesi
;read;	Komut Enkeksiyonu Cümlesi
;netstat -a;	Komut Enkeksiyonu Cümlesi
\nnetstat -a%\n	Komut Enkeksiyonu Cümlesi
jls –la	Komut Enkeksiyonu Cümlesi

5.2. Anormal Tabanlı Denetim (ATD)

Anormal HTTP istekleri normal isteklerinden farklı davranış gösteren istek tipleridir. ATD, web uygulamasının HTTP istek yapısına uymayan HTTP isteklerinin denetimini gerçekleştirmek için yapılan denetim türüdür. ATD için ilk önce HTTP isteği sayılaştırılarak denetim türlerinin girişleri oluşturulmaktadır. HTTP isteğini sayısallaştırmak için Alfanümerik Karakter Analizi (AKA), Harf Frekans Analizi (HFA) ve İstek Uzunluk Analizi (İUA) olmak üzere üç öznitelik seçilmiştir. ATD BS, YSA, BTYSA denetim türleri kullanılarak öğrenme tabanlı olarak gerçekleştirilmiştir. Geliştirilen denetim türü modelleri öğrenme tabanlı olarak HTTP isteklerini anormal veya normal olarak sınıflandırmaktadır. Anormal olarak tespit edilen istekler AİİTD veritabanına eklenmetedir. Normal olarak tespit edilen istekler ise NİİTD veritabanına eklenmektedir. Böylece aynı HTTP istekleri istemci tarafından tekrar talep edildiği zaman, HTTP isteği anormal olarak tespit edildiyse hız performansı bakımında İTD'ye göre çok düşük performansa sahip olan ATD devreye girmeden HTTP isteği engellenmektedir veya normal olarak tespit edildiyse web uygulamasına gönderilerek istemci cevabı web sunucu tarafında oluşturulmaktadır ve web uygulaması çalıştırılmaktadır. Aşağıda ATD için gerçekleştirilen öznitelik çıkarımı ve BS, YSA, BTYSA denetim türlerinin tasarımı ve gerçekleştirilmesi anlatılmıştır.

5.2.1. Öznitelik çıkarımı ve seçimi

Öznitelik, makine öğrenmesi veya örüntü tanıma süreçlerinde bir ölçü parametresidir. Öznitelik çıkarımı verinin, hesaplama gereksiminlerini ve çok boyutluluğundan kaynaklanan dezavantajlarını azaltırken belirleyici performansını artırarak anlaşılabilir ve kullanılabilir olmasını sağlamaktadır [80].

Öznitelik seçimi yöntemleri, makine öğrenmesi veya örüntü tanıma uygulamalarında hesaplama zamanının azaltılmasını, tahmin performansının artırılmasını ve verinin daha iyi anlaşılmasını sağlayan yollar sunmaktadır.

Makine öğrenmesi sürecinde gereksiz değişkenlerin azaltılması sorununu çözmek için öznitelik çıkarımı ve seçimi ile ilgili teknikler geliştirilmiştir. Bu teknikler öznitelik çıkarımı sürecinde ilgili alanda kullanılabilecek özniteliklerin belirlenerek kullanılan filtreleme veya sarmalama yöntemleridir. Genelde uzman görüşü alınarak çıkarımı yapılan özniteliklerin, filtreleme veya sarmalama işlemlerinden sonra seçimi gerçekleştirilmektedir. Sarmal modelde seçilen öznitelikler öğrenme algoritmasının performasıyla değerlendirilir. Dolayısıyla sarmal model en iyi özniteliği bulmak için çok fazla zaman ve hesaplama kaynağı gerektirmektedir. Filtreleme modeli herhangi bir öğrenme algoritmasını dikkate almadan veri kümesinin istatistiksel özellikleri dikkate alınır. Öznitelik seçiminde hesaplama verimliliği nedeniyle filtreleme modeli daha fazla tercih edilmektedir [11].

Bu tez çalışmasında öznitelik seçiminde hesaplama verimliliğinden dolayı filtreleme yöntemi kullanılmıştır. Filtreleme yöntemi değişken seçimi için temel ölçüt olarak sıralama yöntemini kullanmaktadır. Sıralama yöntemi kolay kullanımı ve başarılı sonuç üretmesi nedeniyle tercih edilmektedirler. Sıralama işlemi, değişkenlerin skorlanması ve sınır değerlerin altında kalan değişkenlerin seçiminin iptal edilmesi şeklinde gerçekleştirilmektedir. Sıralama metodu sınıflandırmadan önce uygun olmayan değişkenleri filtrelemek için kullanılmaktadır [81].

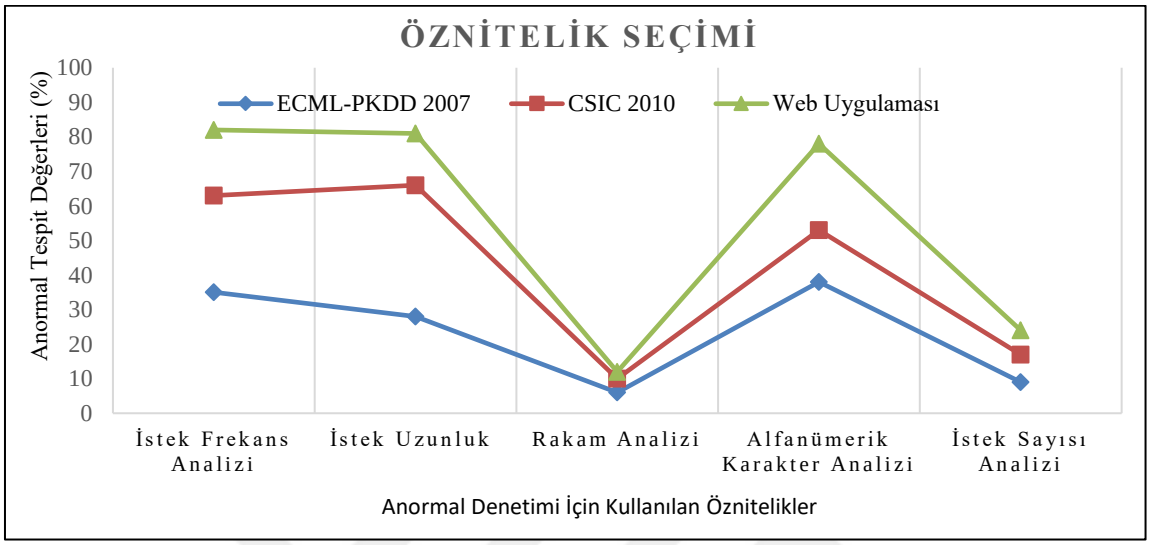
Web tabanlı saldırıların denetiminde, öznitelik seçimi ile ilgili literatürde yapılan en önemli çalışmalardan biri Nguyen ve diğerleri [11] tarafından yapılmıştır. Çalışmada HTTP trafiğinin denetimi için öznitelik seçimi yapılmıştır ve web tabanlı saldırıların tespiti ile ilgili uzman görüşleri alınarak 30 öznitelik belirlenmiştir. Belirlenen özniteliklerin, KDDCUP 1999, CSIC 2010 ve ECML/PKDD 2007 veri kümeleri kullanılarak, denetimler yapılmıştır. Her bir veri kümesinde ayrı ayrı yapılan denetim sonucunda, ortak olarak kullanılabilen özniteliklerin seçimi istatistiksel olarak yapılmıştır. Seçilen özniteliklerden *İstek Uzunluk*, *İstekte Bulunan Rakam Sayısı* ve *İstekte Bulunan Özel Karakterlerin Sayısı* öznitelikleri bütün veri kümelerinde ortak olarak kullanıldığı için bu tez çalışmasında da öznitelik olarak kullanılmak üzere seçilmiştir.

HTTP isteğinin bazı bölümlerinin gereğinden uzun olması bellek taşması saldırısına sebep olabildiği için istek uzunluğunun öznitelik olarak kullanılması önemlidir. Alfanümerik olmayan karakterler SQL enjeksiyonu ve XSS enjeksiyonu gibi web tabanlı saldırı türlerinin gerçekleştirilmesinde kullanılan özel karakterlerdir. Dolayısıyla alfanümerik karakterlerin denetiminin yapılması web tabanlı saldırıların denetiminde öznitelik olarak kullanılması gerekmektedir. HTTP metin tabanlı yapıya sahip olduğu için, HTTP denetimi aynı zamanda metin analizi işlemidir. Metin analizi sürecinde cümleyi oluşturan harf frekans analizinin yapılması cümle içinde bulunan karakterlerin denetiminde anlamlı sonuçlar üretebilmektedir. Dolasıyla HTTP istek denetiminde harf frekans analizinin öznitelik olarak kullanılması web tabanlı saldırı denetiminde önemli bir denetim mekanizmasını oluşturmaktadır. HTTP isteklerini oluşturan bir diğer karakter seti rakamlardır. HTTP isteğinde bulunan rakam sayısının isteğin anormal olup olmadığını belirlenmesinde istatistiksel olarak kullanılmaktadır. HTTP istek sayısı, web uygulamalarına gelen HTTP istekleri, web uygulamasının ziyaretçi sayısına göre sürekli olarak tekrarlanmaktadır. Saldırı içeren isteklerin, saldırı içermeyen isteklere göre tekrarlanma ihtimalleri değişmektedir. Dolayısıyla http istek sayısının web tabanlı saldırıların tespit edilmesinde istatistiksel olarak ayırt edici bir özelliğe sahip olabilmektedir.

Bu tez anormal HTTP istek denetimini gerçekleştirebilmek öznitelik seçimi gerçekleştirilmiştir. Anormal HTTP denetimi, metin tabanlı anormallik tespiti gerçekleştirmek olarak tanımlanabilir. ATD gerçekleştirmek için kullanılan öznitelik sayısı, ATD hızını doğrudan etkilemektedir. Dolayısıyla ne kadar az sayıda öznitelik ile ATD gerçekleştirilebilinirse denetim hızı o kadar yüksek olmaktadır. Dolayısıyla ATD'de, denetim gerçekleştirilen öznitelik sayısı ile denetim hızı ters orantılıdır. ATD ile herhangi bir saldırı türüne karşı değil, denetimi gerçekleştirilen web uygulamasının HTTP istek yapısına uymayan anormal HTTP isteklerinin denetimi gerçekleştirilmektedir.

Bu tez çalışmasında Anormal HTTP istek denetimini gerçekleştirebilmek için alfanümerik karakter analizi (AKA), harf frekans analizi (HFA), İstek uzunluk analizi (İUA), İstek Sayısı Analizi (İSA) ve İstek Rakam Analizi (İRA) olmak üzere beş öznitelik belirlenmiştir. Özniteliklerin ATD'de kullanılması için öznitelik seçimi, filtrelme yöntemi kullanılarak istatistiksel olarak denetim performanslarına göre belirlenmiştir. Şekil 5.2'de belirlenen özniteliklerin, WUGD 2015, CSIC 2010 ve ECML-PKDD 2007 veri kümelerine göre anormal denetim performans değerleri yüzde (%) olarak verilmiştir. Her bir özniteliğin

kullanılan veri kümelerine %30'un altında olan denetim oranları alınmamıştır. Bu durumda İSA ve İRA öznitelikleri anormal tabanlı denetim performansı %30'un altında olduğu için ATD özniteliği olarak seçilmemiştir.



Şekil 5.2. Farklı veri kümelerine ait öznitelik seçim değerleri

Alfanümerik karakter analizi (AKA)

Alfanümerik, latin alfabesindeki harf ve rakam (A-Z, a-z, 0-9) kullanan karakter dizisini tanımlamak için kullanılır. Benzer şekilde bu dizinin elemanlarından her biri de alfanümerik olarak tanımlanır. Bilgisayarların veri saklama anında uygun değer hafıza kullanımına imkân sağlamak için oluşturulmuş bir tanım kümesidir. Genellikle bir byte içinde alfanümerik değerin ASCII karşılığı tutulur. Web uygulamasına gelen istekler belirli karakter dizisine sahiptir. Belirlenen bu karakter dizilerinin dışında karakterler içeren HTTP istekleri anormal olarak belirlenmektedir. Alfanümetik karakter analiz için eşitlik (5.1) kullanılmıştır.

$$T = \sum_{i}^{N} R_i \in E \mid (T + 1) \qquad (5.1)$$

T = Toplam Alfanümerik Karakter Değeri,
R = HTTP İsteği,
N = İsteği Oluşturan Toplam Karakter Sayısı,
E = Evrensel Küme,

İstek uzunluk analizi (İUA)

Web uygulamasının geliştirilme yapısına göre web uygulamasına gelen isteklerin belirli bir istek yapısı vardır. Dolayısıyla bu istek yapısının özelliklerinden biri de istek uzunluğudur. Bellek taşması ve siteler arası betik yazma saldırılarının istek uzunluk değerleri normal isteklerden farklıdır. Kruegel ve Vigna tarafından yapılan çalışmaya göre isteklerin ortalama uzunluk ve varyans değerleri kullanılmıştır. İstek uzunluk analiz için eşitlik (5.2) kullanılmıştır [82].

$$p = \frac{\sigma 2}{(l - \mu)2} \tag{5.2}$$

P = Olasılık
μ = Ortalama (İsteklerin ortalama değerleri)
σ = Varyans (isteklerin varyans değerleri)
l = uzunluk (kontrol edilen isteğin uzunluk değerleri)

(2) Formüle göre isteğinin uzunluk değerinin sıfır (0) olması anormallik sınır değerini belirlemektedir. Her bir isteğin anormal olma olasılık değeri formüle göre hesaplanarak bulunan değer, uzunluk değeri sıfır (0) olan isteğin anormallik değerinden küçük ise, söz konusu istek anormal olarak belirlenmektedir. Dolayısıyla istek uzunluk değeri arttıkça isteğin anormal olma olasılığı artmaktadır.

Harf frekans analizi (HFA)

Karakter dağılımı modeliyle isteklerin karakter (harflerin) frekans değerleri belirlenmektedir. Web uygulamasına gelen normal HTTP isteklerini oluşturan karakterlerin harf frekans değerleri normal olmayan isteklerin harf frekans değerlerine göre yüksektir. Karakter dağılımında ASCII karakterleri kullanılmaktadır. ASCII karakterlerinin 0 ile 255 arasında 8 bit değer alan, yazılabilen harf ve rakam karakterlerdir. İstek frekans analiz için (5.3) nolu eşitlik kullanılmıştır.

$$T = \sum_{i}^{N} R_i \in \text{ASCII} \mid T + F(R_i) \tag{5.3}$$

T = Toplam Frekans Değeri
R = HTTP İsteği
N = İsteği Oluşturan Toplam ASCII Karakter Sayısı
F = İstek Harf Frekans Toplamı

Harf frekans analizi daha çok kriptoanaliz yöntemlerinde kullanılan bir tekniktir. Bu çalışmada gelen isteklerdeki her bir karakterin toplam sayısı ve ortalama değerini tespit etmek için harf frekans analizi yapılmaktadır. Örnek olarak index.php?secim=9&mid=50 gibi bir HTTP isteği ele alındığında bu cümleyi oluşturan harflerin frekansı ve ortalama değerleri elde edilmektedir. Frekans değerleri gelen bütün isteklerdeki her bir karakterin sayısını verirken, Ortalama değer ise her bir karakterin toplam değerinin gelen istek sayınına bölünmesi ile bulunmaktadır.

5.2.2. BS modelinin tasarımı ve gerçekleştirilmesi

Anormal istek denetiminde kullanılan denetim türlerinden biri, veri madenciliği araçlarından olan bayes teoremidir. Bayes teoremi, bir olayın meydana gelmesinde birbirinden bağımsız birden fazla etkenin olması koşulunda, olayın hangi etkenin etkinliği ile ortaya çıktığını gösteren sınıflandırma teoremidir [83].

