
GAZİ ÜNİVERSİTESİ
Gazi
1926

DERİN ÖĞRENME YÖNTEMLERİ KULLANILARAK WEB UYGULAMA GÜVENLİĞİ SAĞLANMASI

MEHMET SEVRİ

DOKTORA TEZİ
BİLİŞİM SİSTEMLERİ ANA BİLİM DALI

GAZİ ÜNİVERSİTESİ
BİLİŞİM ENSTİTÜSÜ

HAZİRAN 2022

MEHMET SEVRİ tarafından hazırlanan "DERİN ÖĞRENME YÖNTEMLERİ KULLANILARAK WEB UYGULAMA GÜVENLİĞİ SAĞLANMASI" adlı tez çalışması aşağıdaki jüri tarafından OY BİRLİĞİ/OY ÇOKLUĞU ile Gazi Üniversitesi Bilişim Sistemleri Ana Bilim Dalında DOKTORA TEZİ olarak kabul edilmiştir.

Danışman: Doç. Dr. Hacer KARACAN

Bilgisayar Mühendisliği, Gazi Üniversitesi

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

Başkan: Unvanı Adı SOYADI

Ana Bilim Dalı, Üniversite Adı (Örnek: Fizik Ana Bilim Dalı, Gazi Üniversitesi)

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

Üye: Prof. Dr. M. Ali AKCAYOL

Bilgisayar Mühendisliği, Gazi Üniversitesi

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

Üye: Prof. Dr. Osman ABUL

Bilgisayar Mühendisliği, TOBB Ekonomi ve Teknoloji Üniversitesi

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

Üye: Unvanı Adı SOYADI

Ana Bilim Dalı, Üniversite Adı (Örnek: Fizik Ana Bilim Dalı, Gazi Üniversitesi)

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

Tez Savunma Tarihi: 15/06/2022

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

……

Prof. Dr. Aslıhan TÜFEKCİ

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.

MEHMET SEVRİ

15/06/2022

DERİN ÖĞRENME YÖNTEMLERİ KULLANILARAK WEB UYGULAMA GÜVENLİĞİ SAĞLANMASI

(Doktora Tezi)

Mehmet SEVRİ

GAZİ ÜNİVERSİTESİ

BİLİŞİM ENSTİTÜSÜ

Haziran 2022

ÖZET

Web uygulamalarının yaygın kullanımı ile birlikte web uygulamalarına yönelik saldırılar artmakta ve daha sofistike hale gelmektedir. Bu saldırıların tespit edilmesi ve engellenmesi ise önemli zorluklar içermektedir. Web saldırılarının tespit edilmesinde etkili olan anomali tabanlı web uygulama güvenlik duvarlarının gerçek zamanlı olarak kullanılabilmeleri için aşılması gereken önemli zorluklar mevcuttur. Bu tez çalışması kapsamında web uygulamalarının güvenliğinin sağlanması ve mevcut zorlukların aşılması amacıyla derin öğrenmeye dayalı yeni yöntemler ve çözümler geliştirilmiştir. Tez çalışması kapsamında ilk olarak, web anomali veriseti eksikliğini gidermek amacıyla GAZI-HTTP isimli yeni ve kapsamlı bir web anomali veriseti oluşturulmuştur. İkinci olarak, web anomali verisetlerindeki dengesiz dağılımdan kaynaklanan problemleri çözmek amacı ile yeni bir veri artırma yöntemi ve derin öğrenme tabanlı yeni bir web uygulama güvenlik duvarı (WAF) geliştirilmiştir. Bu kapsamda özellikle saldırı türü tespitinde yaşanan düşük hassasiyet sorunu çözen DA-SANA veri artırım yöntemi ve Bi-LSTM tabanlı güçlü bir WAF modeli geliştirilmiştir. Üçüncü olarak, web anomali tespitinden kaynaklı gecikmeleri ortadan kaldıran, farklı web uygulamaları için kullanılabilir, genelleştirme gücü yüksek ve gerçek zamanlı kullanılabilecek iki aşamalı, katmanlı topluluk öğrenmeye dayalı derin öğrenme temelli yeni bir WAF sistemi geliştirilmiştir. Geliştirilen sistem sayesinde çok sınıflı saldırı türü tespitinden kaynaklanan problemler ikinci aşamaya ötelenerek, normal web uygulama kullanıcılarının gecikmeden etkilenmesinin önüne geçilmiş ve ikinci aşamadaki topluluk öğrenme modeli ile yüksek hassasiyete sahip bir WAF modeli oluşturulmuştur. Tez kapsamında gerçekleştirilen son çalışmada ise derin öğrenme temelli WAF modellerinin açıklanabilirliğinin sağlanması amacı ile web güvenliği alanına özel yeni bir açıklanabilirlik yöntemi geliştirilmiştir. Derin öğrenme yöntemleri web anomali tespitinde etkin ve başarılı çalışmakla birlikte, modellerin ve model tahminlerinin yorumlanması mümkün olmayıp karakutu modeller olarak adlandırılmaktadır. Bu çalışma ile web güvenliğine özel derin öğrenme temelli karakutu modellerin tahminlerinin açıklanmasını sağlayan, özellikler arasındaki ilişki ve benzerlikleri dikkate alan güçlü bir açıklanabilirlik yaklaşımı ve buna dayalı web tabanlı yazılım çerçevesi geliştirilmiştir.

Bilim Kodu	:	92403
Anahtar Kelimeler	:	Web uygulama güvenliği, web uygulama güvenlik duvarı, derin öğrenme, açıklanabilir yapay zeka, saldırı tespiti sistemi (IDS)
Sayfa Adedi	:	162
Danışman	:	Doç. Dr. Hacer KARACAN

ENSURING WEB APPLICATION SECURITY USING DEEP LEARNING METHODS

(P.h. D. Thesis)

Mehmet SEVRİ

GAZİ UNIVERSITY

INSTITUTE OF INFORMATICS

June 2022

ABSTRACT

With the widespread use of web applications, attacks on web applications are increasing and becoming more sophisticated. Detecting and preventing these attacks contains significant challenges. There are significant problems to be overcome in order to use anomaly-based web application firewalls, which are very effective in detecting web attacks, in real time. In the scope of this thesis, new methods and solutions based on deep learning have been developed in order to ensure the security of web applications and overcome existing challenges. Firstly, a new and comprehensive GAZI-HTTP web anomaly dataset was created to fill the lack of web anomaly datasets. Secondly, a new data augmentation method and a new deep learning-based web application firewall (WAF) have been developed in order to solve the problems arising from the imbalanced distributions in web anomaly datasets. In this context, DA-SANA data augmentation method, which solves the low sensitivity problem, especially in attack type detection, and a powerful WAF model based on Bi-LSTM have been developed. Thirdly, a new two-stage, stacked ensemble learning WAF model based on deep learning has been developed, which can be used for different web applications, has high generalization power and can be used in real-time, eliminating the delays caused by anomaly types detection. Thanks to the developed system, the problems arising from multi-class attack type detection were transferred to the second stage, preventing the normal web application users from being affected without delay, and a high-precision WAF model was created with the ensemble learning model in the second stage. In the last study carried out in the scope of the thesis, a new explainability method specific to the field of web security has been developed in order to ensure the explainability of deep learning-based WAF models. Although deep learning methods work effectively and successfully in web anomaly detection, it is not possible to interpret models and model predictions and they are called black-box models. In this thesis, a strong explainability approach and a web-based software framework, which provides the explanation of the predictions of deep learning-based black box models specific to web security and takes into account the relationships and similarities between the features, has been developed.

Science Code	:	92403
Key Words	:	Web application security, web application firewall, deep learning, explainable artificial intelligence, intrusion detection system
Page Number	:	162
Supervisor	:	Assoc. Prof. Dr. Hacer KARACAN

TEŞEKKÜR

Doktora sürecinde bana rehberlik eden, kıymetli bilgisi ve akademik kişiliği ile daima örnek alacağım, çalışmaların her aşamasında çok değerli katkıları, yol göstericiliği ve destekleri ile beni yönlendiren, bu zorlu süreçte gösterdiği aile sıcaklığı ile motive eden ve naif tavırları sayesinde huzurla çalıştığım, danışmanım olduğu için her zaman onur duyacağım, çok kıymetli ve değerli danışman hocam Doç. Dr. Hacer KARACAN'a en içten sevgi ve şükranlarımı sunarım.

Doktora tezimi yapıcı eleştirileri ve önemli yönlendirmeleri ile geliştirip, iyileştirmemi sağlayan, çalışmalarımla ilgili destekleyici tutumları ile yayım sürecinde çok değerli katkıları olan doktora tez izleme komitesinin değerli üyeleri Prof. Dr. M. Ali AKCAYOL ve Prof. Dr. Osman ABUL hocalarıma saygılarımı ve teşekkürlerimi sunarım.

Gerek iş hayatında, gerekse akademik hayatımda sağladıkları değerli katkılar ve destekler ile samimi dostlukları ve huzurlu bir iş ortamı oluşturdukları için Gazi Üniversitesi Bilişim Enstitüsünde birlikte çalıştığım yönetim, akademik ve idari kadrolardaki tüm değerli mesai arkadaşlarıma sevgi ve teşekkürlerimi sunarım.

Eğitim hayatımın şekilllenmesinde önemli bir role sahip olan, destekleri ile eğitim ve öğretim hayatıma çok önemli katkılar sağlayan, başta değerli hocam Şebnem SAZAK olmak üzere tüm öğretmenlerime minnet ve teşekkürlerimi sunarım.

Bu günlere gelmemde büyük emeği olan çok kıymetli annem Behiye SEVRİ'ye ve rahmetli babam Mustafa SEVRİ'ye minnetarım. Beni her zaman destekleyen değerli aileme sonsuz teşekkürlerimi sunarım.

Doktora sürecinde beni her zaman destekleyen, bu süreçte yükümü paylaşan, moral ve motivasyon kaynağım kıymetli eşim Meliha SEVRİ'ye destek ve sabrı için çok teşekkür ederim.

Son olarak bu tezi, varlığı ile en büyük moral ve motivasyon kaynağım olan çok sevdiğim biricik oğlum Mustafa Kağan SEVRİ'ye ithaf ediyorum.

İÇİNDEKİLER

Sayfa

ÖZET iv
ABSTRACT v
TEŞEKKÜR vi
İÇİNDEKİLER vii
ÇİZELGELERİN LİSTESİ xi
ŞEKİLLERİN LİSTESİ xv
SİMGELER VE KISALTMALAR xviii
1. GİRİŞ 1
2. LİTERATÜR ARAŞTIRMASI 11
2.1. Web Uygulama Güvenliği 11
2.1.1. Web uygulama açıklıkları 11
2.1.2. Web uygulama güvenliğinin sağlanmasında kullanılan yaklaşımlar 19
2.2. İlgili Çalışmalar 24
2.2.1. Geleneksel makine öğrenmesine dayalı çalışmalar 24
2.2.2. Derin öğrenmeye dayalı çalışmalar 26
2.2.3. Açıklanabilir yapay zeka (XAI) ile ilgili anomali tespit çalışmaları 28
3. GAZI-HTTP WEB ANOMALİ VERİSETİNİN OLUŞTURULMASI 31
3.1. Web Uygulamalarının Geliştirilmesi 33
3.2. Balküpü (Honeypot) Sistemlerinin Kurulması 35
3.3. Bulut Tabanlı Sunucular İle Web Trafiğinin Toplanması ve Etiketlenmesi 37
3.3.1. Bulut tabanlı web sunucuya gelen trafiklerin analizi 37
3.3.2. Bulut tabanlı balküpü sistemine gelen web isteklerinin etiketlenmesi 39
3.4. Yerel Sunucular ile HTTP Trafiklerinin Oluşturulması ve Etiketlenmesi 43
3.4.1. Saldırı yüklerinin (payloads) toplanması 44
3.4.2. Web uygulamalarının taranması (spider) 45
3.4.3. Otomatize saldırıların gerçekleştirilmesi 46
3.4.4. Normal web isteklerinin oluşturulması 47

Sayfa

4. DERİN ÖĞRENME TABANLI WAF MODELLERİ İÇİN YENİ BİR VERİ ARTIRMA TEKNİĞİ GELİŞTİRİLMESİ: DA-SANA 51
4.1. Ön Bilgiler 53
4.1.1. Öznitelik çıkarımı 53
4.2. Kullanılan Derin Öğrenme Algoritmaları 55
4.2.1. Çift yönlü kısa uzun zamanlı bellek mimarisi (Bidirectional Long Short Term Memory, Bi-LSTM) 57
4.3. Metodoloji 59
4.3.1. Kendinden uyarlamalı, gürültü eklemeye dayalı veri artırım yöntemi 59
4.4. Çalışmada Kullanılan Web Anomali Tespit Modelleri 65
4.4.1. WAF sistemlerinin teorik arka planı 65
4.4.2. Önerilen sınıflandırma modelin yapısı 67
4.4.3. Hiper-parametre optimizasyonu 67
4.5. Çalışmada Kullanılan Verisetleri ve Doğrulama Yöntemleri 69
4.5.1. CSIC-HTTP veriseti 69
4.5.2. ECML-PKDD veriseti 69
4.5.3. Çalışmada kullanılan araçlar 70
4.5.4. Çalışmada kullanılan model doğrulama yöntemleri 70
4.5.5. Performans metrikleri 71
4.6. Çalışmanın Sonuçları ve Tartışmalar 71
4.6.1. Performans hesaplamaları 72
4.6.2. DA-SANA'nın web anomali tespit modelleri üzerindeki etkinliği 78
4.6.3. Önceki çalışmalar ile kıyaslama 80
4.6.4. Zaman tüketimi ve çalışmanın sınırlılıkları 82
4.7. Çalışmanın Sonuç Bölümü 83

Sayfa

5. İKİ AŞAMALI DERİN ÖĞRENME TABANLI KATMANLI TOPLULUK (ENSEMBLE) ÖĞRENMEYE DAYALI WAF MODELİ.. 85
5.1. Ön Bilgiler 88
5.1.1. Öznitelik çıkarma (Feature extraction) 88
5.1.2. Topluluk öğrenme (Ensemble learning) 89
5.1.3. Çalışmada kullanılan DL algoritmaları 91
5.1.4. Katmanlı ensemble öğrenme modellerinin istatistiksel benzerlikleri ve çeşitliliklerinin analizi 92
5.2. Metodoloji 96
5.2.1. Birinci aşama: iki sınıflı web saldırı tespiti 99
5.2.2. İkinci aşama: web saldırı türlerinin sınıflandırılması için DL tabanlı katmanlı ensemble modeli 100
5.2.3. Çalışmada kullanılan araçlar 103
5.3. Çalışmanın Deneysel Sonuçları ve Tartışmalar 103
5.3.1. Birinci aşama: iki sınıflı sınıflandırıcı sonuçları 104
5.3.2. İkinci aşama: katmanlı ensemble modellerin sonuçları 105
5.3.3. İki aşamalı model sonuçları 108
5.4. Önerilen İki Aşamalı WAF Modelinin DL Algoritmalarına Dayalı Bireysel Modeller ve Önceki Çalışmalar ile Kıyaslanması 109
5.5. Önerilen Modelin Zaman ve Uzay Karmaşıklıkları 111
5.6. Çalışmanın Sonuç Bölümü 114
6. WEB ANOMALİ ALANI ODAKLI YENİ BİR AÇIKLANABİLİRLİK YÖNTEMİNİN GELİŞTİRİLMESİ: XAI-WAF 115
6.1. Açıklanabilir Yapay Zekâ (XAI) Yaklaşımları Hakkında Ön Bilgiler 118
6.1.1. Local Interpretable Model-Agnostic Explanations (LIME) 120
6.1.2. SHapley Additive exPlanations (SHAP) 121
6.2. XAI-WAF Açıklanabilirlik Çerçevesinin (Framework) Temelleri 126
6.3. XAI-WAF Açıklanabilirlik Framework Arayüz ve Açıklanabilirlik Çıktıları ... 135

Sayfa

6.4. Çalışmanın Sonuç Bölümü 144
7. SONUÇ 145
KAYNAKLAR 151
ÖZGEÇMİŞ 159

ÇİZELGELERİN LİSTESİ

Çizelge **Sayfa**

Çizelge 2.1. OWASP Top Ten web uygulama güvenlik riskleri 12
Çizelge 2.2. WAF geliştirme yaklaşımlarının avantajları, dezavantajları ve zorlukları 23
Çizelge 3.1. En çok trafik gelen portlar 38
Çizelge 3.2. Gelen HTTP isteklerindeki metotların dağılımı 39
Çizelge 3.3. Web sunucunun döndüğü cevapların dağılımı 39
Çizelge 3.4. OWASP Modsecurity WAF ile tespit edilebilen saldırılar 40
Çizelge 3.5. Toplanan saldırı yüklerinin dağılımı 45
Çizelge 3.6. GAZI-HTTP verisetinin dağılımı 49
Çizelge 4.1. Ekstra ağaç tabanlı özellik seçimine göre en önemli beş özelliğin isimleri 61
Çizelge 4.2. DA-SANA algoritması 63
Çizelge 4.3. Algoritma ve eşitliklerde kullanılan sembollerin açıklamaları 63
Çizelge 4.4. Önerilen modelin mimari yapısı 67
Çizelge 4.5. ECML-PKDD verisetinin dağılımı 70
Çizelge 4.6. Performans metrikleri 71
Çizelge 4.7. İki sınıflı Bi-LSTM modellerinin yüzde bölmeye göre karmaşıklık matrisleri 72
Çizelge 4.8. İki sınıflı Bi-LSTM modellerinin yüzde bölmeye göre performans değerleri 73
Çizelge 4.9. GAZI-HTTP veriseti için çok sınıflı Bi-LSTM modellerinin yüzde bölmeye göre karmaşıklık matrisi 74
Çizelge 4.10. GAZI-HTTP veriseti için çok sınıflı Bi-LSTM modellerinin yüzde bölmeye göre performans değerleri 74
Çizelge 4.11. ECML-PKDD veriseti için çok sınıflı Bi-LSTM modellerinin yüzde bölmeye göre karmaşıklık matrisi 75
Çizelge 4.12. ECML-PKDD veriseti için çok sınıflı Bi-LSTM modellerinin yüzde bölmeye göre performans değerleri 75
Çizelge 4.13. Bi-LSTM tabanlı tüm modellerin 10-katlı çapraz doğrulamaya göre performans değerleri 76
Çizelge 4.14. Farklı DL modelleri üzerinde DA-SANA yönteminin etkinliği 80
Çizelge 4.15. Önceki çalışmaların sonuçları ile kıyaslama 81

Çizelge **Sayfa**

Çizelge 4.16. Bir web isteğinin eğitilmesinin ortalama çalışma süresi (milisaniye) 83
Çizelge 5.1. Kappa katsayısının yorumlanması 94
Çizelge 5.2. Korelasyon analiz matrisi 95
Çizelge 5.3. Geliştirilen WAF modelinin sözde kodu 98
Çizelge 5.4. İlk aşamadaki iki sınıflı DNN modelinin mimarisi 100
Çizelge 5.5. İkinci aşamada temel öğrenen modellerinin mimarileri. 101
Çizelge 5.6. GAZI-HTTP verisetinin dağılımı 104
Çizelge 5.7. ECML-PKDD verisetinin dağılımı 104
Çizelge 5.8. İlk aşamadaki iki sınıflı sınıflandırmanın karşıtlık matrisleri ve performans değerleri 105
Çizelge 5.9. Yığılmış topluluk modellerinin istatistiksel çeşitlilik analizleri 106
Çizelge 5.10. ECML-PKDD veriseti sonuçlarımızın ile önceki çalışmalarla karşılaştırılması 111
Çizelge 5.11. Önerilen modelin zaman performansları ve tek algoritmaya dayalı WAF modelleri ile karşılaştırılması 113

ŞEKİLLERİN LİSTESİ

Şekil **Sayfa**

Şekil 1.1. Sofistike saldırı araçlarının gelişimi ve ihtiyaç duyulan teknik bilgi arasındaki değişim 1
Şekil 1.2. Taranan web sitelerinin açıklık ve kritik açıklık bulundurma yüzdeleri 2
Şekil 1.3. Ağ ve uygulama katmanı açıklık ve kritik açıklık bulundurma yüzdeleri 3
Şekil 1.4. Tezin organizasyonu 8
Şekil 2.1. SQL enjeksiyon açıklığının sömürülmesi 13
Şekil 2.2. Bozuk kimlik doğrulama açıklığının sömürülmesi 14
Şekil 2.3. Yeterli korunmaya sahip olmayan hassas verilerin açığa çıkarılması 14
Şekil 2.4. XXE açıklığının sömürülmesi 15
Şekil 2.5. Bozuk erişim kontrolü açıklığının sömürülmesi 16
Şekil 2.6. Yanlış güvenlik yapılandırılması açıklığının sömürülmesi 16
Şekil 2.7. XSS açıklığının sömürülmesi 17
Şekil 2.8. Serileştirme ve Serileştirmeden çıkarma işlemleri 18
Şekil 2.9. Bilinen açıklıkları içeren bileşenlerin kullanılması ile ortaya çıkan açıklıkların sömürülmesi 19
Şekil 2.10. Ağ güvenlik duvarının web saldırısı ile atlatılması 20
Şekil 2.11. Standart Firewall ile WAF arasındaki fark 21
Şekil 3.1. GAZI-HTTP verisetinin oluşturulmasına ilişkin topoloji 33
Şekil 3.2. Geliştirilen web uygulamalarının arayüzleri 34
Şekil 3.3. Tshark ağ izleme aracının web trafiğini izlemek için kullanımı 35
Şekil 3.4. Sunucuya gelen ve giden ağ paketlerinin dağılımı (paket/saniye) 38
Şekil 4.1. Tokenizasyon terim sayısının GAZI-HTTP veriseti üzerinde sınıflandırma başarımına etkisi 55
Şekil 4.2. Çalışmada önerilen Bi-LSTM mimarisinin yapısı 57
Şekil 4.3. Ekstra ağaç tabanlı özellik seçimine dayalı özellik önem değerleri ve sıralamaları (en önemli beş özellik) 61
Şekil 4.4. Gürültülü terimlerin miktarının istek uzunluğuna göre farklı oranlarda sınırlandırılması 64
Şekil 4.5. Önerilen modelin akış şeması 68
Şekil 4.6. Bi-LSTM tabanlı tüm modellerin 10-katlı çapraz doğrulamaya göre performans değerleri 77

Şekil **Sayfa**

Şekil 4.7. GAZI-HTTP (a) ve ECML-PKDD (b) verisetleri ile 10-katlı çapraz doğrulamaya göre eğitilen Bi-LSTM modellerinin sınıf bazlı performans değerleri....78
Şekil 5.1. Geliştirilen iki aşamalı DL tabanlı katmanlı ensemble modelinin mimarisi....97
Şekil 5.2. Geliştirilen WAF modelinin eğitim ve anomali tespit aşamalarının akış şeması....98
Şekil 5.3. Yığılmış topluluk modellerinin performans değerlendirmeleri....106
Şekil 5.4. GAZI-HTTP (a) ve ECML-PKDD (b) verisetleri için katmanlı ensemble modellerinin doğru sınıflandırma ve çeşitlilik arasındaki ödünleşim (trade-off) grafiği....107
Şekil 5.5. GAZI-HTTP (a) ve ECML-PKDD (b) verisetleri için iki aşamalı nihai modellerin performans değerleri....109
Şekil 5.6. Önerilen iki aşamalı model ile DL tabanlı tek sınıflandırıcıya dayalı modellerin performans karşılaştırması....110
Şekil 6.1. Açıklanabilir yapay zeka kullanımının amaçları....115
Şekil 6.2. Model açıklanabilirliği ile performans arasındaki ilişki....116
Şekil 6.3. Shapley değerlerine göre NSL-KDD'deki en önemli 20 özellik....123
Şekil 6.4. NSL-KDD üzerinde DNN modeli için en önemli 20 özelliğinin sınıf bazlı etkileri....124
Şekil 6.5. NSL-KDD'deki normal (a) ve anormal (b) örneklerin SHAP' a dayalı yerel açıklamaları....124
Şekil 6.6. LIME ile NSL-KDD' de normal bir örnek için DNN modeli tahmininin açıklaması....125
Şekil 6.7. LIME ile NSL-KDD' de normal bir örnek için DNN modeli tahmininin açıklaması....125
Şekil 6.8. LIME tarafından bir SQLi anomali örneğinin açıklaması....126
Şekil 6.9. LIME tarafından bir ORED anomali örneğinin açıklaması....126
Şekil 6.10. Word2vec modeli için “select” ve “where” kelimelerine en yakın 10 adet kelimenin t-SNE ile iki boyuta indirgenmesi....129
Şekil 6.11. Word2vec modeline göre “select” kelimesine en yakın 5 kelime ve skorları....129
Şekil 6.12. XAI-WAF açıklanabilirlik yaklaşımı akış diyagramı....133

Şekil **Sayfa**

Şekil 6.13. Bir SQLi web isteğinin XAI-WAF yaklaşımına göre sınıflandırılmasında en etkili olan kelime vektörleri 134
Şekil 6.14. Eşik değeri geçen kelime vektörlerinin etki değerleri, skorları ve yorumlanabilirlik matrisi..... 135
Şekil 6.15. XAI-WAF framework ‘ün giriş arayüzü 135
Şekil 6.16. XAI-WAF framework’ün veriseti yükleme arayüzü..... 136
Şekil 6.17. XAI-WAF framework’ ün derin öğrenme modeli eğitim arayüzü..... 137
Şekil 6.18. Eğitimi tamamlanan DL modelinin test edilmesi ve performans değerleri..... 137
Şekil 6.19. XAI-WAF framework’ ün veri görselleştirme arayüzü..... 138
Şekil 6.20. XAI-WAF framework’ ün kelime benzerlik analizi arayüzü..... 139
Şekil 6.21. SQLi isteğinin XAI-WAF modülü ile açıklanabilirliğin sağlanması 141
Şekil 6.22. SQLi isteğinde yer alan en etkili kelimeler ve etki değerleri 141
Şekil 6.23. SQLi isteğinin XAI-WAF modülü ile açıklanabilirliğin sağlanması 142
Şekil 6.24. SQLi isteğinde yer alan en etkili kelimeler ve etki değerleri 142
Şekil 6.25. LFI isteğinin XAI-WAF modülü ile açıklanabilirliğin sağlanması..... 143
Şekil 6.26. LFI isteğinde yer alan en etkili kelimeler ve etki değerleri..... 143

SİMGELER VE KISALTMALAR

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

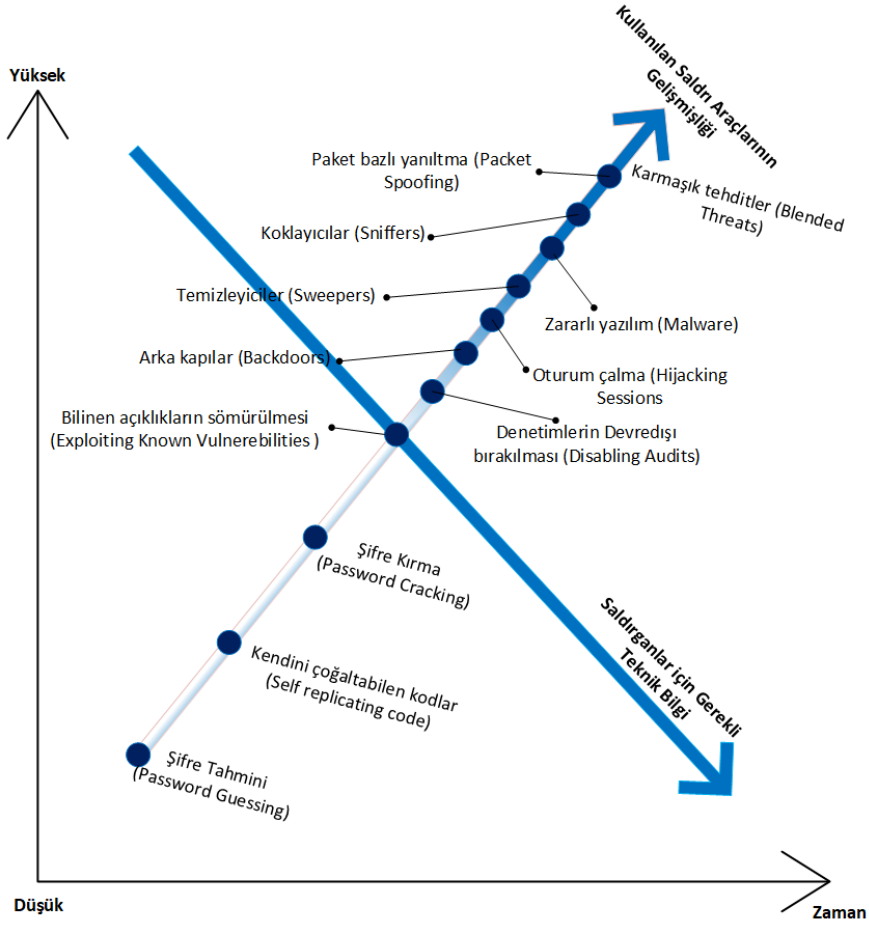
Simgeler	**Açıklamalar**
ms	mili saniye
GHz	Giga hertz

Kısaltmalar	**Açıklamalar**
AE	Otokodlayıcılar (Autoencoders)
Bi-LSTM	Çift yönlü uzun kısa süreli bellek (Bidirectional Long Short Term Memory)
CMS	İçerik yönetim sistemleri (Content Managment System)
CNN	Evrişimli sinir ağı (Convolutional Neural Network)
CRLF	Satır dönüşü satır besleme enjeksiyonu (Carriage Return Line Feed Injection)
CSIC	İspanya Ulusal Araştırma Konseyi (Spanish National Research Council)
CSRF	Siteler arası istek sahtekârlığı (Cross site request forgery)
DA-SANA	Kendinden uyarlamalı gürültü eklemeye dayalı veri artırımı (Data Augmentation technique based on the Self-Adapting Noise Adding)
DFF	İleri beslemeli ağ (Deep Feed Forward)
DNN	Derin sinir ağı (Deep Neural Network)
DT	Karar ağacı (Decision tree)
FAR	Yanlış alarm oranı (False Alarm Rate)
FFNN	İleri beslemeli yapay sinir ağı (Feed-forward neural network)
GLM	Genelleştirilmiş doğrusal model (Generalized Linear Model)
GRU	Geçitli tekrarlayan birim (Gated Recurrent Unit)
HTTP	Hiper metin transfer protokolü (Hyper Text Transfer Protocol)

Kısaltmalar	Açıklamalar
IDS	Saldırı tespit sistemi (Intrusion Detection System)
IoT	Nesnelerin interneti (Internet of Things)
KNN	K en yakın komşu (K-nearest neighbors)
LFI	Yerel dosya yükleme saldırısı (Local File Inclusion)
LIME	Yerel olarak yorumlanabilir model agnostik açıklamalar (Local Interpretable Model-agnostic Explanations)
LSTM	Kısa uzun süreli bellek (Long Short Term Memory)
NLP	Doğal dil işleme (Natural language processing)
ORED	Yeniden yönlendirme saldırısı (Open Redirect)
OS	İşletim sistemi (Operating System)
OWASP	Açık Web Uygulama Güvenliği Projesi (Open Web Application Security Project)
PCA	Temel bileşen analizi (Principal Component Analysis)
RBM	Sınırlı Boltzman makineleri (Restricted Boltzmann Machine)
RCE	Uzaktan komut çalıştırma saldırısı (Remote code execution)
RF	Rastgele orman (Random Forest)
RNN	Yinelemeli sinir ağı (Recurrent Neural Network)
SHAP	Shapley katkı açıklamaları (SHapley Additive exPlanations)
SQLi	SQL enjeksiyonu (SQL Injection)
SSI	Sunucu tarafı yükleme enjeksiyonu (Server Side Include)
SVM	Destek vektör makineleri (Support Vector Machine)
t-SNE	T-dağıtılmış stokastik komşu gömme (t-Distributed Stochastic Neighbor Embedding)
WAF	Web uygulama güvelik duvarı (Web application firewall)
XAI	Açıklanabilir yapay zeka (Explainable Artificial Intelligence)
XSS	Siteler arası betik çalıştırma (Cross-site scripting)
XXE	XML harici girdiler (XML external entities)
YSA	Yapay sinir ağı

1. GİRİŞ

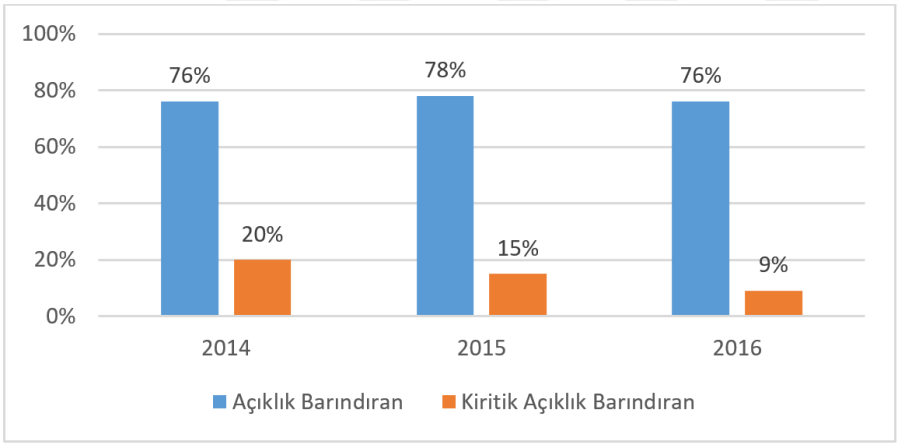
Web uygulamaları; insanların günlük yaşamında önemli bir yere sahip olup, hayatı kolaylaştırması, ekonomik olması, zaman tasarrufu sağlaması, mekândan bağımsız olması gibi özellikleri ile özel ve kamu birçok alanda önemli hizmetler/uygulamalar olarak karşımıza çıkmaktadır. Web uygulamaları bu kadar yaygın kullanımla birlikte bilgisayar korsanları (hacker) tarafından en çok saldırdığı hedeflerin başında gelmeye başlamışdır. Web uygulamalarına yönelik SQL injection (SQLi), cross site scripting (XSS), remote code execution (RCE), local file inclusion (LFI), broken authentication, sensitive data exposure, XML external entities (XXE), cross site request forgery (CSRF), gibi önemli saldırılar gerçekleştirilebilmektedir. Bu web saldırıları sonucunda hassas veriler ifşa olabilmekte, sistemler ele geçirilebilmekte ve önemli gizlilik ihlalleri veya finansal kayıplar yaşanabilmektedir. Birçok ekonomik ve kritik işlem web uygulamaları aracılığıyla gerçekleştirildiği için, bu uygulamalarına saldıran bilgisayar korsanlarının (hackerlar) sayısı her geçen gün artmakta ve saldırılar daha sofistike hale gelmektedir.



Şekil 1.1. Sofistike saldırı araçlarının gelişimi ve ihtiyaç duyulan teknik bilgi arasındaki değişim [1]

Günümüzde web saldırılarında kullanılan araçların sayısı artmış olup, saldırı araçları daha karmaşık saldırıları destekler hale gelmiştir. Bununla birlikte saldırı yapmak için gereken bilgi ve yetkinlik ihtiyacı azalış göstermektedir [1]. Böylece, temel bilgilere sahip ve lamer olarak adlandırılan bilgisayar korsanlığına özenen kişiler tarafından dahi çok karmaşık ve zarar veren saldırılar gerçekleştirilebilmektedir. Sürekli gelişen ve daha sofistike hale gelen saldırı teknikleri ile web uygulamalarında bulunabilecek açıklıklar sömürülerek; devletlere, kurumlara, şirketlere veya kişilere çok önemli zararlar verebilmektedir. Şekil 1.1'de saldırı araçlarının gelişimindeki artış ve buna bağlı olarak saldırı gerçekleştirmek için ihtiyaç duyulan bilgi/yetkinlik seviyesindeki azalış görülmektedir.

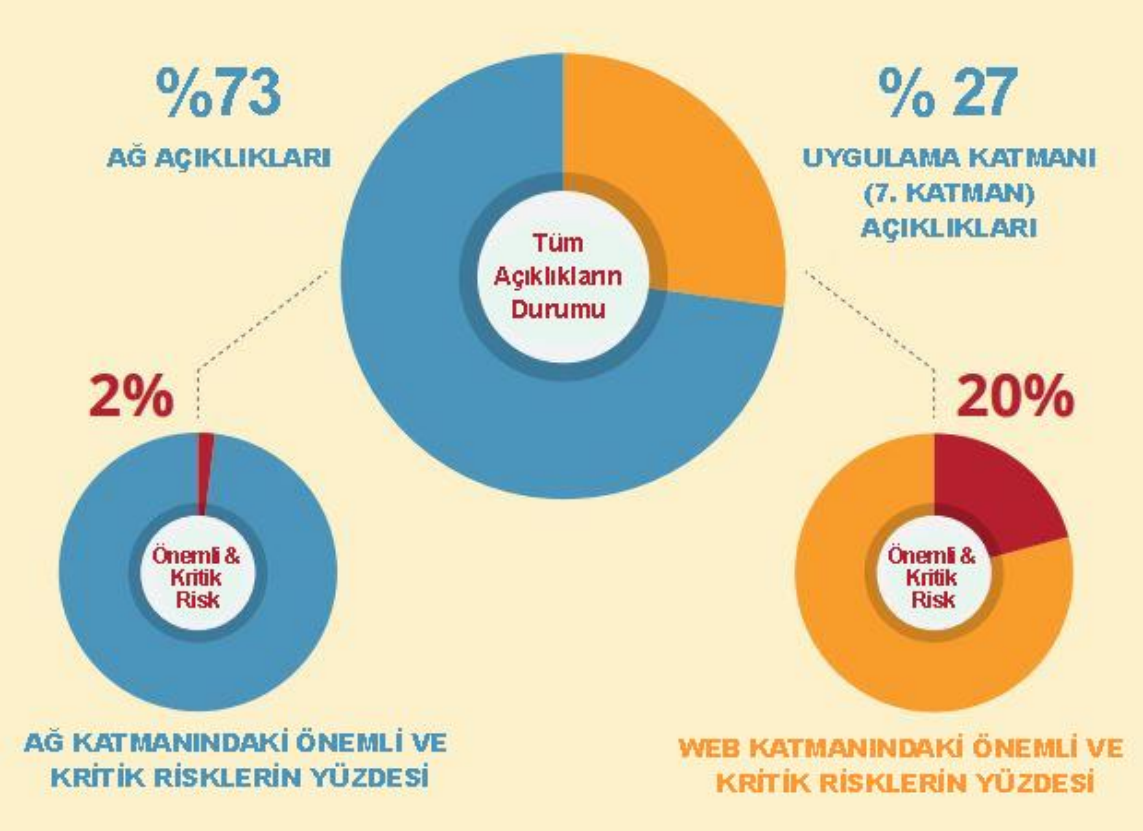
Symantec firmasının 2017 yılsonu internet güvenliği tehdit raporuna [2] göre, web uygulamaları internet güvenliği alanında en büyük hedeflerden biri olmaya devam etmekte olup, ortalama olarak günlük 229.000 siteye saldırı gerçekleşmektedir. Şekil 1.2'de görüldüğü üzere taranılan web sitelerinden %76' sı açıklık barındırırken, %9' u ise kritik sayılan açıklıklar barındırmaktadır.



Şekil 1.2. Taranan web sitelerinin açıklık ve kritik açıklık bulundurma yüzdeleri [2]

Siber güvenlik alanında gerçekleştirilen araştırmalara bakıldığında en çok çalışmanın ağ güvenliği üzerine yapılan çalışmalar olduğu görülürken, web güvenliği alanında gerçekleştirilen çalışma sayısı sınırlı kalmaktadır. Bununla birlikte, siber güvenlik firması Edgescan tarafından yayınlanan 2018 yılı açıklık analizi raporuna (vulnerability statistics report) göre, tespit edilen ihlallerin %73' ü ağ tabanlı iken bunların sadece %2' si kritik açıklık olup, tespit edilen % 27 uygulama seviyesi açıklığının tamamı web uygulamaları ile ilgili olup, bunların %20' si kritik açıklıktır [3]. Buna göre uygulama seviyesinde kritik açıklık barındırma oranı %5.7 iken bu oran ağ tabanlı olarak % 1.46 olup; uygulama

seviyesinde ortaya çıkan kritik açıklıkların miktarı, ağ tabanlı kritik açıklık miktarının yaklaşık 4 katıdır. Bu durum da web güvenliğinin siber güvenlik açısından ne kadar kritik olduğunu göstermektedir. Web uygulamalarına gelen saldırıların tespit edilmesi ve önlenmesi önemlidir. Bununla birlikte web saldırılarının tespit edilmesinde önemli zorluklar bulunmaktadır.



Şekil 1.3. Ağ ve uygulama katmanı açıklık ve kritik açıklık bulundurma yüzdeleri [3]

Bu tez çalışması kapsamında web güvenliği alanındaki zorluklar kapsamlı şekilde ele alınmış ve çözümler ortaya konulmuştur. Bu kapsamda web saldırılarının tespit edilmesi ve önlenmesinde karşılan zorlukları aşmak amacı ile farklı derin öğrenme yöntemlerinin uygulandığı çözümler geliştirilmiştir. Derin öğrenmeye dayalı yöntemler kullanılarak, web uygulama güvenliğin sağlanması amacı ile yeni ve kapsamlı çalışmalar gerçekleştirilmiştir. Tez çalışmasında web güvenliği alanındaki ana problemler tespit edilerek, bu problemler derin öğrenme teknikleri dâhil olmak üzere farklı çalışmalar ile kapsamlı olarak ele alınarak yenilikçi çözüm önerileri sunulmuştur. Tez çalışması kapsamında gerçekleştirilen çalışmalara ilişkin ön bilgiler ve alana sağlanan başlıca katkılara aşağıda verilmiş olup, çalışmaların detayları ayrı bölümler halinde sunulmaktadır.

Tez çalışması kapsamında öncelikle web güvenliği alanındaki mevcut web anomali veri setlerinin yetersiz olduğu tespit edilmiş olup, bu problemin çözümüne yönelik çalışmalar gerçekleştirilmiştir. Bu problemi çözmek amacı ile yeni ve kapsamlı bir web anomali veriseti oluşturulmuştur. Oluşturulan GAZI-HTTP isimli web anomali veriseti ile web güvenliği alanına sağlanan başlıca katkılar aşağıda maddeler halinde sunulmaktadır.

- Anomali tabanlı web güvenliği çalışmalarında kullanılan kıyaslama (benchmark) verisetleri incelendiğinde, az sayıda veriseti olduğu ve mevcut verisetlerinin güncel olmadığı tespit edilmiştir. Tez çalışması kapsamında oluşturulan GAZI-HTTP veriseti ile web güvenliği alanına güncel saldırı türlerini kapsayan bir benchmark veriseti kazandırılmıştır.

- GAZI-HTTP veriseti OWASP Top Ten kapsamında yer alan web açıklık türlerini kapsamakta olup, güncel ve kritik web saldırı tiplerinin tamamını içermektedir. Veriseti hem iki sınıflı hem de çok sınıflı sınıflandırmaya uygundur.

- Verisetinin oluşturulmasında hem herkese açık balküpü sistemlerinden toplanan web istekleri hem de yerel sunucular ile manuel oluşturulan istekler kullanılmıştır.

- Verisetinin oluşturulmasında internette yer alan saldırı payloadları, açık kaynak saldırı araçlarının barındırdıkları saldırı payloadları dahil olmak üzere geniş kapsamlı bir saldırı payload veritabanı oluşturulmuş ve verisetinin oluşturulması sürecinde kullanılmıştır. Bilgimiz dahilinde en geniş kapsamlı web anomali veriseti oluşturulmuştur.

- Tez kapsamında kurulan balküpü sistemi, oluşturulan GAZI-HTTP veriseti, web trafiklerinin etiketlenmesi yöntemi tamamen özgün ve yenilikçidir.

Tez çalışması kapsamında ikici olarak, dengesiz (imbalanced) veri dağılımına sahip web anomali verisetleri için sınıflandırma hassasiyetini ve başarımını artıran DA-SANA (data augmentation technique based on the self-adapting noise adding method) isimli yeni bir veri artırım yöntemi geliştirilmiştir. Yapılan çalışma ile GAZI-HTTP veriseti dâhil olmak üzere farklı web anomali verisetleri üzerinde geliştirilen farklı derin öğrenme algoritmalarına dayalı modellerin başarım ve hassasiyetlerinin artırıldığı gösterilmiştir. Bu çalışmanın web güvenliği alanına sağladığı başlıca katkılar aşağıda sunulmuştur.

- Web anomali tespit hassasiyetini güçlendirmek için DA-SANA isimli gürültü eklemeye dayalı yeni bir veri artırım yöntemi geliştirilmiştir. DA-SANA yönteminde, gürültü katsayısı ilerde detaylı olarak açıklandığı üzere saldırı isteklerinin istatistiksel olarak belirleyici özelliklerine dayalı olarak belirlenmektedir. Adaptif gürültü eklemesi yapılarak, normal ve saldırı trafiğinin gizli uzayda (hidden space) birbirinden daha da ayrılması sağlanarak, tespit oranının iyileştirilmesi sağlanmaktadır.

- Gerçekleştirilen çalışma ile gerçek zamanlı çalışabilen derin öğrenmeye dayalı özgün web uygulama güvenlik modelleri geliştirilmiştir. Çalışma kapsamında geliştirilen DA-SANA'ya dayalı WAF modellerinin performans değerleri hem iki sınıflı hem de çok sınıflı sınıflandırma için bizim araştırmalarımıza göre en iyi performans değerlerini içermektedir.

- *Çalışmanın kaynak kodları ve önişleme gerçekleştirilmiş verisetleri yeniden üretilebilirlik ve gelecekteki çalışmalara katkı sağlamak amacıyla herkese açık olarak paylaşılacaktır.

  * https://github.com/mehmetsevri/Deep-Learning-based- WAF

Anomali tabanlı WAF sistemlerinin geliştirilmesinde önemli zorluklardan birisi düşük sıklıkta gerçekleştirilen saldırıların tespit edilmesi iken, diğer önemli bir zorluk ise zaman gereksinimi açısından son kullanıcıyı etkilemeyen ya da minimum düzeyde etkileyen performanslı sistemler geliştirilmesidir. Tez kapsamında gerçekleştirilen üçüncü çalışma ile gerçek zamanlı web saldırı tespitinde yer alan bu problemleri çözen ve yüksek tespit başarımını düşük zaman ve kaynak gereksinimi ile sağlayabilen derin öğrenmeye dayalı iki aşamalı ensemble WAF modeli geliştirilmiştir. Gerçekleştirilen çalışmanın web güvenliği alanına sağladığı başlıca katkılar aşağıda yer almaktadır.

- Çalışmada kullanılan iki aşamalı yapı sayesinde normal kullanıcı istekleri sadece ilk aşamada işlenmekte, bu aşamadaki modelin basitliği ve hızı ile çok sınıflı sınıflandırmanın getireceği zaman gecikmesinden kurtarılarak, hızlı ve yüksek doğrulukta tespit edilebilmektedir. Böylece normal ve saldırı isteklerinin sınıflandırılması ilk aşamada gerçekleştirilerek, saldırı türü tespitinin gerektirdiği karmaşık yapı gereksinimi ikinci aşamaya ötelenmektedir.

- Saldırı tipleri, ikinci aşamada kullanılan katmanlı (stacked) ensemble model sayesinde yüksek hassasiyet ve doğrulukta sınıflandırılmaktadır.

- Çalışma da geliştirilen modeller daha önce aynı kıyaslama (benchmark) verisetlerini kullanılan çalışmaların sonuçları ile karşılaştırılmıştır. Geliştirilen iki aşamalı katmanlı ensemble model ile bilgimiz dâhilinde en yüksek başarım performanslarına ulaşılmıştır.

- *Çalışmanın kaynak kodları ve önişleme gerçekleştirilmiş verisetleri yeniden üretilebilirlik ve gelecekteki çalışmalara katkı sağlamak amacıyla herkese açık olarak paylaşılacaktır.
  *https://github.com/mehmetsevri/Two-Stage-Deep-Learning-Based-Stacked-Ensemble-Model-for-Web-Application-Security

Saldırı tespit sistemleri gibi kritik alanlarda çalışan anomali tabanlı modellerin şeffaf ve açıklanabilir olması önemli bir gereksinimdir. Bununla birlikte derin öğrenme yöntemleri gibi yüksek başarıma sahip yapay öğrenme yöntemleri genelde karakutu modeller olup, model tahminlerinin nasıl oluşturulduğu ve karar verme sürecinin nasıl gerçekleştiği modele bakılarak şeffaf olarak açıklanamamaktadır. Bu sebeple başarılı çalışan anomali tabanlı karakutu modellerin saldırı tespit sistemi olarak gerçek hayatta kullanımı yeterince yaygın değildir. Doktora tezi kapsamında gerçekleştirilen dördüncü çalışma ile başarılı çalışan derin öğrenme tabanlı web anomali tespit modellerini açıklayabilen yeni bir açıklanabilirlik yöntemi geliştirilmiştir. Bu çalışmanın sağladığı başlıca katkılar aşağıda sunulmaktadır.

- Geliştirilen XAI-WAF açıklanabilirlik yöntemi web uygulama güvenliği alanına özgü ilk yöntemdir. XAI-WAF, web isteğindeki kelimelerin birliktelik ve ilişkilerini, kelime gömme (word embedding) vektörlerindeki benzerlikleri ve kelimelerin web isteğindeki konumlarını dikkate alan özgün bir hesaplama yapısına sahiptir. Bireysel web isteklerinin karakutu derin öğrenme model tahminlerini web isteğinde yer alan kelime ve kelime grupları üzerinden açıklayabilmektedir.

- XAI-WAF yaklaşımı, web anomali odaklı olduğu için tahmin sınıfına dayalı olarak güçlü görsel açıklamalar sunabilmektedir. Ayrıca gerçek sınıf etiketlerinin verilmesi durumunda derin öğrenme modellerinin optimize edilmesi için yanlış sınıflandırmaya neden olan girdileri açıklayabilmektedir.

- Geliştirilen XAI-WAF açıklanabilirlik yaklaşımı ile literatürde sık kullanılan ve bilinen açıklanabilirlik yaklaşımlarından LIME ve SHAP kıyaslanmıştır. Geliştirilen yöntem alana özgü olduğu için ve kelime gömme (word embedding) de yer alan kelime birlikteliklerini ve ilişkilerini ele aldığı için web isteklerinin analizinde daha stabil çalıştığı görülmüştür.

- Farklı derin öğrenme algoritmalarına dayalı geliştirilen modeller ile yeni açıklanabilirlik yönteminin bir arada kullanılabildiği, web isteklerini gerçek zamanlı sınıflandırabildiği bir yazılım iskeleti (freamework) yapısı oluşturularak, kullanıma sunulmuştur.

- Farklı derin öğrenme algoritmalarına dayalı, başarılı ve etkili WAF modelleri geliştirilmiş ve derin öğrenme yöntemlerinin web güvenliği alanındaki etkinliği gösterilmiştir.

- *Çalışmanın kaynak kodları ve önişleme gerçekleştirilmiş verisetleri yeniden üretilebilirlik ve gelecekteki çalışmalara katkı sağlamak amacıyla herkese açık olarak paylaşılacaktır.
  *https://github.com/mehmetsevri/XAI-WAF-A-Novel-Explainable-AI-Framework-for-Deep-Learning-Based-Web-Application-Security

Yukarıda kısaca açıklanan çalışmaların ayrıntılı detayları ve sonuçları aşağıda ayrı bölümler halinde sunulmaktadır.

Tezin Organizasyonu

Tez giriş bölümü dâhil toplam 7 bölümden oluşmaktadır. Tezin organizasyonu ve akışı Şekil 1.4' de görülmektedir. Tezin ikici bölümünde web uygulama güvenliği alanına ilişikin literatür araştırmasına yer verilmiştir. Bu kapsamda web uygulama güvenlik açıklıkları, web güvenliğinin sağlanmasında kullanılan temel güvenlik yaklaşımları ve web güveliği alanında gerçekleştirilen mevcut çalışmalar ele alınmıştır. Bu bölüm kapsamında web uygulamalarında ortaya çıkan en kritik güvenlik açıklıkları ele alınmış, nedenleri ve saldırganlar tarafından nasıl sömürüldükleri incelenmiştir. Tez kapsamında OWASP Top Ten' de yaralan tüm web uygulama açıklıkları ele alınmaktadır. Ayrıca bu bölümde literatürde yer alan çalışmalara dayalı olarak web uygulamalarının güvenliğinin sağlanmasında kullanılan temel yaklaşımlar ele alınmıştır. Tez kapsamında gerçekleştirilen çalışamalar ile ilişkili olan mevcut çalışmalar sunulmuştur.



Şekil 1.4. Tezin organizasyonu

Tezin üçüncü kısmında tez kapsamında oluşturulan GAZI-HTTP isimli web anomali veriseti oluşturulma süreci ve özellikleri kapsamlı olarak açıklanmıştır. Web anomali verisetinin oluşturulmasında hem herkese açık olarak oluşturulan balküpü sistemleri ile dış kaynaklardan toplanan web istekleri hem de tez kapsamında yerel ağda kurulan balküpü sistemlerine yönelik otomatize araçlar ve scriptler ile oluşturulan web istekleri kullanılmıştır. Bu kapsamda bal küpü sistemlerinin geliştirilmesi, web saldırılarının oluşturulmasında kullanılan yüklerin (payload) toplanması, normal kullanıcı isteği ve farklı saldırı türleri içeren web isteklerinin oluşturulması ve etiketlenmesi aşamaları detaylı olarak açıklanmıştır.

Tezin dördüncü bölümünde web uygulama güvenliğinin sağlanması ve dengesiz veriseti dağılımından kaynaklanan hassasiyet probleminin ortadan kaldırılması amacıyla özgün olarak geliştirilen DA-SANA isimli gürültü eklemeye dayalı veri artırım yöntemi ve derin öğrenme tabanlı saldırı tespit modelleri sunulmuştur. Bu çalışma ile dengesiz veri dağılımından kaynaklanan düşük hacme sahip sınıf etiketlerinin tespit zorluğu ortadan kaldırılarak yüksek hassasiyete sahip derin öğrenme tabanlı web uygulama güvelik duvarlarının oluşturulması sağlanmamıştır.

Tezin beşinci bölümünde çok sınıflı sınıflandırmadan kaynaklanan karmaşık model kullanımı ve buna dayalı zaman gecikmesinin önlenmesi amacıyla iki aşamalı katmanlı ensemble öğrenme ile derin öğrenme tabanlı WAF yaklaşımı geliştirilmiştir. Bu çalışma ile web saldırı tespitini iki aşamalı olarak gerçekleştirebilen yeni bir yöntem geliştirilmiştir. İlk aşama da geliştirilen basit, hızlı ve yüksek başarım performansına sahip derin sinir ağı (DNN) tabanlı WAF modeli ile normal ve anormal web trafiklerinin iki sınıflı olarak ayırt edilmesi sağlanmaktadır. İkinci aşamada ise geliştirilen yüksek hassasiyete sahip katmanlı ensemble öğrenmeye dayalı derin öğrenme modelleri içeren bir yapı ile ilk aşamada saldırı olduğu belirlenen web trafiklerin saldırı türlerinin tespit edilmesi sağlanmaktadır. Bu sayede normal web kullanıcılarının saldırı türü tespitinden kaynaklanan karmaşıklık ve zaman gecikmesinden etkilenmelerinin önüne geçilebilmektedir. Ayrıca tasarlanan ikinci aşamadaki yapı sayesinde yüksek ayrıklık ve düşük korelasyona sahip, dolayısıyla yüksek genelleştirebilme gücü olan bir WAF sistemi oluşturulmuştur.