BS teoremi ile öznitelikler arasında ilişki bulunarak her bir özniteliğin değerine göre HTTP isteğinin anormal olup olmadığı önceden tahmin edilmektedir. AKA, İUA ve HFA öznitelikleri BS denetimi için giriş verileri olarak kullanılmıştır. Çizelge 5.5'de BS için kullanılacak özellikleri bulunmaktadır. Çizelge 5.5'deki verilere göre isteğin anormal olup olmaması belirlenmektedir. Değeri 1 (bir) olan öznitelik anormal, 0 (sıfır) olan öznitelik normal istek olarak belirlenmiştir. Her bir isteğin üç farklı özelliğe göre anormal veya normal olarak belirlenmesi uzman görüşü alınarak tespit edilmiştir.

Çizelge 5.5. BS için önsel kullanılan veriler

Sıra	**İUA**	**HFA**	**AKA**	**Sonuç**
1	0	0	0	Normal
2	1	0	0	Normal
3	0	1	0	Normal
4	1	1	0	Anormal
5	0	0	1	Normal
6	1	0	1	Anormal
7	0	1	1	Anormal
8	1	1	1	Anormal

Anormal denetim gerçekleştirilen WUGD 2015 veri kümesinden alınan sonuçlara göre 35208 tekrarlı veri bulunmaktadır. Bu verilerden tekil olan 1533 veride BS gerçekleştirilmiştir. Anormal denetim özelliklerine göre 1533 veriden 390 veri anormal olarak 1143 veri ise normal olarak belirlenmiştir.

P(İstek Anormal) = 1143/1533 = 0,25
P(İstek Anormal Değil) = 390/1533 = 0,75

Çizelge 5.6. BS veri dağılımları

		İUA	HFA	AKA	Toplam
Anormal	**Uygun**	159	60	7	390
	Uygun Değil	231	330	383	
Normal	**Uygun**	1143	1143	1143	1143
	Uygun Değil	0	0	0	
Toplam		1533	1533	1533	1533

Çizelge 5.6'de BS teoremi kullanılarak gerçekleştirilen makine öğrenmesi süreci verilmiştir. İUA özniteliğine göre anormal olarak belirlenen 390 HTTP isteğinden 159 istek İUA özniteliğine göre anormal olduğu için anormal olarak belirlenirken, 231 isteğin ise anormal olarak belirlenmesinin İUA özniteliği dışındaki diğer özniteliklerden ortaya çıkmıştır.

Normal olan 1143 verinin ise istek uzunluk özelliğine, 1128 isteğin istek uzunluğuna göre normal istek, 15 verinin ise normal olma özelliğinin istek uzunluk özelliğinden kaynaklanmadığı ortaya çıkmıştır.

P(İstek Uzun | Anormal İstekler) = 231/390 = 0.59
P(İstek Uzun Değil | Anormal İstekler) = 159/390 = 0.40
P(İstek Uzun | Normal İstekler) = 15/1143 = 0.01
P(İstek Uzun Değil | Normal İstekler) = 1128 / 1143 = 0.98

İstek frekans analizi özelliğine göre anormal olarak belirlenen 390 veriden 330 veri istek frekans özelliğine göre anormal olarak belirlenirken, 30 isteğin ise anormal olarak belirlenmesi frekans analizi özelliği dışındaki diğer özelliklerden ortaya çıkmıştır.

Normal olan 1143 verinin ise frekans analizi özelliğine, 1059 frekans analizine göre normal istek, 84 verinin ise normal olma özelliğinin frekans analizi özelliğinden kaynaklanmadığı ortaya çıkmıştır.

P(Frekans Uygun Değil | Anormal İstekler) = 330/390 = 0.85
P(Frekans Uygun | Anormal İstekler) = 60/390 = 0.155
P(Frekans Uygun Değil | Normal İstekler) = 84/1143 = 0.075
P(Frekans Uygun | Normal İstekler) = 1059 / 1143 = 0.93

Alfanümerik karakter analizi özelliğine göre anormal olarak belirlenen 390 veriden 383 veri istek alfanümerik karater analizine göre anormal olarak belirlenirken, 7 isteğin ise anormal olarak belirlenmesi alfanümerik karakter analizi özelliği dışındaki diğer özelliklerden ortaya

çıkmıştır. Normal olan 1143 verinin ise alfanümerik karakter analizi özelliğine göre normal olarak belirlenmiştir.

P(Alfanümerik Uygun Değil | Anormal İstekler) = 383/390 = 0.98
P(Alfanümerik Uygun | Anormal İstekler) = 7/390 = 0.02
P(Alfanümerik Uygun Değil | Normal İstekler) = 0/1143 = 0
P(Alfanümerik Uygun | Normal İstekler) = 1143 / 1143 = 1

Geliştirilen uygulama CSIC 2010 ve ECML-PKDD 2007 farklı veri kümeleri ile test edilmiştir. Farklı veri kümelerinden alınan sonuçlar Çizelge 5.7 ve Çizelge 5.8'de verilmiştir.

Çizelge 5.7. CSIC 2010 veri kümesi BS dağılımı

		İUA	**HFA**	**AKA**	**Toplam**
Anormal	**Uygun**	163	230	232	278
	Uygun Değil	115	48	46	
Normal	**Uygun**	19	19	19	19
	Uygun Değil	0	0	0	
Toplam		297	297	297	297

Çizelge 5.8. ECML-PKDD 2007 veri kümesi BS dağılımı

		İUA	**HFA**	**AKA**	**Toplam**
Anormal	**Uygun**	230	1114	189	1359
	Uygun Değil	1129	245	1170	
Normal	**Uygun**	2590	2590	2590	2590
	Uygun Değil	0	0	0	
Toplam		3949	3949	3949	3949

Çizelge 5.9'da BS denetim türünün, WUGD2015, CSIC 2010 ve ECML-PKDD 2007 verikümeleri ile yapılan denetim sonuçları ve yanlış pozitif oranları verilmiştir. WUGD 2015 veri kümesinde bulunan anormal http isteklerinin %97,6'sı, CSIC 2010 veri kümesinde bulunan anormal HTTP isteklerinin % 94,5'i ve ECML-PKDD 2007 veri kümesinde bulunan HTTP isteklerinin % 93,3'ü tespi edilmiştir. Yanlış pozitif oranları normal olarak tespit edilen anormal HTTP isteklerini ifade etmektedir. WUGD 2015 veri kümesinde bulunan anormal HTTP isteklerinin % 0,2'si, CSIC 2010 veri kümesinde bulunan anormal HTTP isteklerinin % 0,3'ü ve ECML-PKDD 2007 veri kümesinde bulunan anormal HTTP isteklerinin % 0,6'sı normal olarak tespit edilmiştir.

Çizelge 5.9. BS veri kümeleri denetim başarı oranı karşılaştırması

Denetim Türü	**Veri Kümesi**	**%**	**YP**
BS	**WUGD 2015**	%97,6	0,2
	CSIC 2010	%94,5	0,3
	ECML-PKDD 2007	%93,3	0,6

5.2.3. YSA modelinin tasarımı ve gerçekleştirilmesi

Geliştirilen WUGD sistemi, YSA modelinin performans özellikleri kullanılarak oluşturulmasına imkân sağlayacak şekilde geliştirilmiştir. Sistemde YSA performans özellikleri değiştirilerek daha iyi YSA modelin üretilmesi sağlanabilmektedir. Tez kapsamında en iyi YSA modelinin oluşturulması için, web uygulamalarına gelen HTTP istekleri kullanılarak YSA ile en iyi öğrenme sonuçları üretebilecek YSA modeli ortaya çıkarılmıştır. Öncelikle, giriş ve çıkış verileri sonlu sayılarla ifade edebilmek için 0-1 arasında normalize edilmiştir. Normalizasyon işleminde Min-Max yöntemi kullanılmıştır. Min-Max yöntemi, verileri doğrusal olarak normalize etmektedir. Minimum; bir verinin alabileceği en düşük değeri ifade ederken, maksimum; verinin alabileceği en yüksek değeri ifade etmektedir. Bir veriyi Min-Max yöntemi ile 0 ile 1 aralığına indirgemek için eşitlik (5.4) kullanılmıştır.

$$x' = \frac{x_i - x_{min}}{x_{max} - x_{min}} \tag{5.4}$$

x' = Normalize edilmiş veriyi,
x_i = Girdi değerini,
x_{min} = Girdi seti içerisinde yer alan en küçük sayıyı,
x_{max} = Girdi seti içerisinde yer alan en büyük sayıyı, ifade etmektedir.

En iyi YSA modelinin oluşturulması için Çizelge 5.10'te belirtilen performans değerleri kullanılmıştır. Kullanılan performans değerleri denemeler sonucunda oluşturulmuştur.

Çizelge 5.10. Yapay sinir ağı modeli performans değerleri

İterasyon Sayısı	1000
Eğitim Verisi	% 80
Test Verisi	% 20
Performans Kontrolü	Mean Square Error (MSE)
Gizli Katman Sayısı	1
YSA Modeli	3-10-1
Öğrenme Oranı	0,05
Momentum	0,01
Gizli Katman Nöron Sayısı	10
Hata Payı	0,01

YSA eğitim işleminin amacı, minimum hataya az sayıda gizli katman ve nöronlar ile ulaşmaktır. YSA ağırlıklarının eğitiminde, ileri beslemeli geri yayılım (feed-forward backpropagation) algoritması seçilmiştir. Bu noktada en önemli işlem, gizli katman

sayısının, katmanlardaki nöron sayısının ve nöron aktivasyon fonksiyonlarının belirlenmesi işlemidir. Gizli katman ve nöron sayıları en iyi YSA sonucunu alabilmek için deneme yoluyla belirlenmektedir. Katman sayısı, problemin zorluğuna göre artırılabilir, ancak katman sayısının fazla olması işlem süresinin uzamasına ve ağın ezberlemesine sebep olmaktadır.

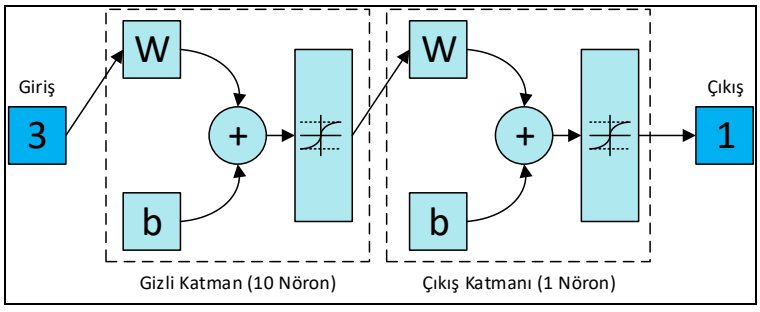
En iyi YSA modelinin belirlenmesinde, elde edilen sonuçlar içerisinden hataların kareleri ortalaması (Mean Squared Error – MSE) değeri en düşük (0'a yakın) ve regresyon değeri ise en yüksek (1'e yakın) olan model kullanılmaktadır.

Özetle, veri kümeleri ile kurulan farklı modeller ile denemeler sonucunda en iyi ağ modelini belirlemek için;

- Normalize yöntemi olarak "min maks" kullanılmıştır.
- Transfer fonksiyonu (Aktivasyon Fonksiyonu) olarak gizli katman ve çıkış katmanı için hiperbolik tanjant kullanılmıştır.
- Eğitim algoritması olarak ileri beslemeli geri yayılım algoritması kullanılmıştır.
- Performans fonksiyonu olarak ise "Hataların Kareleri Ortalaması - MSE" kullanılmıştır.

Ağ eğitimini gerçekleştirmek için, tüm verilerin %80'i eğitim, %20'si ise test verisi olarak belirlenmiştir. Oluşturulan ağ 1 gizli katmanda 10 nörona sahiptir.

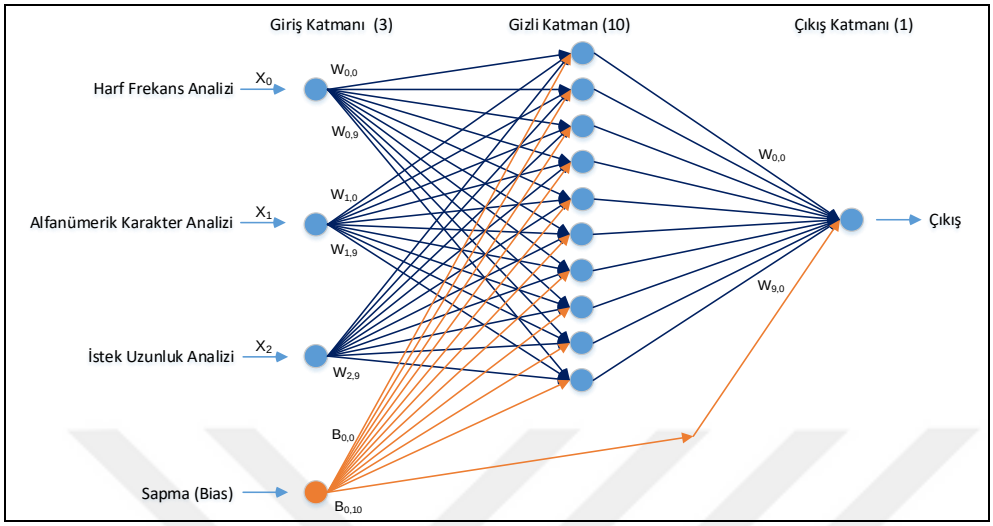
Oluşturulan sinir ağı modelinin yapısı Şekil 5.3'te verilmiştir. YSA modeli 3 giriş, 10 nörona sahip 1 gizli katman ve 1 çıkış katmanından oluşmaktadır. Şekil 5.3'te verilen "W" giriş değerleri ile çarpılarak toplanan ağırlık değerlerini, "b" ise ağ eğitimini iyileştirmek için kullanılan bias değerini ifade etmektedir.



Şekil 5.3. Tasarlanan yapay sinir ağı modeli

Geliştirilen YSA modelinin düğümler bazında gösterimi şekil 5.4'te verilmiştir. YSA eğitiminin en önemli aşaması giriş katmanı, gizli katman ve çıkış katmanı arasındaki ağırlıkların belirlenmesi aşamasıdır. Önerilan YSA modelinde, 3 giriş, 1 çıkış, 1 gizli katman

ve 10 gizli katman nöronu bulunmaktadır. Şekil 5.4.'te oluşturulan ağın topolojik yapısı verilmiştir. Ayrıca her nörona katkı sağlayacak sapma (bias) değeri eklenmiştir.



Şekil 5.4. Yapay sinir ağı modeli (3-10-1)

Oluşturulan ağ için ağırlık vektörleri ilk başta -10 ile 10 arasında rasgele üretilmektedir. Çizelge 5,11'de tasarlanan ağın kısımlarına göre sahip olduğu toplam ağırlık sayıları verilmiştir. 3-10-1 mimarisinde tasarlanan ağın toplam 51 ağırlık değeri bulunmaktadır. Ağırlık sayısı aşağıdaki gibi hesaplanmaktadır.

Toplam Ağırlık Sayısı = (Giriş Sayısı * Gizli Katman Nöron Sayısı) + (Gizli Katman Nöron Sayısı * Çıkış Sayısı) + (Sapma Sayısı * Gizli Katman Nöron Sayısı) + (Çıkış Sayısı)

Toplam Ağırlık Sayısı = (3*10) + (10*1) + (1*10) + (1*1) = 30 + 10 + 10 + 1 = 51

Çizelge 5.11. Geliştirilen YSA modelindeki toplam ağırlık sayısı

	1	**2**	**3**	**4**	**5**	**6**	**…**	**29**	**30**	**Toplam**
Girişten Gizli Katmana	$W_{0,0}$	$W_{0,1}$	$W_{0,2}$	$W_{0,3}$	$W_{0,4}$	$W_{0,5}$	…	$W_{2,8}$	$W_{2,9}$	30
Gizli Katmandan Çıkışa	$W_{0,0}$	$W_{0,1}$	$W_{0,2}$	$W_{0,3}$	$W_{0,4}$	$W_{0,5}$	…			10
Sapmadan Gizli Katmana	$W_{0,0}$	$W_{0,1}$	$W_{0,2}$	$W_{0,3}$	$W_{0,4}$	$W_{0,5}$	…			10
Sapmadan Çıkışa	$W_{0,0}$									1
Toplam	4	3	3	3	3	3	…	1	1	51

YSA modeli; web uygulamasına gelen isteklerin anormal veya normal olarak belirlenmesi amacına yönelik öğrenme yöntemini belirlemek için geliştirilmiştir. WUGD yazılımı

Microsoft Visual Studio 2015 aracı kullanılarak .NET Framework platformunda, C# ASP.NET programlama dili kullanılarak geliştirilmiştir.

Belirlenen ağırlıkların eğitiminde ileri beslemeli geri yayılım algoritması kullanılmıştır. Geriyayılım algoritması, bir hedef fonksiyonu minimize etmek üzere tasarlanmış en uygun şekle sokma tekniğidir. En sık kullanılan hedef fonksiyon hataların karelerinin ortalamasıdır. $\mathcal{E}$, hata terimini göstermek üzere, hata (5.5 a) ve hedef fonksiyonun tanımlanmasında kullanılan hatanın karesi (5.5 b) eşitliğiyle verilmiştir.

$$\varepsilon = \varepsilon_q = \left[T_q - \Phi_{qk}(\mathrm{I})\right] \tag{5.5 a}$$

$$\varepsilon^2 = \varepsilon_q^2 = \left[T_q - \Phi_{qk}(\mathrm{I})\right] \tag{5.5 b}$$

Yukarıda kullanılan $\mathcal{E}$, k katman numarası, q nöron numarasını göstermektedir. Eşitlik (5.6 a) Delta kuralıyla ifade edildiği üzere, ağırlık değerindeki değişim, hatanın karesinin ağırlığa göre değişim oranıyla orantılıdır.

$$\Delta w_{pq,k} = -\eta_{p,q} \frac{\partial \varepsilon_q^2}{\partial w_{pq,k}} \tag{5.6 a}$$

Kısmi türev, zincir kuralı kullanımıyla açılarak eşitlik (5.6 b) ile hesaplanmaktadır.

$$\Delta w_{pq,k} = -\eta_{p,q} \frac{\partial \varepsilon_q^2}{\partial \Phi_{q,k}(\mathrm{I})} \frac{\partial \Phi_{q,k}(\mathrm{I})}{\partial \mathrm{I}_{q,k}} \frac{\partial \mathrm{I}_{q,k}}{\partial w_{pq,k}} \tag{5.6 b}$$

Bu eşitlikte,

$$\frac{\partial \varepsilon_q^2}{\partial \Phi_{q,k}(\mathrm{I})} = -2\left[T_q - \Phi_{q,k}(\mathrm{I})\right] \tag{5.7 a}$$

$$\frac{\partial \Phi_{q,k}(\mathrm{I})}{\partial \mathrm{I}_{q,k}} = \alpha \Phi_{q,k}(\mathrm{I})\left[1 - \Phi_{q,k}(\mathrm{I})\right] \tag{5.7 b}$$

$$\mathrm{I}_{q,k} = \sum_{p=1}^{n} w_{pq,k} \Phi_{p,j}(\mathrm{I}) \tag{5.7 c}$$

$$\frac{\partial \mathrm{I}_{q,k}}{\partial w_{pq,k}} = \Phi_{p,j}(\mathrm{I}) \qquad (5.7\ \mathrm{d})$$

(5.7) eşitlikleri (5.6 b) eşitliğine yerleştirilirdiğinde eşitlik (5.8) elde edilerek ağırlıkların eğitimleri gerçekleştirilirken elde edilen hata farklıları tespit edilmektedir.

$$\Delta w_{pq,k} = -\eta_{p,q}(-2\alpha)\left[T_q - \Phi_{q,k}(\mathrm{I})\right]\Phi_{q,k}(\mathrm{I})\left[1 - \Phi_{q,k}(\mathrm{I})\right]\Phi_{p,j}(\mathrm{I}) \qquad (5.8)$$

Geriyayılacak hata terimi olan δ aynı zamanda kısa bir gösterim elde etmek için eşitlik (5.9) elde edilmektedir.

$$\delta_{pq,k} = (2\alpha)\left[T_q - \Phi_{q,k}(\mathrm{I})\right]\Phi_{q,k}(\mathrm{I})\left[1 - \Phi_{q,k}(\mathrm{I})\right] \qquad (5.9)$$

Dolayısıyla eşitlik (5.9), eşitlik (5.8)'de yerine konursa eşitlik (5.10) elde edilerek geriye yayılımlarda hata oranları farkı hesaplanmaktadır.

$$\Delta w_{pq,k} = -\eta_{p,q}\frac{\partial \varepsilon_q^2}{\partial w_{pq,k}} = -\eta_{p,q}\delta_{pq,k}\Phi_{p,j}(\mathrm{I}) \qquad (5.10)$$

N, iterasyon sayacı olmak üzere, (N+1). adım için ayarlanacak ağırlık değeri eşitlik (5.11) ile hesaplanmaktadır.

$$w_{pq,k}(N+1) = w_{pq,k}(N) - \eta_{p,q}\delta_{pq,k}\Phi_{p,j}(\mathrm{I}) \qquad (5.11)$$

Eşitlik (5.11)'de tanımlanan işlem, uygun ağırlık değerlerine ulaşabilmek için çıkış katmanının tüm nöronlarına uygulanmaktadır. Hedef değerlere ulaşılamamasının bir sebebi hatalı çıkış katmanı ağırlıklarıyken diğeri, gizli katmanın ürettiği hatalı çıkışlardır. Gizli katmanın ağırlıklarının ayarlanmasında kullanılan eşitlikler, hedef değer olmaksızın hesaplanması gereken hata terimi $\delta_{hp,j}$ dışında aynı olmaktadır. $\delta_{hp,j}$, çıkış katmanında bağlantılı olduğu her nöronun hata terimine katkı yapan gizli katman nöronlarının herbiri için hesaplanmaktadır.

YSA eğitimi gerçekleştirilirken eğitim başarısını etkileyen özellikler sapma, momentum, öğrenme katsayısı, transfer fonksiyon, eğitim ve test verilerini dağılımı ve yerel minimum değerleridir. Geliştirilen ağ modelinde bu değişkenlerin YSA eğitimine etkisi anlatılmıştır.

Sapma (bias): Her bir nöron için bir sapma elemanı ilave edilebilir. Aktivasyon fonksiyonunun apsisi kestiği noktayı öteleyerek nöronun eşik seviyesinde değişiklik etkisi yaratılır. Sapma genellikle, eğitim hızını olumlu etkiler. Giriş elemanlarının sapması +1 olması durumunda, diğer sapmalar herhangi bir değer alabilir ve eğitilebilir.

Momentum: Hareketli bir cismin momentumunun etkisine benzer ve eğitim sürecinin yönünün korunmasını sağlar. Bunun için, ağırlık ayarlaması sırasında, önceki ağırlık değişimiyle orantılı bir terim ilave edilir. μ, momentum terimi olmak üzere eşitlik (5.12) elde edilir:

$$w_{pq,k}(N+1) = w_{pq,k}(N) - \eta_{p,q}\delta_{pq,k}\Phi_{p,j}(\mathrm{I}) + \mu\Delta w_{pq,k}(N) \qquad (5.12)$$

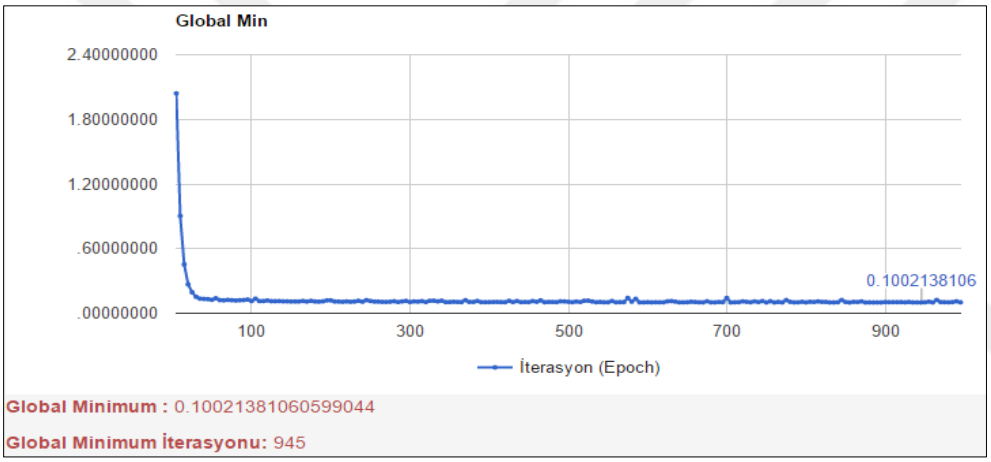
Yerel minimumdan kurtulmayı sağladığı için ağırlık eğitimde etkilidir.

Öğrenme Oranı: Öğrenme oranı için 0-1 aralığı uygun aralıktır. Bu aralıkta seçilen oranın büyüklüğü, öğrenme adım aralığı ile doğru orantılıdır. Öğrenme oranı pozitif olmak zorundadır ve öğrenme oranı 2'den büyük olduğu zaman YSA'nın kararsızlığına, 1'den büyük olduğu zaman ise salınım yapmasına sebep olmaktadır. Eğitim verisinin değişim oranına uygun olarak öğrenme oranı seçilmelidir.

Transfer Fonksiyon: Tanjant-sigmoid fonksiyonu eğiminin(türevinin) çok küçük değerli bölgelerinde işlem yapmasını sağlar. Geri yayılımda hata terimi türevle orantılı olduğundan, düşük eğimde yeterli eğitim gerçekleşmez. Eğimin ayarlanması, eğitim süresini ve başarısını doğrudan etkiler. Tanjant hiperbolik fonksiyonu, sigmoid fonksiyonuna benzer bir fonksiyondur. Sigmoid fonksiyonunda çıkış değerleri 0 ile 1 arasında değişirken hiperbolik tanjant fonksiyonunun çıkış değerleri -1 ile 1 arasında değişmektedir.

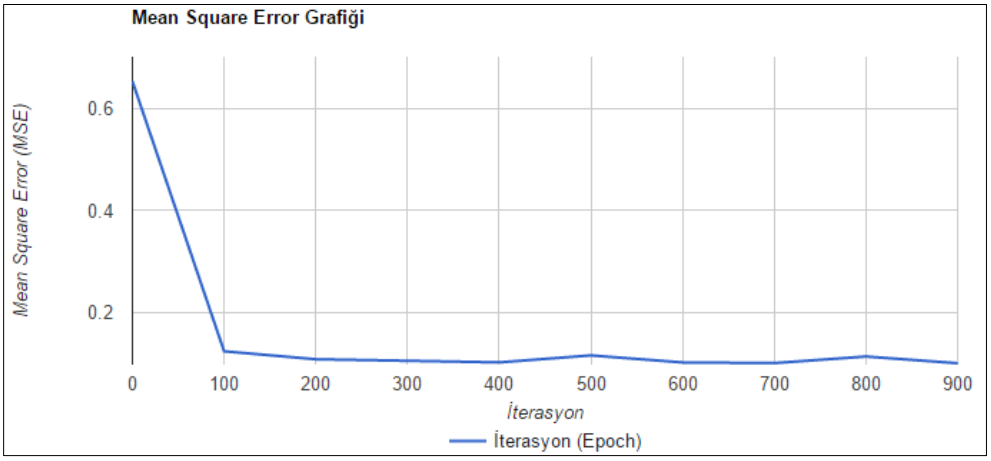
Eğitim ve Test Verileri: YSA eğitimi, eğitim ve test süreçlerinden oluşmaktadır. YSA modelini oluşturmak ve ağırlıkları eğitmek için eğitim veri kümesi kullanılır. Oluşturulan modelin doğruluğunu test ekmek için ise test verikümesi kullanılır. Eğitim veri kümesinin kendi içinde doğrusal dağılım göstermesi modelin oluşturulmasında ve ağırlıkların eğitiminde daha iyi başarı sağlamaktadır. Test veri kümesi eğitim veri kümesinden farklı verileri içermelidir.

Global Minimum: Geriyayılım algoritmasının dezavantajı, yerel minimuma takılmasıdır. Algoritma, gradyan azaltma yöntemini kullandığı için hata yüzeyinin eğimi negatif olduğu sürece ağırlıkların minimuma ulaşmayı sağlayacak şekilde ayarlandığı kabul edilir. Hata yüzeyi, kolaylıkla düşülebilecek fakat kurtulmanın mümkün olamayabileceği tepe ve çukurları barındırabilir. Atanan ilk ağırlık değerleri civarında, azalan gradyan yönünde arama yapılacak noktayı belirler. Rasgele seçilen ilk noktadan, global minimuma kadar olan mesafede yerel minimum problemini yaratabilecek hata değerleriyle karşılaşmak muhtemeldir. Şekil 5.5'da yerel minimumlar göstermektedir. Seçilen performans değerleri neticesinde ağ eğitiminin gerçekleştiği yerel minimum 0.1002138106 olarak tespit edilmiştir.



Şekil 5.5. Karşılaşılan yerel minimum değerleri

Şekil 5.4'te geliştirilen YSA modeline göre eğitilen ağın performans grafiği Şekil 5.6'da verilmiştir. Şekil 5.6'da verilen hataların karelerinin ortalaması, her 100 iterasyonda hesaplanarak her bir eğitim için tekrar sayısına (iterasyon) göre değişimi verilmiştir. Eğitimin hızını artırmak için her 100 iterasyonda hesaplanmıştır.



Şekil 5.6. YSA modelden elde edilen performans

Çizelge 5.12'te YSA denetim türünün, WUGD2015, CSIC 2010 ve ECML-PKDD 2007 verikümeleri ile yapılan denetim sonuçları ve yanlış pozitif oranları verilmiştir. WUGD 2015 veri kümesinde bulunan anormal http isteklerinin %98,52'si, CSIC 2010 veri kümesinde bulunan anormal HTTP isteklerinin % 96,74'ü ve ECML-PKDD 2007 veri kümesinde bulunan HTTP isteklerinin % 94,53'ü tespi edilmiştir. Yanlış pozitif oranları normal olarak tespit edilen anormal HTTP isteklerini ifade etmektedir. WUGD 2015 veri kümesinde bulunan anormal HTTP isteklerinin % 0,4'ü, CSIC 2010 veri kümesinde bulunan anormal HTTP isteklerinin % 1,1'i ve ECML-PKDD 2007 veri kümesinde bulunan anormal HTTP isteklerinin % 1,3'sı normal olarak tespit edilmiştir.

Çizelge 5.12. YSA veri kümeleri ile karşılaştırma

Denetim Türü	**Veri Kümesi**	**%**	**YP**
YSA	**WUGD 2015**	98,52 %	0,4
	CSIC 2010	96,74 %	1,1
	ECML-PKDD 2007	94,53 %	1,3

5.2.4. BTYSA modelinin tasarımı ve gerçekleştirilmesi

BTYSA, belirlenen sinir ağı ağırlıklarının eğitiminin bayes sınıflandırma kullanılarak optimize edilmesiyle ağ eğitiminin gerçekleştirmesidir. BTYSA modelinin geliştirimesi YSA modelinin geliştirisi ile benzer süreçlerden oluşmaktadır. BTYSA için de Çizelge 5.11'de verilen performans değerleri kullanılmıştır. Şekil 5.4'te verilen sinir ağı modeline göre belirlenen ağırlıkların eğitimi BTYSA kullanılarak gerçekleştirilmiştir. Ağın eğitiminin amacı E_D hata oranının azaltılmasıdır. Ağın çıkış değeri ile istenen çıkış d_l arasındaki fark

E_D hata oranını oluşturmaktadır. BTYSA ile E_D hata oranının en aza indirilmesi amaçlanmaktadır.

$$E_D = \frac{1}{2}\sum_{l=1}^{n}\{d_l - \varphi[\sum_{j=1}^{h} w_{jl}\phi(\sum_{i=1}^{d} v_{ij}x_i + a_j) + b_l]\}^2 \quad (5.15)$$

Şekil 5.4'te verilen sinir ağı modeline göre belirlenen ağırlıkların toplam sayıları Çizelge 5.12'de verilmiştir. Belirlenen ağırlıkların eğitiminde veri madenciliği yöntemlerinden bayes sınıflandırma kullanılmıştır. Bayes yaklaşımında, ağ eğitilirken tahmin edilen parametrelerdeki belirsizliklerin, belli bir dağılımı takip edilmektedir. BTYSA'nın amacı, YSA'da ağ eğitmek için girişler kullanılarak çıkışlar tahmin edilmektedir. Bu aşamada girişlerin ağırlıkları rastgele atanmaktadır. Bayes yöntemi ile ağırlıkların istatistiksel olarak seçimi gerçekleştirilerek, YSA ağırlıklarının bulunduğu çok geniş bir veri kümesi içerisinde iyileştirme yapılmaktadır.