Tezin altıncı bölümünde karakutu derin öğrenme modellerinin gerçek zamanlı WAF sistemi olarak kullanılabilmelerini sağlamak amacıyla, derin öğrenme tabanlı WAF modellerinin açıklanabilirliğini ve yorumlanabilirliğini sağlayan yeni bir açıklanabilirlik yöntemi geliştirilmiştir. Derin öğrenme modelleri karakutu modeller olarak isimlendirilmekte olup,

nasıl çalıştıklarının ve nasıl karar verdiklerinin şeffaf olarak açıklanamadığı yapılara sahiptirler. Şeffaf yapıda modeller oluşturulamadığı için web anomali tespiti gibi kritik sistemlerde gerçek zamanlı olarak kullanımlarından kaçınılmaktadır. Tez kapsamında gerçekleştirilen çalışma ile bu problemi çözen, web anomali tespiti odaklı geliştirilen yeni bir açıklanabilirlik yöntemi ile karakutu derin öğrenme modellerinin tahminlerinin açıklanabilirliği sağlanmaktadır. Geliştirilen açıklanabilirlik yöntemi web anomali tespitinde kullanılan derin öğrenme modellerinin herhangi bir vekil model kullanmadan açıklanabilmesini sağlayan alana özgü bir yöntemdir. Geliştirilen yöntem web isteklerinde yer alan kelimelerin birliktelik ilişkilerinin yanı sıra diğer açıklanabilirlik yaklaşımlarının içermediği kelime gömme yöntemlerini sağladığı ilişki durumlarını ve kelime konumlarını da hesaba kattığı için güçlü bir açıklanabilirlik sağlamaktadır.

Tezin son bölümü sonuç bölümünden oluşmaktadır. Bu bölümde tezin amacı ve gerçekleştirilen çalışmaların genel olarak yorumlamaş, tezin web güvenliği alanına ve bilime katkıları özet olarak açıklanmış, genel olarak kısıtlılıklarından bahsedilmiş ve gelecek çalışmalara ışık tutması bakımından yapılabilecek çalışmalara değinilmiştir.

2. LİTERATÜR ARAŞTIRMASI

Bu bölümde tez çalışması kapsamında ele alınan web uygulama güvenlik açıklıkları, web güvenliğinin sağlanmasında kullanılan yaklaşımlar ve bu alanda diğer araştırmacılar tarafından gerçekleştirilen mevcut çalışmalar ele alınmaktadır. Bu kapsamda öncelikle web uygulamalarında bulunan en kritik güvenlik açıklıkları ve web uygulama güvenliğinin sağlanmasında kullanılan temel yaklaşımlar hakkında önbilgiler sunulmaktadır. İkinci olarak, tez kapsamında gerçekleştirdiğimiz çalışmalar ile ilişkili daha önce gerçekleştirilmiş web uygulama güvenliği çalışmaları sunulmaktadır.

2.1. Web Uygulama Güvenliği

Bu bölümde web uygulamalarında bulunan en kritik güvenlik açıklıkları ve web uygulama güvenliğinin sağlanmasında kullanılan temel yaklaşımlar hakkında önbilgiler sunulmaktadır.

2.1.1. Web uygulama açıklıkları

Web uygulamaları, yaygın kullanılmaları, hassas ve kritik veriler içermeleri ve genel erişime açık olmaları sebebiyle saldırganların (hackers) öncelikli hedefleri haline gelmiştir. Web uygulamalarında bulunan açıklıklar farklı saldırı yöntemleri ile sömürülebilmektedir. Web uygulamalarına gerçekleştirilen saldırılar; genel olarak gizli verilerin ele geçirilmesi, değiştirilmesi, yetki yükseltilmesi, kullanıcı bilgilerinin veya kimliklerinin ele geçirilmesi, sistemlerin hizmet dışı bırakılması ya da web sunucuların tamamen ele geçirilmesi gibi kritik problemlere neden olabilmektedir. Web güvenlik açıklıklarına ilişkin olarak Açık Web Uygulama Güvenliği Projesi (Open Web Application Security Project - OWASP) tarafından belli aralıklar ile en kritik on adet web güvenlik açıklığı (OWASP Top Ten) belirlenerek raporlanmaktadır [4]. Owasp Top Ten' in mevcut versiyonundaki en önemli güvenlik açıklığı, SQL, NoSQL, OS (Operating system) enjeksiyonlarını içeren enjeksiyon açıklıklarıdır [4]. İkinci en önemli açıklık ise oturum kırılması (session hijacking), oturum tokenlerinin çalınması ve yetkilendirme kırma açıklıklarıdır [4]. Bunların yanı sıra, zayıf yapılandırılmış XML yorumlayıcılarını hedef alan XXE ve girdilerin yeterli doğrulanmaması ve filtrelenmemesi ile sömürülebilen XSS saldırıları da en önemli saldırılar arasında yer almaktadır. OWASP tarafından 2017 [4] yılında sıralanan en kritik 10 web uygulama güvenlik riski Çizelge 2.1'de görülmekte olup, bu açıklıklara ilişkin bilgiler aşağıda mevcuttur.

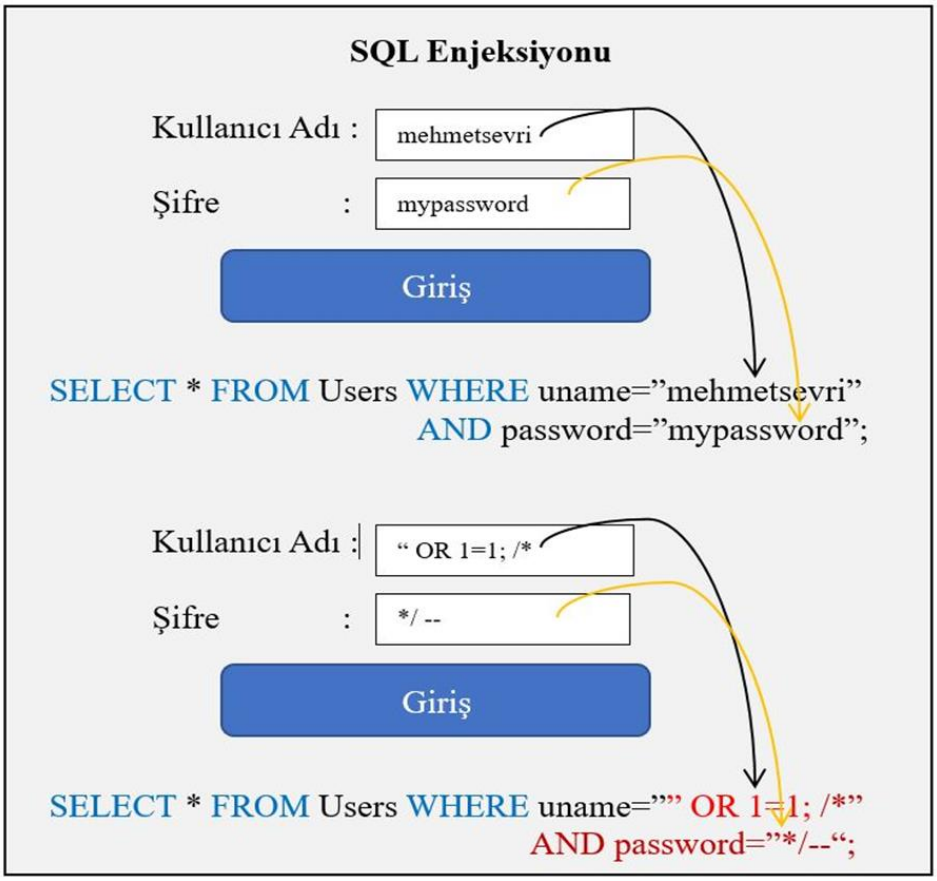
Çizelge 2.1. OWASP Top Ten web uygulama güvenlik riskleri [4]

Risk Sırası (Risk Rank)	**Güvenlik Riski (Security Risk)**	**(Kaynağı) Source**	**Hedefi (Destination)**	**Sömürebilirlik Düzeyi (Exploitability)**	**Zayıflığın Prevelansı (Weakness Prevalence)**	**Zayıflığın Tespit Edilebilirliği (Weakness Detectability)**	**Teknik etkisi (Technical Impacts)**
1	Enjeksiyon (Injection)	Uygulama (Application)	Sunucu / Veritabanı (Server / Database)	Kolay (Easy)	Yaygın (Common)	Kolay (Easy)	Yüksek (Severe)
2	Bozuk kimlik doğrulama (Broken authentication)	Uygulama (Application)	Kullanıcılar (Clients)	Kolay (Easy)	Yaygın (Common)	Orta (Average)	Yüksek (Severe)
3	Hassas veriyi açığa çıkarma (Sensitive data exposure)	Uygulama (Application)	Kullanıcılar (Clients)	Orta (Average)	Çok yaygın (Widespread)	Orta (Average)	Yüksek (Severe)
4	XML harici girdiler (XML external entities – XXE)	Uygulama (Application)	Sunucu / Ağ (Server / Network)	Orta (Average)	Yaygın (Common)	Kolay (Easy)	Yüksek (Severe)
5	Bozuk erişim kontrolü (Broken access control)	Uygulama (Application)	Kullanıcılar (Clients)	Orta (Average)	Yaygın (Common)	Orta (Average)	Yüksek (Severe)
6	Yanlış güvenlik yapılandırması (Security misconfiguration)	Sunucu (Server)	Server	Kolay (Easy)	Çok yaygın (Widespread)	Kolay (Easy)	Orta (Moderate)
7	Siteler arası betik çalıştırma (Cross-site scripting, XSS)	Uygulama (Application)	Kullanıcılar (Clients)	Kolay (Easy)	Çok yaygın (Widespread)	Kolay (Easy)	Orta (Moderate)
8	Güvensiz serileştirme (Insecure deserialization)	Uygulama (Application)	Sunucu (Server)	Zor (Difficult)	Yaygın (Common)	Orta (Average)	Yüksek (Severe)
9	Bilinen açıklıkları içeren bileşenlerin kullanılması (Using components with known vulnerabilities)	Uygulama (Application)	Sunucu / Kullanıcılar (Server / Clients)	Orta (Average)	Çok yaygın (Widespread)	Orta (Average)	Orta (Moderate)
10	Yetersiz kayıt tutma ve izleme (ınsufficient logging & monitoring)	Sunucu / Ağ (Server / Network)	Sunucu / Kullanıcılar (Server / Clients)	Orta (Average)	Çok yaygın (Widespread)	Zor (Difficult)	Orta (Moderate)

OWASP Top Ten'de yer alan web uygulama açıklıkları

Enjeksiyon (Injection)

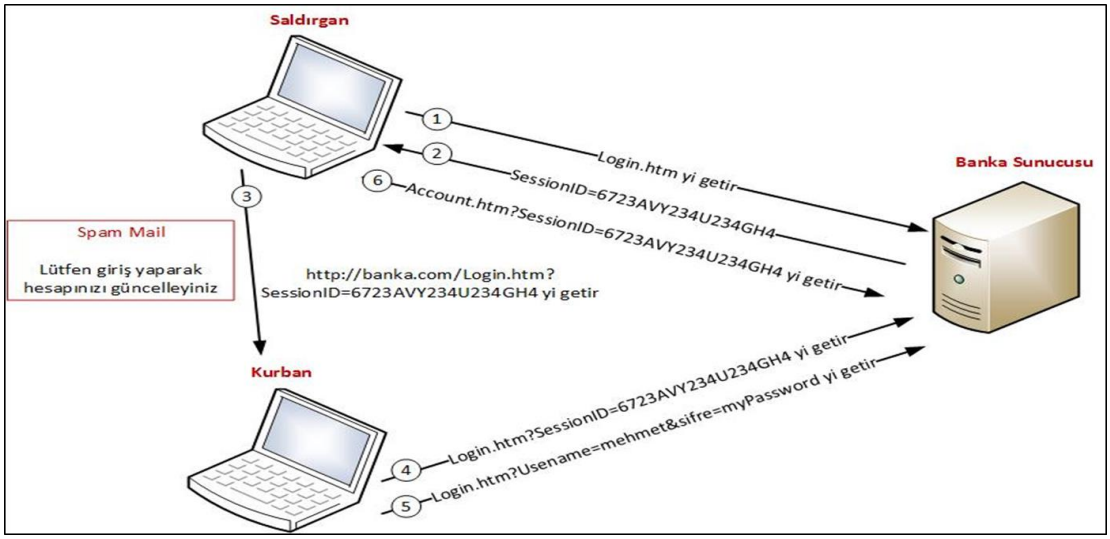
SQL, NoSQL, OS ve LDAP enjeksiyonu gibi enjeksiyon açıklıkları, güvenilmeyen verilerin, bir komutun veya sorgunun bir parçası olarak bir yorumlayıcıya gönderildiğinde ortaya çıkmaktadır (bkz. Şekil 2.1). Saldırganın amaca yönelik zararlı verileri (payload) ile yorumlayıcıyı istenmeyen komutları çalıştırma veya uygun yetkilendirme olmadan verilere erişme konusunda kullanmasına neden olmaktadır [4, 5].



Şekil 2.1. SQL enjeksiyon açıklığının sömürülmesi

Bozuk kimlik doğrulama (Broken authentication)

Kimlik doğrulama ve oturum yönetimi ile ilgili uygulama işlevlerinin yanlış yapılandırılması, genellikle şifrelerin, anahtarların veya oturum tokenlarının riske atılmasına ve saldırganlar tarafından ele geçirilmesine veya saldırganların diğer uygulama açıklıklarından yararlanabilmesine imkân sağlayarak kullanıcıların kimlik bilgilerinin geçici veya kalıcı olarak ele geçirilmesine sebep olabilmektedir (bkz. Şekil 2.2) [4, 6].



Şekil 2.2. Bozuk kimlik doğrulama açıklığının sömürülmesi

Hassas veriyi açığa çıkarma (Sensitive data exposure)

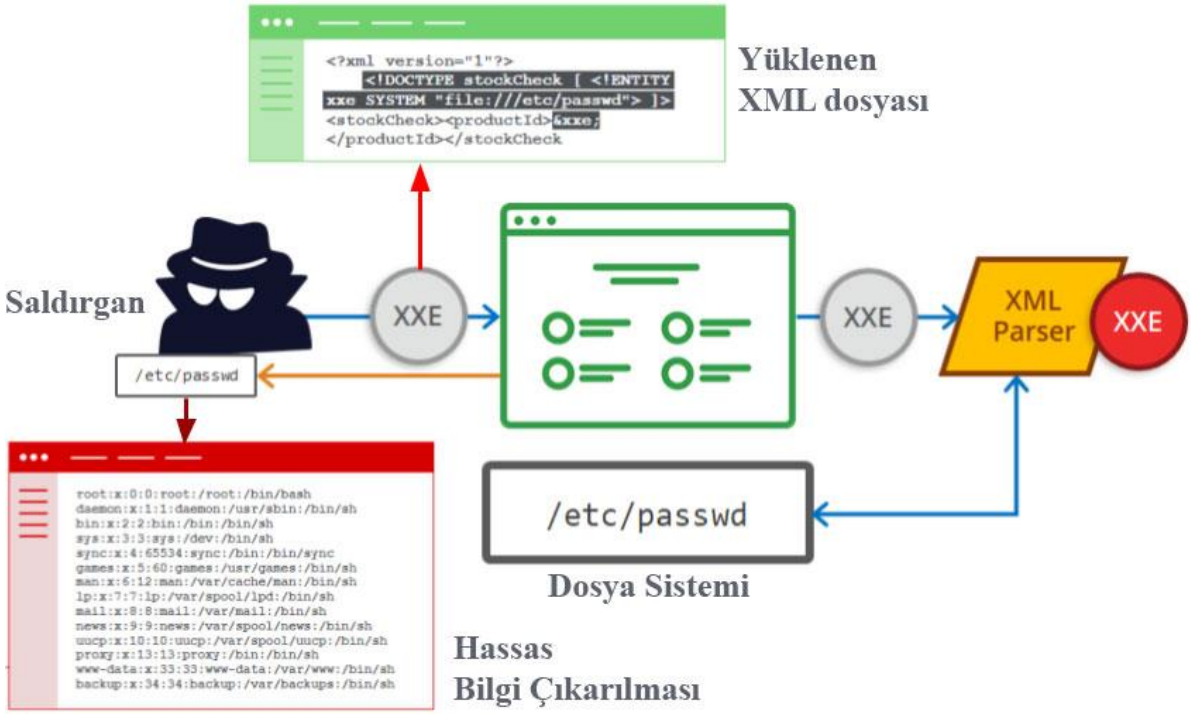
Birçok web uygulaması ve uygulama programlama arayüzü (API), finansal, sağlık hizmeti ve kişisel bilgiler gibi hassas verileri doğru şekilde korumaz. Saldırganlar, kredi kartı sahtekârlığı, kimlik hırsızlığı veya diğer suçları işlemek için zayıf korumalı verileri çalabilir veya değiştirebilirler (bkz. Şekil 2.3). Hassas veriler, saklanırken veya gönderilirken şifreleme gibi ekstra koruma olmadığı durumlarda tehlikeye atılabilmekte ve bu verilerin tarayıcı ile değiştirilmesinin gerektiği durumlarda özel önlemler alınmasını gerektirmektedir [4, 7].



Şekil 2.3. Yeterli korunmaya sahip olmayan hassas verilerin açığa çıkarılması

XML harici girdiler (XML external entities - XXE)

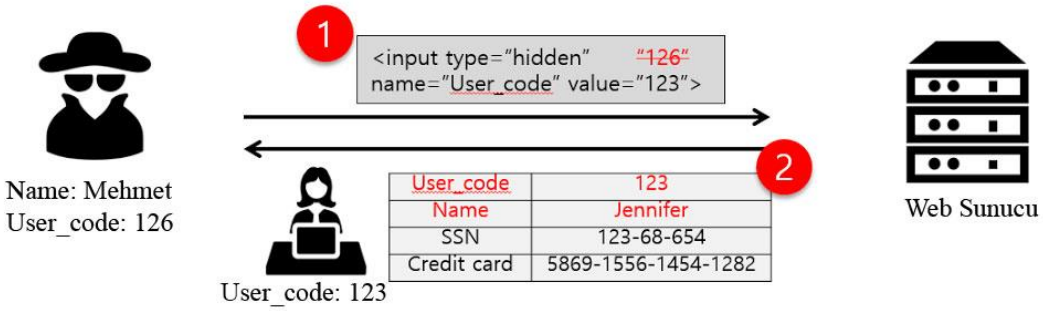
XXE dinamik olarak XML dokümanlarını dosya sistemi üzerinden veya uzaktan sisteme eklemek için kullanılan XML özelliklerinden kaynaklanan bir açıklıktır. Saldırganlar bu açıklık nedeni ile dosya sistemi üzerindeki önemli dosyalara erişim sağlayabilmekte, harici sistemlere bağlantı kurabilmekte ve/veya yerel ağ içindeki sistemlere bağlantı yapmak amacıyla kullanabilmektedirler (bkz. Şekil 2.4). XXE açıklığı, hedef sistemin tamamen ele geçirilmesi, servis dışı bırakılması ve yerel ağ kaynaklarına sızma girişimlerinde bulunulması için kullanılabilmektedir [8].



Şekil 2.4. XXE açıklığının sömürülmesi

Bozuk erişim kontrolü (Broken access control)

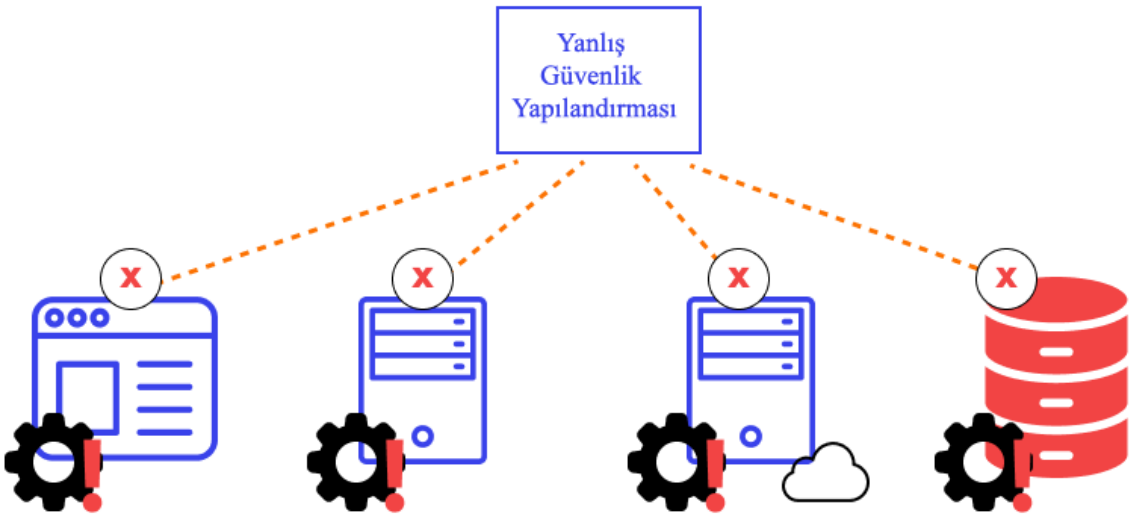
Kimliği doğrulanmış kullanıcıların yapmasına izin verilen işlemlerin ve kısıtlamaların genellikle doğru şekilde yapılandırılmaması sonucunda ortaya çıkmaktadır. Saldırganlar, diğer kullanıcıların hesaplarına erişmek, hassas dosyaları görüntülemek, diğer kullanıcıların verilerini değiştirmek, erişim haklarını değiştirmek vb. gibi yetkisiz işlevlere ve / veya verilere erişmek için bu açıklıkları kullanabilmektedir (bkz. Şekil 2.5) [4, 9, 10].



Şekil 2.5. Bozuk erişim kontrolü açıklığının sömürülmesi

Yanlış güvenlik yapılandırması (Security misconfiguration)

OWASP' a göre [4] güvenliğin yanlış yapılandırılması en çok görülen açıklıktır. Bu açıklık genellikle tamamlanmamış yapılandırma ayarlarının veya geçici yapılandırma ayarlarının kullanılması, açık bulut depolamalarının kullanılması, yanlış yapılandırılmış HTTP başlıklarının ve hassas bilgiler içeren ayrıntılı hata iletilerinin verilmesi gibi güvensiz olan varsayılan yapılandırmaların kullanılması sonucu ortaya çıkmaktadır (bkz. Şekil 2.6). Tüm işletim sisteminin yanı sıra, frameworkler, kütüphaneler ve uygulamalar güvenli bir şekilde yapılandırılmalı, patchler sürekli takip edilerek gerekli güncellenmeler gerçekleştirilmelidir [4, 11].

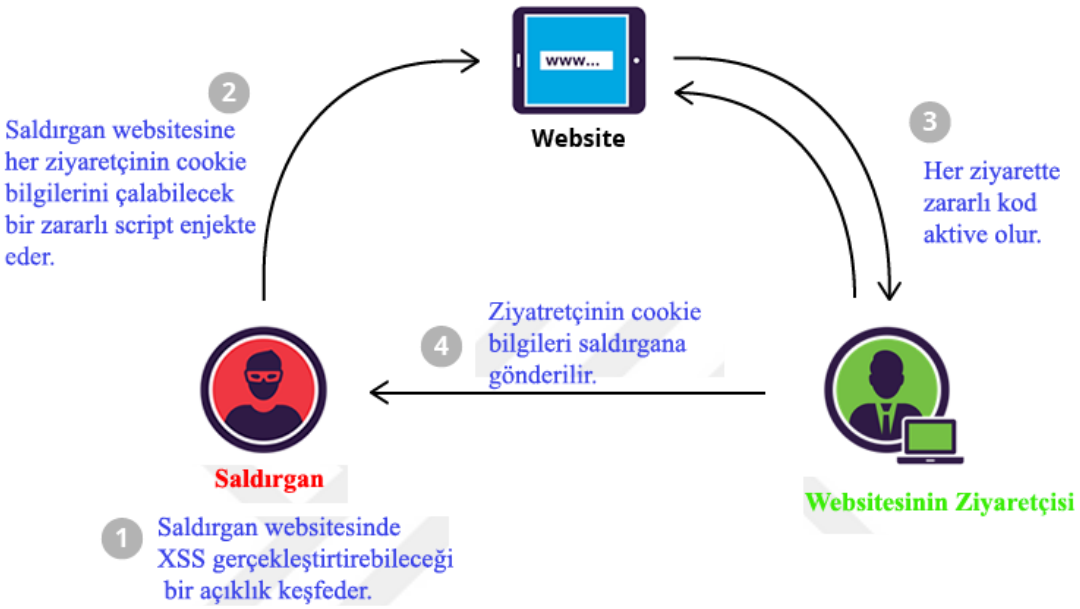


Şekil 2.6. Yanlış güvenlik yapılandırılması açıklığının sömürülmesi [4]

Siteler arası betik çalıştırma (Cross-site scripting, XSS)

XSS açıklığı bir web uygulamasına uygun doğrulama yapılmadan başka bir web sayfasından güvenilmeyen veriler getirildiğinde veya mevcut bir web sayfasının kullanıcı tarafından sağlanan JavaScript tabanlı verilerle güncellediği durumlarda ortaya çıkmaktadır [4, 12].

XSS açıklıkları saldırganların kurbanın tarayıcısındaki kullanıcı oturumlarını ele geçirmelerine, web sitelerinin görünüşlerini bozmalarına veya kullanıcıyı kötü amaçlı sitelere yönlendirebilecek komut dosyaları çalıştırmalarına imkân sağlamaktadır (bkz. Şekil 2.7) [4, 13].



Şekil 2.7. XSS açıklığının sömürülmesi

Güvensiz serileştirme (Insecure deserialization)

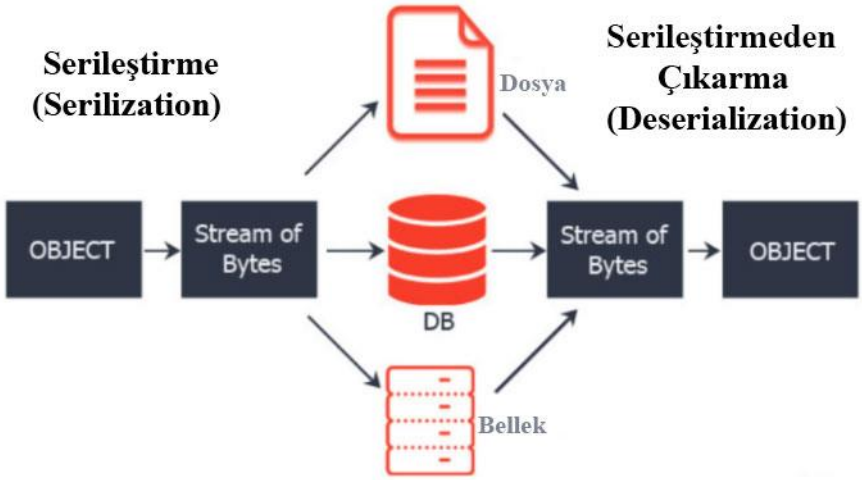
Güvensiz serileştirme genellikle uzaktan kod yürütülmesine yol açmaktadır. Güvensiz serileştirme açıklıkları uzaktan kod yürütme ile sonuçlanmasa bile, yeniden gönderme saldırıları, enjeksiyon saldırıları ve yetki yükseltme saldırıları dahil olmak üzere farklı saldırıları gerçekleştirmek için kullanılabilmektedir [4, 14].

PHP tabanlı bir web forum uygulamasının PHP nesne serileştirme (object serialization) ile kullanıcıya ilişkin kullanıcı id si, role, şifre, hash ve diğer bilgileri "süper" cookie ye kaydettiğini varsayalım. Bu durumda eğer serileştirme işlemi güvensiz şekilde gerçekleştirilirse, saldırgan Şekil 2.8' de göründüğü üzere yetki yükseltme saldırı gerçekleştirebilecektir.

a:4:(i:0;i:132;i:1; s:7:"Mehmet";i:2;s:4:"user"; i:3;
s:32:"b6a8b3bea87fe0e05022f8f3c88bc960";)

Saldırgan bu serileştirme işlemini kendisine admin yetkisi vermek için aşağıda gibi değiştirebilir:

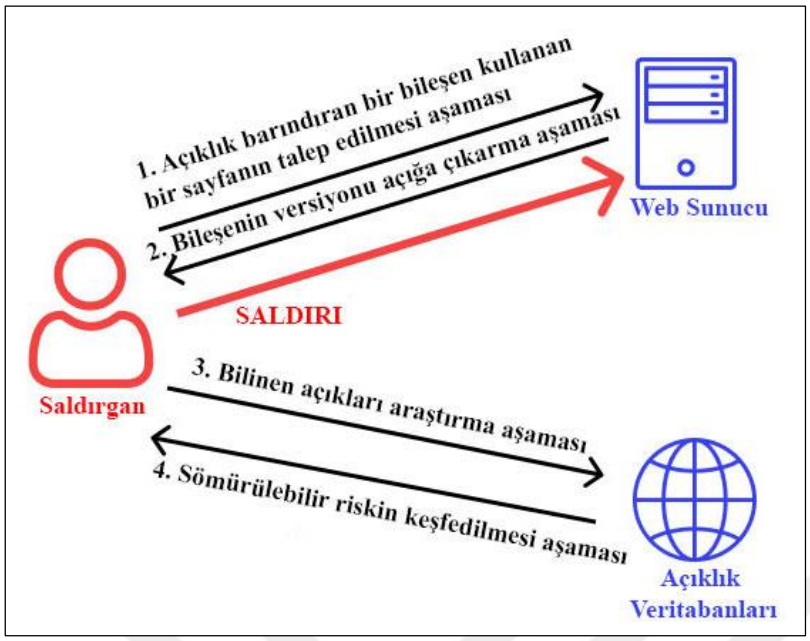
a:4:(i:0;i:1;i:1;s:5:"Yönetici";i:2;s:5:"admin"; i:3;
s:32:"b6a8b3bea87fe0e05022f8f3c88bc960";)



Şekil 2.8. Serileştirme ve Serileştirmeden çıkarma işlemleri

Bilinen açıklıkları içeren bileşenlerin kullanılması (Using components with known vulnerabilities)

Kütüphaneler, frameworkler ve diğer yazılım modülleri gibi bileşenler, web uygulaması ile aynı ayrıcalıklarla çalışmaktadır. Açıklık barındıran bir bileşenin kullanılması durumunda, veri çalınması veya sunucunun ele geçirilmesi gibi ciddi bir saldırının gerçekleştirilebilmesi mümkün hale gelmektedir [4, 15]. Bilinen güvenlik açıklarına sahip bileşenleri kullanan web uygulamaları, uygulamanın güvenliğini zayıflatabilmekte ve çeşitli saldırıların gerçekleştirilmesine imkân sağlayan açıklıkları etkinleştirebilmektedir [4, 16]. Bilinen açıklıkları içeren bir bileşen kullanılması durumunda bir web uygulamasına gerçekleştirilebilecek saldırının aşamaları Şekil 2.9'da görülmektedir.



Şekil 2.9. Bilinen açıklıkları içeren bileşenlerin kullanılması ile ortaya çıkan açıklıkların sömürülmesi

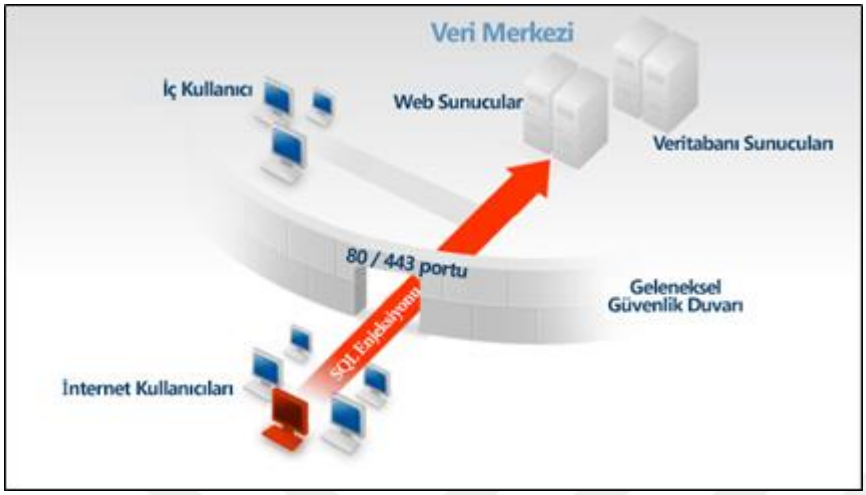
Yetersiz kayıt tutma ve izleme (Insufficient logging & monitoring)

Yetersiz kayıt tutma (logging) ve izleme (monitoring), yanlış entegre edilmiş veya hiç bulunmayan olay müdahale sistemi ile birleşince, saldırganlara ve saldırı sistemlerine, verileri ele geçirme, silme, değiştirme, kalıcı olarak müdahale etme hususlarında imkan sağlayabilmektedir. Çoğu ihlal çalışmasında, ihlali tespit etme süresi 200 günü aşmakta, bu tespit işleminin ise genellikle iç süreçler veya izleme sonucunda değil harici taraflarca tespit edildiği görülmektedir [4, 17].

2.1.2. Web uygulama güvenliğinin sağlanmasında kullanılan yaklaşımlar

Kurum içi ağlar (network) tasarlanırken ağın giriş/çıkış kısmına güvenlik duvarları (firewall) konumlandırılmaktadır. Farklı ağ katmanlarında (layer) çalışabilen güvenlik duvarları, 7. seviye olan uygulama katmanında hizmet veriyor olsa dahi, doğru ve yeterli yapılandırma ayarları yapılmadığı durumlarda iç ağı dışardan gelen saldırılara karşı yeteri kadar koruyamayacaktır. Kurumlar genel olarak klasik güvenlik duvarlarında OSI 4. katmanda güvenlik kuralları yazarak, kaynak IP, hedef IP, port numarası ve paket türü (TCP veya UDP) özelliklerini kullanarak güvenlik kuralları oluşturmaktadırlar. Bir web uygulaması son kullanıcılar ile TCP 80 (HTTP) veya 443 (HTTPS) portundan haberleşeceği için bu portlardan gelen trafiğe izin verilmesi gerekmektedir. Bu durumda web uygulamasında bulunan bir açıklık, güvenlik duvarı tarafından fark edilmeden kolayca sömürülebilmektedir

(bkz. Şekil 2.10). Web uygulamalarına yönelik gerçekleştirilen saldırıların geleneksel güvenlik duvarları ile engellenmesi mümkün değildir.



Şekil 2.10. Ağ güvenlik duvarının web saldırısı ile atlatılması [18]

Günümüzde yeni nesil güvenlik duvarları (Next Generation Firewalls) olarak adlandırılan ve OSI 7. seviye olan uygulama katmanında hizmet veren, port numaralarından ziyade protokol tabanlı (http, ftp, snmp, telnet vb.) kurallar oluşturulabilen güvenlik duvarları kullanılmaya başlanmış olmakla birlikte bu güvenlik duvarları da web uygulamalarının barındırdıkları açıklıklar karşısında yetersiz kalmaktadır. Web uygulamalarına yönelik tehditlerin engellenmesin de genel olarak aşağıdaki üç farklı yaklaşım kullanılmaktadır:

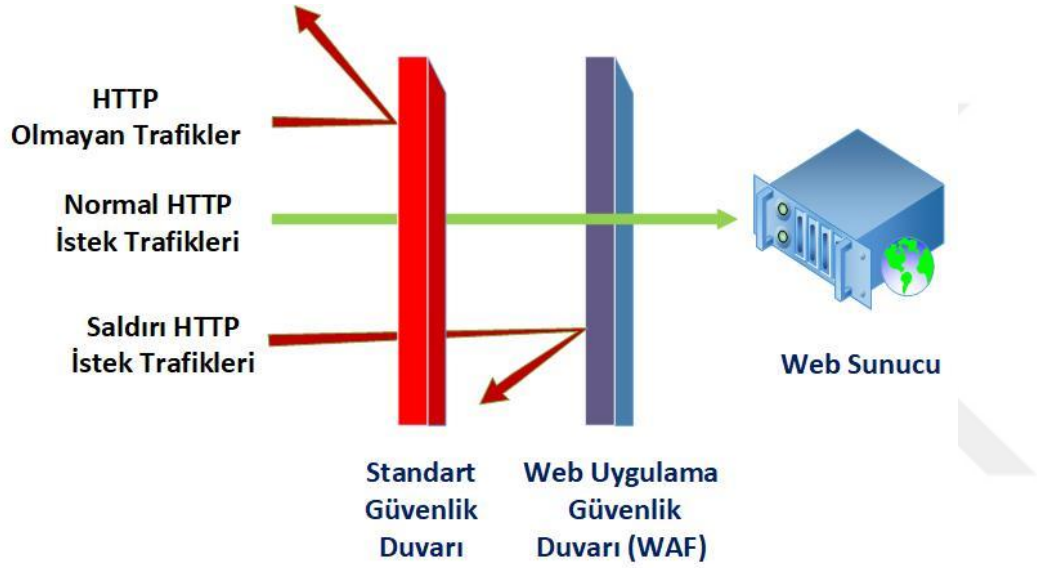
- Direkt yaklaşım: Web uygulamaları geliştirilirken direkt tümüyle güvenli olarak yazılımın geliştirilmesinin sağlanması

- Tespit edici yaklaşım: sıkılaştırma (penetration) testleri, karakutu (blackbox) ve beyaz kutu (whitebox) testleri, sistemde hataya sebep olabilecek verilerin gönderilmesine dayanan bulanıklaştırma (fuzzing) testleri ya da kodların statik/dinamik analizi ile web uygulamasında bulunan açıklıkların tespit edilmesinin sağlanması

- Önleyici yaklaşım: saldırı tespit sistemleri ve web uygulama güvenlik duvarları (WAF) ile web istekleri geldiği anda saldırıların tespit edilmesinin sağlanması

Bu yaklaşımların tek başına başarılı olması mümkün olmayıp, kritik web uygulamalarının güvenliğinin sağlanması için her üç yaklaşımında birlikte kullanılması gerekmektedir. Bununla birlikte, bu tezin ana konusunu önleyici bir yaklaşım olan WAF sistemleri

oluşturmaktadır. WAF sistemleri, ağ tabanlı saldırı tespit sistemlerinden (Intrusion Detection System - IDS) farklı olarak, web isteklerini derinlemesine analiz ederek, web saldırılarını tespit edebilmektedir. Web uygulamalarında bulunan açıklıkların, saldırganlar tarafından bulunması ve kullanılmasının önüne geçebilmek için web isteklerini daha detaylı ve derin analiz edip, web saldırılarını tespit ederek, engelleyebilen, sistem sorumlularına raporlayabilen WAF sistemlerinin kullanılması gerekmektedir (bkz. Şekil 2.11). Web isteklerinin karakteristik özellikleri genel olarak birbirine çok benzediği için web saldırı tespiti oldukça zor bir problemdir.



Şekil 2.11. Standart Firewall ile WAF arasındaki fark

WAF çalışma modelleri

WAF sistemlerinin geliştirilmesinde imza tabanlı ve anomali tabanlı olmak üzere başlıca iki yöntem kullanılmaktadır. İmza tabanlı yöntem, güvenlik firmaları tarafından en sık kullanılan yöntem olup, bilinen saldırı paternleri imza veritabanına kaydedilerek, gelen istek ve/veya dönen cevabın içeriğine bakılarak saldırı olup, olamadığına karar verilmektedir. Burada çözülmesi gereken önemli zorluklar ortaya çıkmaktadır. Bu yöntem her yeni çıkan saldırıya yönelik olarak imza veritabanın güncellenmesini gerektirdiği için maliyetli olup, güncel ve sofistike saldırılar karşısında yetersiz kalmaktadır. İmza veritabanın güncel olmaması veya yeni bir saldırı paterni ortaya çıkması durumunda saldırıların önlenmesi mümkün olmamaktadır [19]. Ayrıca imza tabanlı WAF sistemleri, doğrudan erişim (WAF girdi doğrulamasının atlatılması), dolaylı erişim (payload mutasyonu ve kodlama teknikleri) ve istisnaları keşfetme (farklı yük limitleri için ististanların bulunması) gibi yöntemler ile

bypass edilebilmektedir [20]. Başka bir zorluk ise alan uzmanın (domain expert), imza tabanlı web güvenlik duvarlarının yönetimi ve kuralların oluşturulması için sahip olması gereken yetkinlik düzeyidir. Yetkinlik düzeyi yeterli olmayan uzmanlardan kaynaklanan önemli yanlış yapılandırma ve hatalı güvenlik kuralı problemleri ile karşılaşılabilmektedir. İmza tabanlı WAF sistemlerindeki önemli problemlerden bir diğeri ise yüksek yanlış pozitif (False Pozitive -FP) oranıdır. Kritik sistemlerde tanımlanan sıkı güvenlik kuralları nedeniyle meşru (legitimate) kullanıcı istekleri de yüksek oranda engellenebilmektedir. FP oranının düşük tutulması sunulan hizmetin kalitesi ve kullanıcı memnuniyeti açısından önemlidir.

İmza tabanlı sistemler (Signature Based)

- Saldırı imza veri tabanında yer alan saldırı imzalarını kullanırlar.
- Sadece saldırı veri tabanında yer alan saldırıları engelleyebilirler.
- Yeni saldırıları engelleme şansları yoktur.

Beyaz liste modeli (Positive security model): İzin verilen belirli işlemler hariç her şeyin engellenmesine dayalı bir yaklaşımdır. Sadece izin verilen objelerin belirlenmesi ve geri kalanların engellenmesini amaçlanmaktadır.

Kara liste modeli (Negative security model): Yasaklanmış belirli işlemler hariç her şeye izin verilmesine dayalı bir yaklaşımdır. Sadece yasaklanan objelerin belirlenmesi ve bunların dışındakilere izin verilmesine dayalı olarak çalışmaktadır.

Anomali tabanlı model (Anomaly-based model)

Gelen ve giden istekleri inceleyerek saldırı imzaları kullanmadan web uygulamalarına yönelik anomalileri göz önünde bulundurarak saldırı engellemektedir. Anomali tabanlı saldırı engelleme sistemleri herhangi bir saldırı imzası kullanmadıkları için bu sistemlerin yanlış alarm (false positive) üretme ihtimali vardır. Anomali tabanlı saldırı engelleme sistemleri ile mevcut saldırı türlerine benzer yeni saldırı türlerinin tespit edilmesini ve engellenmesi de mümkün hale gelmektedir.

Yukarıda açıklanan iki ana yöntemden türetilmiş dört farklı yaklaşıma dayalı olarak web uygulama güvenliğinin sağlanması ve saldırıların tespiti gerçekleştirilebilmektedir. Bunlar imza tabanlı (kötüye kullanım) [21-24], anomali tabanlı [25-29], politika tabanlı [30] ve hibrit saldırı tespit sistemleridir [31]. Bu dört WAF yaklaşımının avantajları, dezavantajları ve zorlukları Çizelge 2.2'de sunulmuştur.

Çizelge 2.2. WAF geliştirme yaklaşımlarının avantajları, dezavantajları ve zorlukları

Yaklaşımlar	Avantajları	Dezavantajları	Zorluklar
İmza tabanlı WAF [21-24]	Bilinen saldırıları tespit etmede çok etkili	Yeni ortaya çıkan saldırıları tespit edememesi; Kodlama tekniklerini kullanarak atlatma [29]	İmza veritabanının güncellenmesi
Anomali tabanlı WAF [25-29]	Bilinen ve yeni ortaya çıkan saldırıları tespit etmede etkilidir	Yüksek yanlış pozitif (FP) oranı; yüksek varyans (overfitting) ve yüksek önyargı (underfitting) [32]	Özellik çıkarma ve seçme; gerçek dünya için genelleme
Politika tabanlı WAF [30]	İmza ve anomali tabanlı sistemlerin dezavantajlarının ortadan kaldırılması	Politikaların yanlış sıralanmasından kaynaklanan güvenlik açıklıkları [30]	Alan uzmanlarına olan gereksinim, Büyük ölçekli sistemlerde politikaların işletilmesindeki zorluklar
Hibrit WAF sistemleri [31]	İmza tabanlı ve anomali tabanlı sistemlerin gücünü birleştirmek	İmza tabanlı ve anomali tabanlı sistemlerin tüm dezavantajlarını içerir	Politika kurallarının oluşturulması ve yönetilmesi için alan uzmanına olan gereksinim [31]

İmza tabanlı sistemler, web isteğinin içeriği ile imza tabanında yer alan saldırı vektörlerini karşılaştırarak, web isteğin saldırı içerip içermediğini belirlemektedir. Bu yöntemin başlıca zorluğu imza veritabanının sürekli güncellenmesi ihtiyacı olup, farklı kodlama (encoding) yöntemleri ve yeni çıkan saldırı atakları karşısında koruma sağlayamamaktadırlar.

Anomali tabanlı sistemler ise normal kullanıcılara ait web istekleri ya da saldırı istekleri veya her ikisi ile eğitilmiş yapay zeka algoritmalarına dayalı WAF sistemleri ile yeni gelecek web isteklerinin türünün tespit edilmesine olanak sağlanmaktadır. Anomali tabanlı sistemler mevcut saldırı türlerinden türetilmiş yeni saldırı tekniklerinin tespitinde imza tabanlı yaklaşımlara göre daha etkili çalışmaktadır. Bununla birlikte, anomali tabanlı yöntemlerin önemli dezavantajlarından birisi normal kullanıcı isteklerinin de saldırı olarak sınıflandırma (yüksek yanlış pozitif oranı) olasılıklarıdır. Ayrıca karakutu olarak nitelendirilen yapay zeka algoritmalarına dayalı anomali tabanlı WAF sistemlerinin kullanılması, WAF sisteminin nasıl çalıştığının, saldırı türüne nasıl karar verdiğinin şeffaf olarak belirlenememesi sebebiyle kritik web uygulamalarının gerçek zamanlı olarak korunmasında sorunlara yol açmaktadır.

Üçüncü yaklaşım olan politika tabanlı (policy-based) saldırı tespit sistemlerinde ise her iki yöntemin belirtilen dezavantajlarını ortadan kaldırmak amacıyla normal ve anomali durumlarının her biri için politikalar belirlenerek, bu politikaların sıralı olarak işletilmesi ile web güvenliği sağlanmaktadır. Bu yöntemin en önemli dezavantajı ise barındırılan web

uygulamalarına çok hâkim olan ve tüm web uygulamalarına göre ayrı ayrı politikaları oluşturup, ihtiyaca göre güncelleyebilecek uzmana olan gereksinimdir. Büyük ölçekli web uygulamalarında bu politikaların alan uzmanları tarafından oluşturulması ve birbiri ile çelişmeyecek sıralar ile işletilmesinde önemli problemler ortaya çıkarabilmektedir.

Dördüncü yaklaşımda ise imza tabanlı sistemler ile anomali tabanlı tespit sistemleri entegre hale getirilerek, her iki yaklaşımın sağladığı avantajlardan faydalanan güçlü hibrit sistemlerin oluşturulması amaçlanmaktadır. İyi kurgulanmış hibrit sistemler ile güçlü bir WAF sistemi oluşturabileceği gibi, her iki yaklaşımın dezavantajlarını birlikte barındıracakları için önemli problemler oluşabilecektir. Bu tezin ana yöntemini anomali tabanlı yaklaşım oluşturmaktadır.

2.2. İlgili Çalışmalar

Bu bölümde, tez kapsamında gerçekleştirilen çalışmalar ile ilişkili olan ve daha önce gerçekleştirilmiş akademik çalışmalar ele alınmaktadır. Mevcut makine öğrenimi tabanlı siber güvenlik çalışmalarına bakıldığında, ağ güvenliği (network security) ve kötü amaçlı yazılım (malware) tespitine odaklanan çalışmaların web güvenliği odaklı çalışmalardan daha çok olduğunu görülmektedir. Bununla birlikte, farklı geleneksel makine öğrenmesi algoritmaları ve derin öğrenme yöntemlerinin kullanıldığı web uygulaması güvenlik çalışmaları da bulunmaktadır. Bu kısımda, tez kapsamında gerçekleştirilen çalışmalar ile ilişkili olarak öncesinde gerçekleştirilmiş, geleneksel makine öğrenimine dayalı modelleri, derin öğrenme algoritmalarına dayalı modelleri ve açıklanabilir yapay zeka tabanlı anomali tespit modellerini içeren bazı önemli web güvenliği çalışmaları kısaca açıklanmakta ve tartışılmaktadır.

2.2.1. Geleneksel makine öğrenmesine dayalı çalışmalar

Anomali tabanlı web saldırı tespiti, web güvenliği araştırmaları açısında önemli bir ilgi alanıdır. Bu alanda yapılan ilk çalışmalardan birisi 2003 yılında Kruegel ve Vigna tarafından Markov modelleri kullanılarak gerçekleştirilmiştir [24]. Çalışma, literatürde anomali tabanlı web saldırı tespiti üzerine yapılan ilk çalışma olması ve web saldırılarında kullanılan özniteliklerin çıkarılmasına odaklanması nedeniyle önemini korumaktadır. Liu ve diğerleri [33], özellik çıkarımı için istatistiksel ve manuel teknikleri kullanarak, özellik analizine ve SVM algoritmasına dayalı bir web saldırı tespit modeli geliştirdiler. Önerdikleri model, CSIC-HTTP veri setinde, iki sınıflı sınıflandırma problemi için %95,8 tespit oranı ulaşmıştır.

Thang [34] tarafından web isteklerini düzenli ifadeler (Regex) aracılığıyla filtreleyerek enjeksiyon saldırılarını (SQLi, XSS, vb.) algılayabilen ve ardından CSIC-HTTP veriseti üzerinde eğitilmiş bir Rastgele Orman (RF) modeli kullanarak bunları sınıflandırabilen bir WAF modeli geliştirilmiştir. Zhou ve Wang [35] tarafından, yeni ortaya çıkan XSS saldırılarını tespit etmek için kullanılabilecek güncel XSS saldırı paternlerine dayalı tehdit istihbarat süreci ile Bayes Ağı tabanlı sınıflandırıcıyı birleştiren bir topluluk modeli tasarlamışlarıdır. Kar ve diğerleri [36], bir destek vektör makinesini (SVM) eğitmek için SQL sorgularını graph yapısının düğümleri olarak modelleyerek, SQL enjeksiyon saldırılarını tespit etmek için yeni bir model geliştirdiler. İlgili çalışma kapsamında sonrasında, beş farklı web uygulamasından sorgular toplayarak bir veri seti oluşturarak modelleri eğittiler.

Nguyen ve arkadaşları tarafından biri öznitelik seçimine, diğeri ise farklı geleneksel makine öğrenmesi algoritmalarını kullanan bir topluluk öğrenme (ensemble learning) modelini içeren iki farklı web güvenlik çalışması yapılmıştır [25, 26]. Yazarlar öznitelik seçme çalışmalarında [26], genel özellik seçimine (GeFS) ve farklı makine öğrenimine dayalı WAF modelleri geliştirmişlerdir. Bu çalışmalarında [26], web isteğinden manuel olarak 30 farklı özelliği çıkarmış ve önerilen GeFS ile özellikleri azaltarak farklı modellerin performansını ölçülmüştür. Çalışmalarında [26], ECML-PKDD veri seti üzerinde iki sınıflı sınıflandırma yapan dört farklı geleneksel makine öğrenmesi algoritmasını kullanarak, GeFS ile özelliklerin %63' ü çıkarıldığı durumda ortalama %97,04 tespit oranı elde etmişlerdir. İkinci çalışmalarında [25], farklı IDS modellerinin çıktılarını ağırlık değerlerine göre birleştiren uyarlamalı öğrenmeye dayalı bir topluluk öğrenme (ensemble learning) WAF modeli (A-ExIDS) WAF modeli önermişlerdir. İlgili çalışmada temel öğrenici modeller olarak Naive Bayes, Bayes Ağı, Karar ağacı ve RBFNetwork algoritmaları kullanılmıştır. A-ExIDS yöntemi majör oylama ve Hedge/Boosting ensemble birleştirme yöntemlerinden daha yüksek başarı elde etmiş ve ECML/PKDD için iki sınıflı sınıflandırma da %92,56 tespit oranına ulaşılmıştır [25].

Benzer şekilde, Tama ve diğerleri [37], temel öğreniciler olarak makine öğrenimi algoritmaları kullanmak yerine farklı topluluk modellerinin kullanıldığı katmanlı bir ensemble WAF modeli geliştirdiler. Yazarlar, üç farklı topluluk modelinin çıktılarını birleştiren bir dizi topluluk modeliyle web anomalisi tespiti gerçekleştirdiler [37]. Yazarlar temel sınıflandırıcılar olarak rastgele orman, gradyan artırma ve XGBoosting topluluk

modellerini kullanmış ve çıktıları genelleştirilmiş doğrusal model (Generalized Linear Model, GLM) ile birleştirmişlerdir [37]. Bu çalışmanın sonuçlarını [37] incelendiğimizde, bazı temel modellerin performansının önerilen modele göre daha yüksek veya çok yakın olduğu ve önerilen modelin önemli bir iyileştirme sağlayamadığı görülmüştür. Ayrıca, temel öğreniciler olarak kullanılan modellerden bazıları doğrusal olmasa da, çıktılar doğrusal bir modelle birleştirildiğinden önerilen WAF modelinin genelleştirebilme gücünün zayıf olacağını düşünüyoruz.

Luo ve arkadaşları [38], web isteklerindeki yüklere dayalı olarak XGBoosting ve pozitif etiketsiz (positive-unlabelled, PU) öğrenme modellerini sırasıyla birleştiren bir yöntem önerdiler. Yazarlar, önce yükleri ASCII değerlerine göre vektörleştirerek ensemble modeline göndermekte ve ensemble modelin normal olarak sınıflandırdığı istekleri PU öğrenme modeline göndererek, bilinmeyen saldırıların tespit edilmesini amaçlamaktadırlar. Yazarlar [38] tarafından önerilen modelde sadece yükler dikkate alındığından, dizin dolaşma veya XXE gibi web isteklerinin farklı bölümlerindeki saldırılar tespit edilememektedir.

Yukarıdaki çalışmaları detaylı olarak analiz ettiğimizde, geleneksel makine öğrenimi algoritmalarına dayalı WAF modellerinin geliştirilmesinde bazı büyük zorluklar olduğunu görülmektedir. Bu kapsamda en önemli zorluklardan birisi özellik seçim sürecidir. Geleneksel makine öğrenimi algoritmalarının başarısının, özellik seçim sürecine bağlı olduğu ve HTTP isteklerinden özelliklerin çıkarılması ve seçilmesinde alan bilgisinin önemli hale geldiği görülmektedir [37]. Özellik seçim işlem adımında, eğer alakasız veya gereksiz özelliklerin barındırıldığı bir seçim yapılırsa, aşırı öğrenme (yüksek varyans) veya az öğrenme (yüksek bias) meydana gelebilmektedir. Derin öğrenme algoritmaları, özellik seçimi olmadan büyük miktarda veriden oluşan veri kümelerini kullanarak sağlam ve etkili sınıflandırma modelleri oluşturabilmektedir [39]. Böylece, derin öğrenme algoritmaları veri önişleme adımında bir özellik seçim sürecine ihtiyaç duymadıkları için bu zorlukları (diğer sebepler hariç olmak üzere) önemli ölçüde çözebilmektedir.

2.2.2. Derin öğrenmeye dayalı çalışmalar

Siber güvenlik alanındaki çalışmalar incelendiğinde derin öğrenme yaklaşımları kullanılmasına 2015 yılından itibaren başlandığı görülmekte olup, yeni ve güncel bir çalışma alanını oluşturmaktadır [40]. Derin öğrenme temelli bu çalışmalar web saldırı tespiti için birçok derin öğrenme yönteminin başarıyla kullanılabildiğini göstermektedir. Web güvenliği

alanında derin öğrenme yöntemlerinin uygulandığı çalışmaların çoğu 2017 yılından itibaren gerçekleştirilmiştir [41].

Literatürdeki derin öğrenmeye dayalı çalışmalar incelendiğinde, bazı çalışmalar da CNN tabanlı web saldırı tespit sistemlerinin önerildiği görülmektedir [39, 42, 43]. CNN algoritmasını ve dâhili kelime gömmeyi (word embedding) kullanan bazı çalışmalarda [39, 43], öznitelikler web isteklerindeki veri yüklerinden çıkarılırken, bazı çalışmalar da [42, 43] özniteliklerin URL bölümlerinden çıkarıldığı görülmektedir. Benzer şekilde, çalışma [44] deki yazarlar, word2vec kelime gömme yöntemini ve Bi-LSTM mimarisini kullanarak bir derin öğrenmeye dayalı WAF modeli önermektedirler. [39, 42-44] referansında yer alan dört çalışma da, CSIC-HTTP veri seti kullanılmakta ve sırasıyla %96.13 [39], %97.07 [43], %96.49 [42] ve %98.35 [42] tespit oranlarına eriştikleri görülmektedir. Bu durumda Bi-LSTM mimarisinin yapısı sayesinde web saldırılarını daha iyi temsil edebileceğini göstermektedir. Bizim ilk çalışmamızda da nihai model olarak önerilen Bi-LSTM mimarisinin yapısı ve başarılı çalışmasının nedenleri aşağıda ayrıntılı olarak açıklanmaktadır.

Vartonui ve diğerleri [29], katmanlı otomatik kodlayıcı ve DBN tabanlı özellik seçimine ve tek sınıf sınıflandırma algoritmalarına (one-class algorithm) dayalı iki sınıflı web anomali modelleri geliştirdiler. İlgili çalışmada, web isteklerinden özellik çıkarımı için N-gram kullanılmıştır. Yazarlar, ECML-PKDD veri seti üzerinde gerçekleştirilen model testlerinde iki sınıflı sınıflandırma için %84,13'lük bir tespit oranı elde ettiler [29]. Benzer bir başka web anomali çalışmasında [27], CNN ve DNN tabanlı tek algoritmaya dayalı derin öğrenme modellerinde kelime gömme yöntemini kullanılmasının sınıflandırma performanslarını arttırdığı gösterilmiştir. Ancak ilgili çalışmada [27] ECML-PKDD üzerinde yapılan testlerde saldırı türleri için ağırlıklı ortalama tespit oranları %84,0 olarak hesaplanmıştır. Buda tek bir modelin web saldırı türlerini sınıflandırmada yeterli başarıyı yakalayamayacağını göstermektedir.

Web güvenliği ile ilgili bazı çalışmalarda [35, 39, 44] sadece parametre değerlini (payload) kullanılırken, bazı çalışmalarda [34, 42, 45] sadece URL'ye dayalı özellikler kullanılmaktadır. GET yöntemi ile gönderilen web isteklerinde yer alan istemci yükleri URL içinde yer almakta ve uzunluk kısıtlamaları bulunmaktadır. Bununla birlikte POST yöntemi kullanılarak web sunucusuna gönderilen istemci yükleri, HTTP isteğinin gövdesinde yer almakta, bu nedenle URL görünmemekte ve sınırsız sayıda karakter gönderilebilmektedir.

Yetkilendirme verileri gibi hassas ya da uzun boyutlu girdiler gönderilirken POST metodu tercih edilmektedir. Bu sebeple, yalnızca URL tabanlı özelliklerle oluşturulan modeller kullanıldığında POST yönteminde yer alan parametre değerleri (payload) kullanılarak gönderilen web saldırılarını tespit etmek imkânsızdır; çünkü yükler URL'ye dâhil edilmemektedir. Buna ek olarak, URL ve payload özellikleri, web saldırıları hakkında çok önemli bilgiler içerse de, saldırı kalıpları isteğin diğer kısımlarında da bulunabilmektedir. Örneğin XXE, yetkilendirme kırma (broken authentication) veya oturum çalma (session hijacking) gibi bazı saldırı türlerini sadece URL ve parametre değeri çiftlerinden çıkarılan özelliklerle tespit etmek mümkün değildir. XXE veya shell script enjeksiyonu gibi dosya yükleme güvenlik açıklarından yararlanıldığı durumlarda, analiz için dosyaların içerik ve meta bilgilerine ihtiyaç bulunmaktadır. Kötü amaçlı dosyalar ile gerçekleştirilen işletim sistemi komut çalıştırma (RCE) saldırılarında genellikle dosyalar işletim sistemi komutları içermekte ve bu saldırılar dosya içeriğinin ayrıntılı bir analiziyle tespit edilebilir. Oturum çalma saldırılarında, istemcilerin web isteğinde yer alan oturum çerezleri (session cookies) çalınmaktadır. Sadece URL ve parametrelerin aldıkları istemci değerleri (payload) kullanılarak tüm bu saldırıları tespit edilmesi mümkün değildir. Web isteklerindeki tüm kısımların saldırı yüklerini içermesi mümkündür. Bu tez çalışması kapsamında web isteklerinin tamamı hem kelime tokenizasyonu hem de kelime gömme işlemlerinden geçirilerek HTTP isteklerindeki tüm özellikler kullanılmaktadır. Böylece URL ve payload bölümlerine ek olarak diğer özelliklerde bulunan saldırı vektörlerinin de sınıflandırmaya katkı vermesi sağlanmıştır.

2.2.3. Açıklanabilir yapay zeka (XAI) ile ilgili anomali tespit çalışmaları

Açıklanabilir yapay zeka alanı güncel çalışma alanlarından birisidir. Anomali tabanlı saldırı tespit çalışmalarına XAI' yı da dâhil eden az sayıda çalışma mevcuttur. Wang ve diğerleri [46] tarafından gerçekleştirilen çalışmada geleneksel makine öğrenme algoritmalarına dayalı saldırı tespit modellerinin SHAP ile açıklanması ele alınmıştır. İlgili çalışmada saldırı tiplerinin belirlenmesinde hem yapay sinir ağı tabanlı çok sınıflı modeller hem de One-Vs-All sınıflandırıcı modelleri geliştirilmiştir [46]. Çalışmaları kendi iddialarına göre siber güvenlik alanındaki ilk açıklanabilir makine öğrenmesi çalışması olup, bu yönü ile önemlidir [46].

Sarhan ve arkadaşları [47] nesnelerin internetine ilişkin (IoT) farklı ağ anomali verisetleri üzerinde eğitilen rastgele orman ve ileri beslemeli ağ (Deep Feed Forward, DFF) tabanlı

karakutu modelleri açıklayabileyen bir yapı önermişlerdir. İlgili çalışmada [47] açıklanabilirlik yöntemi olarak SHAP kullanılmıştır. Benzer şekilde çalışma [48]' da, yazarlar üç farklı IoT ağ anomali veriseti üzerinde eğittikleri karar ağacı (DT) ve rastgele orman algoritmalarına dayalı ensemble anomali modelini SHAP açıklanabilir yaklaşımına dayalı heatmaps ile açıklayan bir çalışma gerçekleştirmişlerdir.

Çalışmada [49]' de yazarlar anomali tabanlı IDS sistemlerine güveni artırmak amacıyla karar ağacına dayalı bir IDS modeli ve açıklanabilirlik yapısı önermişlerdir. İlgili çalışmada [49] KDD veriseti kullanılarak eğitilen basit karar ağacının düğüm ve yapraklarına dayalı kural çıkarımları ile model açıklanmaya çalışılmıştır. Çalışma [50]' de, yazarlar destek vektör makinesi (SVM), rastgele orman, ekstra ağaç ve naive bayes gibi geleneksel makine öğrenmesi algoritmaları ile geliştirdikleri ağ anomali tespit modellerini açıklayabilmek amacıyla özellik üreticiye dayalı bir açıklanabilirlik yaklaşımı önermişlerdir. Özelliklerin model bağımsız seçimi ve önemlilik analizleri, veriseti hakkında bilgi vermekte olup, modellerin çalışması ile dolaylı bağlantı bulunmaktadır. Verisetindeki yer alan özelliklerin kendi aralarındaki ve sınıf etiketi ile olan korelasyonu önemli olmakla birlikte, önem değerlerinin model tahminlerinden bağımsız analiz edilmesi modele güven açısından yeterli değildir.

Çalışma [51]' de yazarlar derin öğrenme tabanlı ağ trafik sınıflandırma modellerini açıklamak amacıyla genetik algoritmaya dayalı bir yapı önermişlerdir. İlgili çalışma da yazarlar sınıflandırıcı model olarak Resnet modeli kullanılmıştır. Çalışmada Resnet model çıktıları genetik algoritma yapısındaki kromozomlara aşılanarak (grafting), genetik algoritmanın baskın özellik seçimi yapmasına dayalı bir açıklanabilirlik yöntemi geliştirilmiştir [51].

Mariano ve diğerleri [52] yaptıkları çalışmada DNN modeline dayalı bir ağ anomali modelinin yanlış verdiği kararları düşmanca (adversarial) yaklaşımla açıklayan bir yöntem önermişlerdir. İlgili çalışmada NSL-KDD ağ anomali veriseti kullanılmıştır. Yazarlar DNN modelinin yanlış karar verdiği girdilerdeki (instance) özelliklerin değerlerini minimum sayıda değiştirerek doğru karara ulaşmaya çalışan bir düşmanca yöntem önermişlerdir [52]. Yanlış kararlara sebep olan özelliklerin görselleştirilmesine dayalı bir açıklanabilirlik yaklaşımı önerilmiştir [52]. İlgili çalışma [52] derin öğrenme tabanlı anomali modellerinin performansının açıklanabilirlik yaklaşımı ile iyileştirilebileceğini gösterdiği için önemlidir.

3. GAZI-HTTP WEB ANOMALİ VERİSETİNİN OLUŞTURULMASI

Literatür incelendiğinde web güvenliğine ilişkin az sayıda hazır veriseti [53, 54] bulunduğu, bunların da otomatize araçlar ile üretildikleri ve yeterli kalitede olmadıkları görülmüştür. Veriseti oluşturma işlemi meşakkatli ve zaman alan bir süreç olduğu için, araştırmacılar genel olarak hazır verisetlerini kullanma eğilimindedirler [55]. Mevcut veri setleri ise güncel saldırı paternlerini içermedikleri için eleştirilmekte, bu veri setleri kullanılması durumunda güncel saldırıları tespit edebilecek saldırı tespit sistemlerinin geliştirilmesi mümkün görünmemektedir [56-58].

Bu tezin önemli çıktılarından birisi web güvenliği alanına yeni, güncel ve benchmark olabilecek bir verisetinin kazandırılmasıdır. Bu kapsam da kurulan bal küpü sistemler ve kendi özgün yöntemlerimiz ile saldırı ve normal HTTP isteklerinden oluşan, web saldırılarının tespitinde kullanılabilecek güçlü ve stabil bir veriseti oluşturulmuştur. Tez kapsamında oluşturulan web anomali verisetinin, farklı formatlarda hazırlanarak tüm araştırmacıların genel kullanımına açılması planlanmaktadır. Böylece literatüre siber güvenlik alanında araştırma yapanların kullanabileceği güncel ve organik verilerden oluşan büyük bir hacme sahip, alanda kabul görecek ve kıyaslama amaçlı kullanılabilecek (benchmark) bir veriseti kazandırılmış olacaktır. GAZI-HTTP web anomali veriseti temel olarak 2 farklı yaklaşım ile oluşturulmuştur. Birinci yaklaşımda farklı web uygulamaları yerel sunucular üzerinde hazır hale getirilerek, toplanan saldırı payloadları ve normal web istek parametre değerleri ile bir veri kümesi elde edilmiştir. İkinci yaklaşımda ise aynı web uygulamaları genel erişime açık sunuculara kurularak balküpü sistemler (honeypots) hazırlanarak, dışardan gelen saldırı ve normal web istekleri kayıt altına alınmıştır. Her iki yaklaşımın detaylarına aşağıda detaylı olarak yer verilmektedir.

Tez kapsamında oluşturulan GAZI-HTTP isimli web anomali veriseti oluşturulması genel olarak 5 adımda gerçekleştirilmiş olup, aşağıda özet olarak sunulmaktadır.

1. **Adım:** Balküpü (Honeypot) sistemleri için gerekli olan web uygulamaları geliştirilmiştir. Bu kapsamda saldırganların dikkatini çekeceği düşünülen çeşitli alışveriş, kurumsal web sitesi, bitcoin borsası ve cüzdanı hizmeti benzeri web uygulamaları oluşturulmuştur. Yerel sunucuların yanı sıra, kiralanan web sunucusunun üzerine ilgili web uygulamalarının kurulumları ve yapılandırılmaları

gerçekleştirilmiştir. Web sunucusunun önünde DDOS güvenliği dışında saldırıları engelleyen bir güvenlik aracı konumlandırılmamıştır.

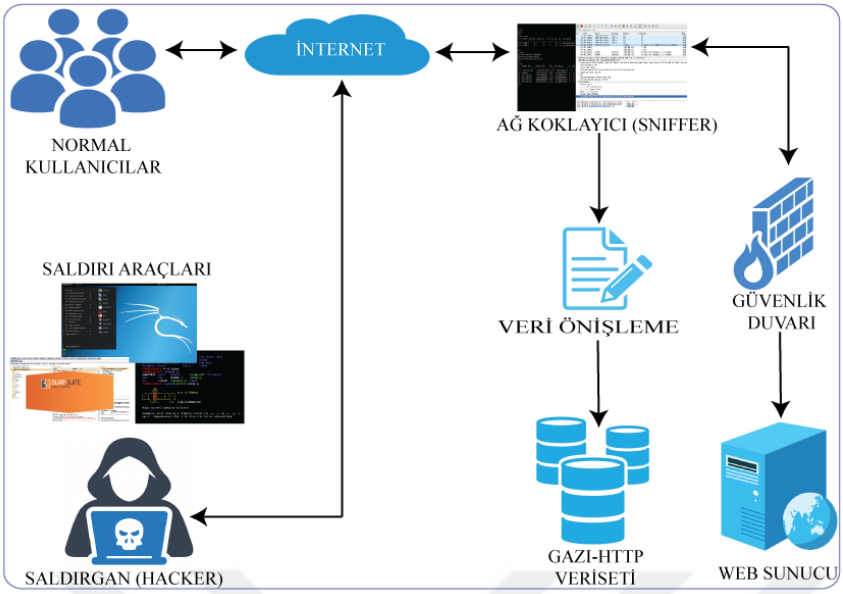
2. Adım: Bu adımda balküpü sistemlerinin kurulması ve yapılandırılması gerçekleştirilmiştir. İlgili web uygulamalarına gelecek web trafiklerinin, WAF sisteminde kullanılacak özellikler belirlenerek kaydedilmesi (loglama) ve raporlanması için gerekli koklayıcı sistemi (network snifer system) açık kaynak çözümler (opensource) kullanılarak, Linux temelli betikler (script) ile geliştirilmiş ve yapılandırılmıştır.

3. Adım: Yeterli miktarda verinin toplanmasının ardından, farklı web uygulamalarına gelen kayıtların birleştirilmesi gerçekleştirilmiştir. Bu kapsamda ilgili web isteğinin hangi web uygulamasına geldiğini belirten bilgiler eklenerek, tüm web uygulamalarına gelen web isteklerinin tek bir veritabanında birleştirilmesi işlemi gerçekleştirilmiştir. Bu işlemler için Python programlama dili kullanılmıştır.

4. Adım: Toplanan web trafiklerinin normal veya saldırı olarak etiketlenmesi ve saldırı tipinin belirlenmesi gerekmektedir. Bu aşamada yerel sunucular ile elde edilen web isteklerinin etiketleri aşağıda detayları açıklandığı üzere gönderilen yüklere (payloads) göre belirlenmektedir. Genel erişime açık bulut tabanlı web sunucuda yer alan web uygulamalarına gelen web isteklerinin etiketlenmesinde OWASP tarafından sunulan imza tabanlı ModSecurity WAF sisteminden yararlanılmıştır. Etiketleme sürecinin detaylarına aşağıda ayrıntılı olarak yer verilmiştir.

5. Adım: Bu adımda veri temizleme ve önişleme adımları gerçekleştirilmiştir. Veritabanında yer alan kayıtlardan tekrarlayan veriler silinmiş ve derin öğrenme için gerekli veri dönüştürme, normalizasyon ve önişleme adımları gerçekleştirilmiştir. Bu işlemlerin gerçekleştirilmesinde kullanılan yöntemlerin detayları aşağıda sunulan çalışmalarda açıklanmaktadır.

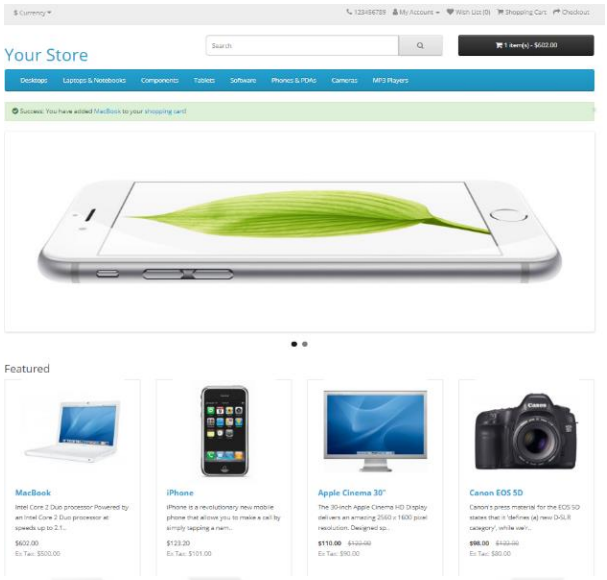
GAZI-HTTP web anomali verisetinin oluşturulmasında kullanılan genel yapı ve çalışma topolojisi Şekil 3.1’ de görülmektedir.



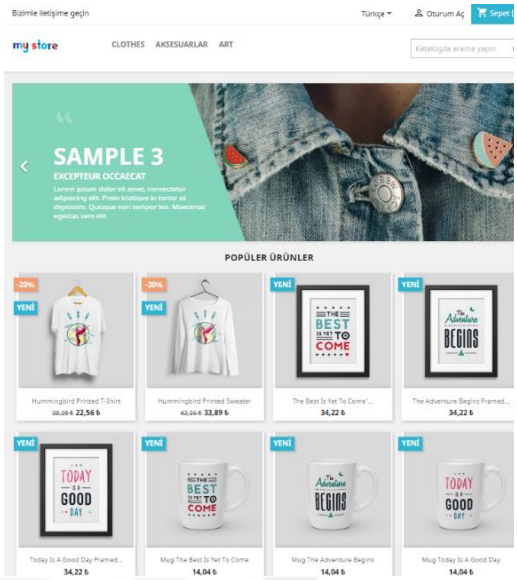
Şekil 3.1. GAZI-HTTP verisetinin oluşturulmasına ilişkin topoloji

3.1. Web Uygulamalarının Geliştirilmesi

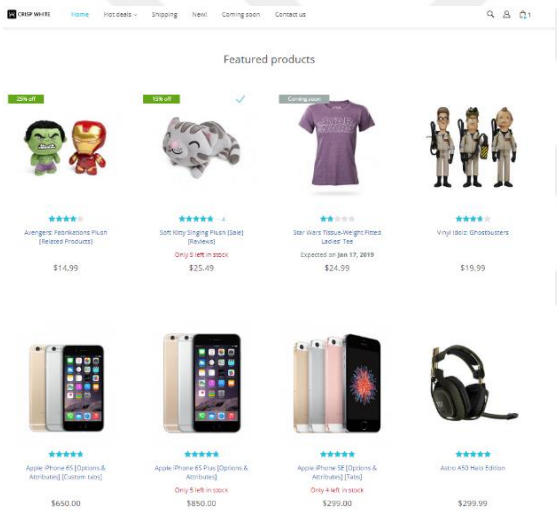
Tez kapsamında veri toplama amacıyla kullanmak üzere farklı web uygulamaları yerel ve bulut tabanlı sunucular üzerinde hazır hale getirilmiştir. Web uygulamaları balküpü (tuzak) sistemler olarak kullanılacağı için gerçek hayatta sık kullanılan e-ticaret siteleri, içerik yönetim sistemleri ve benzeri açık kaynaklı web uygulamaları ve bu uygulamalarda kullanılan pluginler araştırılmıştır. Bu kapsamda Opencart, Prestashop, Xcart açık kaynak kodlu e-ticaret (e-commerce) uygulamaları, Wordpress, Joomla ve Drupal içerik yönetim sistemleri (content managment system - cms) hem yerel sunucu hem de bulut tabanlı sunucular üzerinde kurularak, web uygulamalarının aktif görünmesini sağlayacak veri girişleri yapılmıştır. Oluşturulan verisetinin genel olması amacıyla tüm sistemler İngilizce olarak hazırlanmıştır. Sunucular üzerinde web uygulamalarının çalıştırılması için yine sık kullanılan Apache web sunucu sistemi ve Mysql veritabanı kullanılmıştır. Hazırlanan web uygulamalarına ilişkin arayüzler Şekil 3.2'de görülmektedir. Bu kapsamda 3 adet alışveriş web sitesi (bkz. Şekil 3.2. - (a,b,c)), wordpress tabanlı bir adet bitcoin borsası ve cüzdanı görünümlü web uygulaması (bkz. Şekil 3.2. - d), drupal tabanlı bir adet yemek tarifi web uygulaması (bkz. Şekil 3.2. - e) ve joomla tabanlı bir adet kurumsal firma web uygulaması (bkz. Şekil 3.12. - f) hazırlanmıştır.



(a) Opencart E-ticaret



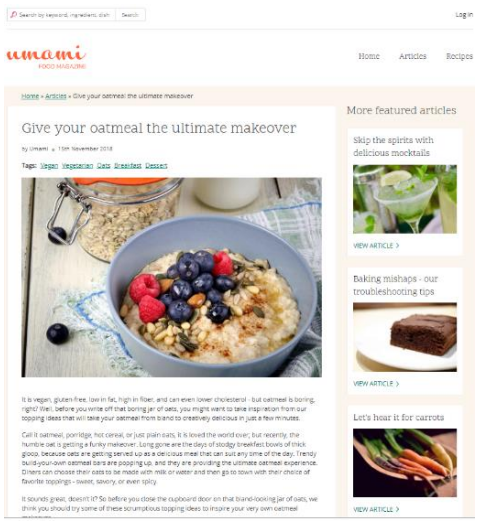
(b) PrestaShop E-ticaret



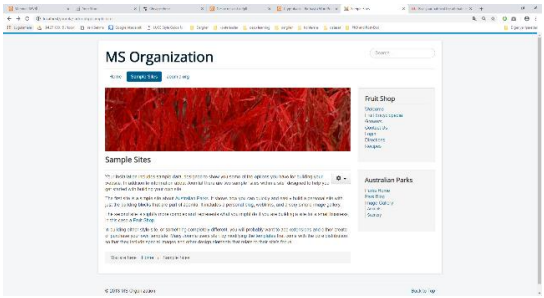
(c) Xcart E-ticaret



(d) Wordpress CMS



(e) Drupal CMS



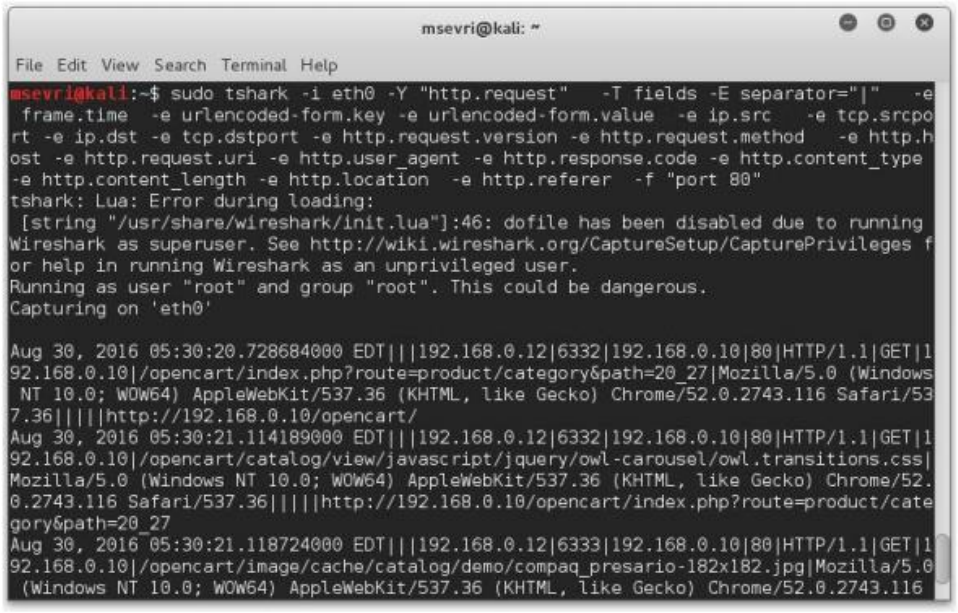
(f) Joomla CMS

Şekil 3.2. Geliştirilen web uygulamalarının arayüzleri

3.2. Balküpü (Honeypot) Sistemlerinin Kurulması

Web uygulamalarına gelen HTTP ve HTTPS isteklerini kaydetmek için bir ağ koklayıcı (sniffer) sisteminin oluşturulması gerekmektedir. Bu amaçla alanda en çok bilinen wireshark yazılımının komut tabanlı çalışan versiyonu olan tshark ağ analiz yazılımını kullanan bir shell script oluşturulmuştur. Tshark temelde tcpdump ile benzer şekilde çalışmakla birlikte tcpdump' a göre daha uyumlu (stateful) çalışmaktadır. Diğer bir özelliği ise uygulama katmanı seviyesinde analiz ve izleme yapmaya olanak sağlamasıdır. Tshark' ın protokol ve paket analizinde kullanılabilecek, işlevsel çok sayıda parametresi bulunmaktadır. Tshark' ın doğal dosya formatı, pcap dosya formatıdır. Web sunucu sistemine gelen tüm ağ trafiklerini dinleyerek, HTTP ve HTTPS isteklerini filtreleyen ve kütük (metin) dosyasına belirlenen özelliklerine göre kaydeden bir Shell betik (Shell script) geliştirilmiştir. Gelen tüm web istekleri, aşağıda açıklanan özellikleri ile birlikte kayıt altına alınmaktadır. Bir web isteğinin tshark ağ koklayıcı ile dinlenmesi Şekil 3.3'de görülmektedir. Tshark aracı ile bir web isteğinin dinlenmesi ve kaydedilmesi için kullanılabilecek örnek komut aşağıda yer almaktadır.

```
tshark -i eth0 -Y "http.request" -T fields -E separator="|" -e frame.time -e urlencoded-form.key -e urlencoded-form.value -e ip.src -e tcp.srcport -e ip.dst -e tcp.dstport -e http.request.version -e http.request.method -e http.host -e http.request.uri -e http.user_agent -e http.response.code -e http.content_type -e http.content_length -e http.location -e http.referer >> web_kayit.txt
```



Şekil 3.3. Tshark ağ izleme aracının web trafiğini izlemek için kullanımı

Oluşturulan Shell Script aracılığıyla web sunucusuna gelen ve kaydedilen web istekleri incelendiğinde, bir web trafiğin de yer alan parametreler genel olarak aşağıda yer almaktadır;

- ***Frame.time:*** Zaman bilgisi olup, isteğin web sunucusuna geldiği saat, dakika, mikro saniye cinsinden detaylı saniye bilgisinin yanı sıra gün, ay ve yıl bilgilerini içermektedir. Bu verinin kaydedilmesinin sebebi sınıflandırmanın yanı sıra raporlama esnasında saldırı olarak tespit edilen bir HTTP zaman verilerini gösterebilmektedir.

- ***Urlencoded-form.key:*** HTTP isteğinde POST metodu ile geçilen parametrelerin isimlerini göstermektedir. Bu özellik GET metodu ile geçilen parametrelerin isimlerini göstermemektedir. GET metodu ile geçilen parametreler ise aldıkları değerler ile birlikte URL'de içinde bulunmaktadır.

- ***Urlencoded-form.value:*** HTTP isteğinde POST metodu için parametrelere geçilen değerleri (payloads) göstermektedir.

- ***Ip.src:*** Web sunucuna gelen HTTP isteğinin kaynak IP adresini göstermektedir.

- ***Tcp.srcport:*** HTTP isteğinin kaynak port adresini göstermektedir.

- ***Ip.dst:*** HTTP isteğinin hedef IP adresini göstermektedir. Bu IP adresi aynı zamanda web sunucusunun adresidir.

- ***Tcp.dstport:*** HTTP isteğinin hedef port adresini göstermektedir.

- ***Http.request.version:*** HTTP isteğinin versiyonunu göstermektedir.

- ***Http.request.method:*** HTTP isteğinin metodunu göstermektedir. Bu özellik POST, GET, PUT, OPTIONS vb. değerleri almaktadır.

- ***Http.host:*** HTTP isteğinin hedefindeki web sitesinin host adresini içermektedir.

- ***Http.request.uri:*** HTTP isteğinin URI bilgisini, yani host adresinde itibaren yol bilgisini göstermektedir.

- ***Http.user_agent:*** İstekte bulunan kullanıcının veya saldırganın kullandığı internet tarayıcısının ve işletim sisteminin bilgilerini içermektedir.

- ***Http.content_type:*** HTTP isteğinin içeriğinin türünü belirtmektedir. Gelen istek URL bilgisi ve parametreler olabileceği gibi, resim, video içeren medya ya da word, pdf, txt vb. bir dosya da içerebilmektedir. Gelen isteğin içeriğinin türünü göstermektedir.

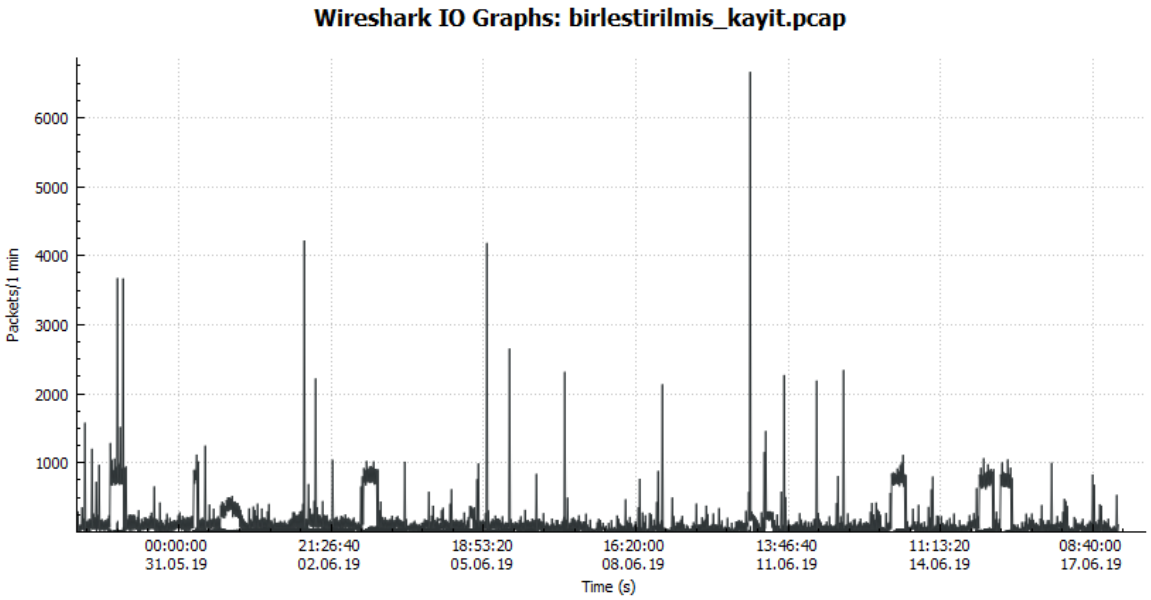
- ***Http.referer:*** Bu özellik ile gelen web isteğin önceki referans web sayfası belirtmektedir.

3.3. Bulut Tabanlı Sunucular İle Web Trafiğinin Toplanması ve Etiketlenmesi

Web isteklerinin yanı sıra sunucuya yönelik farklı ağ trafiklerini ve saldırıları tespit edebilmek amacıyla kiralanan sunucuya gelen ve giden tüm internet trafiği paketleri tcpdump network aracı ile pcap dosyası olarak ayrıca kayıt altına alınmıştır. Sunucunn tüm portları açık bırakılırken, sunucunun önüne DDOS engelleyici dışında aktif olarak engelleme yapan herhangi bir güvenlik duvarı konumlandırılmamıştır. Sunucu kurulduğu ve aktif edildiği ilk 10 dakika içinde ilk trafiğini almaya başlamış olup, port taraması, bazı port ve uygulama seviyesine yönelik kaba güç (brute force) şifre saldırılarının gerçekleştirildiği görülmüştür. İlgili web uygulamalarına farklı IP adreslerinden aşağıda detayları sunulan saldırıların geldiği görülmüştür.

3.3.1. Bulut tabanlı web sunucuya gelen trafiklerin analizi

Sunucu kurulduktan sonra birkaç saniye içinde ilk trafiğini sunucunun portlarını tarayan 185.235.244.244 IP adresinden ve Rusya üzerinden aldığı görülmüştür. İkinci trafik ise 1 dakika sonra, 164.132.98.75 IP adresinden ve Fransa üzerinden gelmiş olup, 22. porta (SSH portu) yönelik olarak tarama ve devamında SSH' a yönelik brute-force şifre saldırısı içerdiği görülmüştür. Sunucunun kurulumundan itibaren 18 günlük ağ trafik kayıtları incelendiğinde toplam 937 MB kayda ulaşıldığı ve sunucunun toplam 300021 (yaklaşık 300 bin) farklı IP adresi ile haberleştiği tespit edilmiştir. Bu trafiğin çoğunluğunu dünya üzerindeki yeni erişeme açılan sunucuları sürekli tarayan ve açık bulduğu portlar ile uygulama katmanı servislerine otomatize saldırılar gerçekleştiren sistemlerden geldiği görülmüştür. Bu durum, internete yeni açılan bir sunucunun ne kadar hızlı ve ne derece çok saldırıya maruz kalabileceğini göstermektedir. Bulut sisteminde kurulan web sunucusuna gelen ve giden network trafiğinin grafiği (18 günlük) Şekil 3.4' de görülmektedir.



Şekil 3.4. Sunucuya gelen ve giden ağ paketlerinin dağılımı (paket/saniye)

Toplanan trafiğin çoğunluğu 22. port üzerinden SSH protokolüne yönelik olarak gerçekleştirilmiştir. Web sunucusuna gelen trafiklere ilişkin olarak en çok paket gelen portlar Çizelge 3.1'de görülmektedir.

Çizelge 3.1. En çok trafik gelen portlar

SN	Port	Toplam Gelen Paket
1	22	1585365
2	80	198067
3	445	27190
4	23	10575
5	3389	2785
6	443	1117
7	8089	1701
8	8080	1259
9	81	2893
10	5555	905
11	8545	572
12	139	691
13	3128	409
14	5038	494

Sunucuya gelen HTTP trafiği (port 80) hem pcap dosyasına hem de tshark network aracılığıyla belirlenen özelliklere göre metin dosyasına kaydedilmiştir. 18 günlük süreçte toplam 68336 adet HTTP paketi alınmış ve gönderilmiştir. Web isteklerinde kullanılan

metotlar ve oranları Çizelge 3.2' de görülmektedir. Çizelge 3.3'de ise gelen isteklere dönen cevap kodlarının dağılımı görülmektedir.

Çizelge 3.2. Gelen HTTP isteklerindeki metotların dağılımı

Metot / Açıklama	**Sayı**	**Yüzde**
HTTP İstek Paketleri	**34324**	**%50.23**
▪ SEARCH	160	%0.47
▪ POST	10395	%30.28
▪ OPTIONS	1	%0.00
▪ HEAD	21	%0.06
▪ GET	23740	%69.16
▪ CONNECT	7	%0.02
HTTP Cevap Paketleri	**34012**	**%49.77**
Toplam Gelen / Giden HTTP Paketleri	**68336**	**%100**

Çizelge 3.3. Web sunucunun döndüğü cevapların dağılımı

Cevap Kodu / Açıklama	**Sayı**	**Yüzde**
5xx: Server Error	**5**	**%0.01**
▪ 500 Internal Server Error	5	%100
4xx: Client Error	**17508**	**%51.48**
▪ 405 Method Not Allowed	8	%0.05
▪ 404 Not Found	17122	%97.80
▪ 403 Forbidden	9	%0.05
▪ 400 Bad Request	369	%2.11
3xx: Redirection	**133**	**%0.39**
▪ 304 Not Modified	48	%36.09
▪ 302 Found	21	%15.79
▪ 301 Moved Permanently	64	%48.12
2xx: Success	**16366**	**%48.12**
▪ 206 Partial Content	15	%0.09
▪ 200 OK	16351	%99.91
HTTP Cevap Paketleri	**34012**	**%49.77**

3.3.2. Bulut tabanlı balküpü sistemine gelen web isteklerinin etiketlenmesi

Bulut tabanlı sistemlere gelen web isteklerinin normal veya saldırı (saldırı türü ile birlikte) olarak etiketlenmesi önemli bir zorluktur. Bu amaçla kullanılabilecek yöntemler araştırılmış ve literatürde sıklıkla iki yöntemin kullanıldığı görülmüştür. Bunlardan birisi, trafik

verilerinin uzman kişiler tarafından tek tek işaretlenmesi işlemi ile gerçekleştirilmesi iken, diğer yöntem bilinen bir saldırı tespit sistemi ile gelen trafiğin analiz edilmesi ve bu sistem tarafından üretilen alarmlara göre etiketleme işleminin gerçekleştirilmesidir. İlk yöntem maliyetli ve uzmanlık seviyesine bağlı olarak yanlış işaretlemelere açık olup, trafik miktarı çok fazla artığında gereken zaman ve iş gücü maliyeti buna paralel olarak katlanmaktadır. Derin öğrenme tekniklerinin uygulanabilmesi ve tüm olası saldırı paternlerinin barındırılması gerekmekte, buda çok fazla HTTP trafiği oluşturulmasını gerekmektedir. Bu sebeplerle, tez kapsamında ikinci yöntem tercih edilmiştir.

Web güvenliği konusunda otoritelerden birisi olarak kabul edilen OWASP tarafından geliştirilen açık kaynak WAF sistemi olan Modsecurity sistemi detaylı olarak incelenmiş ve etiketleme amaçlı olarak kullanılmasına karar verilmiştir. Modsecurity, kural tabanlı ve skorlamaya dayalı çalışan bir WAF sistemi olup, kural listeleri güncel olarak OWASP ModSecurity Core Rule Set (CRS) ismiyle açık olarak paylaşılmaktadır. Modsecurity Apache servisi ile birlikte çalışıp, gelen isteklere ve bunlara web sunucunun döndüğü cevaplara göre kural listesinden oluşturulan skorlar ile alarmlar üretebilmektedir. Mosecurity Mod Proxy modülü ile çalışıp, gelen tüm istekleri anlık olarak işleyebilmektedir. Modsecurity isteğe göre, saldırı isteklerinin tespit edildiği negatif güvenlik modeli ve sadece kabul edilen trafiğe izin verilen pozitif güvenlik modeli ile çalışabilmektedir. Modsecurity WAF sistemi ile aktif olarak saldırılar önlenerek raporlanabildiği gibi, monitör moduna alınarak sadece log kayıtları oluşturulabilmektedir. Tez kapsamında OWASP Modsecurity WAF sistemi monitör modunda kullanılmış ve belirlenen formatta özel log kayıtları oluşturulması sağlanmıştır. Modsecurity üzerine, CRS imza tabanı tüm listeleri ile kurulduğunda Çizelge 3.4'de görülen birçok bilinen saldırı türünü tespit edebilmektedir.

Çizelge 3.4. OWASP Modsecurity WAF ile tespit edilebilen saldırılar

SN	**Saldırı Türü**	**SN**	**Saldırı Türü**
1	SQL Injection (SQLi)	8	HTTPoxy
2	Cross Site Scripting (XSS)	9	Shellshock
3	Local File Inclusion (LFI)	10	Session Fixation
4	Remote File Inclusion (RFI)	11	Scanner Detection
5	Remote Code Execution (RCE)	12	Metadata/Error Leakages
6	PHP Code Injection	13	Project Honey Pot Blacklist
7	HTTP Protocol Violations	14	GeoIP Country Blocking

SN: Sıra Numarası

Modsecurity 4 farklı paranoya seviyesi ile kullanılabilmekte olup, kurulduğunda en düşük olan 1. seviye paranoya ile gelmekte ve tüm saldırı tiplerini tespit ederek engellemektedir. Bu paranoya seviyesi artırıldığında daha sıkı bir güvenlik sağlanırken, yanlış pozitif (FP) oranı çok artmaktadır. Bu tez çalışması kapsamında bilgisayar öğrenmesinin doğasında kaynaklı yüksek FP oranını azaltmak amacı ile 1. paranoya seviyesi kullanılmıştır.

Modsecurity sistemi birçok farklı kayıt (log) formatını desteklemekte ve böylece sistem yöneticisine bu hususta önemli esneklik ve istenilen formatta kayıt oluşturma olanağı sağlamaktadır. Modsecurity sistemi ile gelen bir web isteği ve bu istek için dönülen cevabın (response) barındırılmasına dayalı olarak tüm ağ trafiğinin detaylı olarak kaydedebilemesi mümkünken, sadece alarmların kaydedilmesi ya da trafiğin istenilen kısımlarının kaydedilebilmesi de mümkündür. Tüm gelen ve giden trafiğin kaydının alınması kayıt (log) dosyalarının aşırı büyümesine ve gecikmelere sebep olmaktadır. Bu tez çalışması kapsamında web sunucusuna gelen HTTP isteklerine ilişkin olarak aşağıda yer alan 4 temel özellik kayıt altına alınmıştır;

- Log başlığı (Audit log header),
- Gelen isteğin başlığı (Request Header),
- Gelen isteğin gövdesi (Request Body),
- Oluşturulan alarm ve skorlar (Audit log trailer)

Örnek bir web saldırı isteği kaydı aşağıda görülmektedir. Aşağıda yer alan gelen web isteği içerisinde görüleceği üzere GET metodunun payload kısmında yer alan **1' or '2'='2'** terimi kural tabanındaki SQL enjeksiyonu saldırı tipi ile eşleşmiş ve kritik seviyede alarm üretilmiştir. Ayrıca kayıt içerisinde zaman bilgisi, isteğin kendisi ve üretilen tüm alarmlar yer almaktadır. Tez çalışması kapsamında bulut tabanlı web sunuculara gelen web istekleri ve bu isteklere ilişkin Modsecurity kayıt dosyaları kullanılarak web saldırı istekleri için, saldırı tipi, alarmın kritiklik seviyesi gibi bilgilere dayalı bir veriseti oluşturulmuştur. Modsecurity tarafından alarm üretilmeyen web istekleri tshark ve tcpflow araçları kullanılarak ile kayıt altına alınmış ve normal olarak etiketlenmiştir.

--7d7e7f07-A--
[20/Dec/2019:15:24:47 +0000] Xfznv6G9FkzllmQKhCw-kwAAAAc 212.174.34.196 65038 167.172.235.195 80
--7d7e7f07-B--
GET /cart.php?target=main&widget=XLite%5CView%5CSlidebar&_=1576855486727 HTTP/1.1

Host: cheapesthere.xyz
Connection: keep-alive
*Accept: */**
DNT: 1
X-Requested-With: XMLHttpRequest
ajaxRefererTarget: search
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/79.0.3945.88 Safari/537.36
Content-Type: text/html
Referer: http://cheapesthere.xyz/cart.php?target=search&mode=search&substring=1%27+or+%272%27%3D%272%27
Accept-Encoding: gzip, deflate
Accept-Language: tr-TR,tr;q=0.9,en-US;q=0.8,en;q=0.7
Cookie: _ga=GA1.2.955054423.1573046340; ajs_group_id=null; ajs_user_id=%22sevri_27%40hotmail.com%22; NPS_14d74c65_last_seen=1573211427235; ajs_anonymous_id=%2203EAM6osaDF2cUQqm3f1ATsLALzyQcLV%22; __insp_wid=1795024724; __insp_nv=true; __insp_targlpt=WC1DYXJ0IG9ubGluZSBzdG9yZSBidWlsZGVyIC0gKllvdXIgY29tcGFueSBuYW1lKiA_IE9uYm9hcmRpbmcgV2l6YXJk; __insp_targlpu=aHR0cDovL2NoZWFwZXN0aGVyZS54eXovYWRtaW4ucGhwP3RhcmdldD1vbmJvYXJkaW5nX3dpemFyZA%3D%3D; __insp_norec_sess=true; SnapABugHistory=2#; __insp_slim=1573809384172; SnapABugUserAlias=%23; xid=vT5yfzDn4f8lyOL8fPRRRvTdg9Z4YOMc; viewedResources=a%3A5%3A%7Bi%3A0%3Bs%3A32%3A%22a70ab2e9abcbf777d9e7bca635a48010%22%3Bi%3A1%3Bs%3A32%3A%226ff6f4b88abf6f4ae6c5a0f7dcad2dde%22%3Bi%3A2%3Bs%3A32%3A%22b2ef8a0eba375306063ed071c9515986%22%3Bi%3A3%3Bs%3A32%3A%22119721f47c603862636e7619524fcd19%22%3Bi%3A4%3Bs%3A32%3A%22354cd3e7d25c29f54d8ad729738174df%22%3B%7D

--7d7e7f07-E--

--7d7e7f07-H--
Message: Warning. Match of "rx ^%(tx.allowed_request_content_type)$" against "TX:0" required. [file "/usr/share/modsecurity-crs/rules/REQUEST-920-PROTOCOL-ENFORCEMENT.conf"] [line "942"] [id "920420"] [msg "Request content type is not allowed by policy"] [data "text/html"] [severity "CRITICAL"] [ver "OWASP_CRS/3.2.0"] [tag "application-multi"] [tag "language-multi"] [tag "platform-multi"] [tag "attack-protocol"] [tag "paranoia-level/1"] [tag "OWASP_CRS"] [tag "OWASP_CRS/POLICY/CONTENT_TYPE_NOT_ALLOWED"] [tag "WASCTC/WASC-20"] [tag "OWASP_TOP_10/A1"] [tag "OWASP_AppSensor/EE2"] [tag "PCI/12.1"]
*Message: Warning. **detected SQLi** using libinjection with fingerprint 's&sos' [file "/usr/share/modsecurity-crs/rules/REQUEST-942-APPLICATION-ATTACK-SQLI.conf"] [line "68"] [id "942100"] [msg "SQL Injection Attack Detected via libinjection"] [data "Matched Data: s&sos found within REQUEST_HEADERS:Referer: http://cheapesthere.xyz/cart.php?target=search&mode=search&substring=**1' or '2'='2'**"] [severity "CRITICAL"] [ver "OWASP_CRS/3.2.0"] [tag "application-multi"] [tag "language-multi"] [tag "platform-multi"] [tag "attack-sqli"] [tag "paranoia-level/1"] [tag "OWASP_CRS"] [tag "**OWASP_CRS/WEB_ATTACK/SQL_INJECTION**"] [tag "WASCTC/WASC-19"] [tag "OWASP_TOP_10/A1"] [tag "OWASP_AppSensor/CIE1"] [tag "PCI/6.5.2"]*
*Message: Warning. Operator GE matched 5 at TX:anomaly_score. [file "/usr/share/modsecurity-crs/rules/REQUEST-949-BLOCKING-EVALUATION.conf"] [line "91"] [id "949110"] [msg "Inbound Anomaly Score Exceeded (**Total Score: 10**)"] [**severity "CRITICAL"**] [tag "application-multi"] [tag "language-multi"] [tag "platform-multi"] [tag "attack-generic"]*
Message: Warning. Operator GE matched 5 at TX:inbound_anomaly_score. [file "/usr/share/modsecurity-crs/rules/RESPONSE-980-CORRELATION.conf"] [line "86"] [id "980130"] [msg "Inbound Anomaly Score Exceeded (Total Inbound Score: 10 - SQLI=5,XSS=0,RFI=0,LFI=0,RCE=0,PHPI=0,HTTP=0,SESS=0): individual paranoia level scores: 10, 0, 0, 0"] [tag "event-correlation"]
Apache-Error: [file "apache2_util.c"] [line 273] [level 3] [client 212.174.34.196] ModSecurity: Warning. Match of "rx ^%(tx.allowed_request_content_type)$" against "TX:0" required. [file "/usr/share/modsecurity-crs/rules/REQUEST-920-PROTOCOL-ENFORCEMENT.conf"] [line "942"] [id "920420"] [msg "Request content type is not allowed by policy"] [data "text/html"] [severity "CRITICAL"] [ver "OWASP_CRS/3.2.0"] [tag "application-multi"] [tag "language-multi"] [tag "platform-multi"] [tag "attack-protocol"] [tag

"paranoia-level/1"] [tag "OWASP_CRS"] [tag "OWASP_CRS/POLICY/CONTENT_TYPE_NOT_ALLOWED"] [tag "WASCTC/WASC-20"] [tag "OWASP_TOP_10/A1"] [tag "OWASP_AppSensor/EE2"] [tag "PCI/12.1"] [hostname "cheapesthere.xyz"] [uri "/cart.php"] [unique_id "Xfznv6G9FkzllmQKhCw-kwAAAAc"]
Apache-Error: [file "apache2_util.c"] [line 273] [level 3] [client 212.174.34.196] ModSecurity: Warning. detected SQLi using libinjection with fingerprint 's&sos' [file "/usr/share/modsecurity-crs/rules/REQUEST-942-APPLICATION-ATTACK-SQLI.conf"] [line "68"] [id "942100"] [msg "SQL Injection Attack Detected via libinjection"] [data "Matched Data: s&sos found within REQUEST_HEADERS:Referer: http://cheapesthere.xyz/cart.php?target=search&mode=search&substring=1' or '2'='2'"] [severity "CRITICAL"] [ver "OWASP_CRS/3.2.0"] [tag "application-multi"] [tag "language-multi"] [tag "platform-multi"] [tag "attack-sqli"] [tag "paranoia-level/1"] [tag "OWASP_CRS"] [tag "OWASP_CRS/WEB_ATTACK/SQL_INJECTION"] [tag "WASCTC/WASC-19"] [tag "OWASP_TOP_10/A1"] [tag "OWASP_AppSensor/CIE1"] [tag "PCI/6.5.2"] [hostname "cheapesthere.xyz"] [uri "/cart.php"] [unique_id "Xfznv6G9FkzllmQKhCw-kwAAAAc"]
Apache-Error: [file "apache2_util.c"] [line 273] [level 3] [client 212.174.34.196] ModSecurity: Warning. Operator GE matched 5 at TX:anomaly_score. [file "/usr/share/modsecurity-crs/rules/REQUEST-949-BLOCKING-EVALUATION.conf"] [line "91"] [id "949110"] [msg "Inbound Anomaly Score Exceeded (Total Score: 10)"] [severity "CRITICAL"] [tag "application-multi"] [tag "language-multi"] [tag "platform-multi"] [tag "attack-generic"] [hostname "cheapesthere.xyz"] [uri "/cart.php"] [unique_id "Xfznv6G9FkzllmQKhCw-kwAAAAc"]
Apache-Error: [file "apache2_util.c"] [line 273] [level 3] [client 212.174.34.196] ModSecurity: Warning. Operator GE matched 5 at TX:inbound_anomaly_score. [file "/usr/share/modsecurity-crs/rules/RESPONSE-980-CORRELATION.conf"] [line "86"] [id "980130"] [msg "Inbound Anomaly Score Exceeded (Total Inbound ***Score: 10 - SQLI=5,XSS=0,RFI=0,LFI=0,RCE=0,PHPI=0,HTTP=0,SESS=0****): individual paranoia level scores: 10, 0, 0, 0"] [tag "event-correlation"] [hostname "cheapesthere.xyz"] [uri "/cart.php"] [unique_id "Xfznv6G9FkzllmQKhCw-kwAAAAc"]*
Apache-Handler: application/x-httpd-php
Stopwatch: 1576855487641420 54806 (- - -)
Stopwatch2: 1576855487641420 54806; combined=12850, p1=637, p2=11793, p3=79, p4=174, p5=167, sr=67, sw=0, l=0, gc=0
Response-Body-Transformed: Dechunked
Producer: ModSecurity for Apache/2.9.2 (http://www.modsecurity.org/); OWASP_CRS/3.2.0.
Server: Apache/2.4.29 (Ubuntu)
Engine-Mode: "DETECTION_ONLY"

--7d7e7f07-Z--

3.4. Yerel Sunucular ile HTTP Trafiklerinin Oluşturulması ve Etiketlenmesi

Bulut tabanlı ve dış erişime açık web sunucuya gelen trafiklerin genellikle benzer otomatize sistemler tarafından oluşturulduğu ve yüksek benzerliğe sahip web isteklerinin geldiği görülmüştür. Balküpü sistemler aracılığıyla toplanan web anomali verisetinin tüm saldırı yöntemlerini içermediği, manuel ve hedefli saldırılar ile sofistike saldırılar açısında yetersiz kaldığı gözlemlenmiştir. Bu nedenle bulut tabanlı sistemler ile yeterli, kapsayıcı ve genelleştirilebilir bir web anomali veriseti oluşturulmasının zor olduğu görülmüştür. Organik web anomali veriseti oluşturabilmek için büyük ölçekli ve gerçek hizmetler sunan, kritik web uygulama sistemlerinin kullanılması gerekmektedir. Gerçek web saldırı ve normal istek dağılımları ile saldırılarda kullanılan teknik ve parametreler, web uygulama güvenliği alanında çözüm sunan firmalar ve araştırmacılar tarafından yıllık olarak raporlanmaktadır. Web anomali verisetlerinin oluşturulduğu uygulamaların dışında farklı web uygulamalarının güvenliğinin sağlanmasında kullanılabilecek nitelikte olmaları

genelleştirilebilirlik güçlerini göstermektedir. Kapsamlı ve genelleştirebilir olan, web anomali trafiklerinin doğasına uygun bir veriseti oluşturmak amacı ile bulut sistemine kurulan web uygulamalarının birer kopyası yerel sunucu sistemlerine kurularak, manuel bir web anomali veriseti oluşturulmuştur. Bunu sağlamak için öncelikle kapsamlı bir tarama yapılarak aşağıda detayları açıklandığı üzere farklı saldırı tiplerine ilişkin saldırı yük ve paternleri toplanmıştır. Ayrıca normal web isteklerinin oluşturulması amacıyla da web isteklerinin aldığı değerlere uygun şekilde dünya da sık kullanılan ingilizce kelime ve değerler seti oluşturularak, web isteklerinin oluşturulmasında kullanılmıştır. Saldırı ve normal trafiklerin dağılımı, web anomali raporlarında açıklanan dağılıma uygun olarak gerçekleştirilmiştir. Ayrıca saldırı tiplerinin kendi içindeki dağılımları da gerçek web uygulama saldırı sayıları ile benzer şekilde oluşturulmuştur.