Veri kümesindeki ağırlık parametrelerinin olasılık yoğunluklarını p(w|D) yardımıyla bularak bu dağılımı (5.16) eşitliğiyle daraltılarak daha iyi sonuç alabilmek için YSA'da rastgele oluşturulan ağırlık değerleri bayes teoremi ile iyileştirilerek YSA'nın tahmin değerleri iyileştirilmektedir.

$$P(w|D) = \frac{P(D|w)\,P(w)}{P(D)} \quad (5.16)$$

Eşitlik 5.16'daki P(D|w) ağırlık kümesinde bulunan giriş özniteliklerinin komşuluk değerlerini ifade etmektedir. P(D) ağırlık parametre uzayını ifade etmekt üzerinden bir integral tarafından hesaplanan eşitliktir.

Bayes teoremi ağırlıklardaki (w) belirsizliği azaltarak, aşırı iyileştirme (ezberleme) problemini de azaltmaktadır. Bayes ile ağırlıkların bir sonraki dağılımı elde edilerek, bir sonraki ağırlık dağılımının belli bir düzlemde olması sağlanmaktadır.

E_D hata oranını azaltmak için ağırlık değerleri değiştirilmektedir. Böylece, ağırlık değerlerinin ayarlamasını hata değerlerinin azaltılması oranıyla yapılmaktadır.

$$\Delta w_{jk} = -\eta\frac{\partial E}{\partial w_{jk}} \quad (5.17)$$

$$\Delta v_{jk} = -\eta\frac{\partial E}{\partial v_{ij}} \quad (5.18)$$

Eşitlik 5.17 ve 5.18'de η eğitim ifadesinin öğrenme oranını ifade etmektedir. Eğitim süreci geleneksel geri yayılım sinir ağıdır. Geri yayılım algoritmalarını problemi ise aşırı öğrenme ve ağı genelleştirememesidir.

$$E_W(w) = \frac{1}{2}\sum_{i=1}^{m} w_i^2 \tag{5.19}$$

E_w (m ağ ağırlık parametrelerinin toplam sayısını, W_i ise ağırlık parametrelerini ifade etmektedir) ağ ağırlıklarının karesinin toplamını α ve β hiper parametreleri ifade etmektedir. Bayes teoremine göre P(w|α)'nın ön olasılığı ve P(D|w, β) benzerlik fonksiyonu bayes teoremini kullanılarak, P(w|D, α, β) sonsal olasılığı elde edilmiştir. D, eğitim için kullanılan n adet giriş ve çıkıştan oluşan veri kümesidir.

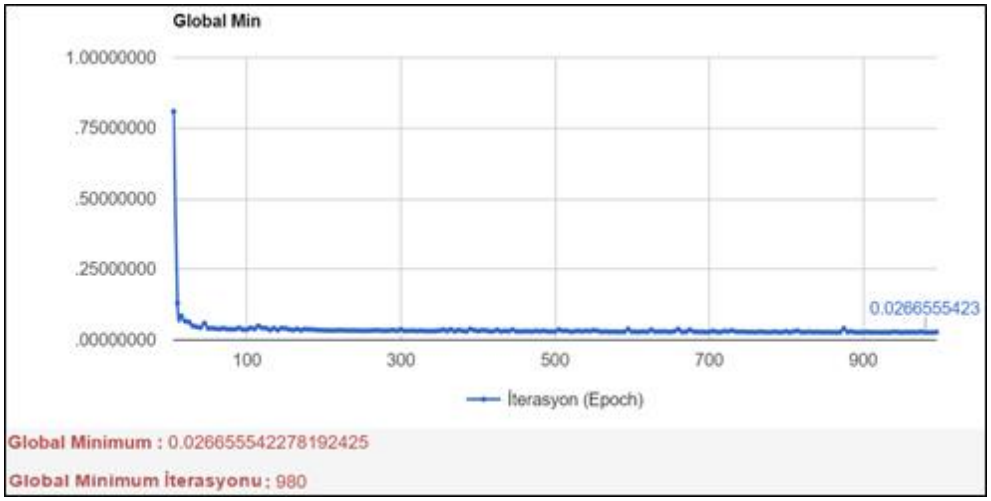
$$P(w|D,\alpha,\beta) = \frac{P(D|w,\beta)P(w|\alpha)}{P(D|\alpha,\beta)} \tag{5.20}$$

P(D | α, β)'nın logaritma farkı kullanılarak eşitlik 5.21 ve 5.22 ifadeleri elde edilmektedir.

$$\alpha = \frac{\gamma}{2E_w} \tag{5.21}$$

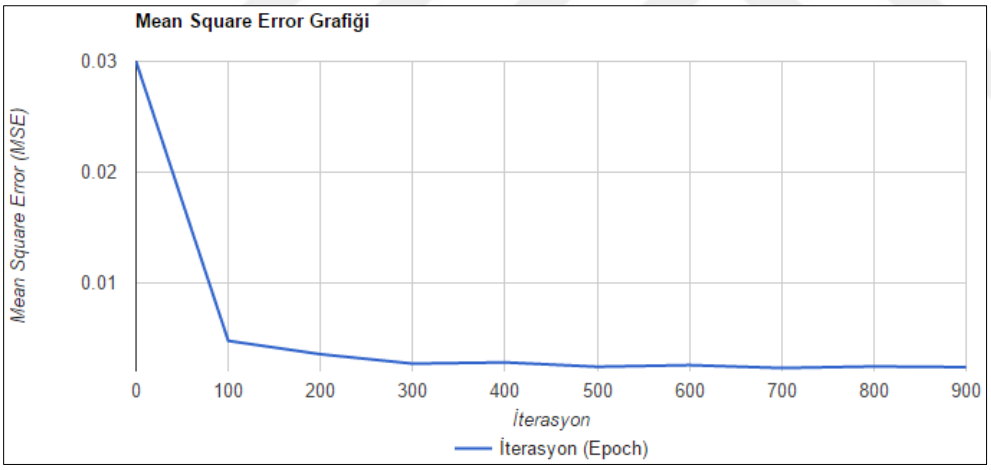
$$\beta = \frac{n-\gamma}{2E_D} \tag{5.22}$$

BTYSA modelinin tasarımı YSA modelinin tasarımı ile benzer süreçlere sahiptir. YSA modelinin tasarım aşamasında kullanılan sapma, momentum, öğrenme oranı, transfer fonksiyon, eğitim ve test verileri vb. performans değerleri BTYSA modelinin tasarımı aşamasında da kullanılmıştır. Kullanılan performans değerleri aynı olmasına rağmen BTYSA modelinin global minimum ve hataların kareleri ortalaması farklı değerlere sahiptir. Şekil 5.7'de BTYSA modeline ait yerel minimumlar göstermektedir. Seçilen performans değerleri neticesinde ağ eğitiminin gerçekleştiği yerel minimum 0.0266555423 olarak tespit edilmiştir.



Şekil 5.7. BTYSA Karşılaşılan yerel minimum değerleri

Şekil 5.4'te geliştirilen BTYSA modeline göre eğitilen ağın performans grafiği Şekil 5.8'de verilmiştir. Şekil 5.8'de verilen hataların karelerinin ortalaması, her 100 iterasyonda hesaplanarak her bir eğitim için tekrar sayısına (iterasyon) göre değişimi verilmiştir.



Şekil 5.8. BTYSA modelinden elde edilen performans

Çizelge 5.13'da BTYSA denetim türünün, WUGD2015, CSIC 2010 ve ECML-PKDD 2007 verikümeleri ile yapılan denetim sonuçları ve yanlış pozitif oranları verilmiştir. WUGD 2015 veri kümesinde bulunan anormal http isteklerinin %97,6'sı, CSIC 2010 veri kümesinde bulunan anormal HTTP isteklerinin % 94,5'i ve ECML-PKDD 2007 veri kümesinde bulunan HTTP isteklerinin % 93,3'ü tespi edilmiştir. Yanlış pozitif oranları normal olarak tespit edilen anormal HTTP isteklerini ifade etmektedir. WUGD 2015 veri kümesinde bulunan anormal HTTP isteklerinin % 0,2'si, CSIC 2010 veri kümesinde bulunan anormal

HTTP isteklerinin % 0,3'ü ve ECML-PKDD 2007 veri kümesinde bulunan anormal HTTP isteklerinin % 0,6'sı normal olarak tespit edilmiştir.

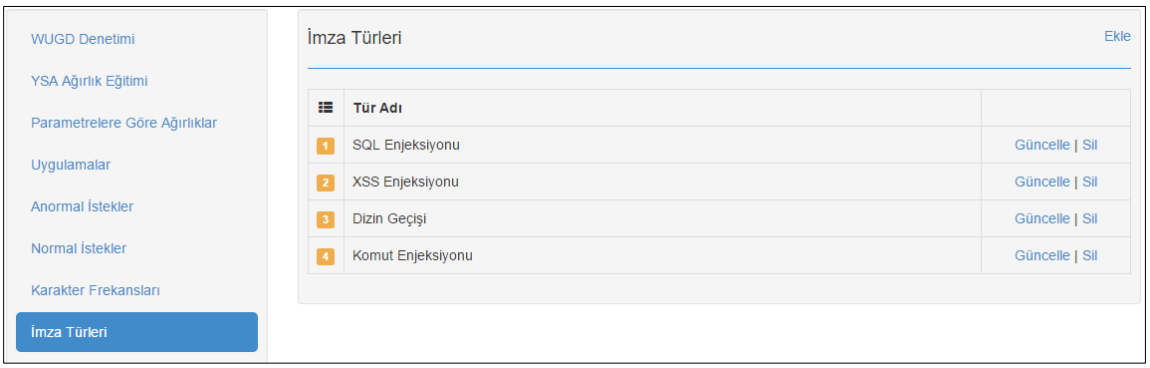
Çizelge 5.13. BTYSA veri kümeleri ile karşılaştırma

Denetim Türü	Veri Kümesi	%	YP
BTYSA	WUGD 2015	98,84 %	0,3
	CSIC 2010	96,83 %	1,4
	ECML-PKDD 2007	95,76 %	1,5

5.3. Geliştirilen Yazılım Arayüzü

Tez kapsamında İTD ve ATD denetim metodlarının beraber kullanıldığı hibrit bir WUGD algoritması geliştirilmiştir. Bu bölümde, geliştirilen algoritmanın, uygulama arayüzü anlatılmıştır. Geliştirilen uygulama iki aşamadan oluşmaktadır. Birinci aşamada bilinen web tabanlı saldırı türlerinin denetiminin yapıldığı İTD gerçekleştirilmektedir. İkinci aşamada ise, anormal HTTP istek denetimi gerçekleştirebilmek için yapay zeka modeli oluşturularak, anormal HTTP istek denetiminin yapıldığı ATD gerçekleştirilmektedir.

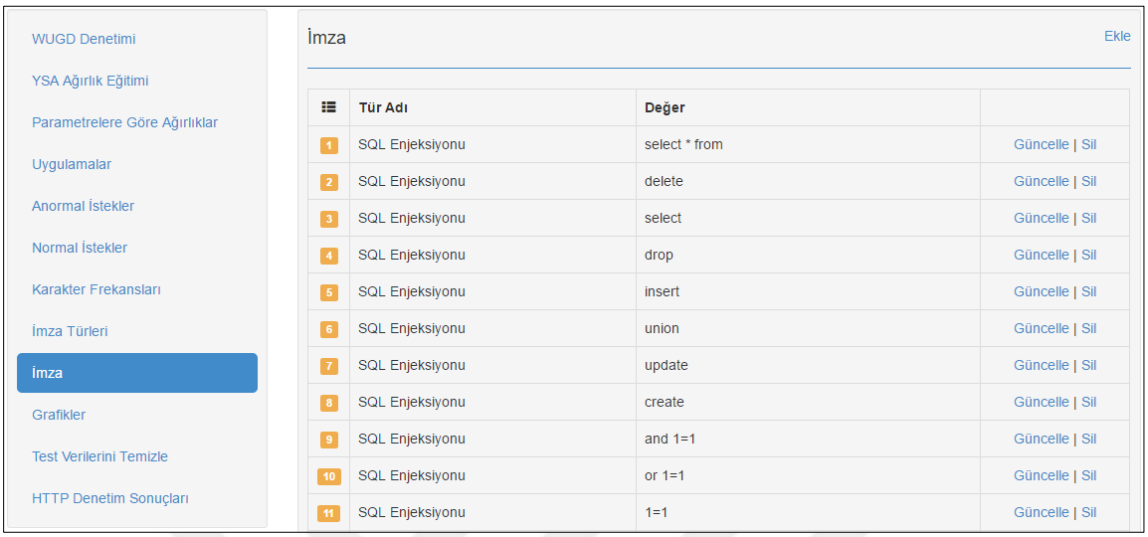
İTD süreci kendi içinde 3 aşamadan oluşmaktadır. Bunlar NİİTD, AİİTD ve BSTİTD süreçleridir. Web uygulamalarına yapılan web tabanlı bilinen saldırı türlerinin denetimini gerçekleştirmek için BSTİTD süreci gerçekleştirilmektedir. Şekil 5.9'da verilen ekranda, SQL enjeksiyonu, XSS enjeksiyonu, dizin geçişi vb. bilinen saldırı türlerinin tanımlamaları gerçekleştirilmektedir.



Şekil 5.9. Denetimi yapılan bilinen web tabanlı saldırı türleri

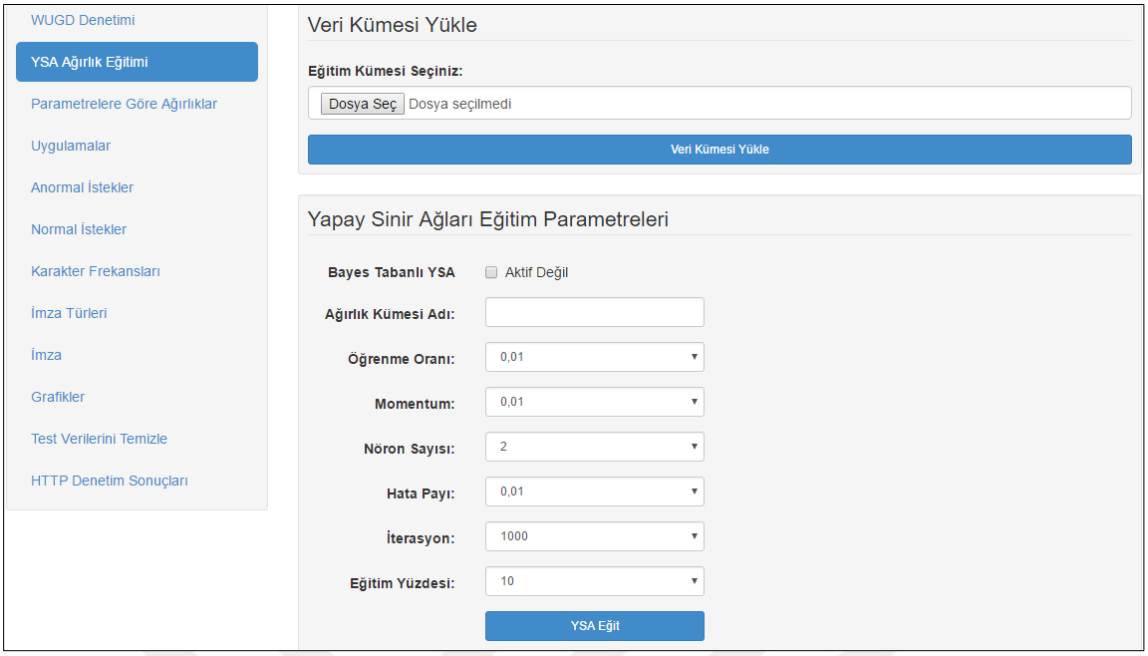
BSTİTD sürecinde denetimi gerçekleştirilecek her bir imza türü için, imza tanımlamaları yapılmaktadır. Bunun için Şekil 5.10'da verilen ekran kullanılmaktadır. Şekil 5.9'da verilen imza türleri ekranında tanımlanan saldırı türlerinin imza tanımlamaları şekil 5.10'da verilen

ekranda tanımlanmaktadır. Tanımlanan her bir imza türü için ayrı ayrı imza tanımlamaları gerçekleştirilerek BSTİTD gerçekleştirilmektedir.