Yerel sunucular ile web anomali veriseti oluşturulması aşağıda maddeler halinde sunulan 3 adımda gerçekleştirilmiştir. Web saldırılarının ve normal web isteklerinin oluşturulmasında kullanılan paternlerin toplanması, web uygulamalarının aldıkları isteklerin belirlenmesi ve bu paternler kullanılarak normal ve saldırı web trafiklerinin oluşturulması ve kaydedilmesi aşamaları aşağıda detaylı olarak alt başlıklar halinde sunulmuştur.

- Saldırı ve normal web isteklerinin oluşturulmasında kullanılacak parametre değerlerinin oluşturulması
- Web uygulamalarının aldıkları isteklerin ve parametrelerin belirlenmesi
- Web uygulamalarına saldırı ve normal web isteklerinin gönderilmesi ve kaydedilmesi

3.4.1. Saldırı yüklerinin (payloads) toplanması

Saldırı web isteklerinin oluşturulmasında kullanmak amacıyla farklı saldırı türlerine ait saldırı yükleri (payload) kapsamlı bir çalışma ile toplanmış ve ön işlemden geçirilmiştir. Bu kapsamda farklı saldırı türlerine ilişkin olarak, özellikle Github olmak üzere birçok farklı kaynaktan yararlanılarak web saldırılarında kullanılan payloadlar toplanarak sınıflandırılmış ve saldırı türlerine göre birleştirilmesi sağlanarak, web saldırı payload verisetleri oluşturulmuştur. Bu amaçla farklı kaynaklardan açık şekilde paylaşılan birçok payload listesi ve açık kaynak saldırı araçlarının kullanıldığı payloadlar toplanarak, amaca yönelik düzenlenmiş ve kullanıma hazır olacak şekilde birleştirilmesi sağlanmıştır. Daha önce bu şekilde hazırlanmış bir payload veritabanına rastlanmamış olup, oluşturulan payload listeleri

çok kapsamlıdır. Tekrar eden payloadlar çıkarılmış, otomatize saldırı araçlarında kullanılabilecek şekilde metin tabanlı kütük dosyaları oluşturulmuştur. Toplanan web saldırı payloadlarının türleri ve dağılımları Çizelge 3.5'de görülmektedir. Bu payloadlar kullanılarak web saldırı araçları ile otomatize saldırılar gerçekleştirilmesi ve payload türüne göre etiketleme işlemilerin gerçekleştirilmesine ilişkin detaylar aşağıda açıklanmıştır. Bu tez çalışmasında benzer saldırı paternine sahip ve aynı yöntemin alt bileşenlerini içeren bazı saldırılar tek bir saldırı türü olarak birleştirilmiştir. Bu kapsamda dizin dolaşma (Directory Traversal), yerel dosya çağırma (Local File Inclusion - LFI), ve uzak dosya çağırma (Remote File Inclusion - RFI) saldırıları çok benzer saldırı paternleri içerdiği için Directory traversal ve File Inclusion olarak tek bir türde birleştirilmiştir.

Çizelge 3.5. Toplanan saldırı yüklerinin dağılımı

SN	Saldırı Türü	Payload Sayısı	Oran
1	Command Injection	9808	% 6.77
2	Carriage Return and Line Feed (CRLF)	107	%0.073
3	Directory Traversal - LFI - RFI	125400	%86.58
4	Open Redirect	875	%0.6
5	Sql Injection	3818	%2.6
6	Crosssite Scripting (XSS)	4550	%3.14
7	XML External Entity (XXE)	275	%0.19
Toplam		**144833**	**%100**

3.4.2. Web uygulamalarının taranması (spider)

Web uygulamalarına yönelik saldırı ve normal web isteklerinin oluşturulabilmesi için öncelikle web uygulamasında yer alan tüm sayfaların URL' leri, bu sayfaların aldığı istekler ile anahtar-payload değerlerinin belirlenmesi gerekmektedir. Bu işlem çeşitli tarama (spider) araçları ile gerçekleştirilebilmektedir. Spider araçları genel olarak kullanıcı tarafından inline modda konumlandırılmakta ve barındırdıkları proxy aracı ile gönderilen istekler için araya girerek bu isteklerden yola çıkarak tüm web sitesinin alabileceği olası tüm HTTP isteklerini ortaya çıkarmaktadır. Web uygulamalarında yer alan giriş (login) sayfalarına yönelik kullanıcı adı şifrelerin gerekiyorsa spider araçlarına tanımlanması gerekmektedir. Bu çalışma kapsamında Paros ve Burp Suite araçları kullanılarak web uygulamalarındaki tüm sayfaların olası alabileceği tüm HTTP istekleri, parametreleri ile birlikte kayıt altına alınmıştır. Sniffer araçları ile kaydedilen web istekleri, normal web isteklerinin oluşturulmasında sadece parametre değerleri uygun değerler ile değiştirilerek kullanılırken, saldırı isteklerinin oluşturulmasında sayfa yolları ve kullanıcı girdi alanları saldırı türüne uygun olarak manipüle edilmiştir.

3.4.3. Otomatize saldırıların gerçekleştirilmesi

Saldırganlar tarafından saldırı araçları kullanılarak iki farklı yöntem ile saldırı işlemi gerçekleştirilmektedir. Bunlardan ilki yukarıda bahsedildiği gibi spider araçları tarafından web uygulamalarının haritalanması sonucunda aldıkları istekler ve parametreler kaydedildikten sonra, bu URL lere ve aldıkları girdi alanlarına yönelik önceden hazırlanan çeşitli saldırı türüne ait payloadların saldırı araçlarında manuel olarak belirlenen girdi alanlarına gönderilmesi yoluyla oluşturulmaktadır. Bu yöntemin başarılı olabilmesi için her web isteği için girdi alanlarının saldırıya özel olarak belirlenmesi gerekmektedir. Bu alan Sql injection veya XSS için kullanıcı girdi alanları iken, dizin dolaşma, LFI, RFI gibi saldırılar için path, olmaktadır. Bu işlem sonucunda etiketleme işlemi gönderilen istekte yer alan payload'un türüne göre kolayca gerçekleştirilebilmekte ve trafiğin saldırı barındırdığı kesinlikle bilinebilmektedir.

İkinci yöntem ise web uygulamalarına yönelik otomatize saldırı gerçekleştirebilen çeşitli araçların kullanılmasıdır. Bu araçlardan bazıları farklı saldırı türlerini desteklerken (Burp Suite, Netsparker, Acunetix vb.), bazı araçlar daha özelleşmiş ve tek bir saldırı türüne yönelik detaylı açıklık analizi ve saldırı gerçekleştirme gücüne sahiptir (Sqlmap, Sqlninja, BBQSQL, BeeF XSS Framework, XSS Scanner vb.). Bu yöntemde web uygulamasının aldığı istekler saldırı aracına verilerek, gerekiyorsa aldığı payload alanları işaretlenerek araçlarda yer alan yöntem ve payloadlar aracılığıyla saldırılar otomatik olarak gerçekleştirilmektedir. Bu yöntemde etiketleme işlemi kullanılan aracın yeteneklerine göre değişmektedir. Eğer tek bir saldırı türü destekleniyorsa ya da saldırı türleri belirlenerek işlem gerçekleştirilebiliyorsa buna göre etiketleme işlemi yapılmaktadır. Manuel payload ile gerçekleştirilen saldırılar ve saldırı anında saldırı türünün belli olduğu otomatize araçlar (Sqlmap, Xss-Strike vb.) ile üretilen trafikler için saldırı etiketinin belirlenmesi manuel olarak gerçekleştirilebilmektedir. Eğer saldırı türü belirlenemiyor ve bir araç ile birden fazla saldırı gerçekleştiriliyorsa etiketleme işlemi zorlaşmaktadır. Otomatize saldırı gerçekleştiren fakat saldırı türlerini o anki isteğe göre otomatik uyarlayan saldırı araçları (Acunetix, Netsparker vb.) kullanıldığında kaydedilen HTTP trafiklerinin etiketlemesinin manuel veya geliştirilecek bir yöntem ile uzman tarafından yapılması gerekmektedir. Bu problem yukarıda açıklanan bulut tabanlı sistemlere gelen web isteklerinin normal ve saldırı olarak etiketlenmesi, saldırı ise saldırı türünün belirlenmesinde kullanılan imza tabanlı OWASP Modsecurity WAF sisteminin kullanılması ile gerçekleştirilebilmektedir. Bununla birlikte, tez çalışması kapsamında etiketleme işleminin doğru ve hassas yapılabilmesi amacı ile yerel

sunuculara sadece manuel yolla oluşturulan payload listeleri ile üretilmiş web istekleri gönderilmiş olup, saldırı tipi öncesinde belirlenemeyen saldırı araçları kullanılmamıştır.

Web uygulamalarının aldığı tüm istek parametrelerine aynı payload listelerinin gönderilmesi ile oluşturulan web anomali verisetleri aşırı öğrenmeye sebep olabilmektedir. Günlük hayatta kullanılabilir bir WAF sisteminin oluşturulması için her uygulamaya diğerlerinden farklı saldırı ve normal parametre yüklerinin gönderilmesi gerekmektedir. Bu tez çalışmasında bu duruma dikkat edilmiştir. İlk aşamada çalışmalarda kullanılmak üzere hazırlanan web uygulamalarının aldıkları web istekleri taranarak, kaydedilmiştir. Aşırı öğrenmeyi önlemek ve daha gürbüz WAF sistemleri oluşturabilmek için gönderilen payload değerleri her bir web uygulamasında sadece bir parametre için tekil olarak belirlenerek, kullanılmıştır. Böylece aynı web uygulamasında yük değerlerinin sürekli tekrar etmesi ve buna dayalı olarak makine öğrenme algoritmalarının aşırı öğrenmesinin önüne geçilmiştir.

3.4.4. Normal web isteklerinin oluşturulması

Normal web isteklerinin oluşturulabilmesi için web uygulamalarının aldıkları web istekleri tek tek incelenerek, her bir web isteğinin aldığı parametrelere uygun farklı yük setleri (payloads) oluşturulmuştur. Normal isteklerin oluşturulabilmesi için web isteğinin aldığı her bir parametre ayrıntılı olarak ele alınıp, aldığı değere göre yük paternleri oluşturulmuştur. Normal isteklerin oluşturulmasında kullanılan payloadların toplanmasında saldırı yüklerinin toplanmasına benzer şekilde titiz bir ön çalışma yapılarak parametrelere uygun olarak dünya da sık kullanılan yük değerlerinin listeleri oluşturulmuştur. Bu kapsamda; eğer bir web isteğindeki iki parametre; isim ve soy isim çiftini girdi alıyorsa, dünyada en sık kullanılan isimlerin ve soy isimlerin listeleri çıkarılmış ve web anomalisinin doğasına uygun sayıda kombinasyonlar ile web istekleri üretilmiştir. Örneğin aşağıda yer alan web isteğinin içerdiği ve belli aralıkta tam sayı değer alan “id_order”, “id_customer” parametreleri için belli bir aralık oluşturularak (web anomali istatistikleri gözetilerek), gönderilecek yükler iteratif olarak belirlenmektedir. Bu işlem benzer şekilde şehir ismi, ülke ismi, kişi ismi ve soy isimi gibi alanlar içinde dünya da sık kullanılan değerler toplanarak gerçekleştirilmiştir. Web isteklerinde yer alan tüm parametrelere gelecek değerlerin (payloads) uygun ve anlamlı olmasına dikkat edilmiştir.

Burp Suite aracı ile web istekleri ve alabileceği farklı payload değerleri ile web sunucularına otomatize web istekleri gönderilmiştir. Web uygulamasına gelen istekler web sunucusu üzerinde oluşturulan tcpflow aracına dayalı script ile kayıt altına alınmıştır. Tcpflow aracı

HTTP protokolü için çok sayıda filtre ve parametre bulunduran hızlı ve etkili bir araçtır. Her isteği ilişkili olduğu diğer istekler ile birlikte isimlendirerek, zaman bilgisi, gönderici ve alıcı bilgileri ile birlikte ayrı ayrı istekler halinde kayıt altına almaktadır. Tcpflow ile kayıt altına alınan örnek bir web isteği aşağıda görülmektedir.

```
POST
/admin555jsxbct/index.php?controller=AdminOrders&vieworder=&id_order=7&token=f0f772a75302d
3984a060d63d40f8beb&token=f0f772a75302d3984a060d63d40f8beb HTTP/1.1
Host: 192.168.0.28
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:71.0) Gecko/20100101 Firefox/71.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: tr-TR,tr;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 91
Origin: http://192.168.0.28
Connection: close
Referer:
http://192.168.0.28/admin555jsxbct/index.php?controller=AdminOrders&vieworder=&id_order=7&tok
en=f0f772a75302d3984a060d63d40f8beb
Upgrade-Insecure-Requests: 1

order_message=speak&visibility=1&message=warn&id_order=15952&id_customer=421&submitMess
age=
```

Oluşturulan GAZI-HTTP verisetinde toplam 9 türde trafik bulunmakta olup, bunlardan birisi Normal etiketi iken, kalanlar ise 8 farklı saldırı türüne ilişkin etiketlerden oluşmaktadır. Saldırı türleri seçilirken, OWASP Top Ten' e göre en sık karşılaşılan ve en kritik oldukları bilinen saldırı türleri seçilmiştir. Bu kadar kapsamlı bir veriseti bilgimiz dâhilinde literatürde yer almamaktadır. Açık ve erişilebilir tüm ortamlarda yer alan saldırı yükleri birleştirilerek, geniş bir saldırı paterni havuzu oluşturulmuştur. GAZI-HTTP verisetinde yer alan sınıf etiketleri ve dağılımları Çizelge 3.6'da görülmektedir. Buna göre verisetinde toplam 108.544 adet web isteği bulunmaktadır. Verisetinde en çok "Normal" etiketli veri bulunmakta olup, toplam 73.528 adet normal etiketli web isteği bulunmaktadır. Saldırı trafiklerinde ise sırasıyla en çok 10.803 adet SQLi, 9.139 adet XSS bulunmakta olup, en az 1963 adet ile SSI saldırısı bulunmaktadır. Veriseti saldırı tespitinin doğasına uygun olarak dengesiz (imbalanced) dağılmaktadır. GAZI-HTTP verisetinde yer alan saldırı türleri ve normal web isteklerinin dağılımı, yukarıda açıklanan ve güncel saldırı verilerini yayınlayan güvenlik araştırma raporları dikkate alınarak oluşturulmuştur. Böylece GAZI-HTTP veriseti gerçek dünya en iyi şekilde simüle edilebilmektedir.

Çizelge 3.6. GAZI-HTTP verisetinin dağılımı

Sınıf	**Sayı**	**Dağılım (%)**
Normal	73528	67,74
CRLF (Carriage Return Line Feed Injection)	2610	2,4
Uzaktan Komut Çalıştırma Saldırısı (Remote Command Execution, RCE)	3235	2,98
Dizin Dolaşma ve Dosya Yükleme Saldırısı (Directory Travesal and Local File Inclusion, DT and LFI)	2885	2,66
Yeniden yönlendirme saldırısı (Open Redirect)	2725	2,51
SQL enjeksiyonu (SQL Injection, SQLi)	10803	9,95
SSI (Server Side Include)	1963	1,81
Siteler arası betik çalıştırma (Cross Site Scripting, XSS)	9139	8,42
XML harici varlık saldırısı (XML external entity attack, XXE)	1656	1,53
Toplam	**108544**	**100**

4. DERİN ÖĞRENME TABANLI WAF MODELLERİ İÇİN YENİ BİR VERİ ARTIRMA TEKNİĞİ GELİŞTİRİLMESİ: DA-SANA

Destek vektör makineleri (Support Vector Machine, SVM), karar ağaçları (Decision Tree, DT), Naive Bayes (NB), rastgele orman (Random Forest, RF), k-en yakın komşu (k-nearest neighbors, KNN) gibi farklı geleneksel (state of the art) makine öğrenme algoritmaları ile geliştirilmiş birçok WAF ve IDS çalışması bulunmaktadır [59, 60]. Makine öğrenmesi tabanlı çalışmalar kapsamında karşılaşılan en önemli zorluklardan birisi özellik seçme işlemidir. Web anomali tespitinde geleneksel makine öğrenme algoritmalarının başarısı önemli oranda özellik seçme işlemine bağlı olup, HTTP isteklerinden özelliklerin çıkarılması ve seçilmesinde ise alan yetkiliği önemli hale gelmektedir [19, 61]. Alakasız veya gerek olmayan özelliklerin seçimi sonucunda aşırı öğrenme (high variance) veya yetersiz öğrenme (high bias) ortaya çıkabilmektedir [62]. Anomali tabanlı saldırı tespit çalışmalarında olayın doğası gereği verisetleri genellikle dengesiz dağılmaktadır (yüksek sayıda normal, az sayıda anormal sınıf etiketli veri) [63]. Bu durumda düşük özgüllük (specificity) oranı ortaya çıkmakta olup, saldırıların tespit edilme oranı düşmektedir (yüksek FN oranı) [64].

Derin öğrenme algoritmaları, çok sayıda veriden oluşan etiketli ve etiketsiz veri setlerinde farklı dağılama sahip olsalar dahi, birbiri ile ilişikli verilerin sınıflandırılması veya kümelenmesinde güçlü ilişkiler içeren danışmalı ve danışmansız modeller oluşturabilmektedir [64]. Derin öğrenme yöntemleri; görüntü işleme, doğal dil işleme (Natural Language Processing, NLP), ses işleme gibi alanlarda kendisini kanıtlaması ile birlikte, siber güvenlik alanında da önemli bir yer edinmeye başlamıştır. Birçok ağ tabanlı saldırı tespit sistemi (N-IDS) ve web güvenliği çalışmasında farklı derin öğrenme yöntemleri kullanılmaya başlanmıştır [65]. Bununla birlikte; web güvenliği alanında derin öğrenme tekniklerinin kullanımına son yıllarda başlanılmış olup, az sayıda çalışmanın bulunduğu çok güncel bir araştırma alanını oluşturmaktadır. Derin öğrenmenin en önemli özelliklerden birisi gizli (latent) ve ayırt edici (distinguishing) özelliklerin ortaya çıkarılması ile güçlü (robust) modellerin oluşturulabilmesidir [66]. Güncel saldırı tespit sistemlerinde sıklıkla kullanılan algoritmalar; öz yinelemeli ağlar (Recurrent Neural Network (RNN), Long Short Term Memory (LSTM), derin sinir ağları (Deep Neural Network, DNN), derin inanç ağları (Deep Belief Network, DBN), evrişimsel sinir ağları, (Convolutional Neural Network, CNN), otokodlayıcılar (Autoencoders, AE), sınırlı Boltzman Makineleri (Restricted

Boltzmann Machine, RBM) olup, özellik seçimi ve sınıflandırma modellerinin oluşturulmasında başarı ile çalışmaktadırlar [59, 65]. Web uygulamalarına yönelik çalışmalar yapıları gereği NLP problemlerine daha çok benzediği için araştırmalarda RNN mimarisine dayalı modellerin daha çok kullanıldığı görülmektedir [59, 65].

Tez kapsamında gerçekleştirilen bu çalışmada web uygulamalarına yönelik saldırıları (iki sınıflı) ve saldırıların ait olduğu sınıfları (çok sınıflı) başarılı şekilde tespit edebilen, derin öğrenme tabanlı güçlü WAF modelleri geliştirilmiştir. Çalışma kapsamında geliştirilen derin öğrenme tabanlı WAF modellerinin geliştirilmesinde öncelikle, metin tabanlı HTTP istekleri kelime tokenizasyonu (word tokenization) ve kelime gömme (word embeding) işlemlerinde geçirildikten sonra, bizim tarafımızdan geliştirilen kendinden uyarlamalı, gürültü eklemeye dayalı veri artırım yöntemi (Data Augmentation technique based on the Self-Adapting Noise Adding, DA-SANA) ile veri miktarı artırılmıştır. İkici aşamada ise normal trafiği, saldırı trafiğinden ayırabilen (two-class) ve saldırı türlerini tespit edebilen (multi-class) modeller geliştirilmiştir. Çalışma kapsamında WAF modellerinin geliştirilmesinde; CNN, DNN, LSTM, geçitli tekrarlayan birim (Gated Recurrent Unit, GRU) ve çift yönlü uzun kısa süreli bellek (Bidirectional Long Short Term Memory, Bi-LSTM) mimarileri kullanılmıştır. Modellerin performansları, biri tez kapsamında oluşturulan GAZI-HTTP web anomali veriseti olmak üzere, üç farklı veri seti kullanılarak değerlendirilmiştir. Diğer iki veriseti ise web güvenliği alanında sık kullanılan, web anomali kıyaslama (benchmark) verisetleri olan CSIC-HTTP [67] ve ECML-PKDD-2007 [68] olup, böylece önerilen modelin performansı daha önce gerçekleştirilen benzer çalışmaların sonuçları kıyaslanabilmiştir. Tez kapsamında bu çalışmanın gerçekleştirilmesinin en önemli üç motivasyonu aşağıda maddeler halinde sunulmuştur.

- Web uygulaması güvenliği, yukarıda açıklandığı üzere kritik olup, acil ve çözülmesi gereken sorunları içermektedir. Önemli zorluklardan birisi de saldırı tekniklerinin her geçen gün artması, bu nedenle güncel çözüm yaklaşımlarına ihtiyaç duyulmasıdır.

- Yukarıda açıklandığı üzere, siber güvenlik alanındaki gerçekleştirilen çalışmaların çoğunu ağ tabanlı güvenlik çalışmaları oluşturmaktadır. Web güvenliği alanında yeterli sayıda çalışma mevcut olmayıp, önemli oranda eksiklik bulunmaktadır.

- Derin öğrenme yöntemlerinin benzer çalışma alanlarındaki etkileyici başarısı, bu çalışmanın yöntemini belirlerken önemli bir motivasyon kaynağı olmuştur.

Gerçekleştirilen çalışmanın web uygulama güvenlik alanına sağladığı başlıca katkılar aşağıda maddeler halinde sunulmaktadır.

- Web anomali alanına özel olarak geliştirilen DA-SANA isimli yeni ve güçlü bir veri artırma tekniği önerilmiştir. DA-SANA yönteminde gürültü katsayıları, tezin bu bölümünün ilerleyen kısımlarında ayrıntılı olarak açıklandığı üzere, saldırı isteklerinin istatistiksel olarak ayırt edici özelliklerine dayalı olarak belirlenmektedir. Uyarlanabilir gürültü eklemesi ile normal ve anormal HTTP trafikleri gizli uzayda (hidden space) birbirinden ayrılarak algılama oranları iyileştirilmektedir.

- Web saldırı tespiti için gerçek zamanlı olarak başarılı bir şekilde çalışabilen yeni WAF modelleri geliştirilmiştir. Bilgimiz dâhilinde, çalışmadaki model performans ölçümleri hem iki sınıflı, hem de çok sınıflı sınıflandırma için literatürdeki en iyi performans değerleri olarak öne çıkmaktadır.

- Bu çalışmada kullanılan kodlar ve önişlemden geçirilmiş veri kümeleri, tekrarlanabilirlik ve gelecekte benzer çalışmalara katkı sağlamak amacıyla herkese açık olarak paylaşılmıştır*.
  *https://github.com/mehmetsevri/Deep-Learning-based-WAF

Yukarıda bahsedilen çalışmalara bakıldığında, genellikle bir [35, 36, 69] veya iki tür [34, 45] saldırı (SQLi ve/veya XSS saldırıları) ile ilgilendikleri görülmektedir. Tez kapsamında gerçekleştirilen bu çalışmada SQLi, XXE, LFI, uzaktan komut çalıştırma (RCE), yeniden yönlendirme (Open Redirect), dosya ifşası, CRLF enjeksiyonu, XSS, SSI olmak üzere birçok saldırı türü ele alınmıştır. Çalışmamız, farklı saldırı türlerini kapsaması, veri ön işleme adımları, yeni bir veri artırma yönetimi içermesi, geliştirilen derin öğrenme modelleri ve yüksek sınıflandırma başarımı açısından diğer çalışmalardan ayrılmaktadır.

4.1. Ön Bilgiler

Bu bölümde çalışma kapsamında kullanılan öznitelik çıkarma yöntemi, kelime tokenizasyonu ve kelime gömme teknikleri hakkında ön bilgiler sunulmaktadır.

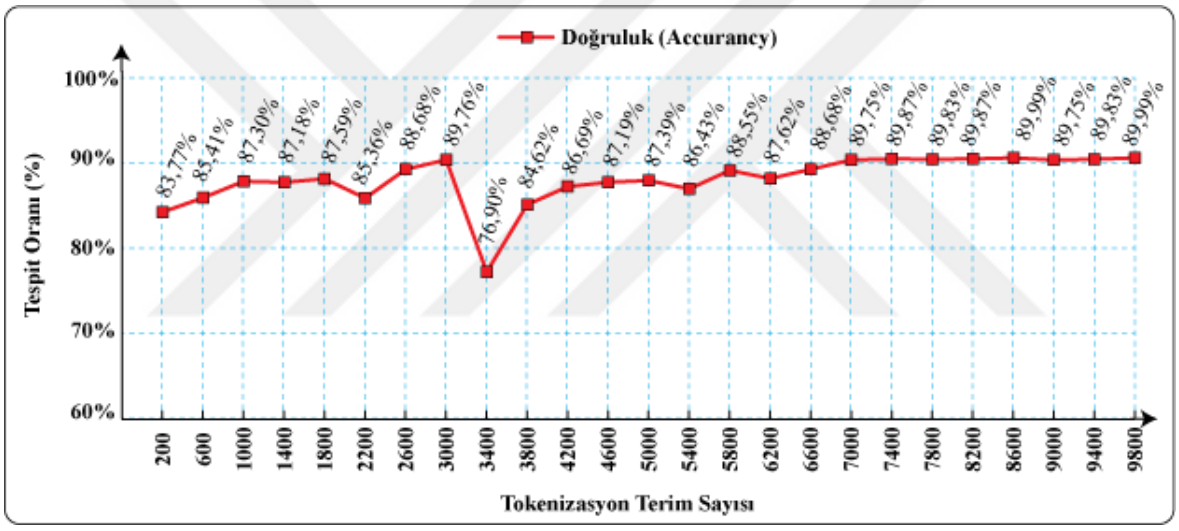
4.1.1. Öznitelik çıkarımı

Geleneksel makine öğrenimi algoritmalarında; ham verilerden özniteliklerin çıkarılması ve sınıflandırmada kullanılacak özniteliklerin seçilmesi olmak üzere üstesinden gelinmesi

gereken iki temel zorluk bulunmaktadır. Bununla birlikte, derin öğrenme gibi güncel yöntemlerde, öznitelik çıkarma ve öznitelik indirgeme gibi işlemler modellerin kendi içinde gömülü olarak gerçekleştirilmekte ve genellikle öznitelik işlemlerinden kaçınılmaktadır [70]. WAF sistemlerinin HTTP trafiğinin bir saldırı içerip içermediğine karar verebilmesi için isteğin içeriğine bakması gerekmektedir. Web istekleri metin tabanlı verilerden oluşmaktadır. Metin verilerinde kullanılan NLP yöntemleri, veri birleştirme, ön işleme ve veri ilişkilendirme süreçlerinde kullanılmaktadır. Web istekleri, alfa sayısal karakterlerin yanı sıra özel karakterler de içerebilmektedir. Veriler işlenirken web isteklerinde yer alan karakterlerin alt varyasyonları kelime grupları olarak ele alınmaktadır. Bazı çalışmalarda [39, 42, 43] sadece harfler dikkate alınmakta ve web isteklerindeki özel karakterler göz ardı edilmektedir. Bununla birlikte saldırı paternleri özel karakterler içerebilmektedir. Bu çalışmada özel karakter temizliği yapılmamış ve tüm karakterler dikkate alınmıştır. Veri setinde yer alan web isteklerindeki karakter gruplarının alt varyasyonlarının frekans ve ilişkilendirme analizine göre kelime tokenleri oluşturulmaktadır. Veri setinde en sık kullanılan kelimeler, tüm veri setindeki karakterlerin birbiri ile ilişkileri ve frekansları dikkate alınarak belirlenmektedir. Bu çalışmada kullanılan veri ilişkilendirme süreci, özel karakterleri de içeren saldırı paternlerini tespit etmede etkili bir şekilde çalışmaktadır. HTTP istekleri, "method", "protocol", "uri", "path", "cookie", "content-length", "content-type" ve “body” gibi metin tabanlı kısımlardan oluşmaktadır. LSTM gibi yapay sinir ağı mimarilerinde, metin veya kategorik veriler eğitim ağına doğrudan girdi olarak verilememekte; girdilerin sayısal vektörlere dönüştürülmeleri gerekmektedir. Bu süreç; istekteki kelime vektörlerinden tokenler oluşturmayı sağlayan kelime tokenizasyon süreci ve kelime vektörlerinin birliktelik ilişkisinin oluşturulmasında kullanılan kelime gömme süreci olmak üzere iki adım dan oluşmaktadır. Metin tabanlı verilerden kelime vektörleri oluşturmak için kullanılabilecek farklı kelime gömme yöntemleri (Word2vec, GloVe, fastText, one hot encoding vb.) bulunmaktadır.

Bu çalışma kapsamında, Keras word tokenizer ve Keras word embedding fonksiyonları kullanılarak metin tabanlı HTTP isteklerinden sayısal vektörler oluşturulmuştur. Tokenizasyon aşamasında, web isteklerinde olası tüm terim vektörleri oluşturularak en sık geçen terimler belirlendikten sonra kategorik tamsayı numaraları verilmektedir. Tokenizasyon işlemi sırasında, sınıflandırmanın başarısını sağlamak için kaç terimin yeterli olacağını belirlemek amacıyla değişen sayıda terim kullanılarak bir etkililik analizi yapılmaktadır. Tokenizasyon işleminde kullanılan terim sayısının GAZİ-HTTP veriseti

üzerinde sınıflandırma başarısına etkisi Şekil 4.1'de görülmektedir. Bu kapsamda tokenizasyon sürecinde en sık kullanılan 3000 terimin modelin öğrenmesi için yeterli olduğu tespit edilmiştir. Tüm isteklerin içerdiği token sayısı aynı olmadığı için, tokenizasyon sürecinden sonra oluşan vektörlerin boyutlarının eşit hale getirilmesi gerekmektedir. Bu amaçla çalışma kapsamında kullanılan post-padding yönetimde tokenlerden oluşan vektörlerin boyutlarının aynı olmasını sağlamak için vektörlerin sonuna sıfır eklenmektedir. Böylece, HTTP isteklerinde yer alan tüm olası saldırı kalıpları, terim uzunluğundan bağımsız olarak ele alınabilmektedir. Çalışma kapsamında kelime çantası (bag of words) tekniğine dayalı bir kelime gömme yöntemi kullanılmadığı için kelimeler arasındaki anlamlı ilişkiler tam olarak ele alınamamıştır. Bunun yerine, çalışma kapsamında karakter gruplarının istatistiksel ilişkilerine dayalı özellikler oluşturulmuştur.



Şekil 4.1. Tokenizasyon terim sayısının GAZI-HTTP veriseti üzerinde sınıflandırma başarımına etkisi

4.2. Kullanılan Derin Öğrenme Algoritmaları

Tez kapsamında gerçekleştirilen bu çalışmada geliştirilen DA-SANA veri artırma tekniğinin etkinliğini ölçmek ve kıyaslayabilmek için güncel derin öğrenme algoritmalarına dayalı web anomali tespit modelleri geliştirilmiştir. Performans kıyaslamasının yapılabilmesi için DA-SANA uygulanan ve uygulanmayan Bi-LSTM, LSTM, GRU, CNN ve DNN algoritmalarına dayalı farklı WAF modelleri geliştirilmiştir. Bu algoritmaların seçilmesindeki ana motivasyon, web istekleri gibi metin tabanlı verilerin işlenmesinde yoğunlukla kullanılmalarıdır. GAZI-HTTP veriseti ile geliştirdiğimiz farklı derin öğrenme algoritmalarına dayalı WAF sistemleri arasında en iyi performansı, Bi-LSTM mimarinse dayalı WAF modeli göstermiştir. Geliştirilen Bi-LSTM tabanlı çok sınıflı web anomali tespit

modeli ECML-PKDD ve GAZİ-HTTP verisetleri üzerinde en yüksek ortalama sınıflandırma başarısına ulaşmıştır. Bu nedenle çalışma kapsamında Bi-LSTM tabanlı bir WAF modeli önerilmiştir. Bi-LSTM mimarisinin genel yapısı ve Bi-LSTM'i temel alınarak geliştirilen ve çalışma kapsamında önerilen model aşağıda detaylı olarak sunulmuştur. Çalışmada kullanılan diğer derin öğrenme yöntemleri hakkında özet bilgiler aşağıdaki sunulmuştur.

GRU ve LSTM, RNN'de kaybolan gradyan (gradyan vanishing) problemini çözmek için geliştirilmiş iki mimaridir. İki mimari birbirlerine çok benzemekte ve ses işleme gibi alanlarda benzer model başarımları göstermektedirler [71, 72]. Bununla birlikte, GRU'nun yapısı LSTM den farklı olup, LSTM'in aksine GRU çıkış kapısı içermemekte ve dolasıyla LSTM' den daha az parametre içermektedir [71]. LSTM mimarisinin yapısı aşağıda ayrıntılı olarak sunulmaktadır.

DNN algoritması, ileri beslemeli bir yapay sinir ağıdır (feed-forward neural network, FFNN). DNN içerdiği çok sayıda katman ve nöron sayesinde (derinlik kavramı) doğrusal olmayan fonksiyonlar ile karmaşık problemleri başarıyla modelleyebilen en temel derin öğrenme algoritmasıdır. CNN mimarisi, konvolüsyonlar ve havuzlama süreçleri aracılığıyla sağlam genelleme ve temsil vektörleri üretebilmektedir [39]. CNN mimarisi, girdilerin özellik haritalarını oluşturan ve bunları bir sonraki katmana aktaran bir dizi evrişim katmanına sahiptir. Genellikle görüntü işleme alanında kullanılsa da web istekleri gibi metin işleme alanında da etkin bir şekilde kullanılmaktadır [42].

Çalışmada kullanılan Bi-LSTM tabanlı model mimarisi Şekil 4.2'de görülmektedir.



Şekil 4.2. Çalışmada önerilen Bi-LSTM mimarisinin yapısı

4.2.1. Çift yönlü kısa uzun zamanlı bellek mimarisi (Bidirectional Long Short Term Memory, Bi-LSTM)

Derin öğrenme mimarilerinde, geleneksel makine öğrenmesi algoritmalarının tek katmanlı yapısında gerçekleştirilen hesaplama işlemleri, farklı seviyelerdeki katmanlar ve çoklu parametreler aracılığıyla kademeli olarak gerçekleştirilmektedir. Her katmanda hesaplanan çıktılar sonraki katmanlara aktarılmakta ve doğrusal olmayan fonksiyonlar ile girdiler, en önemli özelliklerine dayalı olarak indirgenerek amaca yönelik temsilleri çıkarılmaktadır. Bu çalışma kapsamında Bi-LSTM mimarisine dayalı iki sınıflı ve çok sınıflı WAF modelleri önerilmiştir.

Bi-LSTM mimarisinin temel özellikleri ve matematiksel oluşumları aşağıda sunulmaktadır. Tekrarlayan Sinir Ağı (RNN), kısa vadeli bağımlılıklara dayalı bir sonraki diziyi tahmin

etmede başarılı çalışmaktadır. Bununla birlikte, RNN mimarileri gradyan değerinin aşırı küçülmesine dayalı kaybolan gradyan (gradient vanishing) veya gradyan değerinin çok büyümesine dayalı gradyan patlaması problemleri nedeniyle uzun vadeli bağımlılık problemlerinde başarılı olamamaktadır [72]. LSTM [71] mimarisi, RNN mimarisinin özel bir versiyonudur. Uzun ve kısa süreli hafızası sayesinde RNN mimarilerinde var olan uzun süreli bağlantı problemini çözmektedir[71]. LSTM mimarisi, RNN mimarisiyle aynı yapıda olup içerdiği nöronlar farklılık göstermektedir. LSTM mimarisinde yer alan nöronlar, giriş, çıkış ve unutma olmak üzere üç kapı içermekte ve kapı durumlarını güncelleyerek uzun süreli bağlantıların korunabilmesine imkân sağlamaktadır [71]. Bi-LSTM mimarisi [73], tek çıktı üreten iki LSTM katmanının birbirine zıt yönlerde birleştirilmesi ile oluşturulmaktadır (bkz. Şekil 4.2).

Standart bir RNN mimarisinde $\boldsymbol{x} = (x_1, x_2, x_3, \dots, x_n)$ girdi dizisi verildiğinde, $\boldsymbol{h} = (h_1, h_2, h_3, \dots, h_n)$ gizli vektör dizisi ve $\boldsymbol{y} = (y_1, y_2, y_3, \dots, y_n)$ çıktı vektör dizisi $\boldsymbol{t} = 1$' den $\boldsymbol{t} = n$ 'ye kadar $\boldsymbol{n}$ adımda Eş 4.1 ve 4.2' deki gibi hesaplanmaktadır [72]. Burada $\boldsymbol{W}$ ağırlık matrislerini, $\boldsymbol{b}$ bias değerlerini ve $\boldsymbol{H}$ gizli katman fonksiyonunu (genellikle sigmoid fonksiyonu) ifade etmektedir.

$$h_t = H(W_{xh} + W_{hh}x_{t-1} + b_h) \tag{4.1}$$

$$y_t = W_{hy}h_t + by \tag{4.2}$$

Uzun süreli bağlantıları tutmak için geliştirilen ve giriş, çıkış ve unutma kapılarından oluşan LSTM mimarisindeki gizli katman fonksiyonu $\boldsymbol{H}$, Eş. 4.3 - 4.7' deki gibi hesaplanmaktadır [72].

$$f_t = \sigma_a\left(W_{xf}x + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f\right) \tag{4.3}$$

$$i_t = \sigma_a\left(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i\right) \tag{4.4}$$

$$o_t = \sigma_a\left(W_{xo}x_tW_{ho}h_{t-1} + W_{co}c_{t-1} + b_o\right) \tag{4.5}$$

$$c_t = f_tc_{t-1} + i_t\sigma\left(W_cx_t + U_ch_{t-1} + b_c\right) \tag{4.6}$$

$$h_t = o_t\sigma_h\left(c_t\right) \tag{4.7}$$

Bi-LSTM mimarisinde, ileri yönlü hücrelerin (Eş. 4.8) ve geri yönlü hücrelerin (Eş. 4.9) çıkışları tek bir aktivasyon fonksiyonu ile birleştirilmekte ve çıkış dizisi ***y*** Eş. 4.10'daki gibi hesaplanmaktadır [73].

$$h_t = H(W_{\underset{xh}{\rightarrow}} x_t + W_{\underset{hh}{\rightarrow\rightarrow}} h_{t-1} + b_{\underset{h}{\rightarrow}}) \tag{4.8}$$

$$h_t = H(W_{\underset{xh}{\leftarrow}} x_t + W_{\underset{hh}{\leftarrow\leftarrow}} h_{t-1} + b_{\underset{h}{\leftarrow}}) \tag{4.9}$$

$$y_t = W_{\underset{hy}{\rightarrow}} h_t + W_{\underset{hy}{\leftarrow}} h_t + b_y) \tag{4.10}$$

4.3. Metodoloji

Bu bölümde çalışmada geliştirilen DA-SANA yöntemi ve önerilen sınıflandırma modellerinin teorik gösterimleri sunulmaktadır. Önerilen veri büyütme yöntemi ve sınıflandırma modelleri, yapısal terimler kullanılarak ayrıntılı olarak açıklanmıştır. Çalışma kapsamında web isteklerinde yer alan toplam karakter uzunluğuna dayalı DA-SANA yöntemi geliştirilmiş, derin öğrenme tabanlı modellerin duyarlılığı artırılırken, web saldırı tespitinde önemli bir sorun olan yüksek yanlış pozitif (FP) oranı, aşırı öğrenmenin (overfitting) önüne geçilerek azaltılmıştır. Aşağıda ayrıntıları açıklanan DA-SANA süreci ile öğrenme setindeki orijinal veriler, oluşturulan gürültülü verilerle birleştirilmiş; böylece veri miktarı iki katına çıkarılarak veri artırımı gerçekleştirilmiştir.

4.3.1. Kendinden uyarlamalı, gürültü eklemeye dayalı veri artırım yöntemi

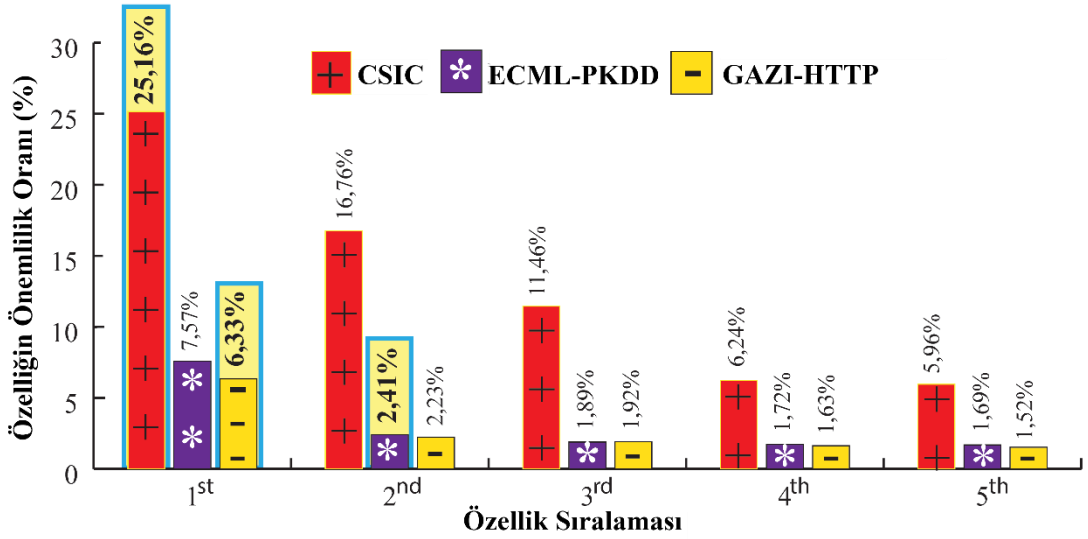
Web isteklerinin birbirine çok benzemesi nedeniyle mevcut NLP yöntemleri ile web saldırı trafiğinin yüksek hassasiyetle belirlenmesi önemli bir sorun haline gelmiştir. Gürültülü veriler ekleyerek model duyarlılığını artırmak, literatürde kabul edilen ve bu sorunla mücadele etmede kullanılan etikili bir yöntemdir. Bu çalışmada kapsamında isteğin uzunluğuna göre kendinden uyarlamalı (her isteğin bireysel özelliklerine dayalı) gürültü ekleyen DA-SANA (Data Augmentation technique based on the Self-Adapting Noise Adding) yöntemini geliştirilmiştir. Geliştirilen yöntemde kullanılan kendinden uyarlama tekniği; DA-SANA' yı normal ve anormal istekleri ayırt etmede etkili bir yöntem haline getirmiştir. Bilgimiz dâhilinde, önerilen yöntem saldırı tespit sistemleri için daha önce hiç kullanılmamıştır. Bu çalışmaya ek olarak literatürde yer alan diğer bazı çalışmalarda [24, 31, 74] gerçekleştirilen istatistiksel analizler ile web saldırı tespitinde HTTP isteğinin uzunluğunun (barındırdığı toplam karakter sayısı) önemli olduğunu açıkça ortaya

konulmaktadır. Bu sebeple, önerilen DA-SANA yöntemi kapsamında eklenecek gürültü miktarının belirlenmesinde kullanılan kendi kendine uyarlama katsayıları için web isteklerinin uzunluklarına dayalı bir yöntem kullanılmıştır.

Web istek uzunluğunun sınıflandırmadaki önemi

Kruegel ve arkadaşları tarafından gerçekleştirilen ve anomali tabanlı web güvenliği alanında gerçekleştirilen ilk araştırmalardan birisi olan çalışmada [24], normal ve saldırı trafiklerinin sınıflandırılmasında web isteklerinde yer alan özellikler ele alınmıştır. Bu çalışmaya göre, bir web uygulamasına yapılan normal HTTP web istekleri benzer bir uzunluk dağılımına sahiptir. Web saldırıları, web uygulamasının normalde alacağından daha uzun veya daha kısa web isteklerinden oluşabilmektedir. Bu durum, web isteklerinin uzunluklarının dağılım varyansında büyük değişikliklere neden olmaktadır [24].

Tez kapsamında gerçekleştirilen bu çalışmada, CSIC-HTTP [26], ECML-PKDD [27] ile tez kapsamında oluşturulan GAZI-HTTP verisetleri üzerinde ekstra ağaç (Extra Tree) sınıflandırıcı tabanlı öznitelik seçim işlemi gerçekleştirilmiştir. Ekstra ağaçlara dayalı sınıflandırıcı modelinde Gini endeksine dayalı olarak ilişkili karar ağaçları oluşturulmaktadır. Ekstra ağaçlar, rastgele orman (Random Forest, RF) algoritmasına çok benzeyen bir topluluk öğrenme algoritmasıdır. Ekstra ağaçlara dayalı oluşturulan Gini indeksleri, özellik seçimi için bir özellik setinin önemi olarak kullanılabilmektedir. Üç farklı web anomali veri kümesi için ekstra ağaç sınıflandırıcısına dayanan en önemli beş özellik Şekil 4.3'te gösterilmektedir. Ayrıca bu özelliklerin adları Çizelge 4.1’ de görülmektedir. Bazı özellikler tokenizasyon sonucunda oluşturulduğundan, web isteklerinin tokenize edilmiş kelime listesi analiz edilerek özellik adları belirlenmiştir.



Şekil 4.3. Ekstra ağaç tabanlı özellik seçimine dayalı özellik önem değerleri ve sıralamaları (en önemli beş özellik)

Çizelge 4.1. Ekstra ağaç tabanlı özellik seçimine göre en önemli beş özelliğin isimleri

Veriseti	En Önemli Beş Özellik				
	1. en-önemli	**2. en-önemli**	**3. en-önemli**	**4. en-önemli**	**5. en-önemli**
CSIC-HTTP	**İstek-Uzunluğu (Request-Length)**	İsteğin aldığı yol değeri (Path)	İsteğin aldığı ilk parametrenin ismi (Payload-Key-First)	İçerik-Uzunluğu (Content-Length)	İsteğin aldığı ilk parametrenin değeri (Payload-Value-First)
ECML-PKDD	İçerik (inContext)	**İstek-Uzunluğu (Request-Length)**	İsteğin aldığı ilk parametrenin ismi (Payload-Key-First)	Başlık (Header)	İsteğin aldığı ilk parametrenin değeri (Payload-Value-First)
GAZI-HTTP	**İstek-Uzunluğu (Request-Length)**	URI'nin son kısmı (URI-End)	URI'nin orta kısmı (URI-Middle)	İsteğin aldığı ilk parametrenin değeri Payload-Value-First)	Tarayıcı bilgisi (User-Agent)

Sonuç olarak, HTTP isteğinin uzunluğu, CSIC-HTTP ve GAZI-HTTP veri kümesi için en önemli özellik iken, ECML-PKDD veri kümesi için ikinci en önemli özellik olarak belirlenmiş olup, sınıflandırmaya Gini endeksi değerlerine göre sırasıyla %25.16, %6.33 ve %2.41'e katkıda bulunduğu tespit edilmiştir. Çalışma da kullanılan gürültü katsayısı, orijinal istekten yeni bir gürültülü istek oluştururken gürültünün yoğunluğunu belirtmektedir.

Kendinden uyarlamalı gürültülü verinin oluşturulmasında kullanılan gürültü katsayılarının web isteklerinin uzunluğu ile doğru orantılı olarak hesaplanmasına karar verilmiştir. DA-SANA yönteminin etkinliği, çalışmanın geri kalanında hem sonuçlar hem de karşılaştırma bölümlerinde görülmektedir.

DA-SANA' nın teorik temelleri

DA-SANA gürültü ekleme algoritmasının sözde kodu (pseudocode) Çizelge 4.2'de gösterilmiştir. Aşağıda sunulan eşitliklerde ve algoritmada kullanılan sembollerin tanımları Çizelge 4.3' de yer almaktadır. DA-SANA ile gürültülü verilerin oluşturulması iki aşamada gerçekleştirilmiştir. Padding (dolgu yapma) işlemi yukarıda açıklandığı üzere tokenizasyon işleminden sonra farklı boyutlardaki verilerin aynı boyuta getirilmesi amacı ile sonuna ya da başına verilerde yer almayan sabit bir değerin (genellikle 0 değeri) eklenmesi ile gerçekleştirilmektedir. DA-SANA kapsamında, orijinal verisetinden gürültü verilerin oluşturulmasında orijinal verinin padding kısmına gürültü ekleme işlemi gerçekleştirilmiştir. İlk olarak, isteklerin uzunluğuna bağlı olarak belirlenen $\boldsymbol{C_i}$ gürültü katsayılarına dayalı olarak gürültülü verilerin yoğunluğu ve yerleri (padding kısmındaki) rastgele olarak belirlenmiştir. İkinci olarak, orijinal veriseti ile aynı boyutta sahip olan $\boldsymbol{N_i}$ gürültü veri kümesi oluşturulmuş ve padding bölümündeki sıfır değerleri belirlenen gürültü konumlarına göre daha önce verisetinde yer alamayan ve tamamen rastgele belirlenmiş yeni tokenler ($k \subseteq (f, 2*f)$) ile değiştirilmiştir. Buradaki $\boldsymbol{f}$ terimi, tokenleştirme sürecinde en sık geçen terimlerin belirlenmesinde kullanılan verisetinde minimum bulunma sayısını (çalışmada 3000 olarak belirlenmiştir) ifade etmektedir. Bir isteğe eklenecek gürültü miktarı Eş. 4.12' de hesaplanan gürültü katsayılarına göre bulunmaktadır. Burada $\boldsymbol{C_i}$, gürültü katsayılarını, $\boldsymbol{L}$ HTTP isteklerinin içerdiği karakter sayısını ve $\boldsymbol{n}$ post-padding sonrasında oluşan dizilerdeki özelliklerin sayısını temsil etmektedir. Dizilerdeki gürültülü değerlerin yerlerini belirlemek için ayrık olasılık dağılımına (0 veya 1) dayalı binom dağılımı (bkz. Eş. 4.11) kullanılmıştır.

Gürültü değerlerini belirlemek için Gauss normal dağılımı (Eş. 4.13'te gösterildiği gibi) kullanılmış olup; buradaki $\boldsymbol{\sigma}$ parametresi standart sapmayı ve $\boldsymbol{\mu}$ dağılımın ortalamasını ifade etmektedir. Gürültülü verileri içeren $\boldsymbol{N_i}$ veriseti, Eş. 4.14'teki gibi $\boldsymbol{S_i}$ orijinal veri dizilerinden $\boldsymbol{C_i}$ gürültü katsayılarına dayalı olarak Gauss dağılımı ile oluşturulmuştur. Son olarak, $\boldsymbol{S_i}$ orijinal veri kümesine $\boldsymbol{N_i}$ gürültülü veri kümesi eklenerek veri büyütme işlemi gerçekleştirilmiştir.

$$P_{Binomal}(x) = \binom{n}{x} p^x q^{n-x} = \frac{n!}{(n-x)!x!} p^x q^{n-x} \quad (4.11)$$

$$C_i = P_{Binomal}\left(\frac{L_i - \min(L)}{\max(L) - \min(L)} * 0.05, size = (n) \right) \quad (4.12)$$

$$P_{Gaussian}(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2} \quad (4.13)$$

$$N_i = S_i + P_{Gaussian}\left(k \subseteq (f, 2*f), size = n\right) * C_i \quad (4.14)$$

Çizelge 4.2. DA-SANA algoritması

Giriş: x HTTP isteklerinden oluşan $\Omega \leftarrow \{x_1, x_2, ..., x_m\}$ eğitim veriseti ve
s tokenizasyon verilerinden oluşan $S \leftarrow \{(s_{11}, s_{12}, ..., s_{1n}), ..., (s_{m1}, s_{m2}, ..., s_{mn})\}$ dizisi
Çıkış: Orijinal ve gürültü verilerden oluşan artırılmış veriseti

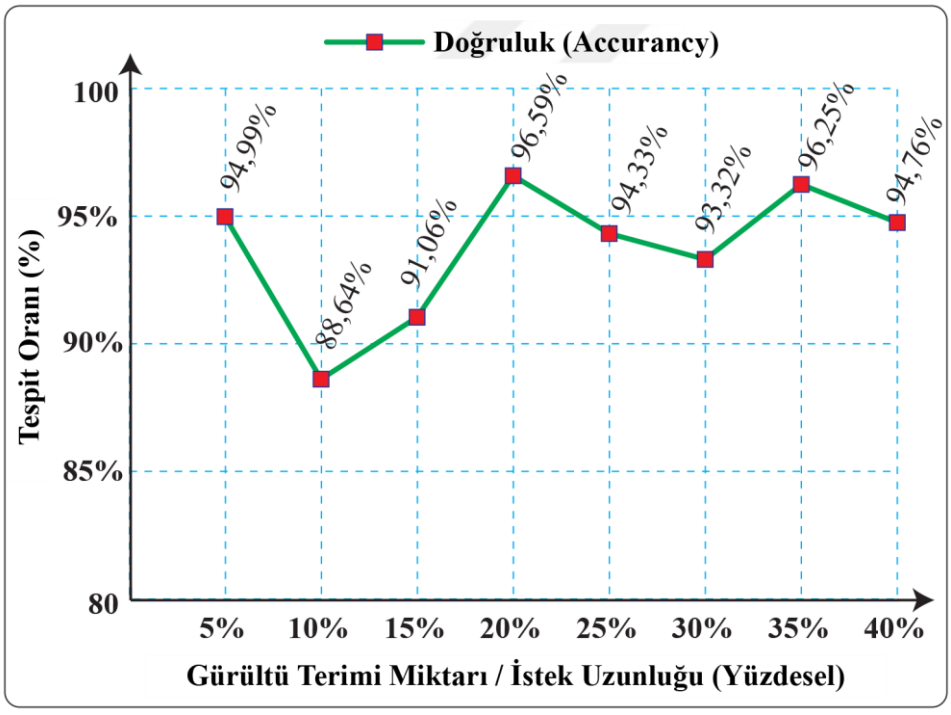
1. **for** $\forall x_i \in \Omega \; \forall xi \in \Omega$ **do**
2. **Set:** $L_i \leftarrow$ x_i isteğinin uzunluğu // x_i isteğinin içerdiği karekter sayısı
3. $L_i \leftarrow \frac{L_i - \min(L)}{\max(L) - \min(L)}$; (0,1) aralığında normalize edilmiştir.
4. **Set:** Gürültülü Veri $N_i \leftarrow$ rastgele üretilmiş n adet integer sayı, sayılar $(f, 2*f)$ aralığından, rastgele olarak Gauss normal dağılımına göre seçilmiştir.
5. **Set:** Gürültü katsayıları Ci$\leftarrow$ rastgele üretilmiş *n* adet integer sayı, sayılar (0,1) kümesinden alınmış ve *n* değerinin %20 si kadardır.
6. **Set:** Gürültülü veriseti $ND_i \leftarrow N_i * C_i + S_i * C_i^{'}$
7. **end for**
8. **Set:** Artılmış eğitim veriseti $S^{'} \leftarrow S_i + ND_i$

Çizelge 4.3. Algoritma ve eşitliklerde kullanılan sembollerin açıklamaları

Semboller	**Tanımları**
x_t	Giriş vektörü
C_i	İstek uzunluklarına dayalı gürültü katsayıları
N_i	Gürültülü veriseti
L_i	HTTP isteklerinin uzunlukları (karakter sayısı)
S_i	Orijinal veriseti (tokenizasyondan sonraki diziler)
m	Girdi verisetindeki HTTP istek sayısı
n	Post-pading işleminden sonraki toplam özellik sayısı
f:	Maksimum sık geçen terim sayısı (çalışmada 3000 olarak ayarlanmıştır)

Eklenecek gürültü miktarı, her talebin uzunluğu ile doğru orantılı olarak belirlenmiş ve dizi uzunluğunun (***n***) maksimum %20' si ile sınırlandırılmıştır. Sınırlama oranı, veri setimiz üzerinde gerçekleştirilen analiz işlemi ile belirlenmiştir. Değişken limit oranlarına dayalı

DA-SANA ile üretilen artırılmış veriseti ile eğitilmiş Bi-LSTM tabanlı çok sınıflı modellerin tespit oranları Şekil 4.4' te görülmektedir. Limit oranının uygulanmasının amacı, dizilerde yer alan token sıklığındaki değişimi kısıtlamak olup, böylece veri kümesindeki temsillerin bozulmasının önüne geçilmiştir. Şekil 4.4’ e göre, önerilen DA-SANA tekniği ile diziye eklenen gürültü terimlerinin sayısı istek uzunluğunun %20'si ile sınırlandırıldığında en yüksek algılama oranının elde edildiği görülmektedir. Düşük miktarda gürültülü terim eklenmesi DA-SANA'nın etkinliğini ortadan kaldırırken, yüksek oranda gürültülü terim eklenmesi gürültü eklenerek artırılan web isteğinin orijinalden uzaklaşmasına neden olabilmektedir. Sonuç olarak, DA-SANA yönteminde dizideki mevcut tokenlerden farklı tokenler gürültü olarak eklenmekte; böylece mevcut tokenlerin sıklığı değişmemektedir. Ayrıca artırılmış veriseti içerisinde orijinal veriseti de yer aldığından eğitim ve test süreçlerinde modellerin orijinal web isteklerini görebilmesi sağlanmaktadır. Gürültü miktarları da sınırlanmakta ve web isteklerinin etiket dağılımında bir değişiklik olmamaktadır. Böylece, önerilen DA-SANA yöntemi, verisetlerinin temsilinde ihmal edilebilir bir değişime neden olmaktadır.



Şekil 4.4. Gürültülü terimlerin miktarının istek uzunluğuna göre farklı oranlarda sınırlandırılması

4.4. Çalışmada Kullanılan Web Anomali Tespit Modelleri

Çalışma kapsamında hem iki sınıflı hem de çok sınıflı sınıflandırma modelleri geliştirilmiştir. Anomali tabanlı WAF sistemlerinin süreçlerini anlamak ve analiz etmek için öncelikle WAF sistemlerinin teorik yapısını aşağıda sunulmaktadır.

4.4.1. WAF sistemlerinin teorik arka planı

Anomali tabanlı WAF sistemleri temel olarak web isteklerini içeren veri kaynaklarının işlenmesi ve sınıflandırılmasından meydana gelmektedir. Anormali tabanlı bir WAF sistemi, ilk üç öğenin veri yapılarıyla ilgili olduğu ve geri kalan üç öğenin ise tahmin modeliyle ilgili olduğu toplam temel altı öğeden (***D, Y, X, P, H, L***) oluşan bir setle temsil edilebilmektedir. Bu kapsamda anormali tabanlı bir WAF sistemi oluşturan altı temel öğenin teorik temsilleri aşağıda sunulmaktadır.

D: WAF modellerinin eğitilmesi, test edilmesi ve analiz edilmesi için verileri sağlayan veri kaynağını ifade etmektedir. Temel olarak, ardışık veri öğelerinden oluşan web isteklerinin akışından oluşmaktadır. Her WAF sisteminin kendi veri temsili bulunmaktadır. Bazı WAF modelleri yalnızca web isteklerini veya bunların belirli bölümlerini (URL, Yük (Payload), Yöntem (Method), Gövde (Body) vb.) kullanırken, bazı WAF modelleri ise web sunucusunun yanıtlarını ve yanıt kodlarını da içerebilmektedir. Genel olarak, $\boldsymbol{D} = (D_1, \ldots, D_N)$ olarak tanımlanmakta olup, buradaki $\boldsymbol{D_i}$, $\boldsymbol{i}$. web isteğinin bölümlerinden oluşan ve WAF tarafından analiz edilecek olan web istek verilerini ifade etmektedir.

Y: $\boldsymbol{Y} = (Y_1, \ldots, Y_N)$ olarak tanımlanan bir etiket kümesi olup, buradaki $\boldsymbol{Y_i}$ ifadesi $\boldsymbol{D_i}$ web isteğinin normal mi yoksa anormal mi olduğunu (veya hangi tür saldırı olduğunu) belirtmektedir. ***Y*** kümesinde yer alan etiket değerleri önceden belirlenmiştir ve tümü doğru kabul edilmektedir. İki sınıflı web anomali tespiti için $\boldsymbol{Y_i}$, ***Y*** kümesindeki $\boldsymbol{i}$. öğe olup; $\boldsymbol{Y_i} \in$ [*Normal, Anormal*] veya $\boldsymbol{Y_i} \in$ [*0, 1*] olmak üzere ***0*** değeri normal, ***1*** değeri ise anormal anlamına gelmektedir. Çok sınıflı sınıflandırma için $\boldsymbol{Y_i} \in$ *[Normal, Saldırı-tipi-1, …, Saldırı-tipi-K]* veya [*0,1, …, K*] ile ***Y*** setini oluşturabilmekte olup, buradaki ***K*** toplam saldırı türü sayısıdır.

X: sonlu sayıda özellik içeren bir özellik vektörü olup, formal olarak ***X*** = *[X_1, ..., X_m]* olarak ifade edilmektedir. Buradaki *m* toplam özellik sayısıdır. Özellikler, her bir WAF modeli için Ngram, Tf-Idf veya kelime gömme gibi farklı yöntemlerle oluşturulabilmektedir.

P: özellik seçimi, veri indirgeme ve temsil oluşturma işlemlerini içeren veri ön işleme aşamasıdır. Özellik seçimi ve veri azaltma işlemleri isteğe bağlıdır ve derin öğrenme teknikleri gibi bazı algoritmalar için yapılmasına ihtiyaç duyulmayabilmektedir. Bununla birlikte, metin tabanlı web isteklerinin, sınıflandırma algoritmasının işleyebilmesi için ***X*** özellik vektör uzayından ***P***: $D \rightarrow X$ işlemi ile temsil uzayına dönüştürülmesi gerekmektedir.

H: sınıflandırma algoritmasıdır. Esasında, çıktı olarak normal ve anormal (veya saldırı türleri) sınıf etiketlerinden birini atamak için girdi özelliklerini tartan bir işlevdir. Problemlere göre sınıflandırıcı olarak farklı makine öğrenmesi algoritmaları seçilebilmekte ya da farklı algoritmaların bir topluluk modeli olarak birlikte kullanıldığı hibrit yapılar oluşturulabilmektedir.

L: Verilen veri kaynağı ***D***' nin vektör temsili ***X*** de yer alan özellikleri kullanarak, ***H*** sınıflandırma algoritması sayesinde ***Y*** kümesinde yer alan etiketlere eşleyen sınıflandırıcı modeldir.

Yukarıda verilen formal tanımlar ışığında, HTTP web isteklerinin sınıflandırılması işleminin matematiksel gösterimi Eş 4.15 – 4.17 de gösterilmektedir. Veriseti önişleme ve bölme işlemleri Eş 4.15'de, sınıflandırıcı model gösterimi Eş. 4.16' da, WAF model tahminlerinin üretilmesi ise Eş. 4.17' de matematiksel olarak gösterilmektedir. Bu eşitliklerdeki ifadelerden $\boldsymbol{D_{train}}$ eğitim verilerini, ***L*** sınıflandırıcı modelini, $\boldsymbol{H(X_i)}$ belirli bir algoritmaya dayalı sınıflandırma sürecini, $\boldsymbol{X_i}$ ***i.*** girdiyi ve $\boldsymbol{Y_i}$ ***i.*** girdinin model çıktısını ifade etmektedir. $\boldsymbol{Y_{test}}$ ise, ***K*** sayıda bilinmeyen web isteği içeren $\boldsymbol{X_{test}}$ test seti için öncesinde eğitilmiş ***L*** sınıflandırıcı modeli kullanılarak üretilen tahminleri ifade etmektedir (bkz. Eş. 4.17).

$$\left(D_{train}, D_{test}\right): P(D) \cup Y \rightarrow X \cup Y = \left\{x_n, y_n\right\}, n = 1,..N \tag{4.15}$$

$$L: D_{train} \rightarrow H\left(X\right), burada\, H\left(X_i\right): X_i \rightarrow \Upsilon_i \tag{4.16}$$

$$Y_{test}: X_{test} \rightarrow L, burada\, X_{test} = \left\{x_k\right\}, k = 1,...K \tag{4.17}$$

4.4.2. Önerilen sınıflandırma modelin yapısı

Önerilen sınıflandırma modeli sırasıyla bir kelime gömme katmanı, iki ardışık Bi-LSTM katmanı ve bir softmax sınıflandırıcı katmanından oluşmaktadır (bkz. Şekil 4.2). İki sınıflı ve çok sınıflı sınıflandırma modelleri için aynı yapıya sahip iki farklı model önerilmiş olup, yapıları Çizelge 4.4' de görülmektedir. İki sınıflı sınıflandırıcı modelinde gömme boyutunun 16 seçilmesinin, başarılı öğrenme için yeterli olduğu belirlenirken, çok sınıflı sınıflandırıcı için 32 boyutuna uygun olduğu tespit edilmiştir. Önerilen modelin akış şeması Şekil 4.5'te görülmektedir. Şekil 4.5' te görüldüğü üzere web anomali sınıflandırma modelinin geliştirilmesi iki aşamada gerçekleştirilmiştir. İlk aşamada, web istekleri ön işleme tabi tutulmuş, kelime tokenleri çıkarılmış ve DA-SANA tabanlı veri artırma işlemi gerçekleştirilmiştir. İkinci aşamada ise, derin öğrenme modeline dayalı olarak on katlı çapraz doğrulama yöntemi ile eğitim ve test işlemleri gerçekleştirilerek WAF modeli eğitilmiş ve test edilmiştir.

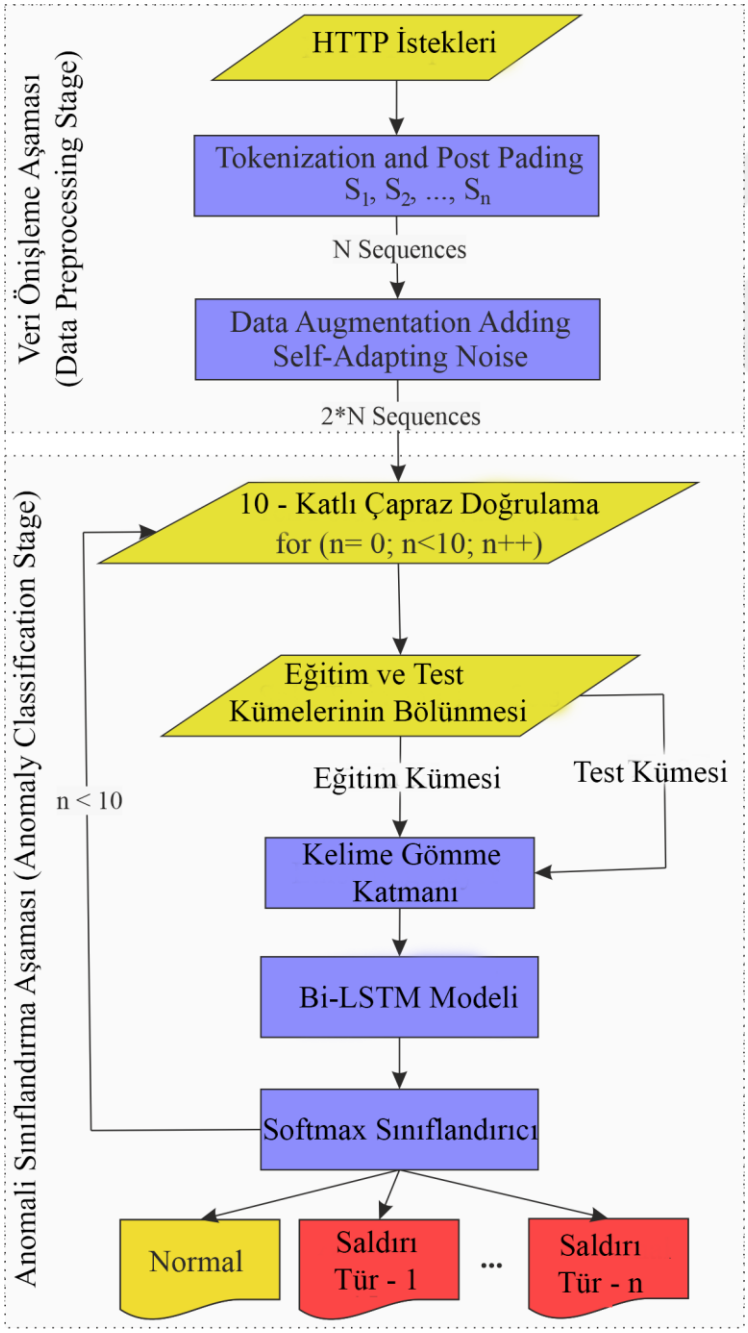
Çizelge 4.4. Önerilen modelin mimari yapısı

İki sınıflı sınıflandırma modeli				**Çok sınıflı sınıflandırma modeli**			
Katman	Giriş Boyutları	Çıkış Boyutları	Aktivasyon Fonksiyonu	Katman	Giriş	Çıkış	Akt. Fonk.
Embedding	(m,n)	$(m,n,16)$	-	Embedding	(m,n)	$(m,n,32)$	-
Bi-LSTM	$(m,n,16)$	$(m,n,256)$	tanh	Bi-LSTM	$(m,n,32)$	$(m,n,256)$	tanh
Bi-LSTM	$(m,n,256)$	$(m,256)$	tanh	Bi-LSTM	$(m,n,256)$	$(m,256)$	tanh
Dense	$(m,256)$	$(m,2)$	softmax	Dense	$(m,256)$	$(m,label_count)$	softmax

4.4.3. Hiper-parametre optimizasyonu

Artırılmış eğitim seti eğitilen modellerin sınıflandırma performansını iyileştirmek için hiper-parametre optimizasyonu gerçekleştirilmektedir. Çalışma kapsamında hiper-parametre optimizasyonu işleminde, modellerin performansını iyileştirmek için farklı parametre ayarlamaları gerçekleştirilmiştir. İlk olarak, Bi-LSTM katmanlarında Relu, sigmoid, tanjant hiperbolik (tanh) gibi lineer olmayan farklı aktivasyon fonksiyonları denenmiştir. Sonuçlar doğrultusunda tanh fonksiyonunun en başarılı olduğunu görüldüğü için tanh' ın aktivasyon fonksiyonu kullanılmasına karar verilmiştir. Sınıflandırmada en başarılı aktivasyon fonksiyonu olarak tespit edilen Softmax fonksiyonu, çıkış katmanında aktivasyon fonksiyonu kullanılmıştır. Bi-LSTM katmanlarında aşırı öğrenmeyi önlemek amacıyla katmanların girişlerinde ve tekrarlayan katmanlarda (recurent state) ayrı ayrı %5 oranında

dropout uygulanmıştır. Veriseti miktarı ve bellek kısıtlamaları nedeniyle mini parti miktarı (batch size) 256 olarak ayarlanmıştır. Eğitim sürecinde eğitim setindeki girdilerin sırası her döngüde (epoch) karıştırılmıştır. Aksi durumlarda, mini partiler halindeki veriler yüksek oranda korelasyon gösterecek ve modelin fazla uyum sağlamasına neden olacaktır [75]. Modelin optimizasyonunda 0.001 öğrenme oranı (learning rate) ile Adam optimizasyon algoritması kullanılmıştır. İki sınıflı ve çok sınıflı modeller için kayıp fonksiyonu (loss function) olarak sırasıyla ikili çapraz entropi ve kategorik çapraz entropi fonksiyonları kullanılmıştır.



Şekil 4.5. Önerilen modelin akış şeması

4.5. Çalışmada Kullanılan Verisetleri ve Doğrulama Yöntemleri

Anomali tabanlı WAF modellerini test etmek için çok fazla güncel veri seti bulunmamaktadır. Mevcutta DARPA tarafından oluşturulan KDD-99 veri seti, saldırı tespit sistemlerini (IDS) test etmek için en bilinen ve en çok kullanılan veri setidir [76]. Bununla birlikte, bu veriseti güncel olmadığı ve güncel saldırı türlerini içermediği için eleştirilmektedir [57]. Bu tez çalışması kapsamında, yukarıda açıklandığı üzere GAZI-HTTP isimli güncel saldırı parametrelerini içeren, çoklu saldırı türleri ile etiketlenmiş özgün bir web anomali veri seti oluşturulmuştur. Modellerin eğitim ve testlerinde GAZI-HTTP veri setimize ek olarak, web güvenliği alanında sıklıkla kullanılan CSIC-HTTP ve ECML-PKDD isimli iki benchmark web anomali veriseti kullanılmıştır. Bu verisetleri doğrudan web güvenliği ile ilgili olmaları, web saldırılarını içermeleri ve herkese açık olmaları nedeniyle tercih edilmiştir. Tez çalışması kapsamında oluşturulan GAZI-HTTP veri setimize ilişkin detaylı bilgiler yukarıda verilmiştir. Çalışma kapsamında kullanılan diğer iki verisetine ilişkin özet bilgiler aşağıdaki gibidir.

4.5.1. CSIC-HTTP veriseti

CSIC-HTTP veriseti, İspanya Ulusal Araştırma Konseyi (CSIC), Bilgi Güvenliği Enstitüsü tarafından bir web uygulamasına gelen normal ve saldırı web trafiklerinin kaydedilmesi ile üretilmiştir [67]. Veri seti, 36000 normal ve 25065 saldırı isteği olmak üzere toplamda 61065 adet HTTP istek verisi içermektedir. HTTP istekleri, normal veya anormal olarak etiketlenmiş olup, veriseti yalnızca iki sınıflı sınıflandırma için uygundur. Çalışma kapsamında HTTP istekleri ön işleme aşamasında içerdikleri özellikler kullanılarak Yöntem, URI, Yol, Yük Anahtarları, Yük Değerleri vb. olmak üzere farklı sütunlara ayrılmıştır.

4.5.2. ECML-PKDD veriseti

Bu veri seti, Web Trafiğinin Analiz Edilmesi (Analyzing Web Traffic ECML/PKDD 2007 Discovery Challenge) isimli yarışma kapsamında 2007 yılında, ECML/PKDD konferansında yayınlanmıştır [68]. Verisetindeki HTTP istekleri; yöntem, protokol, uri, sorgu, başlık ve gövde olmak üzere 6 farklı bileşenden oluşmaktadır. Normal (“Valid”) ve yedi farklı türde web saldırı tipi olmak üzere 8 farklı sınıfta 52296 HTTP isteği bulunmaktadır. Verilerin %66'dan fazlası "Valid" olarak etiketlendiğinden veri kümesi dengesiz dağılıma sahiptir. Veri setindeki etiketlerin dağılımı ve eğitim setinin DA-SANA ile veri büyütme öncesi ve sonrası durumu Çizelge 4.5’ de görülmektedir.

Çizelge 4.5. ECML-PKDD verisetinin dağılımı

Etiket	Miktar	DA-SANA sonrası miktar	Dağılım
Valid	35006	70012	%66.94
Ldap Injection	2279	4558	%4.36
OS Commanding	3420	6840	%6.54
Path Traversal	3029	6058	%5.79
SSI	1898	3796	53.63
SQLi	2560	5120	%4.90
XPath Injection	2279	4558	%4.36
XSS	1825	3650	%3.49
Toplam	**52296**	**104592**	**%100**

4.5.3. Çalışmada kullanılan araçlar

GAZI-HTTP web anomali veri seti oluşturulmasında işletim sistemi, web spider, web saldırı ve ağ sniffer araçları olarak Kali Linux, Burp Suite, Paros ve TcpFlow kullanılmıştır. Çalışmanın yürütülmesinde Keras, Tensorflow açık kaynak platformları ve Python programlama dili kullanılmıştır. Ayrıca veri ön işleme, istatistiksel analiz ve görselleştirme aşamaları için panda, numpy, scikit-learn ve matplot kütüphaneleri kullanılmıştır. Modellerin eğitim ve testleri; Intel 6850K 3.6 GHz i7 işlemci, 16 GB RAM, GTX 1080 Ti 11 GB GPU, 512 GB SSD konfigürasyonuna sahip bir bilgisayar ile gerçekleştirilmiştir.

4.5.4. Çalışmada kullanılan model doğrulama yöntemleri

Bu çalışmada, GAZI-HTTP, CSIC-HTTP ve ECML-PKDD verisetlerine dayalı iki sınıflı anomali tespiti için üç model ve GAZI-HTTP ve ECML-PKDD verisetlerine dayalı çok sınıflı anomali tespiti için iki model olmak üzere toplam beş farklı model geliştirilmiştir. Geliştirilen modelleri doğrulamak ve performanslarını ölçmek için iyi bilinen iki veri bölme yöntemi kullanılmıştır. Doğrulama yöntemlerinden birisi, veri setini eğitim, doğrulama ve test setleri olarak yüzdelik bir dağılımla üç alt sete bölmektir. Çalışma kapsamında, veri setleri eğitim, doğrulama ve test olarak sırasıyla %60, %20 ve %20 rastgele kümelere ayrılmıştır. İkinci yöntem ise tüm veri setinin eğitim ve testlerde kullanılmasına izin veren, k-katlı (genellikle 10-katlı) çapraz doğrulama yöntemidir. K-katlı çapraz doğrulamada, veri seti rastgele k parçaya bölünmekte, k-1 parça eğitim için kullanılırken, kalan bir parça testlerde kullanılmaktadır. Bu işlem, tüm veri setinin eğitim ve test aşamalarında kullanılabilmesi için k kez tekrarlanarak tamamlanmaktadır. Bu çalışma kapsamında hem yüzdesel bölme hem de 10-katlı çapraz doğrulama yöntemleri kullanılmıştır.

4.5.5. Performans metrikleri

Bu bölümde çalışma kapsamında geliştirilen modellerin performans ölçümlerinin yapılmasında kullanılan metriklerin açıklamalarına kısaca yer verilmiştir. Dengesiz verisetleri, verisetinin sınıf dağılımındaki çarpıklık nedeniyle genellikle sınıflandırma yanlılığa (bias) eğilimli olma sorununa neden olmaktadır [77]. Sınıflandırma için kullanılan yapay öğrenime algoritmaları genellikle aynı oranlarda etikete sahip homojen veriler için iyi çalıştığından, dengesiz veri kümelerinin sınıflandırılması önemli bir zorluktur. Genel olarak, web anomali tespiti; çoğunluk sınıfın "normal", azınlık sınıfının ise "anormal" olduğu dengesiz bir sınıflandırma problemidir. Dengesiz sınıflandırma problemlerinde sınıfların dağılımı çarpık olduğundan, model performansı için doğruluk veya hata oranını tek başına bir metrik olarak kullanmak yeterli değildir. Bunlara ek olarak, azınlık sınıflarının duyarlılığını hesaba katan Kesinlik (Precision), Anma (Recall), F-skoru (F-score) ve Yanlış Alarm Oranı (False Alarm Rate, FAR) metriklerinin hesaplanması önemlidir [78]. Çalışmada kullanılan performans metrikleri; TP parametresi Doğru Pozitifi, TN parametresi Doğru Negatifi, FP parametresi Yanlış Pozitifi, FN parametresi ise Yanlış Negatifi göstermek üzere Çizelge 4.6' da sunulmuştur. Performans ölçütlerinin değerlendirilmesinde, çok sınıflı sınıflandırmada daha doğru sonuçlar verdiği için sınıf dağılımına dayalı ağırlıklı ortalama (A. Ort.) değerleri kullanılmaktadır.

Çizelge 4.6. Performans metrikleri

Metrik	**Hesaplanması**
Doğruluk (Accuracy, Acc)	$(TP+FP)/(TP+FP+TN+FN)$
Kesinlik (Precision)	$TP/(TP+FP)$
Anma (Recall)	$TP/(TP+FN)$
F-skoru (F1-Score)	$2*\frac{\mathrm{Pr}\,ecission*\mathrm{Re}\,call}{\mathrm{Pr}\,ecission+\mathrm{Re}\,call}$
Yanlış Alarm Oranı (False Alarm Rate, FAR)	$FP/(FP+TN)$

4.6. Çalışmanın Sonuçları ve Tartışmalar

Bu bölümde, üç farklı veri seti için modellerin performans sonuçları vesonuçların ilişkin tartışmalar aşağıda sunulmaktadır.

4.6.1. Performans hesaplamaları

Çalışma kapsamında yukarıda açıklandığı üzere iki farklı doğrulama yöntemi kullanılmış olup, bunlara ilişkin performans hesaplamaları ayrı ayrı başlıklar halinde sunulmaktadır.

Yüzdesel bölmeye dayalı doğrulama

Her model için eğitim ve test süreçleri deneysel olarak üç kez tekrarlanmış olup; yüzde bölme işlemi üç veriseti için de gerçekleştirilmiştir. Yüzde bölmeye dayalı iki sınıflı sınıflandırma modelleri için karşıtlık matrisleri Çizelge 4.7' de ve performans değerleri ise Çizelge 4.8' de görülmektedir. İki sınıflı sınıflandırma modellerinin, yüzde bölmeye dayalı olarak gerçekleştirilen üç deneyin sonuçlarından elde edilen en az değerlere göre, GAZI-HTTP, ECML-PKDD ve CSIC-HTTP için sırasıyla %98,34, %97,0 ve %99,08 sınıflandırma başarılarına ulaşılmıştır. Modellerin kesinlik, anma ve f-skor değerlerine bakıldığında; modellerin duyarlılıklarının çok yüksek, FN ve FP oranlarının düşük olduğu görülmektedir. Üç deney sonucunda, GAZI-HTTP, ECML-PKDD ve CSIC-HTTP verisetleri için en az ulaşılan F-skor değerlerinin sırasıyla %98.34, %96.95 ve %99.06 olduğu görülmüştür.

Çizelge 4.7. İki sınıflı Bi-LSTM modellerinin yüzde bölmeye göre karmaşıklık matrisleri

			Tahmin Edilen Sınıflar		
	Veriseti	Sınıf	Normal	Anormal	Toplam
Gerçek Sınıflar	GAZI-HTTP	Normal	**29245**	167	29412
		Anormal	552	**13454**	14006
		Toplam	29797	13621	**43418**
	ECML-PKDD	Normal	**6994**	1	6995
		Anormal	312	**3153**	3465
		Toplam	7306	3154	**10460**
	CSIC-HTTP	Normal	**7169**	60	7229
		Anormal	52	**4932**	4984
		Toplam	7221	4992	**12213**

Çizelge 4.8. İki sınıflı Bi-LSTM modellerinin yüzde bölmeye göre performans değerleri

		Performans Metrikleri				
Veriseti	Sınıf	Hassasiyet (Precision)	Anma (Recall)	F-skoru (F-Score)	FAR	Doğruluk (Accuracy)
GAZI-HTTP	Normal	0.9815	0.9943	0.9879	**0.0056**	**%98.34**
	Anormal	0.9877	0.9606	0.9740		
	A. Ort.	**0.9834**	**0.9834**	**0.9834**		
ECML-PKDD	Normal	0.9572	0.9998	0.9780	**0.0001**	**%97.00**
	Anormal	0.9996	0.9098	0.9525		
	A. Ort.	**0.9712**	**0.9687**	**0.9695**		
CSIC-HTTP	Normal	0.9927	0.9917	0.9921	**0.0082**	**%99.08**
	Anormal	0.9879	0.9895	0.9886		
	A. Ort.	**0.9907**	**0.9908**	**0.9906**		

A. Ort. : Ağırlıklı Ortalama

ECML-PKDD ve GAZI-HTTP verisetleri farklı saldırı tipinde etiketlere sahip oldukları için çok sınıflı sınıflandırma yapılabilmektedir. GAZI-HTTP veriseti ile eğitilen çok sınıflı sınıflandırma modelleri için yüzdelik dağılıma dayalı karşıtlık matrisleri Çizelge 4.9' da, performans değerleri ise Çizelge 4.10' da görülmektedir. Benzer şekilde, ECML-PKDD veriseti ile eğitilen çok sınıflı sınıflandırma modelleri için yüzdelik dağılıma dayalı karşıtlık matrisleri Çizelge 4.11' da, performans değerleri ise Çizelge 4.12' da görülmektedir. Çok sınıflı modellerin üç deney sonucunda, GAZI-HTTP veriseti için test kümesindeki 43418 adet HTTP isteğinden en az 41922' sini ve ECML-PKDD test kümesindeki 10460 adet web isteğinden 9824' ünü doğru sınıflandırdığı ve sınıflandırma başarılarının sırasıyla %96,5 ve %93,91 olduğu görülmektedir. Çizelge 4.10'a bakıldığında GAZI-HTTP veriseti için en iyi sınıflandırma oranının %98,6 F-skor değeri ile "Valid" sınıf için elde edildiğini, bunu %96,7 ile Open Redirect saldırısının ve takip ettiği görülmektedir. Bununla birlikte en kötü F-skor değeri %78,2 ile komut enjeksiyonu (Command Injection) saldırısına aittir. Komut enjeksiyon saldırısında rastgele sistem komutları bulunduğundan, sınıflandırma modelinin benzer komutları içeren XSS, SQLi ve LFI saldırı türlerinin içerdiği saldırı kalıpları ile karıştırdığı tespit edilmiştir.

Çizelge 4.9. GAZI-HTTP veriseti için çok sınıflı Bi-LSTM modellerinin yüzde bölmeye göre karşıtlık matrisi

			Tahmin Edilen Sınıflar									
			CRLF	*Command Injection*	*LFI*	*Open Redirect*	*SQLi*	*SSI*	*Valid*	*XSS*	*XXE*	*Toplam*
GAZI-HTTP	**Gerçek Sınıflar**	*CRLF*	**957**	16	4	3	4	18	32	6	4	1044
		Command Injection	5	**1015**	70	11	4	14	133	37	5	1294
		LFI	0	80	**950**	9	1	0	92	16	6	1154
		Open Redirect	4	22	34	**974**	0	4	20	20	12	1090
		SQLi	12	59	2	1	**4071**	6	123	47	0	4321
		SSI	1	12	3	41	0	**693**	21	12	2	785
		Valid	1	48	11	0	5	0	**29301**	22	24	29412
		XSS	1	48	3	0	2	6	266	**3328**	2	3656
		XXE	0	0	0	0	4	0	25	0	**633**	662
		Toplam	981	1300	1077	1039	4091	741	30013	3488	688	**43418**

Çizelge 4.10. GAZI-HTTP veriseti için çok sınıflı Bi-LSTM modellerinin yüzde bölmeye göre performans değerleri

		Performans Metrikleri				
		Hassasiyet (Precision)	Anma (Recall)	F-skoru (F-Score)	FAR	Doğruluk (Accuracy)
GAZI-HTTP	*CRLF*	0.975	0.916	0.945	0.0005	**%96.5**
	Command Injection	0.780	0.784	0.782	0.0067	
	LFI	0.882	0.823	0.851	0.0030	
	Open Redirect	0.937	0.893	0.915	0.0015	
	SQLi	0.995	0.942	0.967	0.0005	
	SSI	0.935	0.882	0.908	0.0011	
	Valid	0.976	0.996	0.986	0.0508	
	XSS	0.954	0.910	0.931	0.0040	
	XXE	0.920	0.956	0.937	0.0012	
	A. Ort.	**0.965**	**0.966**	**0.965**	**0.0077**	

A. Ort. : Ağırlıklı Ortalama

ECML-PKDD sonuçlarına bakıldığında; XPath enjeksiyon saldırısı için en iyi sınıflandırma oranına ulaşıldığı ve tüm saldırıların tespit edildiği; bu saldırı tipi için yanlış negatif (FN) üretilmediği görülmektedir. Valid, Ldap enjeksiyonu ve XSS türlerinin F-skor değerleri 0.99' un üzerinde olduğu ve bu sınıflar için Xpath enjeksiyonundan sonra sırasıyla en yüksek değerlere erişildiği görülmektedir. ECML-PKDD için en kötü sınıflandırma performansı SSI saldırı türünde ortaya çıkmaktadır. Çizelge 4.11'deki karşıtlık matrisinden görüldüğü üzere, SSI türü ile karışıklığa neden olan saldırı türleri sırasıyla işletim sistemi kod çalıştırma (OS

Commanding), dizin dolaşma (Path Traversel) ve SQL enjeksiyonu (SQLi) saldırılarıdır. Bunun nedeni, SSI saldırısının komut dosyasının HTML sayfaları aracılığıyla enjekte edilmesine veya ortak komutları çalıştırmasına dayanması ve yukarıdaki belirtilen saldırı türleri ile çok benzer komut dosyası sözcükleri ile dosya yolları paternleri içermesidir. Bununla birlikte bu saldırıların türleri yanlış sınıflandırılmış olsa dahi, çok sınıflı sınıflandırmada saldırıların normal olarak sınıflandırılmasının çok düşük olduğu görülmektedir.

Çizelge 4.11. ECML-PKDD veriseti için çok sınıflı Bi-LSTM modellerinin yüzde bölmeye göre karşıtlık matrisi

			Tahmin Edilen Sınıflar								
			Ldap Injection	OS Com.	Path Trav.	SSI	SQLi	Valid	XPath Inj.	XSS	Toplam
ECML-PKDD	**Gerçek Sınıflar**	Ldap Injection	**466**	0	0	0	0	0	0	0	466
		OS Commanding	0	**587**	4	34	42	13	0	0	680
		Path Transversal	0	3	**486**	103	0	6	0	0	598
		SSI	0	146	127	**71**	41	0	0	0	385
		SQLi	2	53	0	26	**392**	18	0	0	491
		Valid	3	5	0	0	4	**6983**	0		6995
		XPath Injection	0	0	0	0	0	0	**463**	0	463
		XSS	0	0	0	0	0	6	0	**376**	382
		Toplam	471	794	617	234	479	7026	463	376	**10460**

Çizelge 4.12. ECML-PKDD veriseti için çok sınıflı Bi-LSTM modellerinin yüzde bölmeye göre performans değerleri

		Performans Metrikleri				
		Hassasiyet (Precision)	Anma (Recall)	F-skoru (F-Score)	FAR	Doğruluk (Accuracy)
ECML-PKDD	Ldap Injection	0.989	1	0.994	0.0005	**%93.9**
	OS Commanding	0.739	0.863	0.796	0.0211	
	Path Transversal	0.787	0.812	0.8	0.0132	
	SSI	0.303	0.184	0.229	0.0161	
	SQLi	0.818	0.798	0.808	0.0087	
	Valid	0.993	0.998	0.996	0.0124	
	XPath Injection	1	1	1	0	
	XSS	1	0.984	0.992	0	
	A. Ort.	0.932	0.939	0.934	0.0080	

10-Katlı çapraz doğrulama

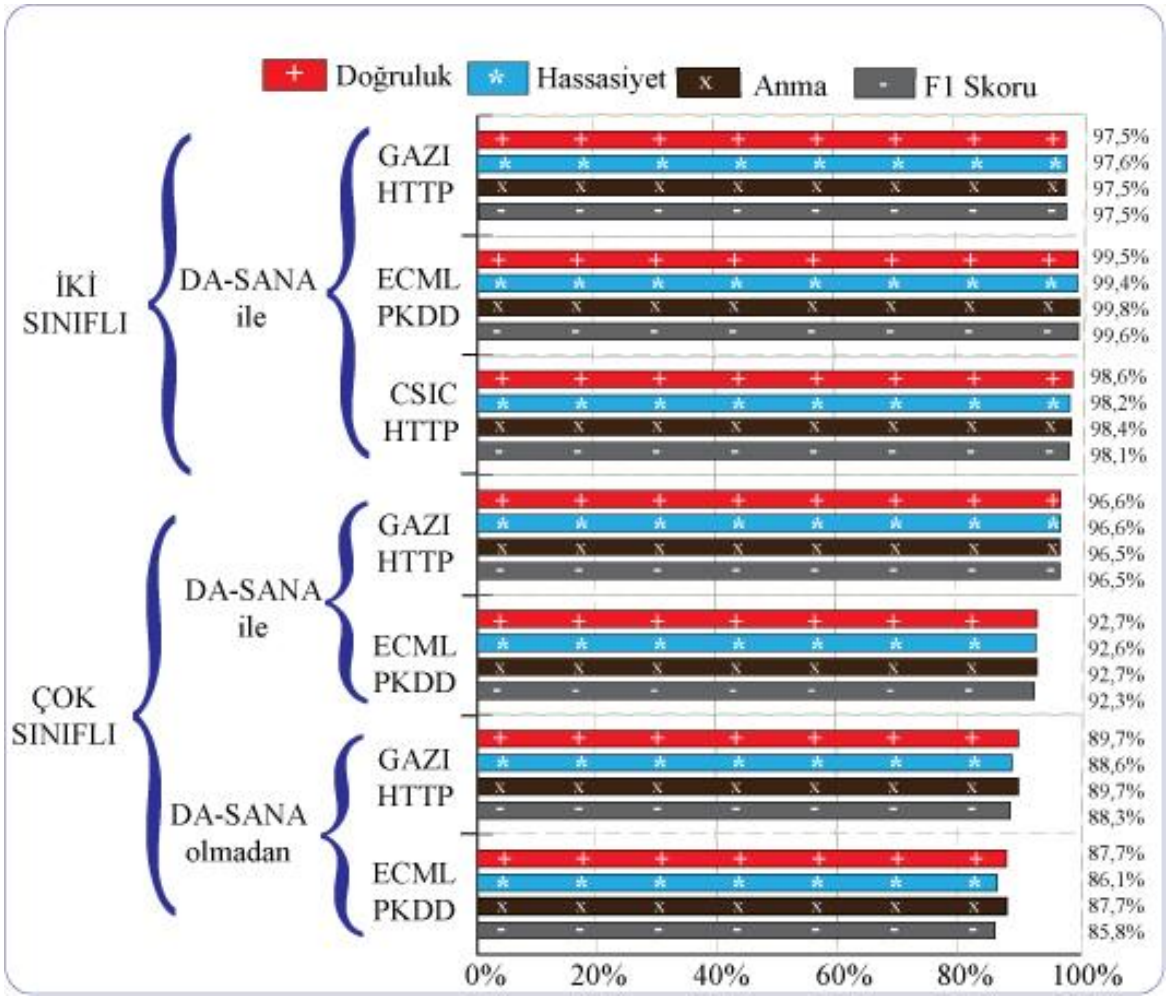
Tüm modeller 10-katlı çapraz doğrulama ile eğitilmiş ve performans testleri gerçekleştirilmiştir. Geliştirilen WAF modellerinin 10-katlı çapraz doğrulamaya dayalı performans değerleri Çizelge 4.13' sunulmuştur. Elde edilen sonuçlara göre iki sınıflı sınıflandırma modelleri, 10-kat çapraz doğrulama ile değerlendirildiğinde, en iyi tespit oranının %99,5 ile ECML-PKDD verisetine, en düşük tespit oranının ise %97,5 ile GAZI-HTTP verisetinde erişildiği görülmüştür. CSIC-HTTP veriseti için %98,6 tespit oranına erişilmiştir. DA-SANA uygulanarak eğitilen çok sınıflı modellerin performansları ile DA-SANA uygulanmadan eğitilen aynı mimarideki modellerin performansları karşılaştırılabilirliği sağlamak amacı ile Çizelge 4.13'de birlikte sunulmuştur. Çizelge 4.13'de görüldüğü üzere, GAZI-HTTP veriseti ile eğitilen aynı yapıdaki model DA-SANA ile veri artırımı gerçekleştirildiğinde çok sınıflı sınıflandırma için %96,6'lık bir doğruluğa erişirken, 10 kat çapraz doğrulamaya dayalı veri kümemiz için DA-SANA olmadan sadece %89,7 tespit oranına erişilmiştir. Benzer şekilde, ECML- PKDD veriseti üzerine DA-SANA uygulanarak ve uygulamandan eğitilen aynı yapıdaki modeller için sınıflandırma doğrulukları sırasıyla %92.7 ve %87.7'dir. Bu durumda, çalışma kapsamında önerilen DA-SANA veri artırım yöntemi ile GAZI-HTTP verisetimiz için %6,8, ECML-PKDD veriseti için ise %6,2 iyileştirme sağlandığı görülmektedir. Bunlara ek olarak, DA-SANA yönteminin doğruluk, kesinlik, anma ve F1-skor performans ölçütleri açısından da önemli katkısının olduğu Çizelge 4.13'de açıkça görülmektedir.

Çizelge 4.13. Bi-LSTM tabanlı tüm modellerin 10-katlı çapraz doğrulamaya göre performans değerleri

	GAZI-HTTP			**ECML-PKDD**			**CSIC-HTTP**
	İki Sınıflı	Çok Sınıflı		İki Sınıflı	Çok Sınıflı		İki Sınıflı
	Da-Sana ile	*Da-Sana olmadan*	*Da-Sana ile*	*Da-Sana ile*	*Da-Sana olmadan*	*Da-Sana ile*	*Da-Sana ile*
Hassasiyet (Precision)	97.6%	88.6%	96.6%	99.4%	86.1%	92.6%	98.2%
Anma (Recall)	97.5%	89.7%	96.5%	99.8%	87.7%	92.7%	98.4%
F Skoru (F Score)	97.5%	88.3%	96.5%	99.6%	85.8%	92.3%	98.1%
Doğruluk (Accuracy)	97.5%	89.7%	96.6%	99.5%	87.7%	92.7%	98.6%

Çalışma kapsamında, her üç veriseti üzerinde DA-SANA ile veri artırımı gerçekleştirilerek ve gerçekleştirilmeden eğitilen Bi-LSTM tabanlı modellerin 10-katlı çapraz doğrulamaya

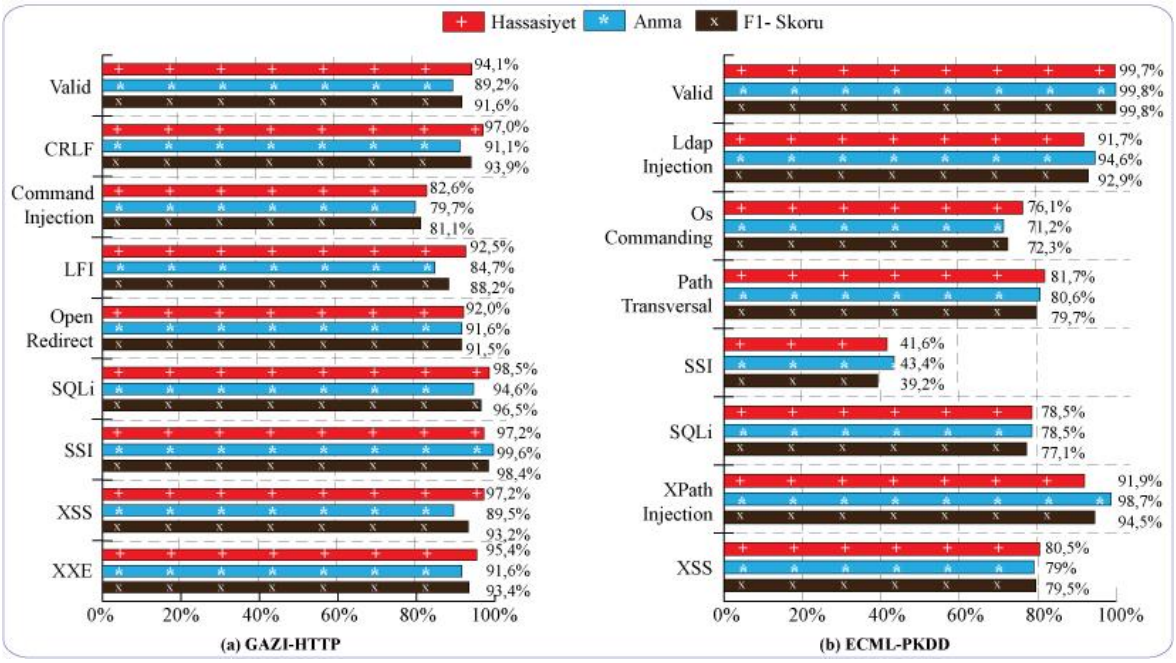
göre performans değerleri grafiksel olarak Şekil 4.6'da sunulmuş olup, ilgili grafik Çizelge 4.13' ün grafiksel halini içermektedir. DA-SANA yönteminin dengesiz dağılıma sahip web anomali verisetleri üzerindeki etkinliği Şekil 4.6'da görülmektedir.



Şekil 4.6. Bi-LSTM tabanlı tüm modellerin 10-katlı çapraz doğrulamaya göre performans değerleri

DA-SANA yöntemi ile artırılmış GAZI-HTTP ve ECML-PKDD verisetleri ile 10 katlı çapraz doğrulama yöntemi ile eğitilen çok sınıflı Bi-LSTM modellerinin sınıf bazlı performans değerleri detaylı olarak Şekil 4.7' da sunulmaktadır. Şekil 4.7 (a)' da görüldüğü üzere GAZI-HTTP verisetimiz için F1-skoruna göre en iyi sınıflandırma performans ölçümleri SSI, SQLi ve CRLF sınıfları için sırasıyla %98,4, %96.5 ve %93,9 olarak elde edilmiştir. Bununla birlikte, %81.1 ile komut enjeksiyonu (Command Injection) saldırı türü, en düşük F1-skor değerine sahiptir. Komut enjeksiyonu saldırısının çoğunlukla LFI, SQL ve XSS saldırılarıyla karıştırıldığı görülmektedir (bkz. Çizelge 4.9). Bunun sebebi, bu saldırı türlerinin benzer komut sözcükleri içermesidir. Bununla birlikte DA-SANA ile veri artırımı gerçekleştirilmeden ve gerçekleştirilerek eğitilen modeller birbiri ile karşılaştırıldığında,

DA-SANA yöntemine dayalı veri artırma ile saldırı türlerinde, özellikle sınırlı sayıda veriye sahip saldırı türlerinde, önemli bir iyileştirme sağlandığı gözlemlenmektedir. Ayrıca F1-skor değerine göre ECML-PKDD verisetinde sınıf bazında en iyi sınıflandırma başarısı Valid (Normal), XPath ve Ldap enjeksiyonu sınıflarında, sırasıyla %99,8, %94,5 ve %92,9 olarak elde edilmiştir (bkz. Şekil 4.7). Bununla birlikte, SSI saldırı türü %39.2 ile en düşük F1-skor değerine sahiptir. Çok sınıflı sınıflandırmada SSI gibi bazı sınıflar için düşük sınıflandırma başarının sebepleri, ECML-PKDD verisetinin aşırı dengesiz dağılıma sahip olması (bkz. Çizelge 4.5) ve yukarıda açıklandığı üzere diğer saldırı türlerine çok benzer saldırı paternlerine sahip olmasıdır. Çalışma kapsamında aşırı dengesiz veri dağılımından kaynaklanan problemleri çözmek için çapraz doğrulama sürecinde veri karıştırma (data shuffling) ve katmanlı örnekleme (stratified sampling) teknikleri kullanılmıştır. Yüzde bölme ve çapraz doğrulama sonuçlarının birbiri ile uyumlu olduğu ve her iki doğrulama yönteminin sonuçlarında anlamlı bir fark olmadığı görülmektedir.



Şekil 4.7. GAZI-HTTP (a) ve ECML-PKDD (b) verisetleri ile 10-katlı çapraz doğrulamaya göre eğitilen Bi-LSTM modellerinin sınıf bazlı performans değerleri

4.6.2. DA-SANA'nın web anomali tespit modelleri üzerindeki etkinliği

DA-SANA'nın web saldırılarının sınıflandırılmasında sağladığı iyileştirmeyi ve etkinliğini analiz etmek amacıyla bu bölümde modellerin sonuçları karşılaştırmalı olarak analiz edilmiştir. Geliştirilen DA-SANA tekniğinin derin öğrenme algoritmaları üzerindeki

etkinliğini incelemek için Bi-LSTM tabanlı modellere ek olarak LSTM, GRU, CNN ve DNN algoritmalarına dayalı WAF modelleri geliştirilmiştir. DA-SANA uygulanarak ve uygulanmadan eğitilen farklı derin öğrenme modellerinin sınıflandırma doğrulukları karşılaştırmalı olarak Çizelge 4.14' te sunulmuştur. Çizelge 4.14' te kullanılan Bi-LSTM modeli tespit oranları 10-kat çapraz doğrulama ve yüzde bölünme ile eğitilen modellerde en yüksek başarıma sahip model başarım değerlerini içermektedir. Bi-LSTM dışındaki diğer dört DL modeli, yüzde bölme yöntemiyle eğitilmiş ve doğrulanmıştır. Çizelge 4.14 incelendiğinde, DA-SANA' nın çalışma kapsamında farklı DL algoritmalarına dayalı olarak geliştirilen tüm modellerde hem iki sınıflı hem de çok sınıflı web anomali tespiti için önemli iyileştirmeler sağladığı görülmektedir. İki sınıflı web anomali tespiti için önerilen DA-SANA yöntemi sayesinde en düşük ortalama iyileştirme oranına %0,62 ile DNN modelinde, en yüksek ortalama iyileştirme oranına ise %2,9' luk iyileştirme ile CNN modelinde ulaşılmıştır. DA-SANA ile çok sınıflı web anomali tespiti için en düşük ortalama iyileştirme oranı %4,52 ile CNN modelinde sağlanırken, en yüksek ortalama iyileştirmeye %7,77 ile LSTM modelinde ulaşılmıştır. Üç web anomali veriseti üzerinde eğitilen tüm DL modelleri için DA-SANA kullanıldığı durumunlar da sağlanan ortalama iyileştirme oranı hesaplandığında, iki sınıflı modellerde ortalama %1,61 ve çok sınıflı modellerde ise ortalama %6,52 iyileştirme sağladığı görülmektedir. DA-SANA' nın zaten yüksek olan iki sınıflı sınıflandırma performanslarında çok yüksek oranda bir iyileştirme sağlayabilmesi mümkün olmayıp, sağlanan iyileştirme oranı iyidir. Bununla birlikte, DA-SANA ile çok sınıflı web anomali tespiti için yüksek bir iyileştirme sağlandığı görülmektedir. Bu iyileştirmenin en önemli nedeni, yeni oluşturulan gürültülü istek dizisine eklenecek gürültü miktarının web isteklerinin uzunluğuna göre belirlenmiş olmasıdır. Web anomali isteklerinin uzunlukları, web uygulamasının beklediği istek uzunluklarının dağılımında önemli bir değişikliğe neden olmaktadır. DA-SANA ile web istek uzunluklarının varyansı, gürültü miktarı olarak sınıflandırma modeline doğru orantılı olarak aktarılabilmekte ve bu iyileştirme sağlanabilmektedir. Nihai olarak, çalışma kapsamında geliştirilen DA-SANA yöntemi ile web saldırı tespitinde yüksek hassasiyet elde edilebildiği gösterilmiştir.

Çizelge 4.14’de yer alan veriler, DA-SANA' nın önerilen Bi-LSTM modeli için iki sınıflı ve çok sınıflı sınıflandırmada sırasıyla %0,6 ve %6,54' lük bir ortalama iyileştirme sağladığını göstermektedir. Bi-LSTM modeli, çok sınıflı web anomali tespitinde GAZI-HTTP veriseti için en yüksek tespit oranına sahipken, ECML-PKDD için ikinci en yüksek tespit oranına erişmiştir. Bi-LSTM, çift yönlü yapısı sayesinde dizi sınıflandırma problemlerinde daha iyi

performans göstermektedir [73]. Bu sayede, Bi-LSTM mimarisi web isteklerinde hem ileri hem de geriye doğru parametreler ve yükler arasında güçlü bağlantılar oluşturmakta ve böylece web anomali tespiti için daha iyi temsili oluşturan modeller oluşturabilmektedir.