Şekil 5.10. Bilinen Saldırı türlerinin imza tanımlamaları

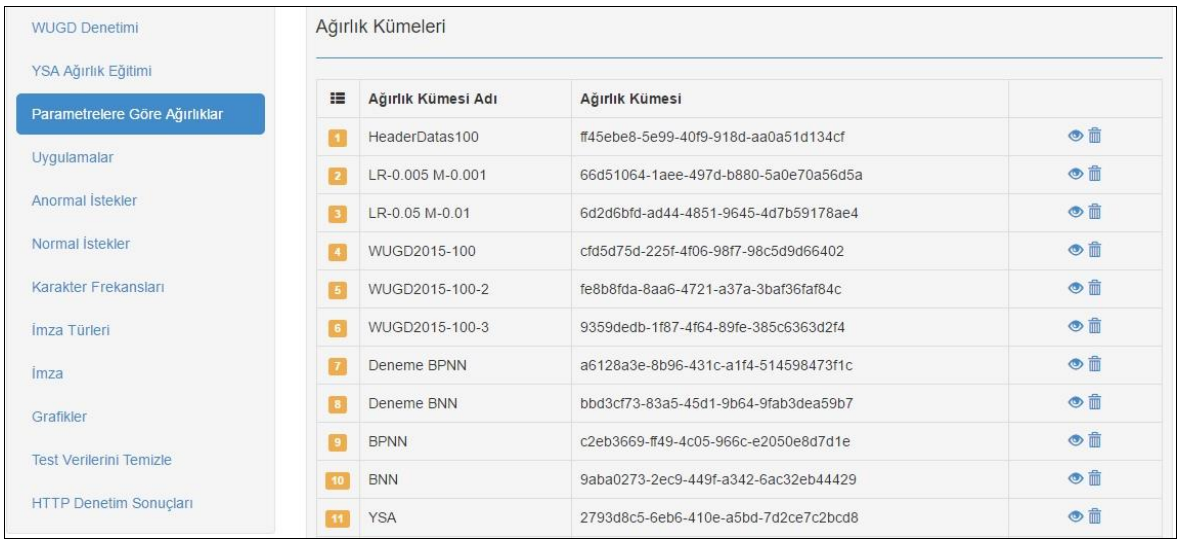
ATD süreci BS, YSA ve BTYSA yöntemleri ayrı ayrı kullanılarak yapay zeka temelli olarak gerçekleştirilmiştir. Şekil 5,11'de ATD denetimini yapmak için kullanılan YSA ve BTYSA modelini oluşturmak için kullanılan arayüz verilmiştir. YSA veya BTYSA modelini oluşturmak için ilk önce ağılıkların belirlenmesi gerekmektedir. 3 giriş, 10 nörona sahip 1 gizli katman ve 1 çıkışa sahip olarak belirlenen sinir ağı modelini toplam 51 ağırlık değeri bulunmaktadır. Sinir ağı modelinin ağırlıklarını belirlenmesi ve eğitilmesi için şekil 5,11'de de verildiği gibi öğrenme oranı, momentum, nöron sayısı, hata payı, iterasyon ve eğitim test oranı değerlerine göre belirlenen ağırlıklar ile ağ eğitimi gerçekleştirilmektedir. BS yöntemi kullanılarak ATD yapmak için farklı bir yazılım geliştirilmiştir.



Şekil 5.11. Sinir ağı eğitim performans parametreleri

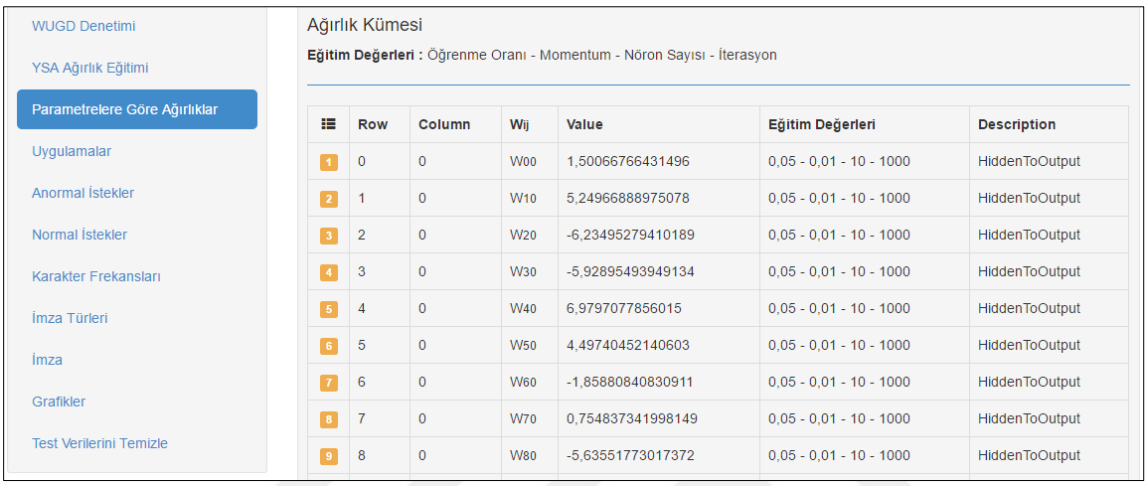
Şekil 5.11'de verilen ekranda hem YSA eğitimi hem de BTYSA eğitimi yapılabilmektedir. Parametre seçimi işlemin BTYSA seçimi yapıldığı zaman seçilen ağırlıklarla BTYSA ağırlık eğitimi yapılmaktadır. Aksi takdirde YSA ağırlık eğitimi yapılmaktadır.

Geliştirilen WUGD ile farklı veri kümeleri ve web uygulamalarının HTTP denetimini gerçekleştirmek için sinir ağı eğitimleri her bir uygulama ve veri kümesi için ayrı ayrı yapılabilmektedir. Bu durum farklı mimariye sahip web uygulamalarının ve veri kümelerinin denetiminin gerçekleştirilmesini sağlamaktadır. Şekil 5.12'de eğitim parametrelerine göre eğitilen ağırlık kümeleri verilmiştir. Ağırlık değerlerinin detayı tıklayarak, şekil 5.13'da verilen ağır değerleri görülebilir.



Şekil 5.12. Eğitilen YSA ve BTYSA modelleri

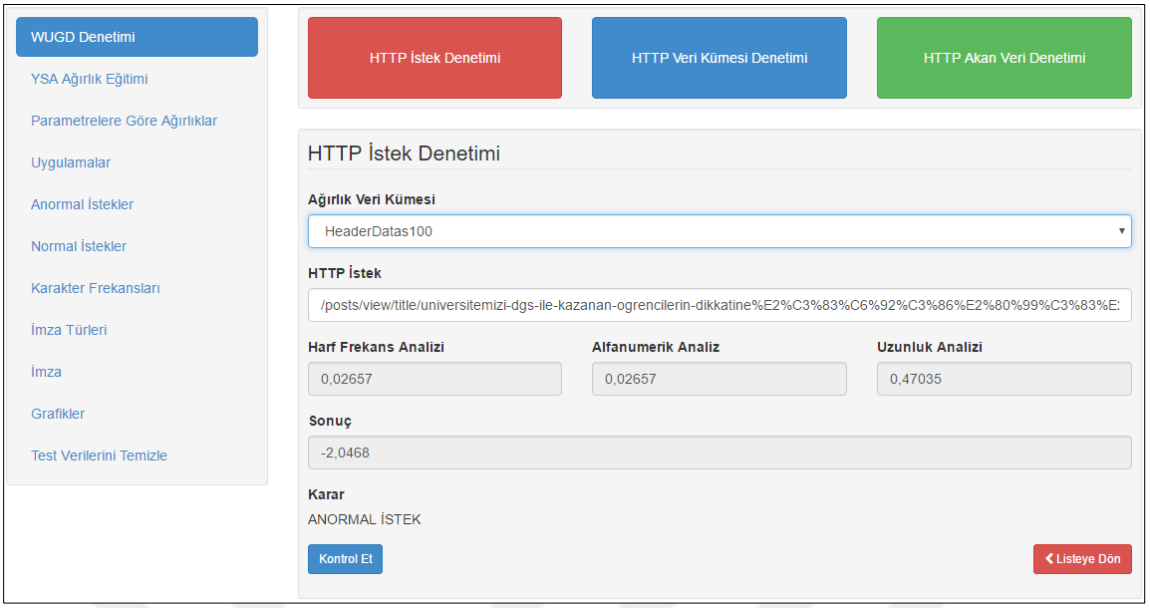
Şekil 5.13'da ağırlık eğitimi sonucu elde edilen ağırlık değerlerinin görüldüğü ekran verilmiştir. Ağırlıklar şekil 5.4'deki modelde gösterildiği gibi, giriş katmanından gizli katmana, gizli katmandan çıkış katmanına, sapmadan gizli katmana ve sapmadan çıkış katmanına olarak gruplandırılarak gösterilmiştir.



	Row	Column	Wij	Value	Eğitim Değerleri	Description
1	0	0	W00	1,50066766431496	0,05 - 0,01 - 10 - 1000	HiddenToOutput
2	1	0	W10	5,24966888975078	0,05 - 0,01 - 10 - 1000	HiddenToOutput
3	2	0	W20	-6,23495279410189	0,05 - 0,01 - 10 - 1000	HiddenToOutput
4	3	0	W30	-5,92895493949134	0,05 - 0,01 - 10 - 1000	HiddenToOutput
5	4	0	W40	6,9797077856015	0,05 - 0,01 - 10 - 1000	HiddenToOutput
6	5	0	W50	4,49740452140603	0,05 - 0,01 - 10 - 1000	HiddenToOutput
7	6	0	W60	-1,85880840830911	0,05 - 0,01 - 10 - 1000	HiddenToOutput
8	7	0	W70	0,754837341998149	0,05 - 0,01 - 10 - 1000	HiddenToOutput
9	8	0	W80	-5,63551773017372	0,05 - 0,01 - 10 - 1000	HiddenToOutput

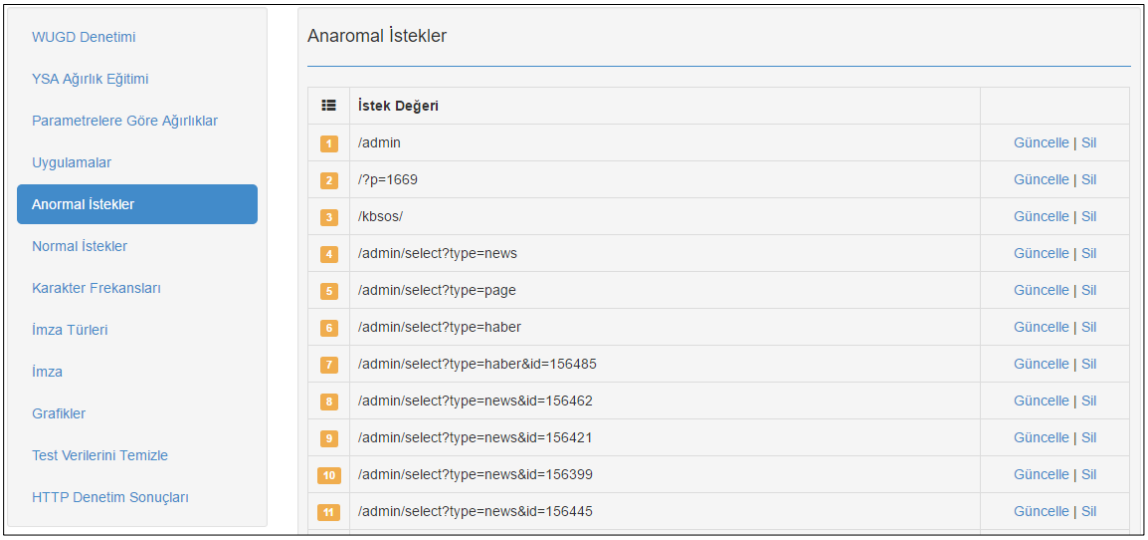
Şekil 5.13. YSA veya BTYSA modelleri için eğitilen ağırlık değerleri

Sinir ağı modelinin oluşturulması ve ağırlık eğitimlerinin yapılmasının ardından anormal HTTP isteklerinin denetimini gerçekleştirebilmek için Şekil 5.14'de verilen WUGD yazılımının denetim arayüzü kullanılmaktadır. HTTP isteklerinin denetimi için üç ayrı denetim mekanizması oluşturulmuştur. Bunlar; tek bir HTTP isteğinin denetimi, HTTP veri kümesi denetimi ve gerçek zamanlı olarak akan HTTP verisinin denetiminin gerçekleştirilmesidir. Her bir denetim mekanizmasında önce sinir ağı ağırlık veri kümesi seçilerek, denetimi yapılacak HTTP isteği, HTTP veri kümesi veya HTTP akan veri denetimi yapılabilmektedir. HTTP istek denetimi tek bir HTTP isteğinin denetimini yaparak için kullanılmaktadır. HTTP veri kümesi denetimi ise bir HTTP veri kümesinin denetimini yapmak için geliştirilmiştir. Akan HTTP denetimi ise web uygulamalarına gelen HTTP isteklerinin gerçek zamanlı olarak denetimini yapmak için geliştirilmiştir.



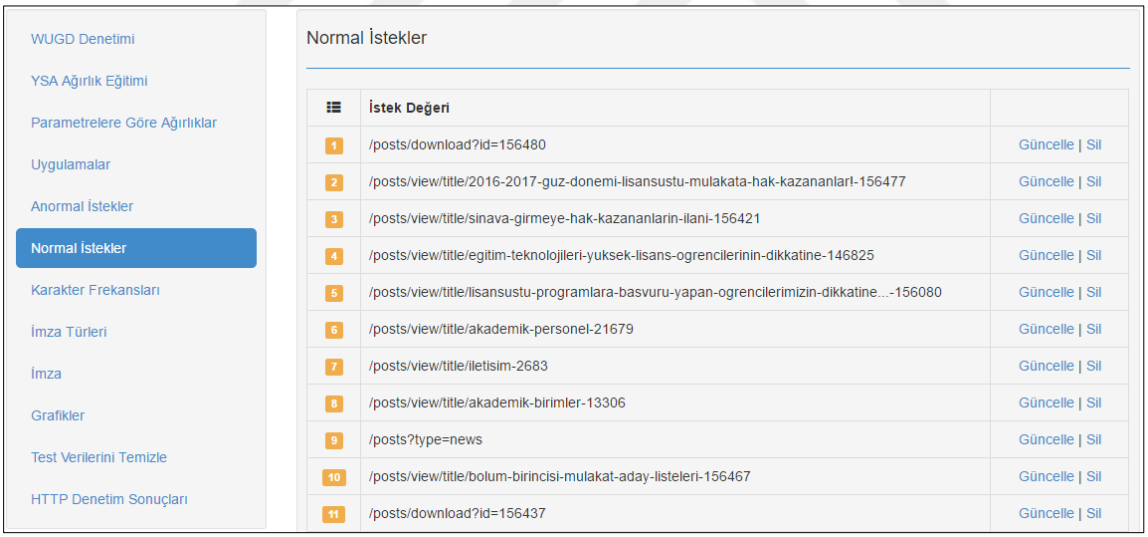
Şekil 5.14. WUGD HTTP istek denetimi

Şekil 5,2'de verilen algoritmaya göre gerçekleştirilen WUGD yazılımı, web uygulamasına gelen HTTP isteklerinin denetimini gerçekleştirmektedir. Denetim sürecinde ilk önce NİİTD daha sonra AİİTD ve daha sonra BSTİTD denetimleri yapılarak İTD sonucunda HTTP isteğinin saldırı içerip içermediği tespit edilemediğinde, HTTP isteği yapay zekâ tabanlı ATD sürecine yönlendirilmektedir. ATD sürecinde HTTP isteğinin anormalik denetimi yapılarak eğer istek anormal ise AİİTD veritabanı güncellenerek istek engellenmektedir. Eğer istek normal ise NİİTD veritabanı güncellenerek web uygulaması çalıştırılmaktadır. Şekil 5.15'de HTTP yapay zekâ denetimi sonucunda anormal olarak tespit edilmiş HTTP isteklerinin işlemlerinin yapıldığı ekran verilmiştir. Sinir ağı eğitimi sonucunda eğitim başarı oranına göre, normal HTTP istekleri anormal olarak, anormal HTTP istekleri ise normal olarak tespit edilebilmektedir. Şekil 5.15'de verilen ekran kullanılarak hatalı olarak tespit edilen anormal HTTP istekleri güncellenebilmektedir. Bu durum aynı HTTP isteği web uygulamasına tekrar geldiği zaman, AİİTD doğru bir şekilde imza denetiminin gerçekleştirilmesini sağlamaktadır.