Çizelge 4.14. Farklı DL modelleri üzerinde DA-SANA yönteminin etkinliği

		İKİ SINIFLI		**ÇOK SINIFLI**	
Model	***Veriseti***	***DA-SANA olmadan***	***DA-SANA ile***	***DA-SANA olmadan***	***DA-SANA ile***
Önerilen Bi-LSTM	GAZI-HTTP	97.65%	98.34%	89.76%	96.59%
	ECML-PKDD	98.94%	99.47%	87.66%	93.91%
	CSIC-HTTP	98.35%	99.08%	N/A	N/A
	Ortalama	98.31%	**98.96%**	88.71%	**95.25%**
	Ortalama İyileştirme		**0.65%**		**6.54%**
LSTM	GAZI-HTTP	97.58%	98.19%	87.47%	95.20%
	ECML-PKDD	98.54%	99.37%	81.92%	89.72%
	CSIC-HTTP	92.87%	95.83%	N/A	N/A
	Ortalama	96.33%	**97.79%**	84.69%	**92.46%**
	Ortalama İyileştirme		**1.46%**		**7.77%**
GRU	GAZI-HTTP	97.83%	98.34%	88.25%	92.51%
	ECML-PKDD	98.64%	99.56	82.81%	91.87%
	CSIC-HTTP	93.27%	96.23%	N/A	N/A
	Ortalama	96.58%	**98.04**	85,53	**92.19**
	Ortalama İyileştirme		**1.46%**		**6,66**
CNN	GAZI-HTTP	97.47%	98.26%	85.44%	91.31%
	ECML-PKDD	95.78%	97.71%	91.06%	94.24%
	CSIC-HTTP	92.98%	93.95%	N/A	N/A
	Ortalama	93.74%	**96,64%**	88.25%	**92.77%**
	Ortalama İyileştirme		**2.9%**		**4.52%**
DNN	GAZI-HTTP	98.06%	98.79%	90.22%	93.00%
	ECML-PKDD	99.75%	99.95%	82.42%	93.87%
	CSIC-HTTP	95.07%	96.00%	N/A	N/A
	Ortalama	97.62%	**98.4%**	86.32%	**93.43**
	Ortalama İyileştirme		**0.62%**		**7.11%**
Beş DL modelinin Ortalama İyileştirmesi		**1.61%**		**6.52%**	

N/A: Bulunmuyor

4.6.3. Önceki çalışmalar ile kıyaslama

Çalışma kapsamında geliştirilen modellerin CISC-HTTP ve ECML-PKDD veri kümelerini kullanan önceki anomali tabanlı çalışmalar ile kıyaslamaları Çizelge 4.15’ de sunulmuştur.

Çizelge 4.15. Önceki çalışmaların sonuçları ile kıyaslama

Yazarlar	Model	CSIC	ECML-PKDD	
		İki Sınıflı	İki Sınıflı	Çok Sınıflı
Nguyen ve Franke [25]	Naïve Bayes	72.78%	85.12%	N/A
Nguyen ve Franke [25]	Bayes Network	82.79%	86.95%	N/A
Nguyen ve Franke [25]	RBF Network	72.46%	87.69%	N/A
Nguyen ve Franke [25]	Hedge Boosting	82.10%	86.30%	N/A
Nguyen ve Franke [25]	A-ExIDS	90.98%	92.56%	N/A
Nguyen ve ark. [26]	C4.5	94.49%	96.37%	N/A
Nguyen ve ark. [26][	CART	94.12%	96.11%	N/A
Nguyen ve ark. [26]	Random Tree	92.30%	96.89%	N/A
Nguyen ve ark. [26]	Random Forest	93.71%	***98.80%***	N/A
Tekerek ve ark. [31]	Hybrid Model	94.50%	93.30%	N/A
Raissi ve ark. [21]	Static Method	N/A	87.93%	***69.00%***
Exbrayat [22]	N-similarity	N/A	90.00%	N/A
Liang ve ark. [45]	C4.5	96.50%	N/A	N/A
Liang ve ark. [45]	Modsecurity	55.20%	N/A	N/A
Liang ve ark. [45]	Naive Bayes	84.08%	N/A	N/A
Kozik ve ark. [79]	AdaBost	83.28%	N/A	N/A
Kozik ve ark. [79]	J48	95.97%	N/A	N/A
Mac ve ark. [80]	Autoencoder	98.23%	N/A	N/A
Hao ve ark. [44]	Bi-LSTM	98.35%	N/A	N/A
Liang ve ark. [45]	RNN	85.15%	N/A	N/A
Liang ve ark. [45]	GRU	97.88%	N/A	N/A
Liang ve ark. [45]	LSTM	98.42%	N/A	N/A
Tekerek [43]	CNN	97.07%	N/A	N/A
Zhang ve ark. [42]	CNN	96.49%	N/A	N/A
Ito ve Iyatomi [81]	CNN	***98.80%***	N/A	N/A
Vartouni ve ark. [29]	DBN	89.24%	84.13%	N/A
Liu ve ark. [66]	PL-RNN	96.13%	N/A	N/A
Yu ve ark. [82]	RCNN	97.91%	N/A	N/A
Yu ve ark. [82]	TextCNN	98.48%	N/A	N/A
Mevcut Çalışma [83]	DA-SANA olmadan Bi-LSTM	98.35%	98.94%	87.66%
Önerilen Model [83]	**DA-SANA ile Bi-LSTM**	**99.08%**	**99.47%**	**93.91%**

N/A: Bulunmuyor

Önerilen modele ilişkin Çizelge 4.15' de yer alan tespit oranları, 10-kat çapraz doğrulama ve yüzde bölme yöntemlerinde en yüksek değerleri içermektedir. Hem iki sınıflı hem de çok sınıflı sınıflandırmada en başarılı modelin çalışma kapsamında önerilen DA-SANA yöntemi artırılmış veriseti ile eğitilen Bi-LSTM modeli olduğu görülmektedir. Daha önce gerçekleştirilen çalışmalar incelendiğinde genel olarak derin öğrenme tekniklerinin web saldırı tespitinde başarılı bir şekilde çalıştığını gözlemlenmektedir. En yüksek sınıflandırma başarısı, CSIC-HTTP veriseti üzerinde %99.08 tespit oranıyla önerilen Bi-LSTM model ile sağlanmıştır. ECML-PKDD veriseti üzerinde iki sınıflı sınıflandırma işleminde, önerilen Bi-

LSTM modeli %99.47 ile en yüksek saptama oranına sahipken, ikinci en yüksek tespit oranı %98.8’ dir [26]. Böylece önerilen model yapısı ile web saldırılarının tespitinde önemli bir gelişme sağlanmıştır.

Web saldırı tespiti için çok sınıflı sınıflandırma konusunda az sayıda çalışma bulunmaktadır. Mevcut çalışmalar ve önerilen model karşılaştırıldığında, sınıflandırma başarımızın ECML-PKDD için %93,91 ve GAZI-HTTP verisetimiz için %96,59 başarı oranı ile diğerlerinden etkileyici bir şekilde yüksek olduğu açıktır (bkz. Çizelge 4.15). Saldırı yükleri çok benzer olduğundan, web saldırı tespitinde çok sınıflı sınıflandırma çok zor hale gelmektedir. Ayrıca veri setleri aşırı dengesiz dağıldığı durumlarda saldırı türlerinin tespiti daha zahmetli hale gelmektedir. DA-SANA yöntemi ile uzun payload içeren bazı tipik saldırı isteklerine daha fazla gürültü eklenerek duyarlılık (TP oranı) ve özgüllük (specificity, TN oranı) artırılarak yukarıdaki başarılı tespit oranları elde edilmiştir. Ayrıca, sonuçlar diğer modellerle karşılaştırıldığında, çalışma kapsamında geliştirilen DA-SANA ile Bi-LSTM modelinin web saldırılarının tespitindeki etkinliği açıkça görülmektedir.

4.6.4. Zaman tüketimi ve çalışmanın sınırlılıkları

Çalışmada kullanılan verisetlerinde bulunan bir HTTP isteğinin model eğitimi için gerektirdiği ortalama çalışma süresi Çizelge 4.16’da görülmektedir. DL modelleri bir kez eğitildikten sonra, yeni gelen HTTP isteklerinin model tahmini için kullanılabilir olmakta ve kaynak verimliliği açısında uzun bir süre tekrar eğitilmelerine ihtiyaç duyulmamaktadır. Çizelge 4.16’ da yer alan verilere göre iki sınıflı DNN modeli 0.043 mili saniye (ms) ile en düşük zaman tüketimini gerektirmektedir. Bi-LSTM modelinin ise 1.8 ms ile en yüksek zaman tüketimini gerektirdiğini görülmektedir. Benzer şekilde, DNN modeli 0,053 ms ile çok sınıflı model eğitimi için en düşük zaman tüketimini gerektirirken, Bi-LSTM modeli 2,5 ms ile en yüksek zaman tüketimini gerektirmektedir. Sonuç olarak, Bi-LSTM mimarisinin karmaşık yapısı nedeniyle yüksek hesaplama kaynaklarına duyulan ihtiyaç ve zaman maliyeti bu çalışmanın kritik sınırlamalarıdır. Doğrulama yöntemleri zaman tüketimi açısında kıyaslandığında, çapraz doğrulama yönteminin daha uzun sürdüğü görülmektedir. LSTM tabanlı modellerde zaman tüketimi önemli bir sorun olmakla birlikte, günümüzde güçlü GPU sistemleri ile bu sorunun üstesinden gelinebilmektedir.

Bu çalışma açısında sınırlılık olabilecek bir diğer faktör ise saldırı yüklerinin kodlanarak ya da farklı yollar (dosya yükleme, uzaktan dosya ekleme) kullanılarak gönderilebilmesidir. Bu

durumlar göz önüne alındığında, tüm HTTP istekleri kodlama işleminin yapılıp yapılmadığı kontrol edilebilmekte ve gerekirse WAF sistemine gelen istekler URL ve/veya Base64 kod çözme sürecine tabi tutulabilecektir. Ayrıca, payload olarak yüklenen dosyaların türü kontrol edilerek, DA-SANA' da istek uzunluğu hesaplanırken komutlar (JavaScript gibi) içeren dosyaların boyutunun da dikkate alınabilmesi mümkündür.

Çizelge 4.16. Bir web isteğinin eğitilmesinin ortalama çalışma süresi (milisaniye)

Model	***Doğrulama Yöntemi***	***Veriseti***	***Ortalama çalışma süresi (ms)***	
			İki Sınıflı	***Çok Sınıflı***
Bi-LSTM	*Yüzde Bölme*	GAZI-HTTP	2,60	2,70
		ECML-PKDD	2,30	2,40
		CSIC-HTTP	0,70	N/A
		Ortalama	**1,80**	**2,50**
	10-Katlı Çapraz Doğrulama	GAZI-HTTP	18,40	19,00
		ECML-PKDD	15,20	18,70
		CSIC-HTTP	9,80	N/A
		Ortalama	**14,40**	**18,80**
LSTM	Yüzde Bölme	GAZI-HTTP	2,12	2,14
		ECML-PKDD	2,11	2,07
		CSIC-HTTP	1,07	N/A
		Ortalama	**1,76**	**2,10**
GRU	Yüzde Bölme	GAZI-HTTP	1,91	1,95
		ECML-PKDD	2,13	1,90
		CSIC-HTTP	0.907	N/A
		Ortalama	**1,64**	**1,92**
CNN	Yüzde Bölme	GAZI-HTTP	0.108	0.121
		ECML-PKDD	0.116	0.118
		CSIC-HTTP	0.075	N/A
		Ortalama	**0.099**	**0.119**
DNN	Yüzde Bölme	GAZI-HTTP	0.038	0.050
		ECML-PKDD	0.046	0.055
		CSIC-HTTP	0.046	N/A
		Ortalama	**0.043**	**0.053**

N/A: Bulunmuyor

4.7. Çalışmanın Sonuç Bölümü

Tez kapsamında gerçekleştirilen bu çalışmada, kendi kendini uyarlanabilir gürültü eklemeye dayalı yeni bir veri artırma yöntemi (DA-SANA) ve Bi-LSTM mimarisine dayalı veri web saldırı tespit modelleri önerilmiştir. Bi-LSTM mimarisine dayalı iki sınıflı ve çok sınıflı web anomali tespit modelleri farklı web anomali verisetleri ile başarıyla eğitilmiş ve test edilmiştir. Bu modellerin eğitiminden önce web isteklerinin uzunluklarına göre çalışan DA-SANA yöntemi ile veri artırma işlemi gerçekleştirilmiştir. Çok sınıflı web anomalilerinin tespitinde, farklı türdeki web isteklerinin birbirine yakın olması ve verisetlerinin dengesiz

dağılmasında kaynaklanan hassasiyet sorununa çözüm bulunmuştur. Bi-LSTM mimarisinin güçlü ilişki çıkarma yapısı ve DA-SANA veri artırma yöntemi kullanılarak yüksek tespit başarımlarına ulaşılmıştır. Ayrıca geliştirilen DA-SANA yönteminin etkinliğinin ölçülmesi amacıyla çalışmada önerilen Bi-LSTM modelinin yanı sıra LSTM, GRU, CNN ve DNN tabanlı olmak üzere farklı DL modelleri geliştirilmiş ve performansları ölçülmüştür. Performans değerlendirmeleri ve karşılaştırmalar, kendi verisetimiz GAZI-HTTP ek olarak web güvenliği alanında benchmark olan ECML-PKDD ve CSIC-HTTP verisetleri kullanılarak yüzde bölme ve 10-katlı çapraz doğrulama teknikleri ile gerçekleştirilmiştir. Önerilen DA-SANA yönteminin çok sınıflı web anomali tespitinde önemli bir iyileştirme sağladığı ve GAZI-HTTP ve ECML-PKDD verisetleri için alanda kabul gören (state-of-the-art) beş farklı DL algoritmasının sınıflandırma doğruluğunda ortalama %6.52 artış sağladığı belirlenmiştir. Çok sınıflı sınıflandırma başarısı açısından ECML-PKDD veri setinde %93,91, iki sınıflı sınıflandırmada ise %99,47 gibi yüksek bir başarı oranı elde edilmiştir. Ayrıca CSIC-HTTP veri seti için bu oran çalışma kapsamında %99,08' e yükseltilmiştir. Literatür incelememiz kapsamında, çalışma kapsamında sağlanan bu sınıflandırma performansları önceki çalışmalarla karşılaştırıldığında en iyi değerlere ulaşılmıştır.

Sonraki çalışmalarda, düşük F-skoruna sahip saldırı türlerinin tespit oranlarını iyileştirmek için DA-SANA ile birlikte hibrit derin öğrenme yöntemlerinin kullanılması planlanabilir. Ek olarak, zaman maliyeti önerilen modelin önemli bir sınırlılığı olduğu için, sonraki çalışmalarda çok sınıflı sınıflandırma başarısında daha fazla iyileştirmeye zaman gereksinimi azaltacak çözümlere odaklanılması faydalı olacaktır. İleride yapılacak çalışmalarda DA-SANA ile otokodlayıcı yapısına dayalı olarak denetimsiz öğrenme yöntemlerine dayalı web anomali tespiti gerçekleştirilebilir, ayrıca HTTP isteklerinin yanı sıra etiketsiz web sunucu logları da model eğitim sürecine dahil edilebilir.

5. İKİ AŞAMALI DERİN ÖĞRENME TABANLI KATMANLI TOPLULUK (ENSEMBLE) ÖĞRENMEYE DAYALI WAF MODELİ

Yukarıda açıklandığı üzere web anomali tespit problemi doğası gereği aşırı dengesiz dağılım problemi içermekte olup, bu durum geliştirilen modellerin hassasiyetinin düşük olmasına sebep olmaktadır. Mevcut WAF geliştirme çalışmaları incelediğinde, çok sınıflı anomali tespiti yapan çalışmalarda genellikle tüm sınıfların birlikte ele alındığı ve tek bir model ile çözülmeye çalışıldığı görülmekte olup, bunun da aşağıda açıklanan çeşitli problemlere yol açtığı görülmektedir. Öncelikle anomali tabanlı bir WAF modelinin gerçek zamanlı olarak kullanılabilmesi için tek başına yüksek tespit oranına sahip olması yeterli olmayacaktır. Web uygulamalarının kullanıcı memnuniyeti açsısından hızlı çalışmaları ve gelen isteklere hızlı cevaplar dönmesi germektedir. Bununla birlikte, hassasiyeti yüksek olan modeller genellikle daha sofistike olup, zaman gecikmelerine yol açmaktadır. Bu durum tezin 4. bölümünde önerilen Bi-LSTM modelinin de önemli kısıtlarından birisidir. Basit yapıya sahip ve hızlı çalışan modellerin kullanımlarında ise düşük dağılıma sahip ya da birbirine çok benzer paternler içeren saldırı tiplerinin doğru tespit edilme hassasiyetleri düşmektedir. Sadece normal ve saldırı isteklerinin tespit edildiği iki sınıflı saldırı tespitinde tek algoritmaya dayalı model kullanılması zaman ve yüksek tespit oranı için yeterli olabilmektedir. Bununla birlikte çok sınıflı saldırı tespitinde daha önce modelin görmediği saldırıların tespit edilebilmesi, modelin genelleştirebilme gücünü göstermekte olup, WAF sistemlerinin gerçek zamanlı kullanımı açsısından önemlidir. Ek olarak, çok sınıflı saldırı tespiti, özellikle yüksek varyansa (aşırı uydurma) eğilimlidir ve bu nedenle modellerde genelleme (daha önce görmediği saldırıları tespit etme) sorunu ortaya çıkmaktadır. Ayrıca, dengesiz dağılımın neden olduğu yüksek yanlılık (yetersiz uyum) nedeniyle yaygın olmayan saldırı türlerini sınıflandırmak zordur. Dengesiz dağılıma sahip saldırı tiplerinin yüksek hassasiyet ile tespit edilebilmesi için karmaşık modellerin tasarlanması gerekmekte olup, bu durum da önemli zaman gecikmelere yol açabilmektedir. Bu nedenle, tek bir algoritma tabanlı sınıflandırma modeli ile çok sınıflı web anomalilerinin gerçek zamanlı olarak tespit edilmesinin zor olacağı açıktır. Bu sebeple farklı ve tekrarlayan veriler ile eğitilmiş çok sayıda zayıf modelin birleşimden oluşacak güçlü bir topluluk öğrenmeye (ensemble learning) dayalı model kullanılması, saldırı türlerinin tespiti için daha uygun olacaktır.

Bu çalışma kapsamında yukarıda açıklanan problemleri çözmek amacı ile iki sınıflı sınıflandırma ile saldırı tiplerinin belirlenmesi için gerekli olan çok sınıflı sınıflandırma

aşamalarını birbirinden ayıran, yüksek hassasiyet sağlamak amacı ile farklı derin öğrenme modellerinin güçlerini birleştiren, hızlı ve genelleştirebilme gücüne sahip bir WAF modeli geliştirilmiştir. Çalışma kapsamında, web saldırılarını iki aşamalı olarak tespit edebilen derin öğrenme temelli katmanlı (stacked) ensemble öğrenmeye dayalı yeni bir WAF modeli geliştirilmiştir. Geliştirilen WAF sisteminde, ilk aşamada web istekleri normal ve saldırı olmak üzere iki sınıflı olarak sınıflandırılmaktadır. Böylece normal web istekleri doğrudan web sunucusuna gönderilirken, saldırı istekleri ikici aşamada yer alan ensemble modele aktarılmaktadır. İkinci aşamada ise ilk aşamada saldırı olarak tespit edilen web isteklerinin ait olduğu saldırı türü, geliştirilen katmanlı ensemble modele dayalı olarak çok sınıflı sınıflandırma işlemi ile belirlenmektedir. Böylece saldırı türlerinin belirlenmesinden kaynaklanan karmaşıklık, ikinci aşamaya ötelenmekte ve web uygulamalarının zaman gecikmesi yaşanmadan normal kullanıcılara hizmet vermeye devam etmeleri sağlanmaktadır. Geliştirilen WAF modelinin her iki aşamasında derin öğrenme tabanlı modeller kullanılmıştır.

Yukarıda sunulan mevcut çalışmaların az bir kısmında çok sınıflı saldırı tipi sınıflandırması gerçekleştirilmekte olup, web saldırı tespiti genellikle tek aşamalı ve çok sınıflı bir sınıflandırma modeli ile gerçekleştirilmektedir. Bununla birlikte, WAF sistemlerinin asıl amacı saldırı ve normal trafiği ayırt etmek olduğundan, böl ve yönet tekniğini uygulayarak ilk aşamada ikili sınıf tespiti yaparak sorunu basitleştirmek daha etkili olacaktır. Tez kapsamında yapılan bu çalışmanın gerçekleştirilmesindeki ana motivasyon, anomali tespitinin gerçek zamanlı performansını etkilememek için saldırı tipi tespit sürecinin karmaşıklığının ikinci aşamaya ertelendiği iki aşamalı bir yapının oluşturulmasıdır. Bu çalışmada oluşturulan iki aşamalı WAF mimarisi sayesinde web uygulamalarının gerçek dünyada kullanılabilecek hızda çalışması sağlanmaktadır. Ayrıca saldırı türlerinin sınıflandırılması için geliştirilen güçlü temel öğrenici modelleri bir araya getiren katmanlı ensemble modeli sayesinde diğer çalışmalara göre daha yüksek doğruluk ve hassasiyet elde edilmiştir. Bunlara ek olarak, bu çalışma kapsamında topluluk modeli oluşturulurken istatistiksel yöntemlerle yüksek çeşitlilik sağlayan temel öğrenici modeller seçilerek gerçek dünya sistemleri için yüksek genelleme gücü hedeflenmiştir. Bu çalışma kapsamında tüm aşamalarda yer alan sınıflandırma modelleri için derin öğrenme algoritmaları kullanılmıştır.

Bu çalışmanın alana sağladığı temel katkılarından bazıları aşağıda özetlenmiştir;

- Çalışmada geliştirilen iki aşamalı yapı sayesinde normal kullanıcı isteklerin çok sınıflı sınıflandırmanın gerektirdiği sofistike model tarafından işlenmesine ihtiyaç kalmamıştır. Böylece normal kullanıcı istekleri çok sınıflı sınıflandırmanın oluşturduğu zaman gecikmesinden ve yüksek yanlış pozitif (FP) oranından kurtarılmış, hızlı ve yüksek doğrulukta sınıflandırabilmeleri sağlanmıştır.

- Çalışmada kullanılan entegre katmanlı topluluk öğrenme (integrated stacked ensemble) modeli sayesinde saldırı türleri yüksek hassasiyet ve doğrulukla sınıflandırılmıştır.

- Tez kapsamında oluşturulan ve bu çalışma da kullanılan GAZI-HTTP verisetinin çok kapsamlı olması sayesinde yüksek genelleme yeteneği olan bir web anomali modelinin oluşturulması sağlanmıştır.

- Bu çalışma kapsamında geliştirilen WAF modelleri, benchmark ECML-PKDD veriseti kullanılarak önceki çalışmalarla kıyaslanmıştır. Ayrıca GAZI-HTTP ve ECML-PKDD verisetleri üzerinde, siber güvenlik alanında en yaygın kullanıma sahip (state-of-the-art) derin öğrenme yöntemlerinde DNN, CNN, GRU ve LSTM mimarileri kullanılarak tek algoritmaya dayalı WAF modelleri geliştirilmiş ve önerilen model ile kıyaslamaları yapılmıştır. Bilgimiz dâhilinde, çalışmada önerilen iki aşamalı, derin öğrenme tabanlı katmanlı ensemble dayalı WAF modeli, web saldırı tespitinden en iyi performans değerlerine sahiptir.

- *Tez çalışması kapsamında oluşturulan ve bu çalışma kapsamında kullanılan GAZI-HTTP veriseti ve çalışmanın kaynak kodları, tekrarlanabilirliğin sağlaması ve gelecekteki araştırmalara katkı sağlanması amacıyla diğer araştırmacılarla kamuya açık olarak paylaşılacaktır.
  * https://github.com/mehmetsevri/Two-Stage-Deep-Learning-Based-Stacked-Ensemble-Model-for-Web-Application-Security

Tez kapsamında gerçekleştirilen bu çalışmanın kalan kısmı aşağıdaki gibi beş bölüm halinde düzenlenmiştir. Çalışmanın bundan sonraki bölümünde web isteklerinden öznitelik oluşturma, derin öğrenme algoritmaları ve çalışmada kullanılan ayrıklık (diversity) analizleri hakkında ön bilgiler sunulmaktadır. Üçüncü bölümde, çalışmanın metodolojisi ve önerilen modelin genel yapısı hakkında detaylı bilgiler sunulmaktadır. Dördüncü bölümde gerçekleştirilen deneysel çalışmaların performans değerlendirmeleri tartışmaları ile birlikte sunulmuştur. Ayrıca önerilen iki aşamalı WAF modelinin performansları benzer çalışmalar ve bilinen (state-of-the-art) derin öğrenme algoritmalarına dayalı modeller ile karşılaştırılmıştır. Son bölümde ise çalışmanın sonucu ve gelecekteki araştırmalar için perspektifler sunulmuştur.

5.1. Ön Bilgiler

Bu bölümde web isteklerinden öznitelik çıkarımı, çalışmada kullanılan algoritmalar ve modellerin genelleme gücünün (generalization power) ölçülmesinde kullanılan çeşitlilik (diversity) analizleri hakkında ön bilgiler verilmiştir.

5.1.1. Öznitelik çıkarma (Feature extraction)

Yapay sinir ağı (YSA) modelleri, metin tabanlı verileri doğrudan girdi olarak kabul etmemektedir. Bu nedenle, metin tabanlı web isteklerinin YSA tabanlı ağa beslenebilmesi için öznitelik çıkarma yoluyla sayısal kelime vektörlerine dönüştürülmesi gerekmektedir. Metin tabanlı verilerden öznitelik çıkarımı kelime tokenizasyonu (word tokenization) ve kelime gömme (Word embedding) olmak üzere iki ana adımda gerçekleştirilmektedir. İlk adımda, kelime tokenizasyonu işlemi ile karakter ve karakter setlerinden sayısal kelime tokenleri çıkarılmaktadır. Özellik çıkarımının ikinci aşamasında, web isteklerindeki benzer kelime tokenlerinin temsillerini oluşturan kelime gömme işlemi gerçekleştirilmektedir.

Web saldırı türleri belirli kalıpları içerdiğinden, öznitelik oluşturma önemli hale gelmektedir. Literatürdeki web güvenliği çalışmaları incelendiğinde, özniteliklerin manuel olarak belirlendiği ya da N-gram [22, 29] veya TF-IDF [23] gibi frekans tabanlı yöntemlerin kullanıldığı görülmektedir. N-gram ile öznitelikleri çıkarmak basit olsa da önemli dezavantajları bulunmaktadır [84]. N-gram, web isteklerindeki kalıpların anlamlarını ve ilişkilerini temsil edememektedir. Ek olarak, ***n*** değerini belirlemek önemli bir zorluktur. Anlamlı karakter grupları küçük ***n*** değerlerinde yeterince temsil edilmezken, yüksek ***n*** değerleri seyreklik sorununa neden olmaktadır. TF-IDF yönteminde kelimelerin sıklığı ve kelimelerin bir arada bulunma oranları hesaplanmaktadır. Bununla birlikte, kelimeler ve anlam yapıları arasındaki ilişkiler dikkate alınmamaktadır. Word2Vec, fastText, Glove ve Keras gibi bilinen (state-of-the-art) kelime torbası (bag-of-words) teknikleri ile önemli semantik yapılar ele alınabilmektedir. Kelime torbası teknikleri ile doğal dil işleme (NLP) problemlerinde çok önemli başarımlar sağlanmıştır. Web isteklerinde kelime torbası tekniklerinin kullanımı ile özellikle URL ve payload kısımlarında kelimeler arasında anlamlı bağlantılar oluşturulabilmektedir. Bu çalışma kapsamında kelime tokenlerinin oluşturulmasında Keras tokenizer kullanılmıştır. Çalışmada, word embedding aşamasında ise web isteğindeki kelimelerin sıklığına ve birlikte bulunmalarına odaklanan Keras kelime

gömme yöntemi kullanılmıştır. Böylece web isteklerinin sınıflandırılmasında kullanılan güçlü özellikler oluşturulabilmiştir.

Öznitelik çıkarım işleminden sonra gerçekleştirilen öznitelik seçimi, geleneksel makine öğrenmesi algoritmalarında başarıyı etkileyen en önemli zorluklardan birisidir. Derin öğrenme algoritmaları yapıları sayesinde öznitelik seçme ve boyut küçültme işlemlerini içsel olarak gerçekleştirebilmektedir. Derin öğrenme modelleri, ağırlıklarını azaltarak her katmandaki önemsiz özellikleri kaldırmakta ve önemli özelliklerin sonraki katmanlara aktarılmasını sağlamaktadır. Bu çalışma kapsamında derin öğrenme tabanlı modeller kullanıldığından öznitelik seçimi veya boyut indirgeme işlemi gerçekleştirilmemiştir.

5.1.2. Topluluk öğrenme (Ensemble learning)

Klasik yapay öğrenmede, en başarılı sınıflandırıcı model, bir problem için farklı güncel algoritmalar kullanılarak ayrı modeller oluşturulması ile belirlenmektedir. Algoritmaların iyi çalışabileceği problemler olduğu kadar çözemedikleri zorluklar da bulunmaktadır. Büyük hacimli, çok sınıflı ve dengesiz dağılımlara sahip veri kümelerini tek bir sınıflandırıcı ile sınıflandırmak oldukça zordur. Toplu öğrenme, birden fazla modelin çıktılarını birleştirerek karar vermeyi sağlayan bir yöntemdir. Toplu öğrenme, Surowiecki'nin [85] kolektif zeka teorisine dayanmaktadır. Başarılı bir kolektif zekanın içermesi gereken dört temel kural bulunmakta olup, bunlar; fikir çeşitliliği (diversity of opinion), bağımsızlık (independence), ademi merkeziyetçilik (decentralisation) ve bütünleşmedir (aggregation) [85]. Aynı yapıya sahip bir temel modelin farklı alt öğrenme kümeleri ile eğitilmesiyle bir ensemble model oluşturulabilmektedir. Aynı zamanda, farklı yapılara sahip eğitim modelleri veya tekrarlı veya tekrarsız alt öğrenme kümeleri ile algoritmalar eğitilerek bir ensemble model oluşturulabilmektedir. Bu şekilde, kolektif eylemde işbirliği sağlanarak karmaşık sorunlar çözülebilmektedir [86]. Başarılı bir ensemble modeli oluşturmak için başarılı temel modeller birleştirilmesi gerekmektedir. Bununla birlikte, çeşitliliği yüksek temel öğrenici modeller kullanılarak, öğrenici modellerin çıktılarının birbirleriyle düşük korelasyonlu olması sağlanmalıdır. Temel öğrenicilerin performansı ve çeşitliliği genellikle ters orantılıdır [86].

Yüksek başarıya sahip temel öğreniciler genellikle düşük çeşitlilik içermekte olup, bunun tam tersi de geçerlidir. Ensemble öğrenme genellikle temel sınıflandırıcı modellerin tahminlerinin birleştirilmesine dayanmakta olup, formal açıklamaları Eş. 5.1 – 5.3 de yer alan eşitlikler ile gösterilebilir. Buradaki $\boldsymbol{D_{train}}$ eğitim verilerini, $\boldsymbol{h(X_i)}$ belirli bir

algoritmaya dayalı sınıflandırıcıyı, $\boldsymbol{X_i}$ i. girdiyi, $\boldsymbol{Y_i}$ ise bu girdiye ilişkin model çıktısını ifade etmektedir.

Eğitim Kümesi: $D_{train} = x_n, y_n; n = 1,2,...,N$ (5.1)

Temel Sınıflandırıcılar: $L : D_{train} \rightarrow h(X)$, burada $h(X_i) : X_i \rightarrow \Upsilon_i$ (5.2)

Ensemble Sınıflandırıcı: $\left\{ \begin{array}{l} L_1 : D_{train} \rightarrow h_1(X_1) \\ L_2 : D_{train} \rightarrow h_2(X_2) \\ \quad ... \\ L_T : D_{train} \rightarrow h_t(X_T) \end{array} \right\} \Rightarrow \{h_1(X_1) \cup h_2(X_2) \cup ... h_T(X_T)\}$ (5.3)

Ensemble öğrenme modelleri oluşturmak için farklı yöntemler bulunmakta olup; en bilinenleri Bagging ve Boosting yöntemleridir. Bagging (torbalama) yöntemi, mevcut öğrenme setinden rastgele seçim ve ikame ile üretilen yeni alt öğrenme setleri ile aynı algoritmaya dayalı temel modellerin eğitilmesine dayanmaktadır. En bilinen ve başarılı bagging algoritmalarından birisi Rastgele Orman (Random Forest, RF) dır. Bagging algoritmaları, rastgele veri seçimine rağmen, web istekleri gibi yüksek benzerlik içeren veri setlerinde yüksek korelasyon ve düşük çeşitlilik göstermektedir [87]. Boosting yönteminde zayıf sınıflandırıcılar kullanılmaktadır. Temel öğreniciler sırayla çalışmakta ve her temel öğrenici bir öncekinin hatasını düzeltmek için eğitilmektedir. En bilinen Boosting algoritmaları, AdaBoost, Gradient-Boosting ve XGBoost'tur. Boosting algoritmaları genellikle gürültüden yüksek oranda etkilenmektedir [87]. Saldırganlar genel olarak saldırı yüklerini kodlama (encoding) eğiliminde olduklarından, bu saldırılara karşı boosting yöntemleri yetersiz kalabilmektedir.

Bu çalışmada, farklı derin öğrenme algoritmalarına dayalı katmanlı (stacked) [88] bir ensemble modeli kullanılmıştır. Katmanlı yapı, öğrenme setinden bağımsız olarak temel öğrenicilere yönelik önyargıyı (bias) engelleyebilmektedir [88]. Tüm sınıflandırma modelleri, katmanlama yapısında temel öğrenici olarak kullanılabilmektedir. Ensemble öğrenme, probleme uygun tek bir algoritmaya dayalı olarak farklı parametrelerle eğitilen modeller ile oluşturulabileceği gibi, farklı algoritmalara dayalı modeller ile de oluşturulabilmektedir. Genelleme özelliği, yeni ortaya çıkan saldırı türlerinin anomali tabanlı saldırı tespit sistemleri tarafından tespit edilebilmesi için önemlidir. Ensemble modellerinin genelleme özellikleri, içerdikleri temel öğrenicilerin benzerlikleri (similarities) ve çeşitlilikleri (diversities) ile doğrudan ilişkilidir. Genel olarak, aynı algoritmaya dayalı

temel öğrenicilerin tahminleri, farklı parametreler ve alt veri kümeleri ile eğitilmiş olsalar dahi yüksek benzerliğe sahip olmaktadır. Yüksek çeşitliliği sağlamanın en etkili yöntemi, farklı algoritmalara dayalı temel öğrenici modeller kullanmaktır [89]. Bu nedenle, çok farklı mimarilerde yapılandırılmış, farklı tahminlerde bulunabilen ve dolayısıyla daha az ilişkili tahmin hatalarına sahip olan bir temel öğrenici takımının kullanılması arzu edilir [89]. Bu çalışmanın temel amaçlarından biri, yüksek çeşitlilik ve yüksek genelleme yeteneği ile gerçek zamanlı bir WAF sistemi tasarlamaktır. Bu nedenle çalışmamız kapsamında genelleme özelliği yüksek bir WAF sistemi oluşturmak için CNN ve DNN tabanlı yüksek doğrulukta iki uygun temel öğrenici modeli geliştirilmiştir.

5.1.3. Çalışmada kullanılan DL algoritmaları

Çalışma kapsamında, birinci aşamadaki iki sınıflı sınıflandırmada ve saldırı tiplerinin belirlendiği ikinci aşamadaki çok sınıflı sınıflandırmada DNN ve CNN tabanlı derin öğrenme modelleri kullanılmıştır. Bu kısımda çalışmada kullanılan algoritmalar hakkında özet bilgiler sunulmaktadır.

Derin sinir ağı (Deep Neural Network, DNN)

DNN en temel derin öğrenme algoritması olup, peş peşe gelen çok sayıda katmanda yer alan birbirine bağlı yapay nöronlardan oluşmaktadır [90]. DNN yüksek doğruluk başarımları ile çalışan ve alanda bilinen (state-of-the-art) bir derin öğrenme modelidir. Derin öğrenme modellerinin eğitilmesi genel olarak temel yapay sinir ağlarına göre daha fazla zaman almaktadır. Derin öğrenme mimarileri içinde en hızlı çalışan modeller genellikle DNN mimarisine dayalı modeller olup, hesaplama karmaşıklığı diğer mimarilere göre çok daha düşüktür. DNN modelleri giriş olarak verilen vektörler ile çıktı olarak beklenen vektörler arasında kullanılan aktivasyon fonksiyonuna göre lineer veya lineer olmayan ilişkilerin, katmanlar ve nöronlar arası bağlantılar ile kurulmasını sağlamaktadır. DNN algoritması YSA yapısını temel alan ileri beslemeli bir ağ olup (Feed Forward Neural Networks, FFNN) olup, katman sayısının ve nöron sayısının artırılması (derinlik kavramı) sayesinde karmaşık problemlerin lineer olmayan fonksiyonlar ile başarılı şekilde modellenmesini sağlayabilmektedir [91]. DNN algoritması FNNN' nin çok katmanlı hali olup; $x=(x_1,x_2,...,x_n)$ giriş vektörü, $y=(y_1,y_2,...,y_m)$ beklenen çıkış vektörü, $\mathcal{E}$ beklenen çıkış ile ağın çıkışı arasındaki hata, $f(x)$ ağ yaklaşma fonksiyonu olmak üzere, ağın ileri besleme için çıkış ve hata değerleri Eş. 5.4 - 5.7 deki gibi hesaplanmaktadır [75].

$$y' = f^*(x) \tag{5.4}$$

$$f(x) = f^n\left(f^{n-1}\left(\ldots f^2\left(f^1(x)\right)\right)\right) \tag{5.5}$$

$$y_i' = b_i + \sum_{k=1}^{m} w_{ik} * x_{ik} \tag{5.6}$$

$$\varepsilon_i = y_i - y_i' \tag{5.7}$$

FFNN için ileri besleme işleminden sonra, hesaplanan hata değerlerinin diskriminantları alınarak, bulunan ε hata değeri geriye doğru yayılarak, yeni w ağırlık değerleri Eş. 5.8 deki gibi hesaplanmaktadır [75].

$$\Delta\varepsilon(w) = \left[\frac{\partial\varepsilon}{\partial w_1}, \frac{\partial\varepsilon}{\partial w_2}, \ldots, \frac{\partial\varepsilon}{\partial w_m}\right]^T \tag{5.8}$$

Evrişimli sinir ağı (Convolutional Neural Network, CNN)

CNN algoritması literatürde sık kullanılan yöntemlerden birisi olup, özellikle görüntü işlemede başarılı çalışmaktadır. CNN mimarisinde girdileri en iyi temsil eden özelliklerin belirlenmesini (feautere map) sağlayan ve bunları bir sonraki katmana akataran bir dizi konvülasyon katmanı bulunmaktadır. Konvülasyon katmanında genellikle matris çarpımları veya farklı nokta çarpımlar kullanılmaktadır. Sıklıkla konvülasyon katmanlarının peşine bağlanan havuzlama (Pooling) katmanı ile parametre sayısı azaltılmakta (downsampling) ve uygunluk kontrol edilmektedir. Havuzlama işleminde, genellikle maksimum havuzlama veya ortalamaya göre indirgeme yapılmaktadır. CNN mimarilerinin genellikle en sonunda tam bağlı bir katman (fully-connected layer) ve bundan sonra gelen sınıflandırma katmanı bulunmaktadır. CNN mimarisi, daha önce görüntü işlemede kullanılan filtrelere benzer yapılar konvülasyon ve havuzlama katmanları ile otomatik olarak oluşturularak en iyi niteliklerin belirlendiği bir mimari ortaya çıkarılabilmektedir. Bu mimari geri besleme ile sürekli parametreleri güncelleyerek en az hata değerine yakınsamaya çalışmaktadır [92].

5.1.4. Katmanlı ensemble öğrenme modellerinin istatistiksel benzerlikleri ve çeşitliliklerinin analizi

Temel modellerin benzerlikleri, korelasyonları ve farklılıkları, ensemble modelinin genelleme yeteneği için en etkili faktörlerdir. Ensemble modelin gerçek dünya problemlerini

simüle etme yeteneği, temel modellerin yüksek başarısına ve çeşitliliğine bağlıdır. Bununla birlikte, iki faktör genellikle birbiri ile ters orantılıdır. Temel modellerin başarıları arttıkça çeşitlilikleri azalır ve bunun tam tersi de durum gerçekleşir. Ensemble modelleri oluştururken performans ve çeşitlilik arasındaki dengeyi / ödünleşimi (trade-off) sağlamak önemlidir.

İki sınıflı modellerin benzerlik ve çeşitlilik analizlerinde kullanılan birçok istatistiksel analiz yöntemi bulunmaktadır. Ancak, bilgimiz dâhilinde, çok sınıflı sınıflandırma modellerin çeşitlilik analizi için genel bir yöntem bulunmamaktadır. Mevcut iki sınıflı modellerin benzerlik ve çeşitlilik yöntemlerinin probleme göre çoklu model ve çoklu sınıfa uyarlanması gerekmektedir. Bu çalışmada, temel modellerin benzerlik ve çeşitliliğini ölçmek için beş farklı istatistiksel yöntem kullanılmıştır. Bunlar; Cohen kappa katsayıları, Pearson korelasyon p katsayısı (Pearson correlation p-value), Yule Q istatistikleri, uyuşmazlık ölçütü (Diversity) ve çift hata (double fault) yöntemleridir. Çalışma kapsamında katmanlı modellerin çeşitliliğinin analiz edilmesinde kullanılan bu yöntemlerin detayları aşağıda sunulmuştur.

Cohen Kappa katsayıları

Sınıflandırma modellerinin birbiri ile uyuşmasının analizde sık kullanılan yöntemlerden birisi Cohen Kappa katsayısıdır. Cohen'in kappa katsayısı iki değerleyici arasındaki karşılaştırmalı uyuşmanın güvenirliğini ölçen bir istatistik yöntemidir. Cohen'in kappa ölçüsü her biri $\boldsymbol{N}$ tane maddeyi $\boldsymbol{C}$ tane birbirinden karşılıklı hariç olan kategoriye ayıran iki değerleyicinin arasında bulunan uyuşmayı ölçmektedir. Ortaya çıkan kategorik bir değişken olduğu için parametrik olmayan bir istatistik türüdür. Kappa katsayısı 0-1 arasında bir değer almakta olup, nasıl yorumlandığı Çizelge 5.1' de görülmektedir. İki sınıflandırıcı arasındaki Cohen kappa katsayısı $\boldsymbol{K_{i,j}}$ Eş. 5.9' daki gibi hesaplanmaktadır [93]. Buradaki, $\boldsymbol{\rho_0}$ sınıflandırıcılar arasında gözlemlenen göreli anlaşmaları, $\boldsymbol{\rho_e}$ ise iki sınıflandırıcının şans eseri anlaşmalarının varsayımsal olasılığını ifade etmektedir. Çalışma kapsamında katmanlı modellerin kappa istatistikleri, temel sınıflandırıcıların ortalama kappa puanları olarak ölçülmüştür. $\boldsymbol{M}$ temel sınıflandırıcıdan oluşan bir topluluk modelinin ortalama kappa katsayısı $\boldsymbol{K'}$ nın hesaplanması Eş. 5.10' da görülmektedir [93].

$$\kappa_{i,j} = \frac{p_0 - p_e}{1 - p_e} = 1 - \frac{1 - p_0}{1 - p_e} \tag{5.9}$$

$$\kappa = \frac{2}{M(M-1)} \sum_{i=1}^{M} \sum_{j=i+1}^{M} \kappa_{i,j} \tag{5.10}$$

Çizelge 5.1. Kappa katsayısının yorumlanması

$\boldsymbol{K}$	Yorum
0	Hiç uyuşma yoktur
0.0 — 0.20	Önemsiz uyuşma var
0.21 — 0.40	Orta derecede uyuşma var
0.41 — 0.60	Ekseriyetle uyuşma var
0.61 — 0.80	Önemli derecede uyuşma olması
0.81 — 1.00	Neredeyse mükemmel uyuşma olması

Pearson korelasyon p katsayısı (Pearson correlation p-values)

Pearson korelasyon katsayısı, iki veri kümesi arasındaki doğrusal ilişkiyi ölçmektedir. Pearson korelasyonu her veri kümesinin normal olarak dağıtılması şartını gerektirmektedir. Diğer korelasyon katsayıları gibi, p değeri de -1 ile +1 arasında değişmektedir. Korelasyon katsayısı p' nin değerinin 0 olması iki sınıflandırıcı arasında korelasyon olmadığını göstermektedir. Bununla birlikte p değeri -1 veya +1 olması durumları ise sırasıyla tam negatif veya pozitif bir doğrusal ilişki anlamına gelmektedir. Pozitif korelasyonlar, x arttıkça y'nin de arttığını gösterir. Negatif korelasyonlar, x arttıkça y'nin azaldığını gösterir. Çalışmada kullanılan Pearson katsayısı, çok sınıflı anomali tespiti için literatürdeki iki sınıflı sınıflandırıcıların benzerliğinin ölçülmesinden yola çıkılarak elde edilmiştir. Katmanlı ensemble modelinin $\boldsymbol{p}$ katsayısı, etiket sayısı ve temel sınıflandırıcı sayısı ile ölçeklenen ikili $\boldsymbol{p}$ katsayılarının ortalaması olarak hesaplanmaktadır. Pearson benzerlik katsayılarının hesaplanmasında Çizelge 5.2'de yer alan korelasyon analiz matrisi kullanılmıştır Pearson p katsayısı aşağıdaki terimler kullanılarak hesaplanır [94].

- $\boldsymbol{N^{TT}}$: Hem Sınıflandırıcı-1 hem de Sınıflandırıcı-2 tarafından birlikte doğru olarak sınıflandırılan toplam tahmin sayısı.

- $\boldsymbol{N^{TF}}$: Sınıflandırıcı-1 tarafından doğru sınıflandırılırken, Sınıflandırıcı-2 tarafından yanlış sınıflandırılan tahminlerin sayısı.

- $\boldsymbol{N^{FT}}$: Sınıflandırıcı-1 tarafından yanlış sınıflandırılırken, Sınıflandırıcı-2 tarafından doğru sınıflandırılan tahminlerin sayısı.

- $\boldsymbol{N^{FF}}$: Hem Sınıflandırıcı-1 hem de Sınıflandırıcı-2 tarafından birlikte yanlış sınıflandırılan toplam tahmin sayısı.

Çizelge 5.2. Korelasyon analiz matrisi

		Sınıflandırıcı-2	
		True	False
Sınıflandırıcı-1	True	N^{TT}	N^{FT}
	False	N^{TF}	N^{FF}

İki sınıflandırıcı arasındaki korelasyon katsayısı $\boldsymbol{\rho_{i,j}}$ Eş. 5.11' daki gibi hesaplanmaktadır. Temel öğrenici model çiftlerinin korelasyonlarına dayalı katmanlı modelin ortalama korelasyon katsayısı $\boldsymbol{\rho}$ Eş. 5.12' deki gibi hesaplanmaktadır [94].

$$\rho_{i,j} = \frac{N^{TT}N^{TF} - N^{TF}N^{FT}}{\sqrt{\left(N^{TT}+N^{TF}\right)\left(N^{FT}+N^{FF}\right)\left(N^{TT}+N^{FT}\right)\left(N^{TF}+N^{FF}\right)}} \tag{5.11}$$

$$\rho = \frac{2}{M(M-1)}\sum_{i=1}^{M}\sum_{j=i+1}^{M}\rho_{i,j} \tag{5.12}$$

Yule'un Q istatistikleri

Q istatistikleri Ki-kare dağılımına dayalı etkili bir çeşitlilik analizi testidir. İki sınıflandırıcı arasındaki Q katsayısı Eş. 5.13'deki gibi hesaplanmaktadır [93, 94]. M adet temel öğrenici modelden oluşan ensemble model için ortalama Q katsayısı Eş. 5.14' deki gibi hesaplanmaktadır [94].

$$Q_{i,j} = \frac{N^{TT}N^{TF} - N^{TF}N^{FT}}{N^{TT}N^{TF} + N^{TF}N^{FT}} \tag{5.13}$$

$$Q = \frac{2}{M(M-1)}\sum_{i=1}^{M}\sum_{j=i+1}^{M}Q_{i,j} \tag{5.14}$$

Anlaşmazlık ölçütü (Disagreement Measure, Diversity)

Anlaşmazlık ölçüsü (disagreement measure) bir sınıflandırıcı ile diğer sınıflandırıcının ayrıklıklarını ölçen önemli bir istatistiksel metriktir. İki sınıflandırıcının birbirlerinin tersine karar verdikleri durumların toplam durum sayısına oranı ile hesaplanmaktadır. İki sınıflandırıcının anlaşmazlık ölçüsü Eş. 5.15' deki gibi hesaplanmaktadır [93, 94]. ***M*** adet temel sınıflandırma modelinden oluşan ensemble modelin ***N*** adet elemandan oluşan test seti üzerindeki ortalama anlaşmazlık ölçüsü Eş. 5.16' daki gibi hesaplanmaktadır [94].

$$dis_{i,j} = \frac{N^{TF} + N^{FT}}{N^{TT} + N^{TF} + N^{FT} + N^{FF}} \tag{5.15}$$

$$dis = \frac{2}{M(M-1)} \sum_{i=1}^{M} \sum_{j=i+1}^{M} dis_{i,j} = \frac{2}{NM(M-1)} \sum_{i=1}^{M} \sum_{j=i+1}^{M} N^{TF} + N^{FT} \tag{5.16}$$

Çift hata (Double Faults)

Temel sınıflandırıcıların aynı girdiler için yanlış etiketle sınıflandırma yapması, topluluk modelinin başarısını en olumsuz etkileyen faktördür. İki sınıflandırıcı arasındaki çift hata oranı Eşit 5.17'deki gibi hesaplanmaktadır. ***M*** adet temel öğreniciden oluşan bir ensemble model için çift hataların ortalaması Eş. 5.18' deki gibi hesaplanmaktadır.

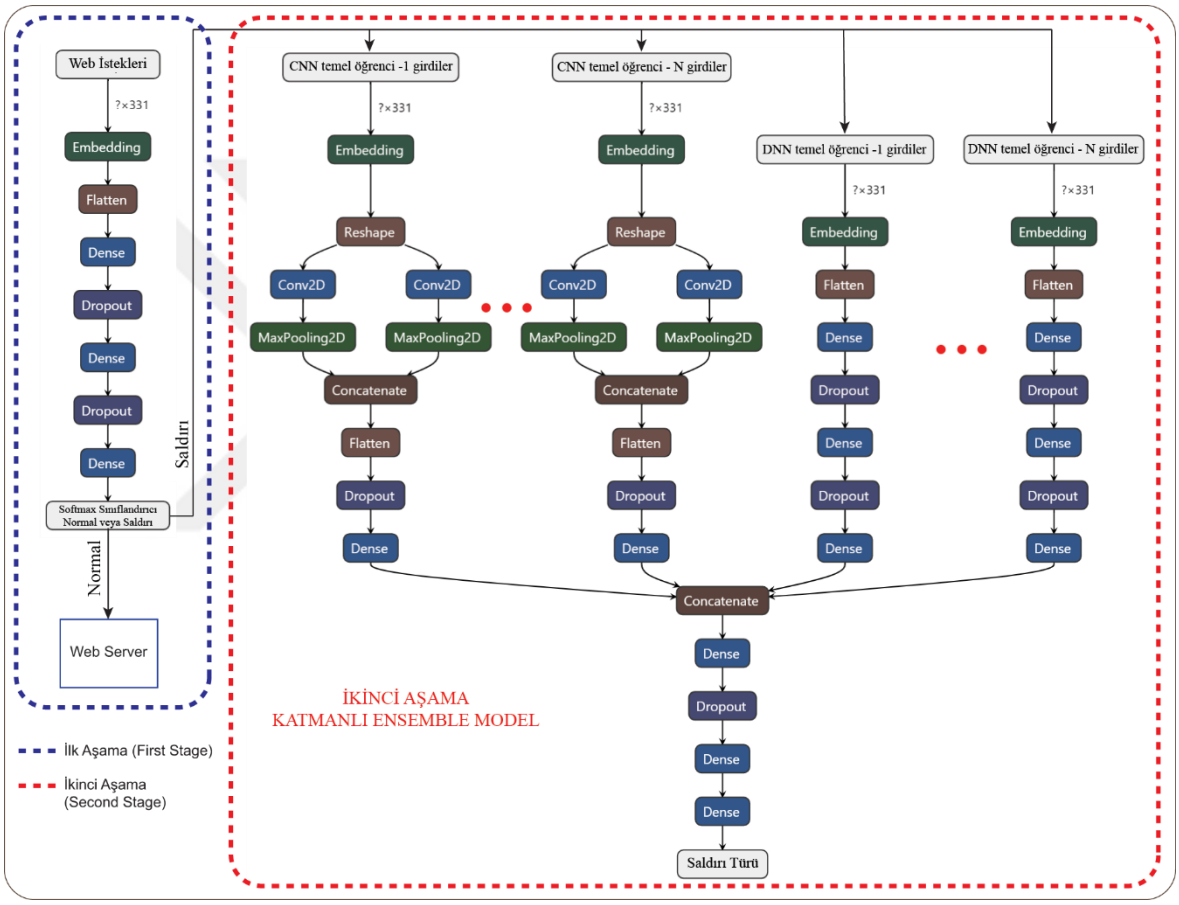
$$DF_{i,j} = \frac{N^{FF(aynı-sınıf)}}{N^{TT} + N^{TF} + N^{FT} + N^{FF}} \tag{5.17}$$

$$DF = \frac{2}{M(M-1)} \sum_{i=1}^{M} \sum_{j=i+1}^{M} DF_{i,j} = \frac{2}{NM(M-1)} \sum_{i=1}^{M} \sum_{j=i+1}^{M} N^{FF(aynı-sınıf)} \tag{5.18}$$

5.2. Metodoloji

Bu çalışma kapsamında web anomalilerini iki aşamada tespit edebilen yeni bir WAF sistemi geliştirilmiştir. Çalışmada önerilen modelin mimarisi Şekil 5.1'de görülmektedir. Önerilen web anomali tespit yönteminin ilk aşamasında, geliştirilen iki sınıflı DL modeli ile HTTP istekleri normal veya anormal olarak sınıflandırılmaktadır. İlk aşamada DNN tabanlı hızlı ve etkili bir model ile iki sınıflı sınıflandırma gerçekleştirilmektedir. Normal etiketi ile sınıflandırılmış kullanıcı istekleri, işlenmek ve kullanıcıya cevap dönülmek amacı ile doğrudan web sunucusuna gönderilmektedir. Saldırı olarak algılanan web istekleri ağdan

düşürülmekte ve saldırı türünün tespit edilmesi amacı ile isteğin bir kopyası ikinci aşamadaki ensemble modele iletilmektedir. Böylece normal kullanıcılar, saldırı türlerini sınıflandırmanın karmaşıklığından kaynaklanan zaman ve performans maliyetlerinden etkilenmemektedir. İkinci aşamada, daha önce anormal olarak tespit edilen saldırı istekleri, daha sofistike bir derin öğrenme tabanlı katmanlı ensemble modeli ile sınıflandırılmaktadır. Önerilen WAF modelinin web anormali tespit aşamalarını içeren sözde kodu (pseudocode) Çizelge 5.3' de görülmektedir.



Şekil 5.1. Geliştirilen iki aşamalı DL tabanlı katmanlı ensemble modelinin mimarisi

Çalışmada önerilen modelin akış şeması Şekil 5.2' de görülmektedir. Şekil 5.2' de görüldüğü üzere web anomali tespit modelinin geliştirilmesi iki aşamada gerçekleştirilmiştir. Öncelikle web isteklerinin ön işlemesi ve kelime tokenizasyonu işlemleri gerçekleştirilmiştir. İlk aşamada, web anomali veri setleri ile eğitim ve test gerçekleştirilerek DNN tabanlı iki sınıflı bir web anomali tespit modeli geliştirilmiştir. İkinci aşamada, sadece saldırı istekleri kullanılarak CNN ve DNN modellerinin eğitimi ve testi gerçekleştirilerek, temel öğrenici modeller oluşturulmuş ve bu temel öğrenici modellerinin tahminleri kullanılarak DNN tabanlı bir meta öğrenici modeli eğitilmiştir.

Çizelge 5.3. Geliştirilen WAF modelinin sözde kodu

Girdi: Web istekleri $\Omega \leftarrow \{x_1, x_2, ..., x_m\}$ $\Omega \leftarrow \{(x_1, x_2 \ldots, x_m)\}$; x HTTP istekleri,

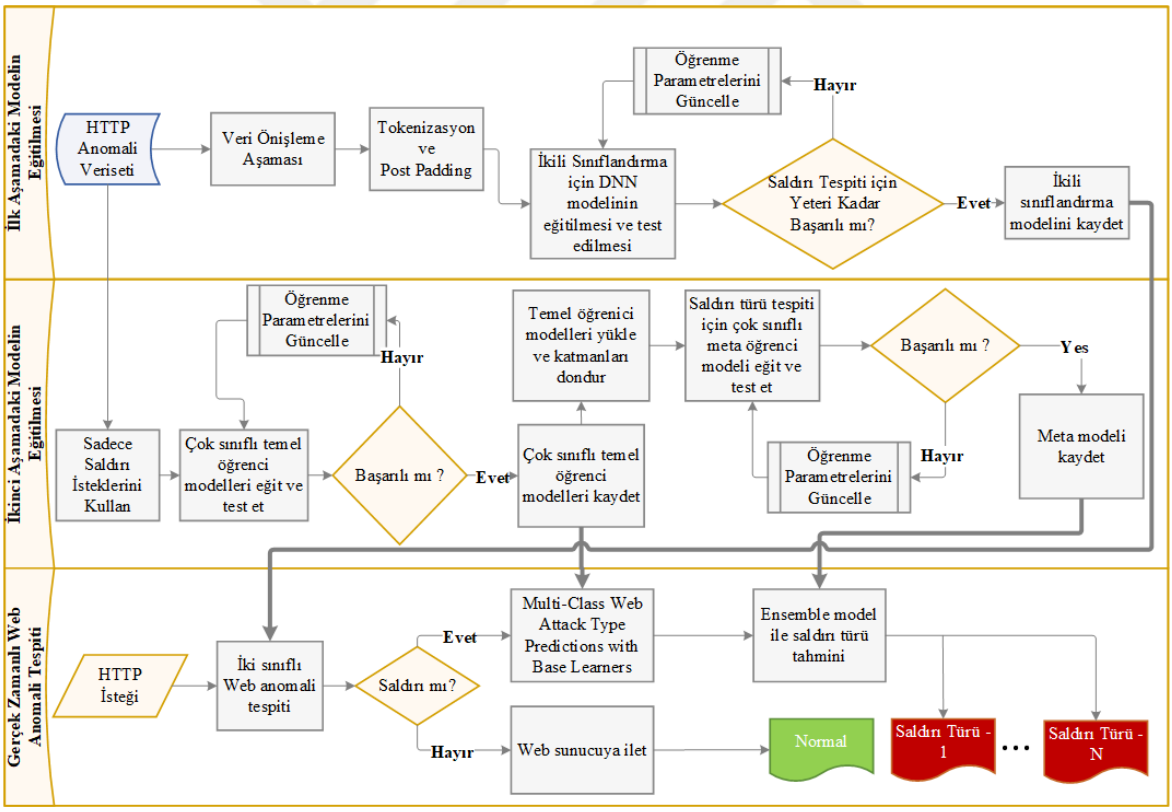
Sequences: $S \leftarrow \{(s_{11}, s_{12}, ..., s_{1n}), ..., (s_{m1}, s_{m2}, ..., s_{mn})\}$; s tokenize edilmiş web istekleri

Embedding vektörü: $E_{m,n,k} \leftarrow$ k embedding boyutuna dayalı oluşturulan kelime gömme vektörleri

$\leftarrow \forall x_i \in \Omega$

Çıktı: P_m ; verilen HTTP istekleri için Normal ya da Saldırı türü etiketleri

1. **for** $\forall R_i \in E_{m,n,k}$ $\forall x_i \in \Omega$ **do**
2. **Predict:** $P_i \leftarrow Model_{FIRST-STAGE}(R_i)$ iki sınıflandırma modeli ile saldırı tespiti
3. **if** $P_i =$ "*Normal*" **do**
4. x_i "Normal" olarak sınıflandırılan istekleri işlenmesi için web sunuuya ilet
5. **else do** // Eğer HTTP isteği x_i ilk aşamada "Saldırı" olarak sınıflandırıldı ise $R_i \leftarrow N_i{*}C_i + S_i{*}C_i'$
6. **Drop:** x_i HTTP isteğini ağdan düşür ve R_i ikinci aşamadaki modele ilet
7. **Predict:** $P_i \leftarrow Model_{SECOND-STAGE}(R_i)$ katmanlı ensemble modeli kullanarak saldırı türünü tahmin et
8. **end for**



Şekil 5.2. Geliştirilen WAF modelinin eğitim ve anomali tespit aşamalarının akış şeması

Modellerin birinci ve ikinci aşamalardaki teorik yapıları ve matematiksel temsilleri aşağıda sunulmuştur. Ayrıca modellerin geliştirilmesinde kullanılan temel yöntem ve teknikler açıklanmıştır.

5.2.1. Birinci aşama: iki sınıflı web saldırı tespiti

Web uygulamaları zamana duyarlı sistemler olduğundan, normal istemcilerin web uygulamasına yönelik isteklerine mümkün olduğunca çabuk yanıt verilmesi önemlidir. Çalışma kapsamında geliştirilen WAF sistemin ilk aşaması, saldırıların tespit edilmesi ve önlenmesi açısından oldukça önemlidir. Web isteklerinin web sunucusuna gönderilmesine ya da ağdan düşülmesine bu aşamada karar verilmektedir. Bu nedenle sınıflandırma başarısı yüksek, FP oranı düşük, istekleri işleme hızı çok yüksek bir model geliştirilmesi hedeflenmiştir. Çalışma kapsamında, CNN, DNN, GRU ve LSTM derin öğrenme algoritmalarına dayalı farklı iki sınıflı modeller eğitilmiş ve test edilmiştir. DNN modeli, her iki veri kümesi için en hızlı ve en etkili model olarak belirlenmiştir. En yüksek hesaplama güç tüketiminin sırasıyla LSTM ve GRU tabanlı modeller için gerekli olduğu görülmüştür. Çalışma kapsamında, ilk olarak veriseti önişleme aşamasında web isteklerindeki yer alan harfler dışındaki sayısal ve özel karakterlerin temizlenmesi gerçekleştirilmiştir. Bu aşamada web isteklerinde yer alan kelimeler özel karakterlere ve boşluklara göre ayrılarak kelime dizileri oluşturulmuştur. DNN modelinin işleyebilmesi için web istekleri kelime tokenizasyon işlemi ile sayısal değerlere dönüştürülmüştür. DNN modelinin eğitim süreci, veri setlerinde normal ve anormal olmak üzere iki farklı etiketle web istekleri kullanılarak gerçekleştirilmiştir. DNN modelinin ilk katmanında, tokenize edilmiş kelimeler arasındaki ilişkilere dayalı olarak kelime vektörlerini oluşturan kelime gömme tabakası bulunmaktadır. İlk aşamada gerçekleştirilecek iki sınıflı saldırı tespiti için geliştirilen DNN modelinin mimari yapısı Çizelge 5.4' de görülmektedir. Birinci aşamada, iki sınıflı sınıflandırmada kullanılacak en iyi modeli oluşturmak için çok sayıda model, farklı sayıda katman ve nörona dayalı olarak test edilmiştir. Çalışmamızda Çizelge 5.4' deki DNN mimarisine sahip modelin ilk aşama için yeterli performansa sahip en hızlı model olduğu tespit edilmiştir. Model, kelime gömme katmanını izleyen iki DNN katmanından ve çıktıda Softmax fonksiyonu içeren bir tahmin katmanından oluşmaktadır.

DNN modelinin optimizasyonu için ilk aşamada ince ayarlama (fine-tuning) yapılmıştır. Aşırı öğrenmeyi önlemek için, her iki DNN katmanından sonra 0.01 oranında dropout uygulanmıştır. DNN katmanlarında aktivasyon fonksiyonu olarak tanh kullanılmaktadır. Modelin optimizasyonu için 0.001 öğrenme oranına sahip Adam optimizasyon algoritması kullanılmıştır. Kayıp fonksiyonu olarak ikili çapraz entropi kullanılmıştır. Eğitim aşamasında her yinelemede kullanılacak parti boyutu (batch size) 1024 olarak ayarlanmıştır.

Çizelge 5.4. İlk aşamadaki iki sınıflı DNN modelinin mimarisi

Katman	**Girdi boyutu**	**Çıktı boyutu**	**Aktivasyon fonksiyonu**
Embedding	(m, n)	$(m, n, 8)$	-
Flatten	(m,n,8)	(m, n*8)	-
Dense-1	$(m, n * 8)$	$(m, 64)$	tanh
Dropout-1	$(m, 64)$	$(m, 64)$	-
Dense-2	$(m, 64)$	$(m, 64)$	tanh
Dropout-2	$(m, 64)$	$(m, 64)$	-
Dense	$(m, 64)$	$(m, 2)$	Softmax

5.2.2. İkinci aşama: web saldırı türlerinin sınıflandırılması için DL tabanlı katmanlı ensemble modeli

Geliştirilen WAF modelinin İkinci aşamasında, ilk aşamada saldırı olduğu belirlenen web isteklerinin saldırı türleri belirlenmektedir. Bu nedenle, ikici aşamadaki modeller sadece web saldırı istekleri kullanılarak eğitilmiş olup, normal web istekleri dâhil edilmemiştir. Katmanlı ensemble öğrenme modelinde kullanılan temel öğrenici (base learner) modellerin ve meta öğrenici (meta learner) modelin mimari detayları aşağıda ayrı ayrı açıklanmıştır.

Temel sınıflandırıcılar (base classifiers)

Temel öğrenicilerin oluşturulmasında CNN, DNN, GRU ve LSTM derin öğrenme algoritmaları ile farklı varyasyonlarda modeller geliştirilmiş ve test edilmiştir. GRU ve LSTM tabanlı temel öğrenici modeller, katmanlı ensemble modelin performansını ve zaman tüketimini olumsuz etkilediği görülmüştür. CNN ve DNN algoritmalarının birlikte kullanılmasının, saldırı türlerinin tespit edilmesinde, performans ve zaman tüketimi açısından daha verimli olduğu görülmüştür. Bu nedenle, temel modeller CNN ve DNN mimarilerine dayalı olarak geliştirilmiştir. Çalışmada kullanılan CNN ve DNN temel öğrenicilerinin mimari yapıları Çizelge 5.5' de görülmektedir. Değişen sayıda temel modelle farklı katmanlı ensemble öğrenme modelleri geliştirilmiştir. Katmanlı ensemble modeller sırasıyla 2, 4, 6, 8 ve 10 adet temel öğrenici model içerek şekilde oluşturulmuştur. Tüm ensemble modellerin içerdiği CNN ve DNN öğrenicilerinin sayısı birbirine eşittir (örneğin, altı adet temel sınıflandırıcı modelin, üçü CNN, kalan üçü ise DNN temel öğrenici modelden oluşmaktadır). Katmanlı ensemble modellerinin performansı ve genelleştirebilme yeteneği yukarıda açıklanan farklı istatistiksel yöntemlerle test edilmiştir. Bir ensemble modeli

oluşturan temel öğrenici modellerde, fazla uyum ve yüksek korelasyondan kaçınmak için öğrenme setinin tekrarlanmayan farklı alt kümeleriyle eğitim işlemi gerçekleştirilmiştir.

CNN temel öğrenici modelleri

Bu çalışmada, web saldırı türünü yüksek tespit oranıyla sınıflandırmak için CNN tabanlı bir temel öğrenici geliştirilmiştir. CNN temel öğreniciler, iki boyutlu evrişim katmanından ve altlarında yer alan maksimum havuzlama katmanından içeren yapıların çıktılarını birleştiren bir birleştirme katmanından oluşmaktadır. Temel öğrenicilerin sonunda, sınıflandırma için bir softmax katmanı bulunmaktadır.

DNN temel öğrenici modelleri

Çalışmada, web saldırı türlerinin sınıflandırılması için güçlü bir DNN temel öğrenici modeli oluşturulmuştur. DNN tabanlı öğrenici modellerinin giriş kısmında, kelime gömme katmanından gelen üç boyutlu vektör çıktılarını iki boyuta indiren bir düzleştirme katmanı bulunmaktadır. Ek olarak, DNN temel öğrenicileri, sırasıyla iki gizli DNN katmanından ve sınıflandırma için bir softmax katmanından oluşmaktadır.

Çizelge 5.5. İkinci aşamada temel öğrenen modellerinin mimarileri.

Temel Öğrenici	**Katman**	**Giriş Katmanı**	**Giriş Boyutu**	**Çıkış boyutu**	**Aktivasyon Fonksiyonu**
CNN	Embedding	Inputs	(m, n)	$(m, n, 16)$	-
	Reshape	Embedding	(m,n,16)	(m, n, 16, 1)	-
	Conv2D - 1	Reshape	$(m, n, 16,1)$	$(m, n, 1, 512)$	ReLU
	Conv2D - 1	Reshape	$(m, n, 1,512)$	$(m, n - 1, 1,512)$	ReLU
	MaxPooling2D - 1	Conv2D - 1	$(m, n, 1,512)$	$(m, 1,1,512)$	-
	MaxPooling2D - 2	Conv2D - 2	$(m, n - 1, 512)$	$(m, 1,1,512)$	-
	Concatenate	Pooling2D – 1 and Pooling2D - 2	$(m, 1,1, 512)$	$(m, 1,2, 512)$	-
	Flatten	Concatenate	$(m, 2,1, 512)$	$(m, 1024)$	-
	Dense	Flatten	$(m, 1024)$	$(m, 8)$	Softmax
DNN	Embedding	Inputs	(m, n)	$(m, n, 16)$	-
	Flatten	Embedding	(m,n,16)	(m, n*16)	-
	Dense-1	Flatten	$(m, n * 16)$	$(m, 32)$	ReLU
	Dropout-1	Dense-1	$(m, 32)$	$(m, 32)$	-
	Dense-2	Dropout-1	$(m, 32)$	$(m, 32)$	ReLU
	Dropout-2	Dense-2	$(m, 32)$	$(m, 32)$	-
	Dense	Dropout-2	$(m, 32)$	$(m, 8)$	Softmax

Meta Öğrenici Model (Meta Learner)

Çalışmada geliştirilen web saldırı türü tespit modeli, entegre bir katmanlı ensemble modelidir. Entegre katmanlı ensemble modelinde, girdilerin bir kopyası her temel öğreniciye ayrı ayrı verilmektedir. Daha sonra temel öğrenicilerin tahminleri birleştirilerek, meta-öğrenci modeline girdi olarak verilmektedir. Katmanlı yöntem, modellerin bir araya toplanmasını ve tek bir büyük model olarak çalışabilmesini sağlamaktadır. Bu yaklaşım, meta-öğreniciyi temel model tahminlerinin en iyi nasıl birleştirileceği konusunda eğitmeye dayanmaktadır. Sabit oylama (hard voting), yumuşak / ağırlıklara dayalı oylama (soft voting), doğrusal regresyon, Naive Bayes (NB) gibi yaygın olarak kullanılan yöntemler, derin öğrenme modelleri veya farklı yapay öğrenme algoritmaları meta öğrenici olarak kullanılabilmektedir. Bu çalışma kapsamında, temel modellerin çıktılarını en başarılı şekilde birleştiren DNN modeli, geliştirilen katmanlı ensemble modeli için meta öğrenici olarak kullanılmaktadır. Meta-öğrenici modelimiz, sırasıyla iki gizli DNN katmanından ve bir softmax çıktı katmanından oluşmaktadır.

Çalışma kapsamında katmanlı ensemble modelin oluşturulması iki adımda gerçekleştirilmiştir. Öncelikle temel (base) modeller eğitildikten sonra ağırlık değerleri ve model parametreleri ile kaydedilmiştir. İkinci adımda, temel modeller ağırlık değerleri ile birlikte yüklenerek, meta sınıflandırıcı ile birlikte katmanlı ensemble model hazır hale getirilmiştir. Temel modellerin tüm katmanları dondurulmuş ve ağırlık değerlerinin değişmesi engellenmiştir. Böylece sadece meta sınıflandırıcı modelin eğitilmesi sağlanmıştır. Meta öğrenici model kendisi için ayrılan eğitim kümesi ile eğitilerek, model parametreleri ve ağırlık değerli kaydedilmiştir.

İkinci aşama için model optimizasyonu ve ince ayarlama (fine-tunning)

Bu çalışma kapsamında, temel ve meta öğrenen modellerin performansını artırmak için farklı optimizasyon yöntemleri kullanılmıştır. Temel modellerde yer alan katmanlarda Relu aktivasyon fonksiyonu kullanılmıştır. Modellerin optimizasyonunda öğrenme oranı 0.001 olan Adam optimizasyon algoritması kullanılmıştır. Modellerin eğitiminde kayıp fonksiyonu olarak kategorik çapraz entropi kayıp fonksiyonları kullanılmıştır. Modellerin eğitim ve testlerinde her yineleme için parti boyutu 1024 olarak belirlenmiştir. Son olarak, çalışmada geliştirilen ensemble modelin içerdiği tüm temel ve meta model katmanlarını birlikte eğitmek ve uyumlu hale getirmek için ensemble modelin ince ayarlama (fine-

tunning) işlemi yapılmıştır. Bu işlemin çok düşük bir öğrenme hızında ve birkaç döngü de gerçekleştirilmesi gerekmektedir. Öncelikle, meta-learner modelinin eğitimi tamamlandıktan sonra katmanlı modelin tüm katmanları tekrar etkin hale (defrozen) getirilmiştir. Çalışma kapsamında ince ayarlama işlemi iki döngüde (epoch), 1×10^{-5} öğrenme oranı ile gerçekleştirilmiştir. Böylece temel öğrenicilerin meta öğreniciye iyi uyum sağlaması sağlanmıştır.

5.2.3. Çalışmada kullanılan araçlar

Çalışmanın yürütülmesinde Keras ve Tensorflow açık kaynak platformları ve Python programlama dili kullanılmıştır. Ayrıca veri ön işleme, istatistiksel analiz ve görselleştirme aşamaları için panda, numpy, scikit-learn ve matplot kütüphaneleri kullanılmıştır. Modellerin eğitim ve testleri; Intel 6850K 3.6 GHz i7 işlemci, 16 GB RAM, GTX 1080 Ti 11 GB GPU, 512 GB SSD konfigürasyonuna sahip bir bilgisayar ile gerçekleştirilmiştir.

5.3. Çalışmanın Deneysel Sonuçları ve Tartışmalar

Bu bölümde geliştirilen yöntemin deneysel sonuçları ve farklı ölçüm metrikleri ile performans değerlendirmeleri hakkında detaylı bilgiler sunulmaktadır. Bu bölümde öncelikle, kullanılan verisetlerinin model dağılımları, çalışma da kullanılan materyal ve araçlar açıklanmıştır. Daha sonra önerilen yöntemin iki farklı web anomali veriseti üzerindeki performans ve çeşitlilik (diversity) analizleri yapılmış ve sonuçlar tartışılmıştır. Son olarak, önerilen iki aşamalı WAF modelinin sonuçları, önceki çalışmaların sonuçları ve mevcut çalışmada geliştirilen tek algoritmaya dayalı bireysel DL modellerinin sonuçları ile kıyaslanmıştır.

Bu çalışma kapsamında önerilen modellerin geliştirilmesi, test edilmesi ve karşılaştırılmasında ECML-PKDD ve GAZI-HTTP veri setleri kullanılmıştır. Çalışmada kullanılan veri setleri ile ilgili bilgiler tezin yukarıdaki bölümlerde detaylı olarak verilmiştir. Çalışma kapsamında, verisetleri rastgele üç alt kümeye bölünmüş olup; %60'ı temel öğrenicilerin eğitimi için, %20'si meta öğrenicinin eğitimi için ve kalan %20'si tüm geliştirilen modelleri test edilmesinde kullanılmıştır. Temel öğrenicilerin eğitim kümelerini meta öğreniciden ayırmak, yüksek varyanstan kaçınabilmek için önemlidir. İlk aşamadaki iki sınıflı model, temel ve meta öğrenen modellerin eğitimin kümelerinin birleşimi olan %80' lik alt eğitim kümesi ile eğitilmiştir. Çalışma kapsamında kullanılan GAZI-HTTP ve ECML-

PKK verisetleri için öğrenme modellerinin eğitim ve testinde kullanılan alt kümelerin sınıf bazlı dağılımları sırasıyla Çizelge 5.6 ve Çizelge 5.7' de görülmektedir.

Çizelge 5.6. GAZI-HTTP verisetinin dağılımı

Etiket	Temel öğrenicilerin eğitim kümesi	Meta öğrenicilerin eğitim kümesi	Test kümesi	Toplam	Dağılım
Valid	44117	14706	14706	73529	67.74%
SQLi	6482	2160	2161	10803	9.95%
XSS	5483	1828	1828	9139	8.42%
Command Injection	1941	647	647	3235	2.98%
LFI	1731	577	577	2885	2.66%
Open Redirect	1635	545	545	2725	2.51%
CRLF	1566	522	522	2610	2.40%
SSI	1178	393	392	1963	1.81%
XXE	994	331	331	1656	1.53%
Toplam	**65127**	**21709**	**21709**	**108545**	**100%**

Çizelge 5.7. ECML-PKDD verisetinin dağılımı

Etiket	Temel öğrenicilerin eğitim kümesi	Meta öğrenicilerin eğitim kümesi	Test kümesi	Toplam	Dağılım
Valid	21003	7001	7002	35006	66.94%
Ldap Injection	1367	456	456	2279	4.36%
OS Commanding	2052	684	684	3420	6.54%
Path Traversal	1818	605	606	3029	5.79%
SSI	1139	380	379	1898	3.63%
SQLi	1536	512	512	2560	4.90%
XPath Injection	1367	456	456	2279	4.36%
XSS	1095	365	365	1825	3.49%
Total	**31377**	**10459**	**10460**	**52296**	**100%**

Bu bölüm kapsamında web anomali tespiti için önerilen iki aşamalı katmanlı ensemble modelin deneysel sonuçları aşağıda sunulmakta ve detaylı olarak tartışılmaktadır. Modellerin her aşamadaki sonuçları ayrı alt bölümlerde sunulmaktadır. Öncelikle, ilk aşamadaki iki sınıflı DNN modelinin sonuçları sunulmuştur. İkinci olarak, temel öğrenicilerin sonuçları ve ikinci aşamadaki saldırı türü sınıflandırması için kullanılan katmanlı ensemble modelin sonuçları sunulmuştur. Son bölümde, tüm web istek tiplerini sınıflandıran nihai birleştirilmiş iki aşamalı modelin sonuçları sunulmuş ve tartışılmıştır.

5.3.1. Birinci aşama: iki sınıflı sınıflandırıcı sonuçları

İlk aşamadaki DNN modellerinin GAZI-HTTP ve ECML PKDD test setlerine dayalı olarak oluşan karşıtlık matrisleri ve performans değerleri Çizelge 5.8' de görülmektedir. Önerilen DNN modeli ile GAZI-HTTP ve ECML-PKDD üzerinde sırasıyla %98.76, %99.81 tespit oranlarına ulaşmıştır. FP oranları ise GAZI-HTTP ve ECML-PKDD için sırasıyla %0,72 ve %0,11' dir. Önerilen iki sınıflı modelin performansı incelendiğinde modelin dengesiz

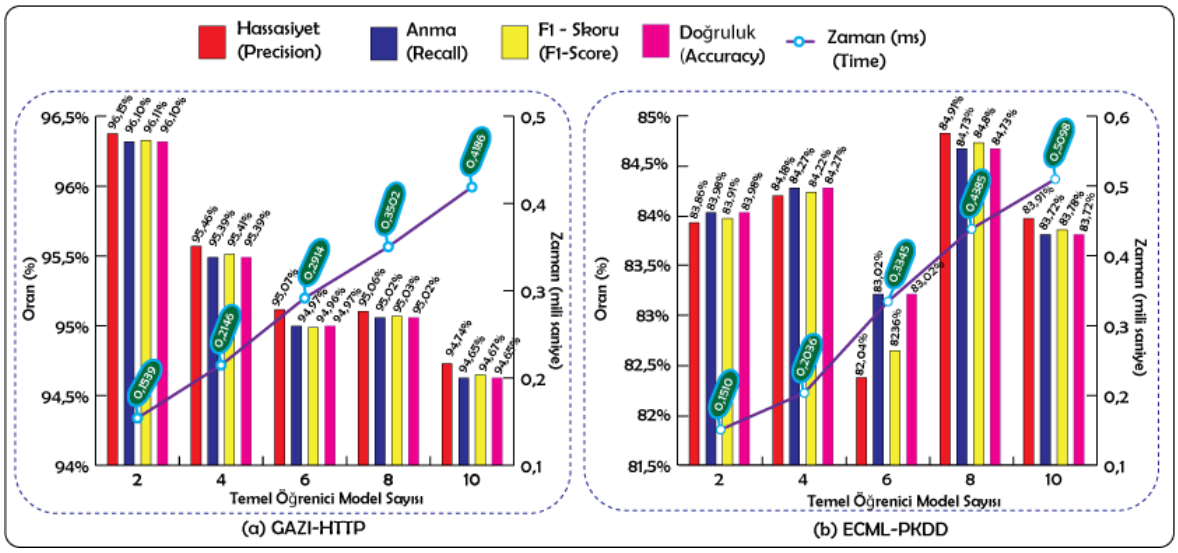
dağılımdan etkilenmediği ve çok düşük FP oranlarına sahip olduğu görülmektedir. Ek olarak, DNN modeli ile yanlılık-varyans ödünleşimi (bias-variance trade-off) yüksek oranda sağlanmaktadır.

Çizelge 5.8. İlk aşamadaki iki sınıflı sınıflandırmanın karşıtlık matrisleri ve performans değerleri

			Tahmin Edilen Sınıf			**Performans Değerleri**			
	Veriseti	Sınıf	Normal	Anormal	Toplam	Hassasiyet (Precision)	Anma (Recall)	F-skoru (F-Score)	**Doğruluk (Acurancy)**
Gerçek Sınıflar	GAZI HTTP	Normal	**14600**	106	14706	0.9847	0.9766	0.9806	
		Anormal	164	**6839**	7003	0.9889	0.9928	0.9908	**%98.76**
		Toplam / A. Ort.	14764	6945	21709	**0.9876**	**0.9876**	**0.9875**	
	ECML PKDD	Normal	**3454**	4	3458	0.9957	0.9988	0.9973	
		Anormal	15	**6987**	7002	0.9994	0.9979	0.9986	**%99.81**
		Toplam / A. Ort.	3469	6991	10460	**0.9982**	**0.9982**	**0.9982**	

5.3.2. İkinci aşama: katmanlı ensemble modellerin sonuçları

Temel sınıflandırıcı modellerinin yarısı CNN ve geri kalan yarısı DNN modelleri olmak üzere 2, 4, 6, 8 ve 10 temel öğreniciden oluşan beş farklı katmanlı ensemble modeli geliştirilmiştir. Katmanlı ensemble modellerin, benzerlik, çeşitlilik ve performansları farklı istatistiksel yöntemlerle ölçülmüş ve karşılaştırılır. Gerçek dünyada kullanılabilir bir WAF sistemi olması için yüksek genelleme ile birlikte yüksek algılama oranlarına sahip en iyi katmanlı ensemble modelin seçilmesi hedeflenmiştir. Katmanlı ensemble modellerinin sınıflandırma performansları ve çeşitlilik analizleri sırasıyla Şekil 5.3’ de ve Çizelge 5.9' da görülmektedir. GAZI-HTTP için en yüksek ve en düşük tespit oranları, sırasıyla %96.10 ve %94.65 ile iki ve on temel öğreniciden oluşan topluluk modelleri ile elde edilmektedir. GAZI-HTTP veri setinde temel öğrenen sayısı arttıkça algılama oranlarının aksi yönde çeşitlilik oranlarının arttığı görülmektedir.



Şekil 5.3. Yığılmış topluluk modellerinin performans değerlendirmeleri

Çizelge 5.9. Yığılmış topluluk modellerinin istatistiksel çeşitlilik analizleri

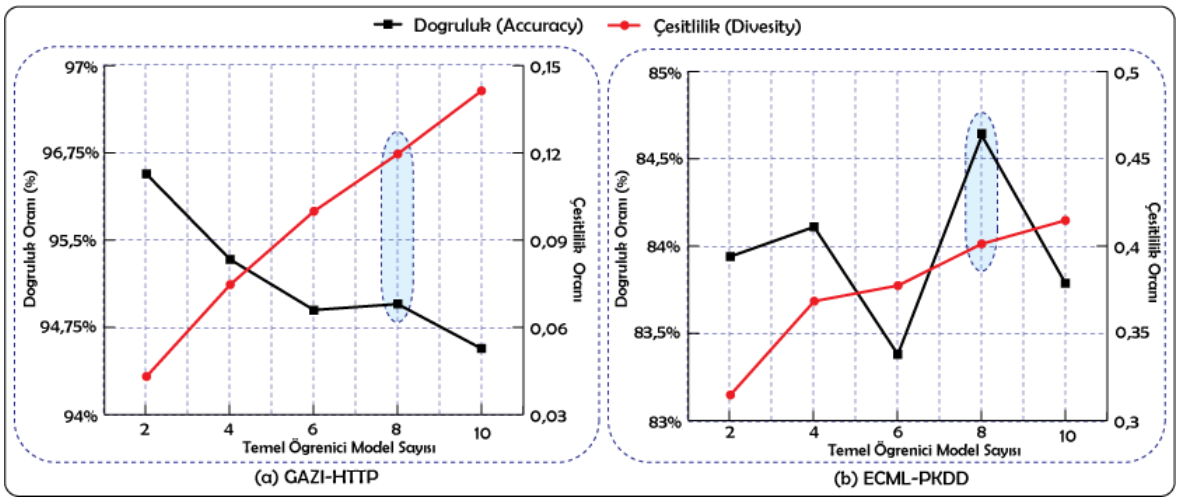
Temel Sınıflandırıcı Sayısı		*Ort. Kappa Skoru	*Ort. Pearson p-değeri	*Ort. Yule Q Benzerliği	Çeşitlilik	Ort. Çift Hata
GAZI HTTP	2	0.9177	0.5579	0.9669	0.0431	**0.0298**
	4	0.8786	0.476	0.9222	0.0747	0.0387
	6	0.8424	0.4289	0.8831	0.0999	0.0448
	8	0.8124	**0.4149**	0.8561	0.1196	0.0522
	10	**0.7764**	0.4153	**0.8347**	**0.1413**	0.0665
ECML PKDD	2	0.5844	0.2364	0.5850	0.3212	**0.0925**
	4	0.4957	0.2239	0.5071	0.3655	0.1259
	6	0.4582	0.2317	0.5038	0.3729	0.1506
	8	0.4262	0.2129	0.4642	0.3926	0.1608
	10	**0.3937**	**0.2067**	**0.4441**	**0.4037**	0.1783

*Çeşitlilik ile ters orantılıdır. **Ort. :** Ortalama

ECML-PKDD için en yüksek ve en düşük tespit oranları, sırasıyla %83,02 ve %84,73 ile altı ve sekiz temel öğreniciden oluşan ensemble modelleri ile elde edilmiştir. Benzer şekilde, ECML-PKDD veri seti için temel öğrenen sayısı arttıkça çeşitliliğin arttığı görülmektedir. Temel öğrenenlerin sayısı arttıkça alt eğitim seti küçülmektedir; bu nedenle, temel sınıflandırıcı modeller daha zayıf (genel veriseti hakkında daha az bilgi) ve farklı hale gelmektedir. Ortalama Kappa skorları, ortalama Pearson p-değerleri ve ortalama Yule Q benzerlik metrikleri çeşitlilikle ters orantılıdır ve bu metrik değerleri azalırken, temel öğrenenler arasındaki benzerlik azalıp ve çeşitlilik artmaktadır. Çizelge 5.9 'da görüldüğü üzere, ortalama Kappa skorları, ortalama Pearson p-değerleri ve ortalama Yule Q benzerlikleri, temel öğrenenlerin sayısıyla doğru orantılı olarak azalmaktadır. Çizelge 5.9

incelendiğinde, katmanlı ensemble modellerinin istatistiksel benzerlik ve farklılıklarının genel olarak en iyi farklılaşmayı temel öğrenen sayısı 10 olduğunda sahip olduğunda sağladığı, temel öğrenen sayısı 2 olduğunda ise en benzer temel öğrenici modellerden oluştuğu görülmektedir.

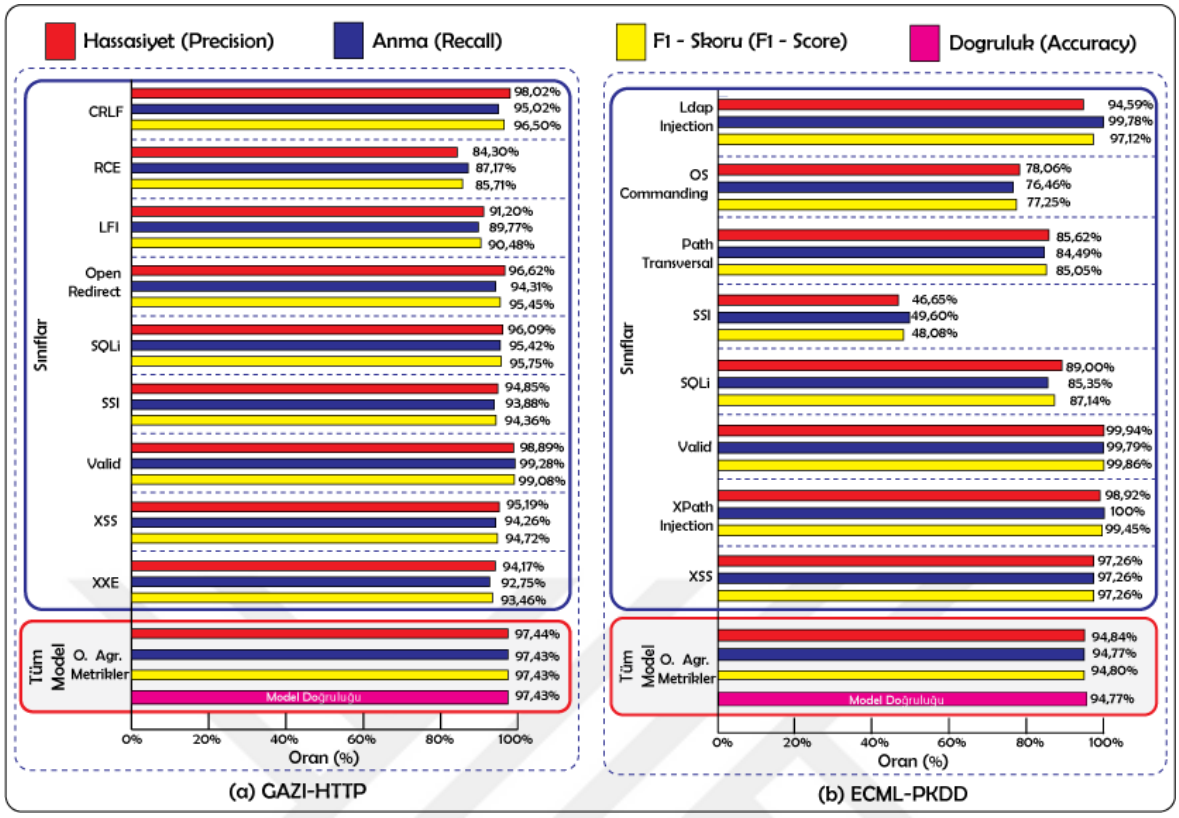
Her iki veriseti için tespit oranları ve çeşitlilik arasındaki ödünleşim (trade-off) grafiği Şekil 5.4' de görülmektedir. Bu grafik incelendiğinde, her iki veri kümesindeki topluluk modellerinin sınıflandırma performanslarının ve çeşitliliklerinin genellikle ters orantılı olduğu görülmektedir. Ensemble modellerde sadece sınıflandırma doğruluklarına odaklanmak doğru bir yaklaşım değildir. Yeni ortaya çıkan web saldırılarını tespit etmek için genelleme yapabilen topluluk modelleri geliştirilmelidir; dolayısıyla yüksek çeşitliliğe sahip model tercih edilmelidir. Şekil 5.4' te yer alan mavi alanlar ensemble modellerin her iki veriseti için birlikte en iyi ödünleşimi sağladığı alanı göstermektedir. Şekil 5.4 incelendiğinde, her iki veriseti için en iyi tespit oranı - çeşitlilik ödünleşimlerinin sekiz temel öğreniciden oluşan topluluk modelleri tarafından sağlandığı görülmektedir (Şekil 5.4'te mavi alanlarda gösterilmiştir). Bu nedenle, çalışma kapsamında nihai WAF modeli için hem yüksek tespit oranı hem de yüksek çeşitliliğe sahip sekiz temel öğreniciden oluşan katmanlı ensemble modellerin kullanılmasına karar verilmiştir.



Şekil 5.4. GAZI-HTTP (a) ve ECML-PKDD (b) verisetleri için katmanlı ensemble modellerinin doğru sınıflandırma ve çeşitlilik arasındaki ödünleşim (trade-off) grafiği

5.3.3. İki aşamalı model sonuçları

Birinci aşama ve ikinci aşama modellerin birleştirilmesiyle oluşturulan nihai iki aşamalı WAF modelinin performansları test setleri üzerinde gerçekleştirilmiştir. Aşağıda yer alan sonuçlar, sekiz temel öğreniciden oluşan iki aşamalı modele aittir. GAZI-HTTP ve ECML-PKDD verisetlerine dayalı iki aşamalı modellerin performans değerleri Şekil 5.5' te sunulmuştur. GAZI-HTTP ve ECML-PKDD için sırasıyla %97,43 ve %94,97 tespit oranları ulaşılmıştır. GAZI-HTTP' de ağırlıklı ortalama hassasiyet (precision), anma (recall) ve F1-skoru (F1-score) metrikleri için sırasıyla %97,44, %97,43 ve %97,43 oranlarına ulaşılmıştır. Benzer şekilde, ECML-PKDD test seti için ağırlıklı ortalama hassasiyet (precision), anma (recall) ve F1-skoru (F1-score) metrikleri için sırasıyla %94.84, %94.77 ve %94.80 oranlarına ulaşılmıştır. Önerilen iki aşamalı modelin genel olarak her iki veri seti için de kabul edilebilir duyarlılık ve doğruluk değerlerine sahip olduğu görülmektedir. Şekil 5.5 sınıf bazında analiz edildiğinde, her iki veri setinde için en iyi F-skoru oranlarına normal istekleri ifade eden "Valid" etiketi için erişilmiş olup, GAZI-HTTP ve ECML-PKDD için sırasıyla %99.08 ve %99.86 F-skoruna ulaşılmıştır. GAZI-HTTP veri seti için en kötü sınıf bazlı sınıflandırma performansı, uzaktan kod çalıştırma (Remote code execution, RCE) etiketinde ortaya çıkmış olup ve bu etiketin F-skor puanı %85,71'de kalmıştır. Benzer şekilde, ECML-PKDD için F-skoru açsısında en kötü olduğu sınıf, %48,08'e oranına erişen SSI etiketidir. Bunun temel nedeni, veri setlerinin dağılımını gösteren çizelgeler incelendiğinde görülmektedir. Her iki veri seti de dengesiz dağılmakta ve en az bulunan web istek sınıflarının en düşük performansa ulaştığı görülmektedir. Ayrıca bazı web saldırı türleri birbirine çok benzer payload paternleri içerebilmektedir. Örneğin, komut enjeksiyon saldırısında rastgele komutlar bulunduğundan, sınıflandırma modelleri tarafından benzer komutları içeren XSS, SQLi ve LFI saldırı kalıpları ile karıştırılabilmektedir.

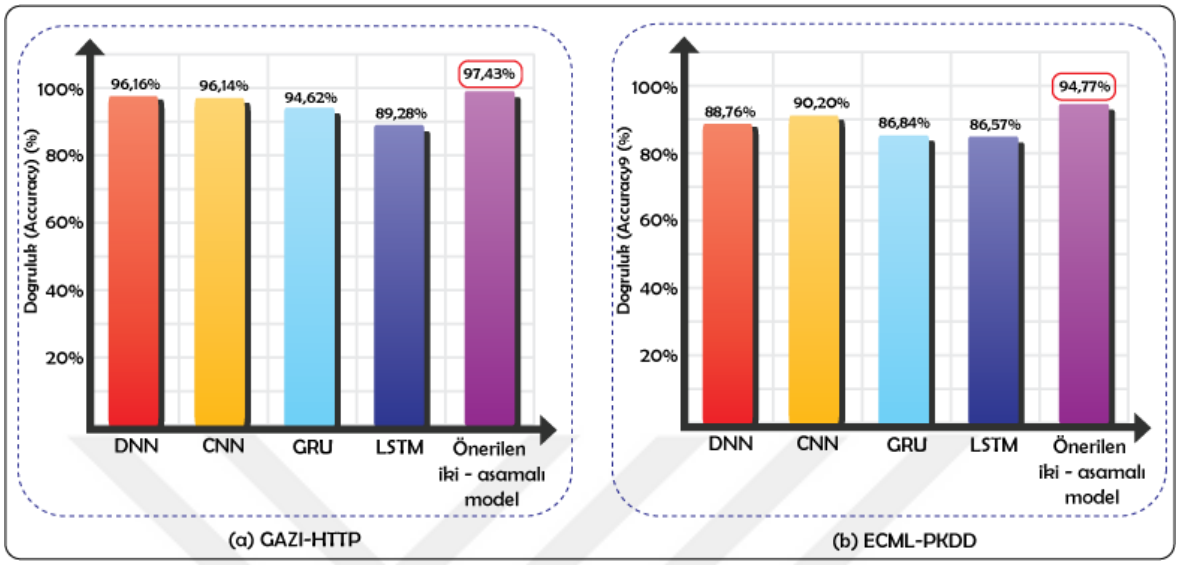


Şekil 5.5. GAZI-HTTP (a) ve ECML-PKDD (b) verisetleri için iki aşamalı nihai modellerin performans değerleri

5.4. Önerilen İki Aşamalı WAF Modelinin DL Algoritmalarına Dayalı Bireysel Modeller ve Önceki Çalışmalar ile Kıyaslanması

Çalışmada önerilen nihai iki aşamalı model ile performans karşılaştırması yapmak amacıyla web güvenliğinde yaygın olarak kullanılan tek algoritmaya dayalı DL sınıflandırıcı modeller geliştirilmiştir. Tek algoritmaya dayalı sınıflandırıcıların geliştirilmesinde DNN, CNN, LSTM ve GRU algoritmaları kullanılmıştır. Önerilen iki aşamalı katmanlı ensemble modelinin, DL modelleri ile çok sınıflı performans karşılaştırmaları Şekil 5.6' da görülmektedir. GAZI-HTTP veriseti üzerinde DNN, CNN, GRU ve LSTM modelleri ile sırasıyla %96,16, %96,14, %94,62 ve %89,28 tespit oranları elde edilmiştir. Önerilen iki aşamalı WAF modelinin tüm modeller içinde en yüksek tespit oranına ulaştığı ve %97,43 ile önemli bir iyileşme sağlandığı görülmektedir. Benzer şekilde ECML-PKDD için DNN, CNN, GRU ve LSTM modelleri ile sırasıyla %88.76, %90.20, %86.84 ve %86.57 tespit oranlarına ulaşılmıştır. Önerilen iki aşamalı WAF modeli ile ECML-PKDD üzerinde %94,77 tespit oranı ulaşılmış ve en başarılı tek DL algoritmasına göre %4,57 daha iyi sınıflandırma başarısı elde edilmiştir. Önerilen modelin, tek bir algoritmaya dayalı DL

modellerine kıyasla çok sınıflı web anomali tespitinde önemli ölçüde iyileştirme sağlandığı görülmektedir.



Şekil 5.6. Önerilen iki aşamalı model ile DL tabanlı tek sınıflandırıcıya dayalı modellerin performans karşılaştırması

ECML-PKDD veri setini kullanan önceki çalışmalar ile önerilen iki aşamalı WAF modeli kıyaslanmıştır. Literatür incelendiğinde çalışmaların büyük çoğunluğunun iki sınıflı model temelli yürütüldüğü, dolayısıyla çok sınıflı çalışmaların az olduğu görülmektedir. Önerilen modelin diğer çalışmalarla karşılaştırılması Çizelge 5.10' da görülmektedir. Nguyen ve arkadaşları [26], ECML-PKDD üzerinde ikili sınıflandırma için %98.80 tespit oranı ile şimdiye kadarki en yüksek sınıflandırma başarısına ulaşmıştır. İlgili çalışmada [26] öznitelik çıkarımı sadece istatistiksel yöntemlerle yapıldığından, web isteklerinin içeriği anlamsal olarak ele alınmamakta, bu nedenle yazarların önerdiği WAF sisteminin gerçek dünya da kullanımı mümkün görülmemektedir. Çalışma [25] da önerilen A-ExIDS modelinin sonuçları incelendiğinde, normal web istekleri için yüksek doğruluk sağlanırken, saldırı türlerini tespit etmede daha düşük olduğu görülmektedir. Sonuç olarak, ilgili çalışmada [25] yüksek FP sorunu ortaya çıkmakta olup, bu durum WAF sistemlerinin gerçek zamanlı kullanılması için uygun değildir. Vartouni ve arkadaşları [29] önerdikleri model ile ECML-PKDD için %84,13'lük bir tespit oranına ulaşmışlardır. Çalışmada [29] doğruluk oranının düşük olmasının önemli nedenlerinden biri, anlamsal temsiller içermeyen N-gram tabanlı öznitelik çıkarımının kullanılmasıdır. Odumuyiwa ve arkadaşları [27] tarafından yapılan çalışmada elde edilen sonuçlar detaylı olarak incelendiğinde, CNN tabanlı sınıflandırıcı modelin saldırı türleri için ağırlıklı ortalama tespit oranı %84 de kalmaktadır. Bu durumda, derin öğrenme kullanılsa bile tek bir algoritma tabanlı sınıflandırıcı ile saldırı türlerinin

yüksek hassasiyetle tespit edilemediği ve dengesiz bir veri setinden dolayı yüksek yanlılık oluştuğu göstermektedir.

Bizim çalışmamızda ise ikili sınıf için bu oran %99,81'e çıkarılmıştır. Bu çalışmada kapsamında geliştirilen bireysel modellerden CNN tabanlı sınıflandırıcı, ECML-PKDD üzerinde çoklu sınıf için %90.20 tespit oranı ile diğer modeller arasında en yüksek başarıyı elde etmiştir. Önerilen iki aşamalı katmanlı ensemble modeli, ECML-PKDD' de %4,57'lik bir iyileştirme sağlayarak %94,77'lik bir algılama oranına ulaşmıştır. Önerilen modelin hem ikili sınıflı hem de çok sınıflı web anomalilerini tespit etmede diğer çalışmalara göre önemli ölçüde daha başarılı olduğu görülmektedir.

Çizelge 5.10. ECML-PKDD veriseti sonuçlarımızın ile önceki çalışmalarla karşılaştırılması

Yazar	***Model***	***İki Sınıflı***	***Çok Sınıflı***
Nguyen ve Franke [25]	Naive Bayes	85.12%	N/A
Nguyen ve Franke [25]	Bayes Network	86.95%	N/A
Nguyen ve Franke [25]	RBF Network	87.69%	N/A
Nguyen ve Franke [25]	Hedge Boosting	86.30%	N/A
Nguyen ve Franke [25]	A-ExIDS	92.56%	N/A
Nguyen ve ark. [26]	Decision Tree	96.37%	N/A
Nguyen ve ark. [26]	CART	96.11%	N/A
Nguyen ve ark. [26]	Random Tree	96.89%	N/A
Nguyen ve ark. [26]	Random Forest	***98.80%***	N/A
Tekerek ve ark. [31]	Hibrit Model	93.30%	N/A
Vartouni ve ark. [29]	DBN	84.13%	N/A
Raissi ve ark. [21]	Statik Metod	87.93%	69.00%
Exbrayat [22]	N-similarity	90.00%	N/A
Pachopoulos ve ark. [28]	Decision Tree	77.00%	N/A
Galagher ve ark. [23]	Tf-Idf	94.00%	N/A
Odumuyiwa ve ark. [27]	DNN	91.90%	N/A
Odumuyiwa ve ark. [27]	CNN	94.70%	N/A
Mevcut çalışma [95]	DNN	99.81%	88.76%
Mevcut çalışma [95]	CNN	98.41%	**90.20%**
Mevcut çalışma [95]	GRU	98.04%	86.84%
Mevcut çalışma [95]	LSTM	97.84%	86.57%
İki Aşamalı WAF Modeli [95]	**İki Aşamalı Ensemble**	**99.81%**	**94.77%**

N/A: Bulunmuyor

5.5. Önerilen Modelin Zaman ve Uzay Karmaşıklıkları

Gerçek zamanlı bir WAF sisteminin geliştirilmesinde, zaman ve uzay karmaşıklığı, hesaplama kaynakları ve zaman gecikmesi açısından önemlidir. Derin öğrenme algoritmalarına dayalı modellerde yoğun matris çarpımlarının gerçekleştirildiği için model derinliğine ve bir dizi parametreye dayalı olarak önemli miktarda hesaplama gücü gerektirmektedir. Derin öğrenme algoritmalarının uzay karmaşıklığı hesaplanırken, her

katmandaki nöronların çıktıları bir sonraki katmana aktarıldığından ve bu değerlerin bellekte tutulması gerektiği dikkate alınmalıdır. Ayrıca ağdaki parametrelerin ve ağırlıkların değerleri de bellekte saklanmaktadır. Bu nedenle, derin öğrenmeye dayalı önerilen bir modelin uzay karmaşıklığı, içerdiği katmanların ve parametrelerin sayısına bağlıdır ve doğrusaldır. Ancak, ağ boyutuyla orantılı olarak derin öğrenme modelinin bellekte tutacağı bellek alanı tahsisi, girdi sayısı, nöronların çıktı sayısı ve modele dâhil edilen parametre sayısına bağlıdır olup, uzay karmaşıklığı yine doğrusaldır [96].

Zaman karmaşıklığı, girdi sayısı, katman sayısı, katmanlardaki nöron sayısı, nöronların çıkış sayısı, nöron ağırlıklarının ve parametrelerinin başlangıç değerleri, kayıp fonksiyonu, katman sayısı, eğitim döngü sayısı veya durdurma kriterine bağlı olarak asenkron olarak değişmektedir. [96]. Derin öğrenme tabanlı modellerin zaman karmaşıklıklarını mutlak olarak hesaplamayabilmek için oluşturulan DL ağının eğitimi sırasında kaç nöron bağlantısının aktif olduğunu belirlemek gerekmektir. Ancak bir derin öğrenme ağında değişen öğrenme parametrelerine göre aktif nöron bağlantılarının sayısı sürekli değişebilmektedir. Önerilen WAF modeli iki aşamalı olduğundan, isteğin normal mi yoksa saldırı mı olduğuna bağlı olarak yalnızca ilk aşamada veya her iki aşamada işlenebilmektedir. Normal istekler, ilk aşamada yalnızca iki sınıflı DNN modelinde işlenmektedir. Bienstock ve ark. [97] yaptıkları deneysel çalışmalar ile farklı derin öğrenme mimarilerinin genel zaman karmaşıklığını hesaplamak için bir yöntem önerdiler. Bu yönteme [97] dayalı olarak önerdiğimiz WAF modelinin ilk aşamasındaki iki sınıflı DNN modelinin zaman karmaşıklığı Eş. 5.18' de görülmektedir. Burada $\boldsymbol{m}$ girdi boyutunu, $\boldsymbol{n}$ çıktı boyutunu, $\boldsymbol{k}$ katman sayısını, $\boldsymbol{N}$ toplam parametre sayısını, $\boldsymbol{D_{train}}$ ve $\boldsymbol{D_{test}}$ öğrenme ve test setlerinin eleman sayısını ve $\boldsymbol{E_{epoch}}$ eğitim döngü sayısını ifade etmektedir.

$$O\left(\left(m\log(m)w^{O\left(k^2\right)}\right)^{n+m+N}\left(D_{train}+D_{test}\right)E_{epoch}\right) \tag{5.18}$$

Benzer şekilde, CNN ve DNN modellerine dayalı temel ve meta öğrenen modellerini içeren ikinci aşama saldırı tipi tespit modelinin genel zaman karmaşıklığı; $\boldsymbol{M_{dnn}}$ meta öğrenici sayısını (çalışma bir meta öğrenici model kullanıldı), $\boldsymbol{B_{dnn}}$ ve $\boldsymbol{B_{cnn}}$ sırasıyla DNN ve CNN algoritmasına dayalı temel öğrenici sayılarını ifade etmek üzere Eş. 5.19’ da görülmektedir [97].

$$O\left(\left(m\left(\left(B_{dnn}+M_{dnn}\right)*\log\left(m\right)+B_{cnn}\right)w^{O\left(k^2\right)}\right)^{n+m+N}\left(D_{train}+D_{test}\right)E_{epoch}\right) \quad (5.19)$$

Önerilen modelin zaman verimliliğini görebilmek için yukarıdaki zaman karmaşıklığına ek olarak test setindeki web isteklerinin ortalama zaman gereksinimleri hesaplanmıştır. Ayrıca tek bir derin öğrenme algoritmasına dayalı modeller ile önerilen iki aşamalı WAF modelinin zaman performansları karşılaştırılmıştır. Yukarıda açıklandığı üzere önerilen iki aşamalı model ile tespit oranının iyileştirilmesine ek olarak, normal web istekleri için gecikmenin azaltılması bu çalışmanın önemli amaçlarından biridir. Önerilen modelin birinci aşama, ikinci aşama, iki aşama için ağırlıklı ortalama tespit süreleri ve bireysel DL algoritmalarına dayalı modellerle karşılaştırmaları Çizelge 5.11' de sunulmuştur. Birinci aşamada, GAZI-HTTP ve ECML-PKDD için bir talebi sınıflandırmak, sırasıyla ortalama 0.041 ve 0.044 mili saniye (ms) sürmektedir. İkinci aşamada, saldırı türlerinin sınıflandırılması sırasıyla 0.339 ve 0.847 ms sürmektedir. GAZI-HTTP ve ECML-PKDD için bilinmeyen bir talebin iki aşamalı olarak çok sınıflı tespitinin ortalama zaman tüketimi sırasıyla 0,129 ve 0,137 ms sürmektedir. Önerilen iki aşamalı model sayesinde normal web istekleri için saldırı türü tespitinden kaynaklanan zaman gecikmesini önüne geçildiği görülmektedir. Çalışmada kullanılan yukarıda konfigürasyonu verilen bilgisayar saniyede yaklaşık 25000 normal kullanıcı talebini işleyebilmektedir. Buna ek olarak, önerilen iki aşamalı WAF sisteminin daha güçlü bir sunucu üzerinde inline mod da konumlandırılması halinde birim zamanda işlenebilecek web isteklerinin sayısı artırılabilecektir.