Şekil 5.15. ATD sonucu tespit edilen anormal HTTP istekleri

Şekil 5.16'de verilen ekran kullanılarak hatalı olarak tespit normal edilen HTTP istekleri güncellenebilmektedir. Hatalı HTTP isteği web uygulamasına tekrar geldiği zaman, NİİTD doğru bir şekilde imza gerçekleştirilmesini sağlamaktadır.

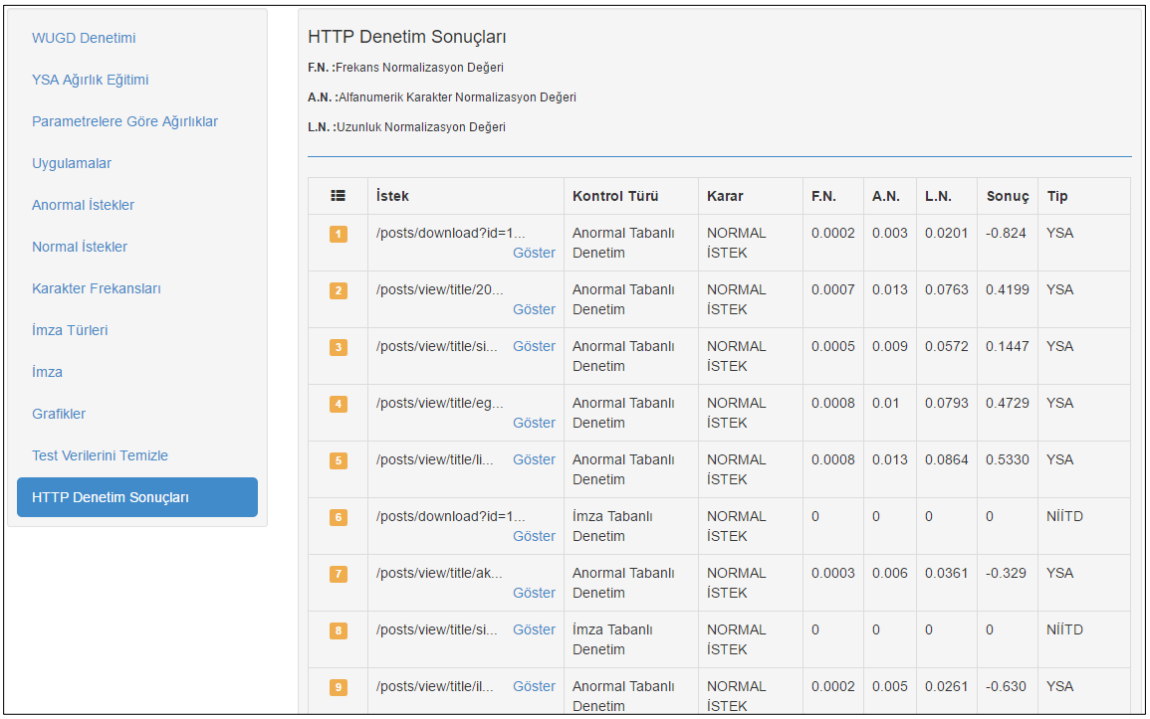


Şekil 5.16. ATD sonucu tespit edilen normal HTTP istekleri

Denetimi gerçekleştirilen toplam HTTP isteklerinin, anormal olarak tespit edilen toplam HTTP isteklerinin, normal olarak tespit edilen toplam HTTP isteklerinin, NİİTD, AİİTD ve ATD sonucunda tespit edilen toplam HTTP isteklerinin istatistiksel sonuçlarının verildiği ekran Şekil 5.17'de verilmiştir.

Şekil 5.17. Denetim sonucu istatistikleri

HTTP denetim sonuçlarını gösteren ekran Şekil 5.18'de verilmiştir. HTTP istekleri geliştirilen algoritmaya göre normal veya anormal olarak sınıflandırmaktadır. Denetim türüne göre ATD veya İTD, karar olarak normal veya anormal olarak sınıflandırılmaktadır. ATD denetiminde, HTTP isteğinin finans analizi, alfanumerik karakter analizi ve istek uzunluk analizi değerleri denetim tipi değerleri verilmektedir.

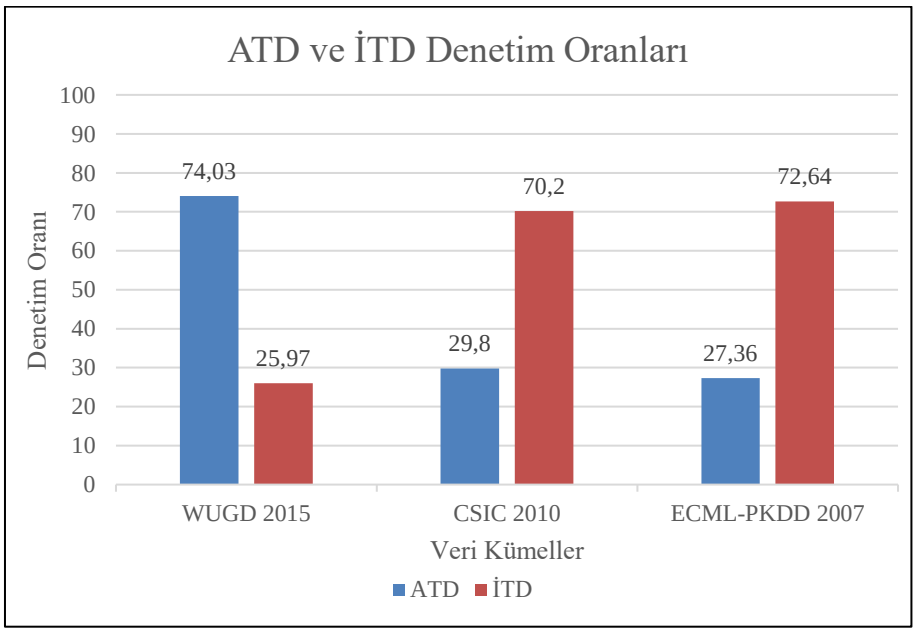


Şekil 5.18. WUGD HTTP istek denetim sonuçları

6. TEST VE DEĞERLENDİRME

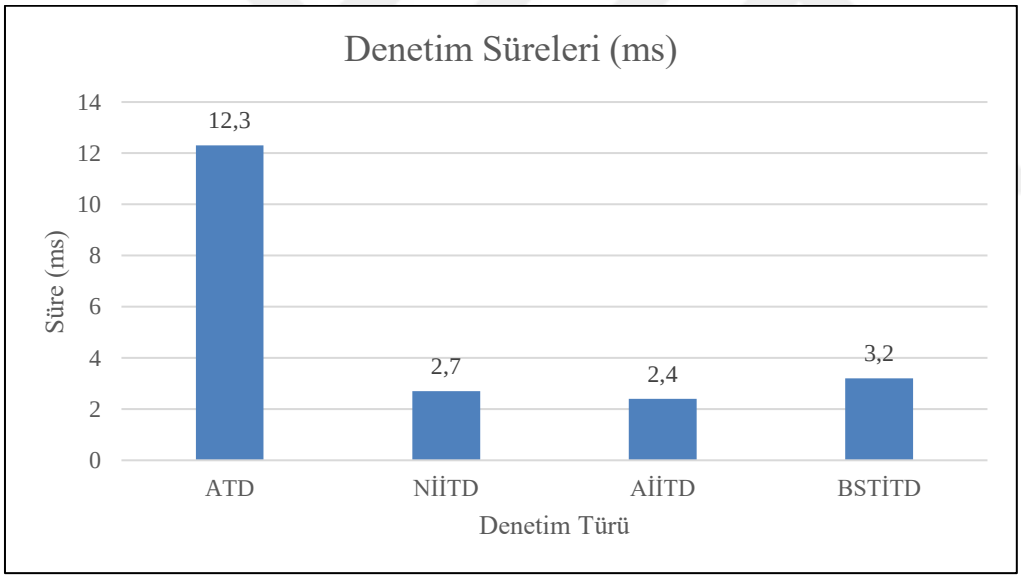
Web uygulamalarının mimarı yapıları birbirlerinden farklı olduğu için, her web uygulaması için ayrı ATD modelinin oluşturulması gereklidir. Geliştirilen WUGD yazılım ile farklı HTTP verikümeleri veya web uygulamaları için farklı öğrenme modelleri geliştirilebilmektedir.

Şekil 6,1'de veri kümelerinin ATD ve İTD denetim oranları verilmiştir. Denetimi gerçekleştirilen veri kümelerinde bulunan HTTP isteklerinin denetim oranları kullanılan denetim yöntemlerine göre değerlendirilmiştir. Buna göre WUGD 2015 veri kümesinin %25,97'si İTD ile ve %74,03'ü ATD ile tespit edilmiştir. CISC 2010 veri kümesinin %29,8'i ATD ile ve %70,2'si İTD ile tespit edilmiştir. ECML-PKDD 2007 veri kümesinin %27,36'sı ATD ile ve %72,64'ü İTD tespit edilmiştir. Web uygulamalarının mimari özelliklerinin birbirinden farklı olması, kullanıcılar tarafından talep edilen HTTP istek mimarisinin farklı olmasına ve veri kümelerinde tespit edilen normal ve anormal HTTP isteklerinin farklı olmasına sebep olmaktadır. Bu durum web uygulamalarına yapılan saldırıların denetimini de zorlaştırdığı için, web uygulamalarının ne ölçüde güvenlik zaafiyetine sahip olduğunu göstermektedir. Ayrıca güvenlik duvarı sistemlerinin saldırı tespit ve engelleme süreçlerinde sadece İTD kullanmaları, uygulamaların güvenlik denetimlerinin düzgün bir şekilde yapılamamasına sebep olacağı için, uygulamalarda güvenlik zaafiyeti ortaya çıkaracaktır.



Şekil 6.1. Kullanılan veri kümelerinin ATD ve İTD olarak denetim oranları

Şekil 6.2‘de, geliştirilen WUGD yazılımının denetim aşamalarına göre ortalama denetim süreleri verilmiştir. NİİTD, AİİTD ve BSTİTD olmak üzere üç aşamalı İTD süresi ve yapay zeka temelli öğrenme yöntemi ile gerçekleştirilen ATD süresi arasındaki farklar ortaya konmuştur. İTD’nin aşamaları olan NİİTD denetim süresi ortalama 2,7 ms, BSTİTD denetim süresi ortalama 3,2 ms, AİİTD denetim süresi ortalama 2,4 ms olarak tespit edilmiştir. Fakat ATD’nin denetim süresi ise ortalama 12,3 ms olarak tespit edilmiştir. ATD denetim süresi İTD’ye göre çok yüksek olduğu için sistemin denetim hız performansı açısından ilk önce İTD yapılmaktadır, daha sonra ATD yapılmaktadır. ATD denetim süresi İTD’ye göre fazla olduğu için ATD sonucu tespit edilen normal HTTP istekleri tekrar web uygulamasına geldiği zaman NİİTD yapılarak doğrudan web uygulaması işletilmektedir. ATD sonucu tespit edilen anormal HTTP istekleri ise tekrar web uygulamasına geldiğinde AİİTD yapılarak, ATD aşamasına gelmeden engellenmektedir. Böylece hibrit bir model önerilmesiyle hız performansı artırılmıştır.



Şekil 6.2. Kullanılan algoritmaların denetim süreleri

Geliştirilen WUGD yazılımının, tez kapsamında geliştirilen WUGD 2015 veri kümesi, literatürde kullanılan CSIC 2010 ve ECML-PKDD 2007 veri kümelerinin, BS, YSA ve BTYSA yöntemleri kullanılarak test edilmesi sonucunda elde edilen değerler ve yanlış pozitif oranları çizelge 6,1’de verilmiştir. Test sonuçlarına göre ortalama olarak YSA yöntemi ile veri kümelerinde bulunan saldırı içeren ve anormal HTTP isteklerinin BS yöntemi ile %95,13’ü, YSA yöntemi ile %96,59’u ve BTYSA yöntemi ile %97,14’ü oranında başarılı denetim elde edilmiştir. Veri kümelerinde yapılan denetim test sonuçlarına göre, %97,14 ile BTYSA denetim türünden daha başarılı sonuç alındığı görülmüştür.

Çizelge 6.1. Farklı denetim türleri ve veri kümeleri ile karşılaştırma

Denetim Türü	Veri Kümesi	%	YP	Ortalama
YSA	WUGD 2015	% 98,52	% 0,4	% 96,59
	CSIC 2010	% 96,74	% 1,1	
	ECML-PKDD 2007	% 94,53	% 1,3	
BTYSA	WUGD 2015	% 98,84	% 0,3	% 96,95
	CSIC 2010	% 96,83	% 1,4	
	ECML-PKDD 2007	% 95,76	% 1,5	
BS	WUGD 2015	% 97,60	% 0,2	% 95,13
	CSIC 2010	% 94,50	% 0,3	
	ECML-PKDD 2007	% 93,30	% 0,6	

Geliştirilen WUGD yazılımı, veri kümesi denetiminin yanında gerçek zamanlı denetim gerçekleştirmek için tasarlanmıştır. Gerçek zamalı denetim için Gazi Üniversitesi web içerik yönetim sistemi (http://www.gazi.edu.tr) kullanılarak gerçek zamanlı olarak test edilmiştir. Gerçek zamanlı denetim için Gazi Üniversitesi içerik yönetim sisteminin seçilmesinin sebebi, günlük olarak ortalama 100.000 HTTP isteğine sahip büyük bir yapı olmasıdır. Gazi Üniversitesi web içerik yönetim sistemi, üniversitenin 650'den fazla olan tüm birimlerinin web içeriklerinin yönetildiği büyük bir web uygulamasıdır.

Geliştirilen WUGD'nın üzerinde koşacağı sunucunun özellikleri gerçek zamanlı denetimin hız performansı açısından çok önemli bir etki oluşturmaktadır. Kullanılan sunucunun merkezi işlem birim (MİB) ve bellek kapasite özellikleri, HTTP isteklerinin denetim sürelerini de doğrudan etkilemektedir. Web sunucuya gelen HTTP anlık istek sayısı ile HTTP istek denetim hızı aynı olmalıdır. Eğer web sunucuya gelen HTTP istek sayısı WUGD denetim hızından daha fazla ise sistem istekleri kuyruğa alarak HTTP istek kaybını ortadan kaldırmaktadır. Çizelge 6.2'de web sunucu özellikleri ve anlık olarak denetlenen HTTP isteklerini sayıları karşılaştırmalı olarak verilmiştir.

Gazi Üniversitesi web uygulamasına 60 sn'de ortalama 45 HTTP isteği gelmektedir. Bu gelen HTTP istekleri 1.7 GHz 8 çekirdekli Xeon işlemci ve 8 GB belleğe sahip sunucu bilgisayar kullanılarak yapılan test sonucuna göre 60 saniyede ortalama, 15 HTTP isteğinin denetimi ATD ile yapılmaktadır, İTD ile ise ortalama 30 HTTP isteği denetlenmektedir. Web uygulamasına gelen HTTP istek sayısı, isteklerin kuyruğa alınmasında önemli etkiye sahiptir. Web uygulamasına anlık olarak gelen istek sayısı arttıkça suncunun özelliklerine göre kuyruğa eklenen istek miktarı da değişecektir. İstek sayısı azaldıkça kuyruğa eklenen istek miktarı azalacaktır veya kuyruğa istek eklenmeden denetim gerçekleştirilecektir.

Çizelge 6.2. Gerçek zamanlı denetim sunucu özellikleri

MİB	**Bellek**	**Gelen HTTP istekleri**	**Süre**	**ATD (Ortalama)**	**İTD (Ortalama)**
1.7 GHz 8 Çekirdekli Xeon	8 GB	45	60 sn	15	30

Çizelge 6.3'te gerçek zamanlı olarak gerçekleştirilen toplam 34070 HTTP isteğini denetim sonuçları verilmiştir. 19 HTTP isteği BSİTD denetimi gerçekleştirilken 3206 http isteğinin ATD gerçekleştirilmiştir. ATD gerçekleştirilen istekler NİİTD ve AİİTD veritabanına eklenerek imza üretimi gerçekleştirilmiştir.

- NİİTD, tekrarlı olarak denetimi yapılan normal HTTP istekleridir.
- NİİTD Tekil, toplam HTTP isteklerinden tekil olarak tespit edilen normal HTTP istekleridir.
- AİİTD, tekrarlı olarak denetimi yapılan anormal HTTP istekleridir.
- AİİTD Tekil toplam HTTP isteklerinden tekil olarak tespit edilen anormal HTTP istekleridir.

Çizelge 6.3. Gerçek zamanlı denetim sonuçları

NİİTD	**NİİTD Tekil**	**AİİTD**	**AİİTD Tekil**	**BSTİTD**	**ATD**	**Toplam**
31770	3159	75	66	19	3206	35070

Çizelge 6,4'da geliştirilen uygulamanın benzer çalışmalarla yapılan karşılaştırmaları verilmiştir. Stephan ve diğerleri [10] yaptıkları çalışmada denetim yönetim olarak YSA kullanmışlardır, kendi ürettikleri verikümesi ile yaptıkları denetim sonucunda %95 oranında başarı elde etmişlerdir. Nguyen ve diğerleri [11] yaptıkları çalışmada CSIC 2010 ve ECML-PKDD 2007 veri kümelerinde C4.5, CART, RandomTree ve RandomForest sınıflandırıcılarını kullanarak denetim gerçekleştirmişlerdir. Denetim sonucunda en iyi sonucu %98,8 oranda RandomForest sınıflandırıcı ile ECML-PKDD 2007 verikümesinde elde etmişlerdir. Kirchner ve diğerleri [84] tarafından yapılan çalışmada denetim yöntemi olarak veri madenciliği yöntemlerinden K-means kullanılarak paylaşılmayan kendi üretikleri veri kümesi ile yaptıkları denetim sonucunda %90,9 oranında denetim başarısı sağlamışlardır. Zolotukhin ve diğerleri [85] yaptıkları çalışmada denetim yöntemi olarak ngram kullanılarak paylaşılmayan kendi üretikleri veri kümesi ile yaptıkları denetim sonucunda %97,7 oranında denetim başarısı sağlamışlardır. Nguyen ve diğerleri [86] tarafından yapılan çalışmada, Navie Bayes, Bayes Network, Decision Stump, RBF Network, Majority Voting ve Hefge/Boosting sınıflandırma araçlarını kullanarak, CSIC 2010 ve ECML-PKDD 2007 veri kümelerinde

denetim gerçekleştirilmiştir. Bu kapsamda geliştirdikleri ve A-IDS isimli adaptif saldırı tespit sistemi modellinde CSIC 2010 veri kümesindan % 90,52 ve ECML-PKDD 2007 veri kümesinde ise % 91,27 oranından başarılı sonuç aldıkları görülmüştür.

Tez kapsamında, önerilen model de ise, WUGD 2015, CSIC 2010 ve ECML-PKDD 2007 veri kümeleri kullanılarak YSA, BTYSA ve BS yöntemleri ile ayrı ayrı yapılan denetim sonuçlarına göre; YSA denetimi sonucunda WUGD 2015 verikümesinde %98,52, CSIC 2010 verikümesinde %96,74 ve ECML-PKDD 2007 verikümesinde % 94,53 oranında saldırı içeren HTTP isteklerinin denetiminde başarı elde edilmiştir. BTYSA denetimi sonucunda WUGD 2015 verikümesinde % 98,84, CSIC 2010 verikümesinde %96,83 ve ECML-PKDD 2007 verikümesinde % 95,76 oranında saldırı içeren HTTP isteklerinin denetiminde başarı elde edilmiştir. BS denetimi sonucunda WUGD 2015 verikümesinde % 97,60, CSIC 2010 verikümesinde %94,50 ve ECML-PKDD 2007 verikümesinde % 93,30 oranında saldırı içeren HTTP isteklerinin denetiminde başarı elde edilmiştir. Yapılan karşılaştırmalara göre tez kapsamında geliştirilen WUGD algoritmasının, diğer çalışmalardan, ortalama olarak veya spesifik sınıflandırma araçları bakımdan daha iyi sonuç ürettiği götülmüştür.

Çizelge 6.4. Benzer çalışmalarla karşılaştırma

Çalışma	**İTD**	**ATD**	**Denetim**	**Veri Kümesi**	**%**
Stephan ve diğ. [10]	x	√	YSA	Veri Kümesi	% 95 ± 4
Nguyen ve diğ. [11]	x	√	C4.5	CSIC 2010	% 94,49 ± 5,9
				ECML-PKDD 2007	% 96,37 ± 3,7
			CART	CSIC 2010	% 94,12 ± 6,2
				ECML-PKDD 2007	% 96,11 ± 4,3
			RandomTree	CSIC 2010	92,3 ± 8,3
				ECML-PKDD 2007	96,89 ± 2,6
			RandomForest	CSIC 2010	93,71 ± 7,2
				ECML-PKDD 2007	% 98,8 ± 1,2
Kirchner ve diğ. [84]	x	√	K-Means	Veri Kümesi	% 90,9 ± 3,2
Zolotukhin ve diğ. [85]	x	√	Ngram	Veri Kümesi	% 97,7 ± 0,2
Nguyen ve diğ. [86]	x	√	Navie Bayes	CSIC 2010	% 72.78 ± 0.01
				ECML-PKDD 2007	% 85,12 ± 0,03
			Bayes Network	CSIC 2010	% 82,79 ± 0,03
				ECML-PKDD 2007	% 86,95 ± 0,025
			Decision Stump	CSIC 2010	% 74,73 ± 0,05
				ECML-PKDD 2007	% 84,27 ± 0,07
			RBF Network	CSIC 2010	% 72,46 ± 0,01
				ECML-PKDD 2007	% 87,69 ± 0,04
			Majority Voting	CSIC 2010	% 81
				ECML-PKDD 2007	% 83
			Hefge/Boosting	CSIC 2010	% 82,1 ± 0,04
				ECML-PKDD 2007	% 86,3 ± 0,05
			A-IDS	CSIC 2010	% 90,52 ± 0,06
				ECML-PKDD 2007	% 91,27 ± 0,01
			A-ExIDS	CSIC 2010	% 90,98 ± 0,05
				ECML-PKDD 2007	% 92,56 ± 0,02
Önerilen Model	√	√	BS	WUGD 2015	% 97,60 ± 0,2
				CSIC 2010	% 94,50 ± 0,3
				ECML-PKDD 2007	% 93,30 ± 0,6
	√	√	YSA	WUGD 2015	% 98,52 ± 0,4
				CSIC 2010	% 96,74 ± 1,1
				ECML-PKDD 2007	% 94,53 ± 1,3
	√	√	BTYSA	WUGD 2015	% 98,84 ± 0,3
				CSIC 2010	% 96,83 ± 1,4
				ECML-PKDD 2007	% 95,76 ± 1,5

7. SONUÇ VE ÖNERİLER

Bu tez çalışmasında, web tabanlı saldırıları önlemek için ATD ve İTD'nin gerçekleştirildiği öğrenme tabanlı hibrit bir WUGD modeli önerilmiştir. Önerilen model doğrultusunda WUGD algoritması tasarlanarak, WUGD uygulaması geliştirilmiştir. Geliştirilen WUGD uygulaması literatürde kaynak gösterilen CSIC 2010 ve ECML-PKDD 2007 veri kümeleri, tez kapsamında üretilen WUGD 2015 veri kümesi ile test edilmiştir. Gerçekleştirilen uygulama ayrıca Gazi Üniversitesi içerik yönetim sistemi kullanılarak gerçek zamanlı olarak da test edilmiştir.

Hibrit çalışmada, İTD ile bilinen web tabanlı saldırı türlerinden olan SQL (Structured Query Language) Enjeksiyonu, Siteler Arası Kod (XSS) Yazma, Komut Enjeksiyonu (KE) ve Dizin Geçişi Saldırı (DGS) saldırı türlerine karşı imza denetimi gerçekleştirilmiştir. ATD ile, standart HTTP istek yapısına uymayan HTTP isteklerinin denetimi gerçekleştirilmiştir. ATD yönteminde AKA, HFA ve İUA olmak üzere üç öznitelik kullanılarak HTTP isteklerinin sayısallaştırılması sağlanmıştır. Özniteliklerden elde edilen sayısal sonuçlar BS, YSA ve BTYSA denetim türleri için giriş parametreleri olarak kullanılarak anormal HTTP isteklerinin denetimi öğrenme tabanlı olarak üç ayrı yöntem ile ayrı ayrı gerçekleştirilmiştir.

Geliştirilen uygulamada BS, YSA ve BTYSA ile elde edilen sonuçlar karşılaştırılmıştır. Karşılaştırma sonucunda BTYSA denetiminin, BS ve YSA yöntemlerinden daha iyi sonuç verdiği görülmüştür. Ayrıca YSA yönteminin ise BS yöntemine göre daha iyi sonuç verdiği görülmüştür.

İTD hızlı çalışırken sıfır gün saldırılarına karşı etkili değildir, ATD yöntemi ise sıfır gün saldırılarında etkilidir. ATD sonucunda tespit edilen normal ve anormal HTTP istekleri tekrar web uygulamasına geldikleri zaman İTD'ye göre daha yavaş çalışan ATD gerçekleştirilmeden, tespit edilen HTTP isteklerinin denetimlerinin gerçekleşrilmesi için, normal HTTP istekleri NİİTD veritabanına, anormal HTTP istekleri ise AİİTD veritabanına eklenmektedir. Bu sayede imza olarak belirlenen anormal HTTP isteği tekrar web uygulamasına geldiği zaman, ATD gerçekleştirilmeden AİİTD ile engellenmektedir. İmza olarak belirlenen normal HTTP isteği tekrar web uygulamasına geldiği zaman, ATD gerçekleştirilmeden NİİTD ile web uygulaması çalıştırılmaktadır. Dolayısıyla ATD gerçekleştirilerek, web uygulamasının istek yapısına göre imza veritabanı güncellenerek İTD'nin dezavantajı olan yeni saldırı türlerine karşı etkili olmama özelliği ortadan

kaldırılmaya çalışılmıştır. Aynı zamanda İTD hız bakımından ATD'den daha hızlı denetim gerçekleştirdiği için modelin ortalama hız performası önerilen hibrit model sayesinde yükselmektedir.

Web uygulamalarının güvenlik zafiyetlerini ortadan kaldırmak için, uygulama düzeyinde ve ağ katmanları düzeyinde güvenlik çalışmaları gerçekleştirilmektedir. Uygulama düzeyinde yapılan çalışmalar uygulamanın geliştirilmesi aşamasında yapılmaktadır. Ağ ortamında yapılan çalışmalar ise özellikle uygulama katmanı düzeyinde HTTP İstek denetimi yapılarak gerçekleştirilen güvenlik uygulamalarıdır. Web uygulamalarının geliştirilmesinde yapılan teknolojik yenilikler, web uygulamalarının sahip olduğu zafiyetlerin bazılarını ortadan kaldırmasına rağmen, aynı zamanda yeni saldırı türlerinin ve web zafiyetlerinin de ortaya çıkmasına sebep olmaktadır. Web uygulamalarının güvenliğini sağlamak için yapılan çalışmalar da, geliştirilen yeni web teknolojilerinden kaynaklanan zafiyetlere yönelik olarak geliştirilmelidir.

KAYNAKLAR

1. Lara, J. ve Gracia, G. (Serrao, Carlos, Aguilera, Vicente, Cerullo, Fabio). (2010). *Building Web Application Firewalls in High Availability Environments. Web Application Security*, Almanya: Springer Berlin Heidelberg, 75-82.

2. Vural, Y. Sağıroğlu, Ş. (2008). Kurumsal Bilgi Güvenliği ve Standartları Bir İnceleme. *Gazi Üniversitesi, Mühendislik-Mimarlık Fakültesi Dergisi*, 23(2), 507-522.

3. Torrano-Gimenez, C. Nguyen, H. Alvarez and G. Franke, K. (2015). Combining expert knowledge with automatic feature extraction for reliable web attack detection", *Security And Communication Networks*, 8(16), 2750–2767.

4. İnternet: Uygulama Güvenlik Duvarlarının (Web Application Firewalls) Analizi, URL: https://www.bilgiguvenligi.gov.tr/web-guvenligi/ugulama-guvenlik-duvarlarinin-web-application-firewalls-analizi.html, Son Erişim Tarihi: 03.10.2016.

5. Valeur, F., Mutz, D. and Vigna G. (2005), *A Learning-Based Approach to the Detection of SQL Attacks*, Proceedings of the Second international conference on Detection of Intrusions and Malware, and Vulnerability Assessment ,Springer-Verlag Berlin, Heidelberg, 123-140.

6. Robertson W. K. (2009). *Detecting And Preventing Attacks Against Web Applications*, Doktora Tezi, University of California, Santa Barbara, CA, USA, 10 -12.

7. İnternet: The MITRE Corporation. Common Exposures and Vulnerabilities Database, URL: http://cve.mitre.org/, Son Erişim Tarihi: 03.10.2016.

8. Namit, G. Abakash, S. Dheeraj, S. (2008). Web Application Firewall, *CS499: B. Tech Project Final Report*, Kanpur.

9. Gaikwad S. K. Shah, V. and Jain, Y. K. (2010). A Secure Network Detection System against Noisy Unlabeled Data, *International Journal of Computer Applications*, 9(9), 7– 11.

10. Stephan, J. J. Mohammed, S. D. Abbas, M. K. (2015) Neural Network Approach to Web Application Protection, International Journal of Information and Education Technology, 5(2).

11. Nguyen, H.T. Torrano-Gimenez, Alvarez, C., Franke, K. Petrovic, S. (2011). Enhancing the effectiveness of Web Application Firewalls by generic feature selection, Logic Journal of IGPL, 21(4), 560-570.

12. Palka, D. Zachara. M. (2011). Learning Web Application Firewall - Benefits and Caveats, Availability, Reliability and Security for Business, Tjoa, A. M. Quirchmayr, G. You, I. Xu, L. (Eds.), Enterprise and Health Information Systems, Springer Berlin Heidelberg, s. 295–308.

13. Basile, C. Lioy, A. (2015). Analysis of Application-Layer Filtering Policies With Application to HTTP, *IEEE/ACM Transactions On Networking*, 23(1), 28-41.

14. Cho, S. Cha, S. (2004). SAD: Web Session Anomaly Detection Based on Parameter Estimation, *Computers & Security*, 23(4), 312-319.

15. Razzaq, A. Ahmed, H. F. Hur, A. Haider, N. (2009, February). Ontology based application level intrusion detection system by using Bayesian filter. Paper presented at the Computer, Control and Communication IC4 2009 2nd International Conference, Karachi, Pakistan.

16. Bremler-Barr, A. Koral, A. (2012) Accelerating Multipattern Matching on Compressed HTTP Traffic, *IEEE/ACM Transactions On Networking*, 20(3).

17. Singh, S. Agrawal, S. Rizvi, M. A. Thakur, R. S. (2011, October). Improved Support Vector Machine for Cyber Attack Detection, Paper presented at the Proceedings of the World Congress on Engineering and Computer Science, San Francisco, USA.

18. Torrano-Gimenez, C. Perez-Villegas and A. Alvarez, G. (2009). Computational Intelligence in Security for Information Systems, Herrero, A. Gastaldo, P. Zunino, R. Corchado E. (Eds). *A Self-learning Anomaly-Based Web Application Firewall*, Springer-Verlag Berlin Heidelberg, s. 85–92.

19. Peng, J. Feng, C. Rozenblit, J. W. (2006, 27-30 March). *A hybrid intrusion detection and visualization system*, Paper presented at the 13th Annual IEEE International Symposium and Workshop on Engineering of Computer Based Systems, Proceedings: Mastering The Complexity Of Computer-Based Systems, Tucson, USA, pp. 505-506.

20. Locasto, M. E. Wang, K. Keromytis, A. D. Stolfo, S. J. (2006). FLIPS Hybrid adaptive intrusion prevention, Recent Advances In Intrusion Detection, *Lecture Notes in Computer Science*, 3858, pp. 82-101.

21. Hwang, K. Cai, M. Chen, Y. Qin, M. (2007). Hybrid intrusion detection with weighted signature generation over anomalous Internet episodes, *IEEE Transactions On Dependable And Secure Computing*, 4(1), 41-55.

22. Hendry, G., Yang, S. (2008, 16-20 March). Intrusion signature creation via clustering anomalies, *Proceedings of SPIE Security and Defense Symposium, Data Mining, Intrusion Detection, Information Assurance, and Data Networks Security Conference*, Orlando, Florida.

23. Gillman, D., Lin, Y., Maggs, B., Sitaraman, R. K. (2015). Protecting Websites from Attack with Secure Delivery Networks, *Computer*, 48(4), 26-34.

24. İnternet: Web Uygulama Güvenlik Duvarı nedir, URL: http://www.tnetworks.com.tr/cozumler/web-application-firewall-waf, Son Erişim Tarihi: 02.07.2016.

25. İnternet: Bilgi Güvenliği, URL: http://www.innova.com.tr/bilgi-guvenligi-iso27001.asp, Son Erişim Tarihi:12.09.2016.

26. Vural, Y. (2007). *Kurumsal Bilgi Güvenliği ve Sızma (Penetrasyon) Testleri*, Yüksek Lisans Tezi, Gazi Üniversitesi Fen Bilimleri Enstitüsü, Ankara, 169.

27. Jia, X. (2006). *Design, Implementation and Evaluation of an Automated Testing Tool for Cross-Site Scripting Vulnerabilities*, Yüksek Lisans Tezi, Darmstadt University of Technology (TUD) Computer Science Department, 2-6.

28. İnternet: Uygulama Güvenlik Duvarlarının (Web Application Firewalls) Analizi, URL:https://www.bilgiguvenligi.gov.tr/web-guvenligi/ugulama-guvenlik-duvarlarinin-web-application-firewalls-analizi.html, Son Erişim Tarihi : 13.06.2016.

29. Foster, J. C., Bhalla, N., Heinen, N., Osipov, V., and ebrary, I. (2005). *Buffer overflow attacks: Detect, exploit, prevent*. Rockland, MA: Syngress Publishing.

30. İnternet: Web Proxy Konfigürasyonu, URL: http://www.ceyhuncamli.com/isa-server-web-proxy-konfigrasyonu, Son Erişim Tarihi: 02.09.2016.

31. İnternet: Application Firewall, URL: http://en.wikipedia.org/wiki/Application_firewall, Son Erişim Tarihi: 10.06.2016.

32. Takahashi, H., Ahmad, H. F. and Mori, K. (2011). Application for autonomous decentralized multi layer cache system. *In Proceedings - 2011 10th International Symposium on Autonomous Decentralized Systems*, ISADS 2011, pp. 113-120.

33. İnternet: IETF, 1999, RFC2616 - Hypertext Transfer Protocol - HTTP/1.1, http://www.ietf.org/rfc/rfc2616.txt, Son Erişim Tarihi: 1810.2006.

34. Auxilia, M., Tamilselvan, D. (2010, October). Anomaly detection using negative security model in web application. *In Computer Information Systems and Industrial Management Applications (CISIM)*, 481-486.

35. Heaton, j. (2007). *HTTP Programming Recipes for Java Bots* (First Edition). USA: Heaton Research, Inc, 42-47.

36. Roesch, M. (1999). Snort: Lightweight Intrusion Detection for Networks, *Proceedings of LISA '99: 13th Systems Administration Conference*, 99(1), 229-238.

37. Pao, D., Or, N.L. and Cheung, R.C.C. (2013). A memory-based NFA regular expression match engine for signature-based intrusion detection. *Computer and Communications*, 36 (10-11), 1255-1267.

38. Thankachan, A., Ramakrishnan, R., and Kalaiarasi, M. (2014). A survey and vital analysis of various state of the art solutions for web application security. *In Information Communication and Embedded Systems (ICICES)*, 1-9.

39. Tekerek, A., Gemci, C. and Bay, O. F. (2014). Development Of A Hybrid Web Application Firewall To Prevent Web Based Attacks, *In Application of Information and*

Communication Technologies (AICT), 1-4.

40. Vigna, G., Valeur, F., Kemmerer, R., A. (2003). Designing and implementing a family of intrusion detection systems, *Proceedings of the 9th European software engineering conference held jointly with 11th ACM SIGSOFT international symposium on Foundations of software engineering*, Helsinki, Finland.

41. Takçı, H. Akyüz, T. Soğukpınar, İ. (2007). Web Atakları için Metin Tabanlı Anormallik Tespiti (WAMTAT), *Gazi Üniversitesi Mühendislik-Mimarlık Fakültesi Dergisi*, 22(2) 247-253.

42. Asmaa, A. Sharad, G. (2011). Importance of Intrusion Detection System (IDS), *International Journal of Scientific & Engineering Research*, 2(1), pp.

43. Gyanchandani, M., Rana, J. L. and Yadav, R. N. (2012). Taxonomy of Anomaly Based Intrusion Detection System: A Review, *International Journal of Scientific and Research Publications (IJSRP)*, 2(12) 1-13.

44. Qayyum, A., Islam, M. H., & Jamil, M. (2005). Taxonomy of statistical based anomaly detection techniques for intrusion detection, *In Proceedings of the IEEE Symposium on Emerging Technologies*, 270-276, IEEE.

45. Narayana, M. S., Prasad, B. V. V. S., Srividhya, A., and Reddy, K. P. R. (2011). Data Mining Machine Learning Techniques–A Study on Abnormal Anomaly Detection System, *International Journal of Computer Science and Telecommunications*, 2(6).

46. Dickerson, J. E., & Dickerson, J. A. (2000). Fuzzy network profiling for intrusion detection, *In Fuzzy Information Processing Society, NAFIPS. 19th IEEE International Conference of the North American*, 301-306.

47. Patcha, A., Park, J. (2007). An overview of anomaly detection techniques: Existing solutions and latest technological trends, *Computer and Networks*, 51(12), 3448-3470.

48. Garcia-Teodoro, P., Díaz-Verdejo, J., Maciá-Fernández, G. and Vázquez, E. (2009). Anomaly-based network intrusion detection: Techniques, systems and challenges, Computer and Networks, 28(1), 18-28.

49. İnternet: Linear discriminant analysis,
URL: https://en.wikipedia.org/wiki/Linear_discriminant_analysis, Son Erişim Tarihi: 15.09.2016.

50. Demirel, S., Bodur, S. (2004). Genetik Danışmada Bayes Teoreminin Uygulamaları, *Erciyes Tıp Dergisi*, 26(2), 81-85.

51. Doya, K., Ishii, S., Pouget, A. and Rao, R. P. N. (2011). *Bayesian Brain: Probabilistic Approaches to Neural Coding*, Boston, USA, The MIT Press.

52. Mukherjee, S., Sharma, N. (2012). Intrusion detection using naive Bayes classifier with feature reduction, *Procedia Technology*, 4, 119-128.

53. Herve, D. Marc, D. Andreas, W. (1999). Towards a Taxonomy of Intrusion Detection Systems, *Elsevier, Computer Networks*, 31, 805-822.

54. Haykin, S. (1999). *Neural networks: a comprehensive fundation*, Prentice Hall, New Jersey.

55. McCulloch, W., Pitts, W. (1943). *A logical calculus of the ideas immanent in nervous activity*, Bulletin of Mathematical Biophysics, 7, 115–133.

56. Hebb, D. O. (1949). *The Organization of Behavior*, Wiley, New York.

57. Rosenblatt, F. (1959). *Principles of Neuradynamics*, New York: Spartan Books.

58. Rosenblatt, F. (1962). *Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms*, Washington D.C.: Spartan Books.

59. Farley, B., Clark, W. (1954). *Simulation of self-organizing systems by digital computer*. Transactions of the IRE Professional Group on Information Theory, 4(4), 76-84.

60. Hopfield, J. J. (1982). Neural networks and physical sistems with emergent collective computational abilities. *PNAS, Proceedings of the National Academy of Sciences*, 79(8), 2554-2558.

61. Kohonen, T. (1989). *Self-Organization and Associative Memory*. Berlin: Springer–Verlag.

62. Rumelhart, D. E., McClelland, J. L. and the PDP Research Group. (1986). *Parallel Distributed Processing: Explorations in the Microstructure*, Cambridge, Massachusetts, London, The MIT Press.

63. Farley, B., Clark W.A. (1954). Simulation of self-organizing systems by digital computer, *IRE Transactions on Information Theory*, 4, 76-84.

64. Taşova O. (2011). Yapay Sinir Ağları İle Yüz Tanıma, Yüksek Lisans Tezi, Dokuz Eylül Üniversitesi Fen Bilimleri Enstitüsü, İzmir, 11-33.

65. Elmas, Ç. (2003). *Yapay Sinir Ağları*, Seçkin Yayıncılık, Ankara.

66. Ataseven, B. (2013). Yapay Sinir Ağları ile Öngörü Modellemesi, *Marmara Üniversitesi Sosyal Bilimler Enstitüsü Öneri Dergisi*, 10(39), 101-115.

67. Nabney, I. (2002). *NETLAB: algorithms for pattern recognition*, Springer Science & Business Media, 325-365.

68. MacKay, D., J. (2003). *Information theory, inference and learning algorithms*, Cambridge university press.

69. David, J. C., (1995). MacKay: Probable Networks and Plausible Predictions: A Review of Practical Bayesian Methods for Supervised Neural Networks, *Network: Computation*

in Neural Systems, 6, 469-505.

70. Bishop C.M. (2006). *Pattern Recognition and Machine Learning*, Jordan, M. Kleinberg, J. and Schölkopf, B., Springer, New York-USA.

71. Ingham, K. (2007). Anomaly Detection for HTTP Intrusion Detection: Algorithm Comparisons and the Effect of Generalization on Accuracy, Doktora Tezi, University of New Mexico, USA.

72. İnternet: HTTP Dataset CSIC 2010, URL : http://www.isi.csic.es/dataset, Son Erişim Tarihi: 04.10.2016.

73. İnternet: Attack Challenge - ECML/PKDD Workshop, URL : http://www.lirmm.fr/pkdd2007-challenge/, Son Erişim Tarihi :04.10.2016.

74. Farid, D. M., Harbi, N., Bahri, E., Rahman, M. Z., and Rahman, C.M., (2010). Attacks Classification in Adaptive Intrusion Detection using Decision Tree, *World Academy of Science, Engineering and Technology*, 86-91.

75. Farley, B. and Clark W. A., (1954). Simulation of self-organizing systems by digital computer, *IRE Transactions on Information Theory*, 4(4), 76-84.

76. İnternet: SQL Injection, URL: https://tr.wikipedia.org/wiki/SQL_Injection, Son Erişim Tarihi: 05.10.2016.

77. İnternet: DOM Based Cross Site Scripting or XSS of the Third Kind, URL : http://www.webappsec.org/projects/articles/071105.shtml, Son Erişim Tarihi : 03.10.2016.

78. Pauli, J. (2013). *Chapter 5 - Web Application Exploitation with Broken Authentication and Path Traversal*, In The Basics of Web Hacking, Syngress, Boston, 87-103.

79. Jourdan, G. V. (2007). Securing large applications against command injections, *IEEE Aerospace and Electronic Systems Magazine*, 24(6), 15-24.

80. Chandrashekar, G. Sahin, F. (2014). A survey on feature selection methods, *Computers and Electrical Engineering* 40, 16–28.

81. Kohavi, R., John, G. H., (1997). Wrappers for feature subset selection, *Artificial Intelligence*, 97, 273–324.

82. Kruegel, C., Vigna, G. (2003). Anomaly Detection of Web-Based Attacks, *Proceedings of the 10th ACM Conference on Computer and Communication Security*, 251-261.

83. Han, J., Kamber, M. (2011). *Data Mining: Concepts and Techniques (Second Edition)*. Elsevier, 310-315.

84. Kirchner, M. (2010), A framework for detecting anomalies in http traffic using instance-based learning and k-nearest neighbor classification, *In Security and Communication*

Networks (IWSCN), 2010 2nd International Workshop on, 1-8.

85. Zolotukhin, M., Hämäläinen, T., and Juvonen, A. (2012). Online anomaly detection by using N-gram model and growing hierarchical self-organizing maps, *In 2012 8th International Wireless Communications and Mobile Computing Conference (IWCMC)*, 47-52.

86. Nguyen, H. T., Franke, K. (2012, December). Adaptive Intrusion Detection System via online machine learning. In Hybrid Intelligent Systems (HIS), *2012 12th IEEE International Conference on* 271-277.

ÖZGEÇMİŞ

Kişisel Bilgiler
Soyadı, Adı : Tekerek, Adem
Uyruğu : T.C.
Doğum Tarihi ve Yeri: 26.10.1981, K. Maraş
Medeni Hali : Bekâr
Telefon : 0 (545) 352 81 09
E-mail : atekerek@gazi.edu.tr

Eğitim

Derece	Eğitim Birimi	Mezuniyet
Doktora	GÜ/Elektronik ve Bilgisayar Eğitimi Bölümü	2016
Yüksek lisans	GÜ/Elektronik ve Bilgisayar Eğitimi Bölümü	2010
Lisans	GÜ/Elektronik ve Bilgisayar Eğitimi Bölümü	2007
Lise	K. Maraş, Anadolu Ticaret Meslek Lisesi	2000

İş Deneyimi

Yıl	Yer	Görev
2005 - 2007	Netkur Bilişim,	Teknik Destek
2007 - 2008	Çağ Ajans,	Web Programlama
2008 - 2009	Forsnet Bilgi Teknolojileri,	Web Programlama
2009 -	Gazi Üniversitesi Rektörlüğü	Uzman

Yabancı Dil
İngilizce

Yayınlar

MAKALE (SCI & SSCI & Arts and Humanities)

1. Tekerek, A., Gemci, C., Bay, Ö.F. (2016). Web tabanlı saldırı önleme sistemi tasarımı ve gerçekleştirilmesi: Yeni bir hibrit model, *Gazi Üniversitesi Mühendislik-Mimarlık Fakültesi Dergisi*, 31(3), 645-653. http://dx.doi.org/10.17341/gummfd.63355

BİLDİRİ

1. Tekerek, A., Gemci, C., Bay, Ö.F. (2014, 15-17 October). Development of a hybrid web application firewall to prevent web based attacks, *Application of Information and Communication Technologies (AICT), 2014 IEEE 8th International Conference on*, Astana, Kazakhstan, 2014, 1-4, http://dx.doi.org/10.1109/ICAICT.2014.7035910

2. Tekerek A., Bay, Ö.F. (2016, 23-25 November). Development of Web Application Firewall Interface (WAFI), *4th International Conference on Advanced Technology & Sciences (ICAT'Rome)*, Roma, Italy.

GAZİ GELECEKTİR..