Çizelge 5.11. Önerilen modelin zaman performansları ve tek algoritmaya dayalı WAF modelleri ile karşılaştırılması

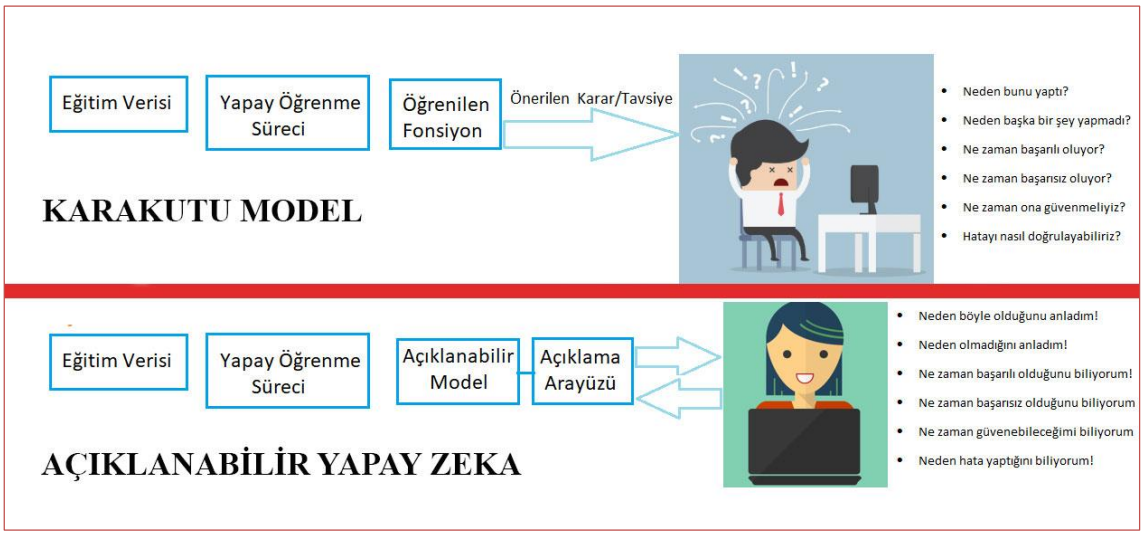
		Tek sınıflandırıcıdan oluşan DL modelleri				**Önerilen iki aşamalı WAF modeli**		
	Veriseti	DNN	CNN	GRU	LSTM	İlk Aşama	İkinci Aşama	İki aşamanın ortalaması
Zaman (ms)	GAZI HTTP	0.035	0.094	12.62	14.97	0.041	0.339	0.129
	ECML PKDD	0.048	0.138	12.849	15.634	0.044	0.847	0.137

5.6. Çalışmanın Sonuç Bölümü

Bu çalışmada, gerçek zamanlı olarak kullanılabilecek bir WAF sistemi için iki aşamalı derin öğrenme tabanlı katmanlı ensemble modeli önerilmiştir. Önerilen modelin ilk aşamasında, normal web istekleri ve saldırı istekleri, güçlü bir DNN modeli ile iki sınıflı olarak sınıflandırılmaktadır. İkinci aşamada DL tabanlı katmanlı ensemble modeli ile saldırı isteklerinin türü belirlenmektedir. Önerilen iki aşamalı yapı sayesinde saldırı tipini belirlemek için kullanılan çok sınıflı sınıflandırmanın gerektirdiği model karmaşıklığı ve zaman gereksinimi ikinci aşamaya aktarılmıştır. Böylece normal kullanıcıların istekleri yüksek doğruluk ve hızla işlenebilmektedir. Çalışmada önerilen modelin genelleme gücünü belirlemek için istatistiksel benzerlik ve çeşitlilik analizleri yapılmıştır. Analizler sonucunda hem yüksek doğrulukta hem de yüksek çeşitliliktliğe sahip bir katmanlı ensemble modeli geliştirilmesi sağlanmıştır. Bu çalışma sayesinde normal istemciler için web isteklerini yüksek hassasiyette ve zaman gecikmesi olmadan çoklu sınıfa göre sınıflandırabilen bir model geliştirilmiştir. Önerilen model ile GAZI-HTTP ve ECML-PKDD için diğer modellere göre sırasıyla %1,27 ve %4,57 oranında iyileştirme sağlanmakta ve sırasıyla %97,43 ve %94,77 tespit oranlarına ulaşılmaktadır. Önerilen modelin, bireysel DL algoritmalarına ve önceki çalışmalara göre önemli bir performans artışına sahip olduğu görülmüştür. Gelecekteki çalışmalarda saldırı türlerinin ayrı ayrı ele alındığı hibrit DL tabanlı modellerin geliştirilmesi planlanmaktadır. Ayrıca, ensemble modelinde kelime yerleştirmeye dayalı temel öğrenicilerle birlikte manuel özellik çıkarımına dayalı temel öğrenicilerin kullanılmasının genelleme gücünü artıracağı düşünülmektedir. Bazı sınıflar için dengesiz veri kümelerinden kaynaklanan düşük sınıflandırma performansı sorununu çözmek için yeniden örnekleme veya veri büyütme gibi tekniklerin uygulanması, hassasiyeti ve tespit oranını artıracaktır.

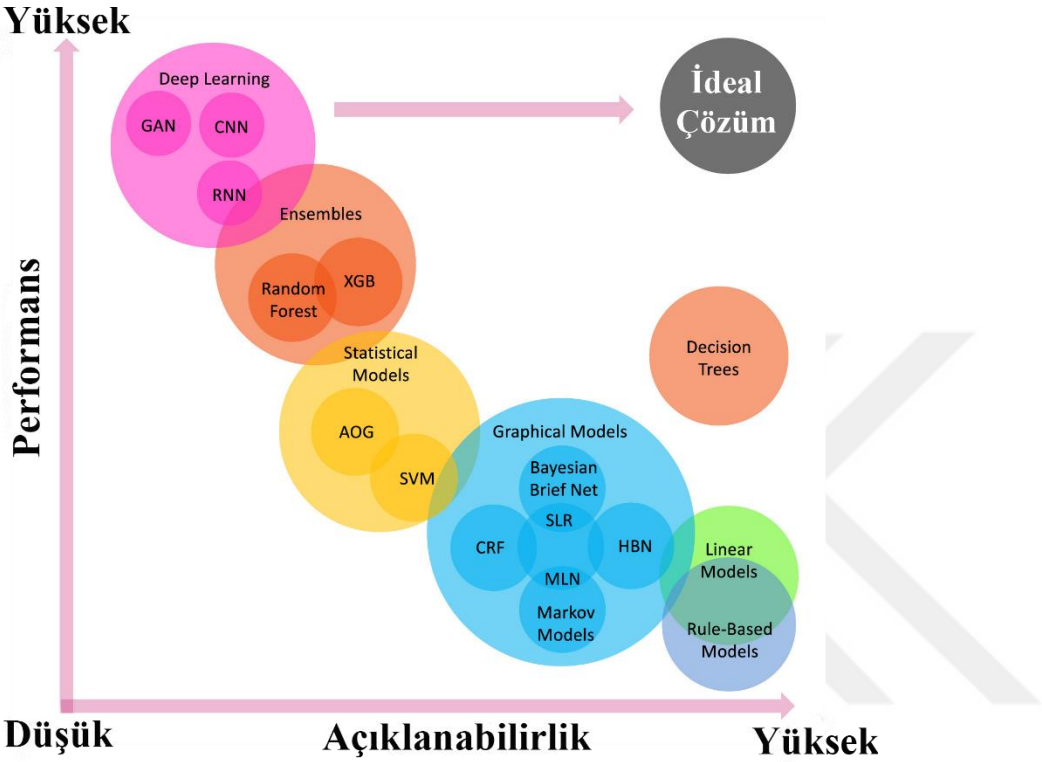
6. WEB ANOMALİ ALANI ODAKLI YENİ BİR AÇIKLANABİLİRLİK YÖNTEMİNİN GELİŞTİRİLMESİ: XAI-WAF

Saldırı tespit sistemlerininin geliştirilmesinde makine öğrenmesi tabanlı modeller sıklıkla kullanılmaktadır. Genellikle denetimli öğrenme tekniklerinin kullanıldığı bu modeller ile yüksek başarıma sahip tespit modellerinin oluşturulduğu görülmektedir. Son zamanlarda derin öğrenme temelli yaklaşımlar gösterdikleri yüksek başarımlar sebebiyle sıklıkla kullanılmaya başlanmıştır. Bununla birlikte derin öğrenme gibi yöntemlerin nasıl çalıştıkları ve model tahminlerini nasıl gerçekleştirildikleri tam olarak açıklanamadığı için, bu yöntemler kara kutu modeller olarak adlandırılmaktadır. Karakutu modellerin bir durum için karar verirken hangi özellikleri ve parametreleri nasıl kullanıldıkları bilinmediği için, model kararlarının yorumlanması ve modelin çalışmasının açıklanması mümkün olmamaktadır. Model başarımlarının yüksek olması, sağlık, finans, insan kaynakları, saldırı tespiti gibi kritik sistemlerde kullanabilmeleri için gerekli güveni sağlamada tek başına yeterli olmamaktadır. Kritik sistemlerde yapay öğrenme temelli modellerin kullanılabilmesi için, modelin şeffaf olması ve aldığı kararların ve modelin çalışmasının açıklanabilmesi önem arz etmektedir. Gerçek zamanlı olarak kritik sistemlerde kullanılması bakımından karakutu modellerin açıklanabilirliğinin sağlanmasından beklenen faydalar genel hatları ile Şekil 6.1' de görülmektedir [98].



Şekil 6.1. Açıklanabilir yapay zeka kullanımının amaçları [98]

Genel olarak anomali tabanlı IDS modellerinde kullanılabilecek yapay zeka algoritmalarının şeffaflığı/açıklanabilirliği ile saldırı tespit performansları ters orantılıdır [99]. Şekil 6.2' de görüleceği üzere, derin öğrenme gibi sofistike yapıya sahip anomali tabanlı IDS modellerinin tespit başarımları artarken açıklanabilirlikleri azalmaktadır [99].



Şekil 6.2. Model açıklanabilirliği ile performans arasındaki ilişki [99]

Mevcut web güvenliği çalışmaları incelendiğinde başarılı anomali tabanlı çalışmaların genellikle karakutu modellerden oluştuğu, model açıklanabilirliği ve tahminlerin yorumlanmasına yönelik çalışmaların az olduğu görülmektedir. Anomali tabanlı modellerin gerçek zamanlı kullanımlarının sağlanabilmesi için saldırı tipi tahminlerinin hangi girdi ve parametrelere göre yapıldığının açıklanması, modele güven ve şeffaflık açısından önemli bir gerekliliktir. Ayrıca anomali tabanlı modellerin basit manipülasyonlar ile kandırılabildiği güncel çalışmalar ile gösterilmektedir [100, 101]. Geliştirilen kapalı kutu modelin bu tür manipülasyonlardan etkilenip etkilenmediğinin anlaşılabilmesi için tahminlerin yorumlanması önemli hale gelmiştir.

Bu çalışma kapsamında web uygulama güvenliğinin sağlanmasına yönelik geliştirilen derin öğrenme modellerinin açıklanabilirliğini sağlamak amacıyla alan orjinli bir açıklanabilirlik yöntemi geliştirilmiştir. Geliştirilen açıklanabilirlik yöntemi web güvenliği alanına özgü, bireysel model tahminleri için yerel yorumlanabilirlik sağlayan güvenilir ve stabil bir

yöntemdir. Önerilen açıklanabilirlik yöntemi ile WAF sistemi olarak kullanılan tüm derin öğrenme algoritma ve modellerin açıklanabilirliği sağlanmakta olup, bu hali ile derin öğrenmeye odaklı model agnostik (bağımsız) bir yöntemdir. Çalışma kapsamında mevcut ve önerilen açıklanabilirlik yöntemlerinin analizi amacıyla farklı derin öğrenme modelleri ile WAF sistemleri geliştirilmiştir. Modellerin eğitim ve testinde tez kapsamında oluşturulan, güncel saldırı türlerini içeren, özgün ve stabil GAZI-HTTP web anomali veriseti kullanılmıştır. Ayrıca mevcut açıklanabilirlik yöntemlerinin ele alındığı çalışma bölümünde GAZI-HTTP verisetinin yanı sıra ağ tabanlı bir benchmark veri seti olan NSL-KDD veriseti kullanılmıştır.

Bu çalışma ile alana sağladığımız başlıca katkılardan bazıları aşağıda sunulmuştur;

- Web güvenliği alanına özgü yeni bir açıklanabilir yaklaşımı geliştirilmiştir. Bilgimiz dâhilde web güvenliği alanına özgü geliştirilmiş mevcut bir açıklanabilirlik yaklaşımı bulunmamaktadır.
- Farklı derin öğrenme algoritmalarına dayalı, başarılı ve etkili WAF modelleri geliştirilmiş ve derin öğrenme yöntemlerinin web güvenliği alanındaki etkinliği gösterilmiştir.
- Önerilen açıklanabilirlik yöntemi ile literatürde sık kullanılan ve bilinen açıklanabilirlik yaklaşımlarından LIME ve SHAP ile kıyaslanmıştır. Mevcut açıklanabilirlik yaklaşımlarına dayalı olarak çalışmada yer alan DL modellerinin açıklanabilirliğinin sağlandığı açıklanabilirlik modülleri oluşturulmuştur. Geliştirilen yöntem alana özgü olduğu için ve kelime gömme işleminde yer alan kelime birlikteliklerini, ilişkilerini, konumlarını ve komşuluklarını ele aldığı için web isteklerinin analizinde daha istikrarlı çalıştığı görülmüştür.
- Farklı derin öğrenme algoritmalarına dayalı geliştirilen WAF modelleri ile yeni açıklanabilirlik yöntemimizin bir arada kullanılabildiği, DL modellerinin eğitildiği, web isteklerinin gerçek zamanlı sınıflandırabildiği ve önerilen XAI-WAF yaklaşımına dayalı açıklanabilirliğin sağlandığı bir framework yapısı oluşturularak, kullanıma hazır hale getirilmiştir.
- Tez çalışması kapsamında oluşturulan ve bu çalışma kapsamında kullanılan GAZI-HTTP veriseti ve bu çalışmanın kaynak kodları, çalışmanın tekrarlanabilirliğin sağlanması ve gelecekteki araştırmalara katkı sağlanması amacıyla diğer araştırmacılarla kamuya açık olarak paylaşılacaktır*.

* https://github.com/mehmetsevri/XAI-WAF-A-Novel-Explainable-AI-Framework-for-Deep-Learning-Based-Web-Application-Security

Tez kapsamında gerçekleştirilen bu çalışma giriş bölümü dâhil toplam beş bölümden oluşmaktadır. Çalışmanın ikinci kısmında mevcut açıklanabilirlik yaklaşımlarının temelleri ve derin öğrenme modellerine ilişkin ön bilgilendirme bölümü yer almaktadır. Üçüncü kısımda çalışma kapsamında geliştirilen XAI-WAF isimli açıklanabilirlik yaklaşımının temelleri ve nasıl çalıştığı açıklanmıştır. Çalışmanın dördüncü kısmında derin öğrenme tabanlı IDS sistemlerinin performansı ve açıklanabilirliklerine ilişkin çıktılar sunulmaktadır. Çalışmanın beşinci kısmında çalışmanın sonucu ve gelecekte yapılabilebilecek çalışmalar ele alınmaktadır.

Yukarıda bahsedilen XAI tabanlı siber güvenlik çalışmaları, genellikle ağ tabanlı çalışmalar olup, web alanında gerçekleştirilen çalışmalar çok kısıtlıdır. Bu çalışmada önerilen XAI-WAF açıklanabilirlik çerçevesi (framework) farklı derin öğrenme algoritmalarına dayalı modellerin açıklanabilirliğini sağlaması açısından önemlidir. Çalışmamız, geliştirilen XAI-WAF yaklaşımının, web uygulama güvenliği odaklı olması, kelime gömmeden gelen benzerlik ilişkilerinin dikkate alınması, kelimeleri konumları ile bireysel ele alması, komşuluk ilişkilerini dikkate alması, web anomali sınıfları arasındaki geçiş ve ilişkileri ele alması açılarından diğer çalışmalardan ayrılmaktadır. Ayrıca çalışma kapsamında sunulan açıklanabilirlik yapısı çalışmada kullanılan DNN ve CNN'nin yanı sıra diğer tüm derin öğrenme modelleri ile geliştirilen IDS modellerini destekleyecek şekilde tasarlanmıştır.

6.1. Açıklanabilir Yapay Zekâ (XAI) Yaklaşımları Hakkında Ön Bilgiler

Makine öğrenme modellerin yorumlanması modelin oluşturulduğu zamana dayalı olarak üç safhada gerçekleştirilebilmektedir. Bunlar sırasıyla; model öncesinde (pre-modelling explainability), model çalıştığı sırada (explainable modelling), ve model oluşturulduktan sonra (post-modelling explainability) gerçekleştirilebilmektedir. Model oluşturulmadan önce (pre-modelling explainability) keşfe dayalı veriseti analizi (Temel Bileşen Analizi (Principal Component Analysis - PCA), t-Distributed Stochastic Neighbor Embedding (t-SEN)), veri setinin özetlenmesi ve yorumlanabilir özellik çıkarım teknikleri (Generalised Linear Models - GLM, extra trees classifier veya Random Forest - RF) gibi özellik önemlilik seçimi modelleri ile gerçekleştirilebilmektedir. İkinci olarak, Karar ağacı, lineer regresyon, K-nn gibi bazı yapay öğrenme modelleri transparan yapıları sayesinde yorumlanabilirliği ve açıklanabilirliği kendiliğinden sağlayan modellerin kullanımları sırasında model

yorumlaması gerçekleştirilebilmektedir. Kendiliğinden açıklanabilir yapılara sahip olmayan makine öğrenme modellerine karakutu modeller denilmektedir. Özellikle son yıllarda birçok farklı alanda sağladığı önemli başarımlar ile derin öğrenme yöntemleri güncel çalışmaların en sık kullanılan yöntemleri olarak öne çıkmaktadır. Bununla birlikte, derin öğrenme modelleri içerdiği çok sayıda katman ve milyonlara ulaşabilen parametre sayısından dolayı modellerin yorumlanması imkansız hale gelmekte ve karakutu modeller olarak adlandırılmaktadır. Karakutu modellerin yorumlanması ve açıklanabilirliğinin sağlanması için model sonrasında (post-modelling explainability, post-hoc) ekstra model ve tekniklerin kullanılması gerekmektedir. Derin öğrenme modellerinin açıklanabilirliğinin sağlanmasında model sonrası yerel açıklanabilirlik (post-hoc local explanation) ve özellik ilişkileri (feature relevance) olmak üzere sıklıkla kullanılan iki yöntem bulunmaktadır. Bunların yanı sıra, model basitleştirme, metin ile açıklama (text explanation), model görselleştirme, özellik ilişkilerinin tahminlenmesi gibi yöntemlerde kullanılabilmektedir [102].

Modellerin yorumlanması amaca yönelik olarak genel açıklanabilirlik (global explanations) ve yerel açıklanabilirlik (local explanation) olmak üzere iki yaklaşım ile gerçekleştirilebilmektedir. Genel yorumlanabilirlik ile modelin genel olarak davranışlarının açıklanması, modelin alt parçalarının öğrenmeye katkısının veya modelin genel olarak tahminlerini en çok etkileyen girdi özelliklerinin gücü ve katkısının ölçülmesi amaçlanmaktadır. Genel yorumlanabilirlikte tüm girdiler (aynı sınıfa ait veya tüm sınıflar birlikte) ve bunların çıktıları dikkate alınarak açıklama getirilmektedir. Yerel yorumlanabilirlik ise tek bir girdiye ait çıktının yorumlanması ve girdi üzerindeki parametrelerin bu çıktıya katkısının ortaya çıkarılmasını amaçlamaktadır. Yerel yorumlanabilirlik sayesinde her bir girdiye ilişkin çıktının model tarafından nasıl oluşturulduğu yorumlanabilmektedir.

Literatür incelendiğinde anomali tabanlı karakutu saldırı tespit sistemi (IDS) modellerinin yorumlanmasında genellikle post-hoc açıklama yöntemlerinin kullanıldığı görülmektedir. En sık kullanılan yöntemler ise LIME (Local Interpretable Model-agnostic Explanations) ve SHAP (SHapley Additive exPlanations) ile karar ağacı kurallarına dayalı yöntemlerdir. Ayrıca yapay sinir ağı modellerini de kullanılabilen Layer-wise Relevance Propagation (LRP) yöntemi mevcuttur. Bu çalışma kapsamında anomali tabanlı saldırı tespit sistemlerinin (IDS) yorumlanmasında kullanılan post-hoc açıklanabilirlik yöntemlerinin genel hatları ve çalışma mantıkları alt başlıklar halinde aşağıda sunulmaktadır.

6.1.1. Local Interpretable Model-Agnostic Explanations (LIME)

LIME yaklaşımı kara kutu modellerin ayrık bireysel tahminlerinin vekil modeller (surrogate models) aracılığıyla açıklanabilirliğini sağlayan bir yaklaşımdır. LIME önceden eğitilmiş bir karakutu modelin bireysel tahminlerinin model bağımsız olarak açıklanmasını sağlanmaktadır. Bunu bir girdiyi oluşturan özelliklerin alt küme permutasyonlarını oluşturarak, tahminler yapılması ve en etkili özellik değerlerinin belirlenmesine dayalı çalışarak gerçekleştirmektedir. LIME yaklaşımında bir girdide yer alan özelliklerin bulunup bulunmama varyasyonlarına dayalı oluşturulan veriseti ve bu verisetindeki öğelerin orijinal karakutu model tahminleri etiket olarak kullanılarak kendiliğinden açıklanabilir bir model (lineer regresyon, Lasso, karar ağacı) eğilmektedir. Açıklanabilir modelin bu verisetine dayalı tahminlerinden en az sayıda özellik kullanarak hedeflenen girdinin orijinal model tahminine en yakın tahminlerine dayalı ayrık lineer açıklamalar oluşturulmaktadır. Bu ayrık lineer açıklamalar kullanılarak istenilen ***K*** adet özelliğe dayalı Lasso hata minimizasyonu gerçekleştirilerek, özelliklerin ağırlık değerlileri hesaplanmaktadır [103]. LIME tablo (tabular) tabanlı verilerin yanı sıra metin ve görüntü verisetleri üzerinde de çalışabilmektedir. LIME yaklaşımın girdi özelliklerine dayalı açıklanabilirlik değerlerinin hesaplanması Eş. 6.1 ve Eş 6.2' de gösterilmektedir.

$$\xi(x) = argmin_{g \in G} L(f, g, \pi x) + \Omega(g) \tag{6.1}$$

$$L(f, g, \pi x) = \sum_{z, z' \in Z} \pi x(z) \, (f(z) - g(z'))^2 \tag{6.2}$$

LIME'ın bireysel girdilerin tahminlerin yorumlanmasını nasıl gerçekleştirdiğine ilişkin çalışma akışı aşağıda maddeler halinde verilmektedir [103, 104].

- Açıklaması alınmak istenilen ve ilgilenilen bir girdi örneği seçilir ve bu girdinin kara kutu model tahminini alınır.
- Bu girdiyi oluşturan özelik değerlerinin yokluğuna dayalı permutasyonlar ile bir veri kümesi oluşturularak, bu sentetik oluşturulan girdi kümesi için kara kutu model tahminleri üretilir.
- Yeni oluşturulan örneklerin, orijinal girdiye uzaklığı ölçülerek ağırlıklandırma işlemi gerçekleştirilir. LIME da metin tabanlı veriler için bu işlemde metinde yer alan her bir kelimenin bulunup bulunmamasına göre oluşturulan permütasyonlar için basitçe,

girdiden kelime çıkarılarak oluşturulan yeni girdide yer alan kelime sayısının orijinal girdi metninde yer alan kelime sayısına oranı ağırlık değeri olarak alınmaktadır.

- Girdinin permütasyonlarına dayalı oluşturulan bu veri kümesi ile kendinden yorumlanabilir bir model eğitilir. LIME da bu yöntem varsayılan olarak lineer regresyon olup, karar ağacı, Lasso gibi dâhili (kendiliğinden) yorumlanabilir modellerde kullanılabilmektedir.
- Kendinden açıklanabilir yerel modelinin çıktıları yorumlayarak istenilen girdi için kara kutu model tahmini açıklanır.

LIME metin, görsel ve tablo tabanlı olmak üzere farklı veri türlerinde çalışması, anlaşılır ve basit açıklama yapısına sahip olması başlıca avantajlarıdır. Ayrıca model bağımsız yapısı sayesinde farklı yapay öğrenme modellerini açıklayabilmekte ve açıklanması istenilen model değişse dahi açıklamalarını yapmaya devam edebilmektedir. Bununla birlikte LIME çekirdek ayarlarının ve kullanımının uzman gözlemi ile gerçekleştirilmesi önemli bir kullanım zorluğu olup, doğru çekirdek (kernel) ayarları seçilmediği durumlar da açıklamalar yetersiz olacaktır. Ayrıca bir girdiden çıkarılacak veriler örneklenirken gauss dağılımı kullanılmaktadır. Bu durumda, verilerin birbiri ile etkileşimi ve birlikteliği, korelasyonları göz ardı edilmektedir. Korelasyonu yüksek verilerin birlikte göz önüne alınmaması model açıklamasının yetersiz yapılmasına yol açacaktır. Ek olarak, çalışma [105]' de gerçekleştirilen çalışma ile LIME'ın stabil çalışmadığı ve birbirine çok yakın iki girdinin açıklamasında önemli değişiklik gösterdiği gösterilmiştir. LIME'ın bir diğer dezavantajı da manipülasyon ile açıklamaların değiştirilebilmesidir [106]. Güvenilebilirliğin en önemli dayanaklarından birisi yorumlama modelinin stabil çalışması ve çok değişiklik göstermemesidir.

6.1.2. SHapley Additive exPlanations (SHAP)

SHAP [107] her bir bireysel girdi için yerel açıklanabilirlik sağlanmasının yanı sıra global açıklanabirlikte sunulabilmektedir. SHAP hesaplama yöntemi sayesinde sunduğu global açıklamalar ile modele en çok katkıyı sağlayan özelliklerin belirlenmesini sağlamaktadır. SHAP açıklama yöntemi LIME' a benzer şekilde model bağımsız (model-agnostic) ve post-hoc olarak çalışan bir yöntemdir. Yerel açıklanabilirlikte LIME'a çok benzer bir yöntem kullanmaktadır. SHAP yaklaşımında modellerin açıklanmasında kendiliğinden yorumlanabilir olan vekil (surrogate) modeller (lineer regresyon, karar ağacı) kullanılmaktadır. Özelliklerin tahmine katkısının hesaplanması oyun teorisine dayalı

Shapley değerlerinin hesaplanması ile gerçekleştirilmektedir. SHAP özelliklerin tahminine katkısını hesaplarken koalisyon oyun teorisi (coalitional game theory) tabanlı olarak, her özelliğin değerinin kendi başına ya da bir grup olarak çok oyunculu bir oyun skoruna katkısının bireysel bazda hesaplandığı bir yöntem kullanmaktadır. SHAP bir girdide bulunan özelliklerin değerlerini değiştirerek ya da ortadan kaldırarak tahmine etkisini hesaplarken, farklı özelliklerin bir arada model tahminine etkisinin ölçülmesinde ise oyun teorisini kullanmaktadır [107]. SHAP özellik değerlerinin olup olmamasına dayalı koalisyon vektörlerini oluştururken özellik değerinin yerine marjinal dağılıma dayalı olarak rastgele farklı özellik değerleri atamaktadır. Yerel açıklanabilirlik için KernelSHAP kullanılmakta olup, en temel SHAP modelini oluşturmaktadır. KernelSHAP yaklaşımın girdi özelliklerine dayalı açıklanabilirlik değerlerinin hesaplanmasında kullanılan hesaplamalar Eş 6.3 ve Eş 6.4' de görülmektedir.

$$g(z') = \phi_0 + \sum_{j=1}^{M} \phi_j z_j' \tag{6.3}$$

$$L(f, g, \pi_x) = \left[f\left(h(z')\right) - g(z')\right]^2 \pi_x(z') \tag{6.4}$$

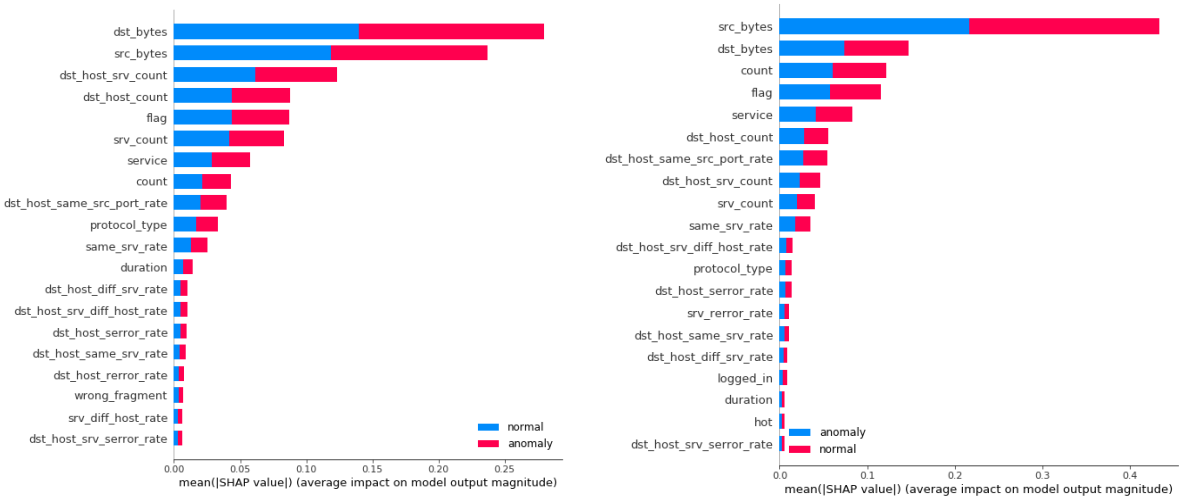
KernelSHAP'ın en önemli dezavantajlarından birisi çok yavaş çalışmasıdır. Shap değerlerinin hesaplanması yüksek hesaplama gücü gerektirmektedir. KernelSHAP özellik değerlerinin etkisini ölçerken özellik değerlerini değiştirmek için gauss dağılımı yerine her ne kadar marjinal dağılım kullansa da farklı özellik değerlerinin korelasyonunu dikkate alınmamakta olup, makine öğrenme algoritmaları için çok önemli olan özelliklerin kendi içindeki bağımlılıkları göz ardı edilmektedir. Ek olarak daha önce görmediği özellik değerleri barındıran yeni bir girdi geldiğinde Shapley değerleri yanlış hesaplanabilmekte olup, yeni gelen özellik değerlerinin, var olan özellik değerlerine dâhil edilip tekrar eğitilmesi gerekmektedir [108]. SHAP'ı manipüle etmek ve yanıltıcı açıklamalar oluşturmak mümkün olup, bu durum da model de yer alan önemli yanlılıkların gizlenmesine neden olabilmektedir [108]. Aşağıdaki bölümlerde SHAP ve LIME yaklaşımları kullanılarak ağ saldırı tespit sistemi (Network Intrusion Detection System, NIDS) ve WAF modellerinin açıklanabilirliği sağlanmıştır.

SHAP ile NIDS ve WAF modellerin açıklabirliğinin sağlanması

Bu bölümde DNN ve CNN derin öğrenme algoritmalarına dayalı geliştirilen iki sınıflı NIDS ve WAF modellerinin hem yerel hem de global olarak açıklanabilirliği ele alınmaktadır. İlgili modellerden NIDS modelinin eğitildiği NSL-KDD veriseti tabular verilerden

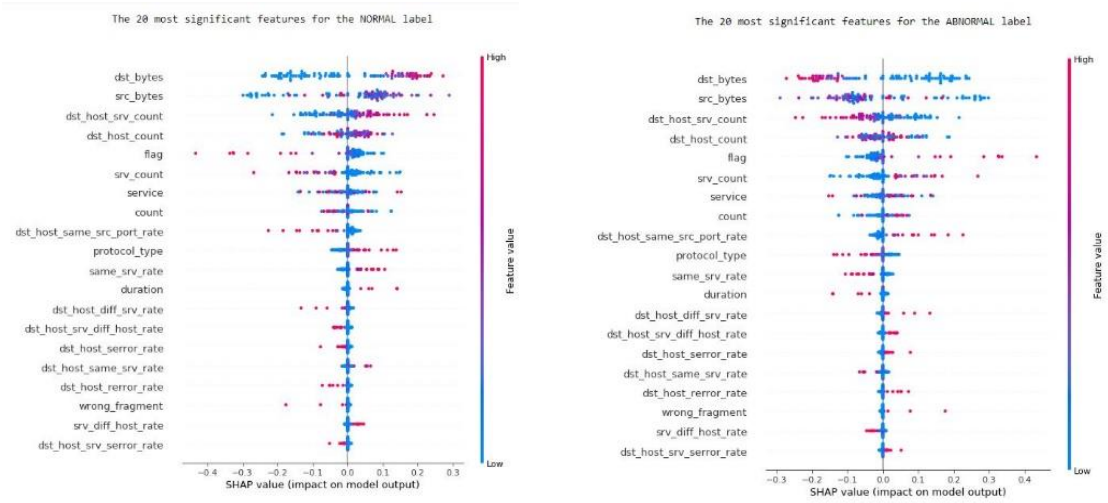
oluşurken, WAF modellerinin eğitildiği GAZI-HTTP ise metinsel web isteklerinden oluşmaktadır. Modellerin global açıklanabilirlirliğinin gerçekleştirilmesin de SHAP kullanılmıştır. SHAP bir verisetinde yer alan özelliklerin, modele katkısını ölçerek, modeli en çok etkileyen özellikleri sunabilmektedir. Global açıklanabilirliğin gerçekleştirilmesinde koalisyon oyun teorisi tabanlı hesaplanan Shapley değerleri kullanılmaktadır. Bir özelliğin modele katkısı Shapley değeri ile doğru orantılıdır. Bu kapsamda en yüksek Shapley değerine sahip özellik, modelin karar vermesine en çok katkı sağlayan, en önemli özelliktir. Bir özelliğin ortalama Shapley değeri; verisetinde ya da alt verisetinde yer alan her bir girdi için hesaplanan Shapley değerlerinin ortalaması alınarak hesaplanmaktadır. Aşağıda verilen SHAP grafikleri aksi belirtilmedikçe tüm veriseti üzerinde hesaplanan ortalama Shapley değerlerini ifade etmektedir.

NSL-KDD veriseti ile eğitilen NIDS modellerine, ortalama Shapley değerlerine göre en yüksek katkıyı sağlayan 20 özellik Şekil 6.3’ de görülmektedir. Şekil 6.3’ de görüldüğü üzere geliştirilen DL tabanlı NIDS modellerinde anomali tespitinde kullanılan en önemli özellik “src_bytes” ile ifade edilen (number of data bytes from source to destination) özellik olarak belirlenmiştir. İkinci en önemli özellik ise “dst_bytes” (number of data bytes from destination to source) özelliğidir. Her iki özellikte veri miktarı ile ilgili olduğu için, NSL-KDD veriseti üzerinde geliştirilen DL tabanlı NIDS modelinin saldırı tespitinde en çok veri miktarından etkilendiğini söyleyebiliriz. NSL-KDD verisetindeki en önemli 20 özelliğin normal (Şekil 6.4. a) ve saldırı (Şekil 6.4. b) tiplerinin belirlenmesinde özellik değerlerinin büyüklüğüne dayalı etkisi ise sınıf bazlı olarak Şekil 6.4’ de görülmektedir.



(a) DNN (b) CNN

Şekil 6.3. Shapley değerlerine göre NSL-KDD'deki en önemli 20 özellik



(a) Normal (b) Abnormal

Şekil 6.4. NSL-KDD üzerinde DNN modeli için en önemli 20 özelliğinin sınıf bazlı etkileri

Shap explainer ile bireysel girdilerin yerel açıklamaları da gerçekleştirilebilmektedir. NSL-KDD veriseti üzerinde normal ve saldırı olduğu bilinen girdilerin SHAP ile sağlanan yerel açıklamaları sırasıyla Şekil 6.5 (a) ve Şekil 6.5 (b)' de görülmektedir.



(a)

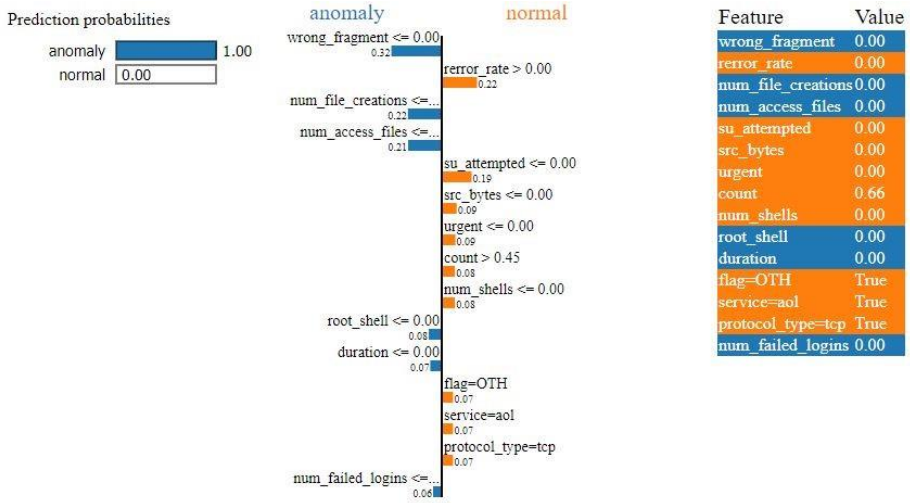
(b)

Şekil 6.5. NSL-KDD'deki normal (a) ve anormal (b) örneklerin SHAP' a dayalı yerel açıklamaları

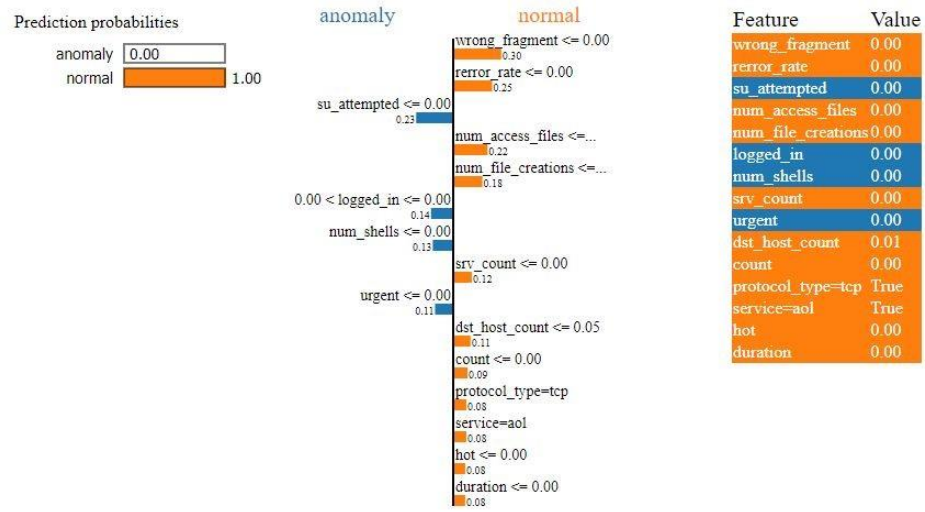
LIME ile NIDS ve WAF modellerin açıklabirliğinin sağlanması

LIME, bireysel girdilerin yerel açıklamalarını oluşturmak için kullanılmaktadır. NSL-KDD veri seti üzerinde eğitilen DNN modeli tarafından normal ve saldırı olarak sınıflandırılan iki örneğin açıklamaları sırasıyla Şekil 6.6 ve Şekil 6.7'de gösterilmektedir. LIME, bireysel girdilerdeki özelliklerin yokluğuna dayalı olarak doğrusal bir model ile açıklama yapmaktadır. LIME basit bir yapıya sahip olmasına rağmen açıklama gücü yüksektir. Bir saldırı örneği için özniteliklerin sınıflandırma sürecine etkilerini gösteren özniteliklerin ağırlık değerleri Şekil 6.7' de gösterilmiştir. Bu örnekler için, etki değerleri en önemli 15

özellik üzerinden hesaplanmış olup, tüm özelliklerin etki değerlerinin toplamı sınıf etiketlerinin tahmin olasılıklarını oluşturmaktadır.

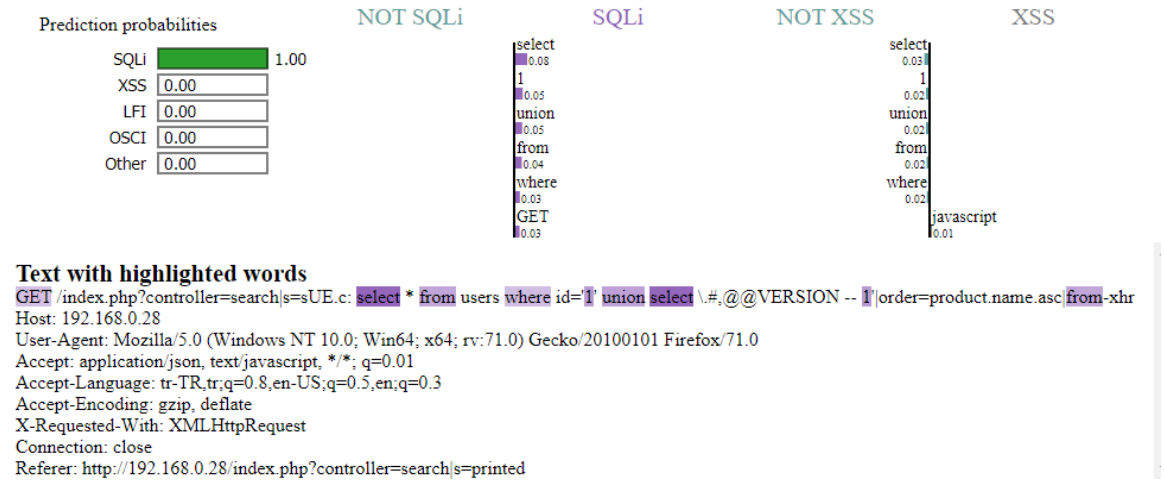


Şekil 6.6. LIME ile NSL-KDD' de normal bir örnek için DNN modeli tahmininin açıklaması

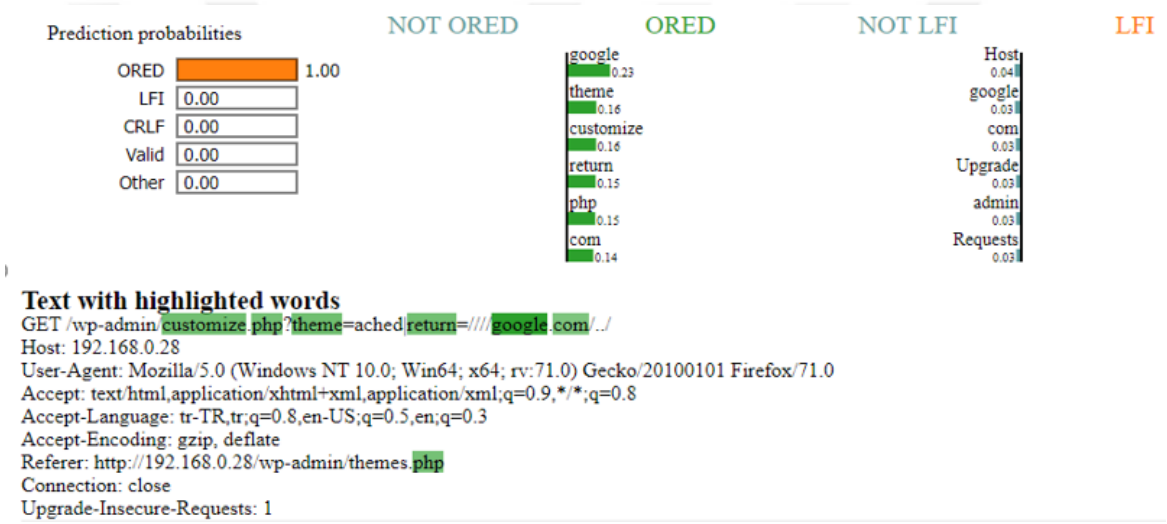


Şekil 6.7. LIME ile NSL-KDD' de normal bir örnek için DNN modeli tahmininin açıklaması

GAZI-HTTP veri setinde SQLi saldırı paternleri içeren bir web isteğinin DNN model tahmininin LIME açıklaması Şekil 6.8' de gösterilmiştir. Şekil 6.8 incelendiğinde LIME'nin SQLi saldırısı için doğru kelimeleri işaret ettiği görülmüştür. Çalışmada LIME ile oluşturulan yapı sayesinde web isteklerinin açıklanmasında metinsel grafiklerin gücünden yararlanılmıştır. Benzer şekilde, LIME ile Open Redirect (ORED) saldırısının açıklaması Şekil 6.9'de gösterilmektedir.



Şekil 6.8. LIME tarafından bir SQLi anomali örneğinin açıklaması



Şekil 6.9. LIME tarafından bir ORED anomali örneğinin açıklaması

6.2. XAI-WAF Açıklanabilirlik Çerçevesinin (Framework) Temelleri

Önerilen XAI-WAF framework, web uygulama güvenliğin alanına özel derin öğrenme tabanlı WAF modellerinin açıklanmasında kullanılabilecek yeni bir açıklama yaklaşımı içermektedir. Geliştirilen açıklanabilirlik yaklaşımı yerel açıklanabilirlik sağlamakta olup, bireysel girdilere ilişkin model tahminlerinin açıklanmasında kullanılmaktadır. Karakutu derin öğrenme model tarafından saldırı olup olmadığı veya saldırı tipi tahmin edilen bir web isteğinin, neden bu sınıfa ait olduğu web isteğindeki kelime ve kelime grupları ile açıklanmaktadır. Geliştirilen yöntemde model tahminleri, kelime komşuluk ilişkileri ve kelime gömme (word embedding) vektörlerinde yer alan kelimelerin korelasyonları kullanılarak kelimelerin tahmine etkileri bireysel ve topluluk halinde hesaplanmaktadır. Geliştirilen açıklanabilirlik yöntemi alana özel olduğu için model tahmini bir saldırı sınıfı

olduğunda, bu saldırı türüne en çok etki eden kelime / kelime grupları daha doğru belirleyebilmektedir. Geliştirilen yöntemin önemli avantajlarından birisi de açıklama modülüne model tahmini ile birlikte gerçekte olan doğru sınıf etiketinin verilmesi durumunda yanlış tahminlere sebep olan özelliklerin belirlenebilmesidir. Ayrıca geliştirilen yöntem ile saldırı tespit sistemleri için çok kritik sayılacak olan, saldırı türlerine ilişkin isteklerden hangi kelimeler çıkarıldığında normal isteğe dönüştüğünün yanı sıra saldırı türleri arasında geçişe neden olan kelimelerin belirlenmesi mümkün hale gelmektedir. Geliştirilen açıklanabilirlik modülü sınıf türleri arası değişim ve sınıflandırma olasılıklarına dayalı skorlar oluşturmakta ve görselleştirmede bu skorlara dayalı kelime vurgulamaları ve renklendirmeler ile gerçekleştirilmektedir. Geliştirilen yöntemde skorların hesaplanması ve web isteklerinin açıklanması için modelden bağımsız vekil sınıflandırıcılar kullanılmadığı için açıklanabilirlik yönteminin manipüle edilmesi mümkün olmayıp, stabil çalışabilmektedir. Kelime skorlarının oluşturulmasında ve kelimelerin filtrelenmesinde, öğrenme kümesinin istatistiksel analizine dayalı yöntemler kullanılmakta olup, yöntemin teknik detayları aşağıda ayrıntılı olarak ele alınmaktadır. Geliştirilen XAI-WAF yöntemi tüm derin öğrenme temelli WAF modellerinin açıklanmasında kullanılabilecek şekilde geliştirilmiştir. Önerilen yöntem web uygulamasının yapısına ve kullanıcı isteklerine bağlı olarak ayarlamalar yapılarak, kısa anlaşılır açıklamaların sunulması ile açıklanabilirlik hassasiyetinin yükseltilmesini açıklayıcıya gönderilen fonksiyon parametreleri ile gerçekleştirilebilmektedir.

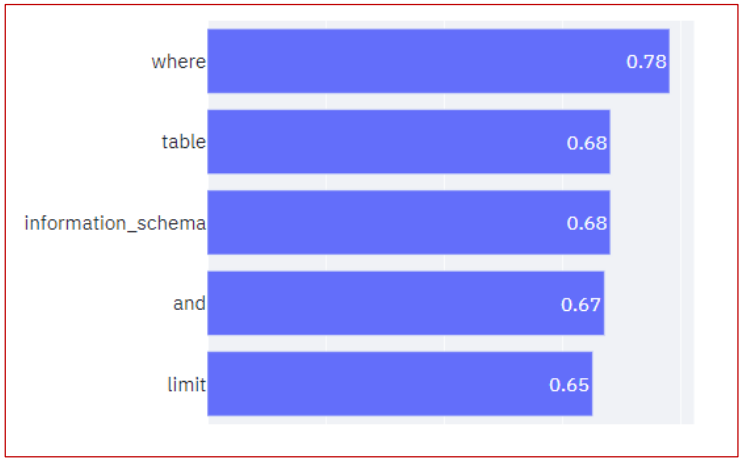
Web uygulamalarına yönelik saldırılar genellikle kullanıcı girdi parametrelerine gönderilen zararlı yükler (malicious payloads) ile gerçekleştirilmektedir. Gönderilen zararlı yükler tek bir kelimeden oluşabileceği gibi farklı karakter veya kelimelerin birleşiminden de oluşabilmektedir. Bu durumda özellikle yük kısmı olmak üzere WAF sistemleri için bir istek içinde yer alan komşu kelimelerin ilişkisi ve korelasyonu önemli hale gelmektedir. Buna ek olarak özellikle derin öğrenme yöntemleri başta olmak üzere web isteği gibi metin tabanlı verilerin işlenmesinde, metin içinde yer alan kelimelerin birbiri ile ilişkisi ve korelasyonuna dayalı vektörler oluşturulması için kelime gömme teknikleri yoğun olarak kullanılmaktadır. Mevcut permutasyona dayalı yöntemler özelliklerin birbiri arasındaki bağımlılıklarını ve korelasyonunu dikkate almadığı için özelliklerin ağırlıklandırılmasında önemli sorunlar ortaya çıkabilmekte ve stabil açıklanabilirlik sistemlerin geliştirilmesi güçleşmektedir. Önerilen yöntem özellik değerlerinin komşuluklarını dikkate aldığı ve word2vec gibi kelime benzerlik analizi yapılmasına imkân sağlayan kelime gömme tekniklerinin kullanılması

durumunda, aynı web isteğinde birden çok kez yer alan kelime ya da birbirine çok benzer kelimelerin gruplanarak model tahminine etkisinin ölçülebilmesi geliştirilen yöntemin önemli avantajlarından birisidir. Önerilen XAI-WAF yaklaşımında bir web isteğinde birden çok kez yer alan farklı konumlardaki aynı kelimeleri konumları ile birlikte değerlendirilmektedir. Bazen web isteğinin bir konumundaki aynı kelime model tahminin etkilemez iken, farklı konumdaki aynı kelime saldırı payload'u olarak değerlendirilebilmektedir. Bu işlemi kelimelerin bireysel ve komşuları ile ele alınması ile gerçekleştirmekte olup, mevcut açıklanabilirlik yöntemlerinde bu özellik mevcut değildir. Böylece özelliklerin ve aldıkların değerlerin bireysel önemlerinin yanı sıra birbiri arasındaki ilişkiler, konumları ve özellik değerlerinin birliktelik durumlarının karakutu modellerin tahminlerine katkısı ölçülebilmektedir.

Kelime gömme işleminin sınıflandırma açısından ne kadar etkili olduğunun görülmesi açısından GAZI-HTTP veriseti kullanılarak kelime gömme işlemi sonucunda oluşan vektörler üzerinden boyut indirgeme ve görselleştirme işlemi yapılmıştır. Word2vec modelini tez kapsamında bizim oluşturduğumuz GAZI-HTTP veriseti ile eğiterek kelime vektörleri oluşturulmuştur. Boyut indirgeme konusundaki güncel ve en iyi çözümlerden birisi t-Distributed Stochastic Neighbor Embedding (t-SNE) tekniğidir. Word2vec kelime vektörleri gibi çok boyutlu vektörlerin indirgenmesi ve görselleştirilmesinde kullanılan bir yöntemdir. GAZI-HTTP web anomali veriseti ile eğitilmiş Word2vec modeli ile "select" ve "where" kelimelerine en yakın olan 10'ar adet kelimenin t-SNE iki boyutlu görselleştirilmesi Şekil 6.10' de görülmektedir. Bu kelimelerin seçilme nedeni, veritabanı sorgularında sıklıkla yer almaları ve SQLi saldırısında sıklıkla geçmeleridir. Word2vec modeline göre "select" kelimesinin en yakın 5 komşusu ve benzerlik skorları Şekil 6.11' de görülmektedir. Bu benzerlik skorları kelime gömme işlemi ile oluşturulmakta ve derin öğrenme modellerinin başarısında çok önemli rol oynamaktadır. Geliştirilen XAI-WAF açıklanabilirlik yaklaşımı model tahminlerinin açıklanmasında bu ilişki ve korelasyon değerlerini de kullanmaktadır.



Şekil 6.10. Word2vec modeli için "select" ve "where" kelimelerine en yakın 10 adet kelimenin t-SNE ile iki boyuta indirgenmesi



Şekil 6.11. Word2vec modeline göre "select" kelimesine en yakın 5 kelime ve skorları

Geliştirilen XAI-WAF açıklanabilirlik yaklaşımının nasıl çalıştığı adım adım detaylı olarak aşağıda sunulmuştur.

- XAI-WAF framework 'ün sağladığı web arayüzü üzerinden açıklanılması istenilen bir web isteği ya da alarm seçilir ve bu web isteğinin karakutu model tahminini alınır.

- Bu web isteğinde yer alan ***N*** adet kelime sırası ile teker teker çıkarılarak, kalan ***N-1*** kelime için kara kutu model tahminleri üretilir. Modelin tüm kelimeler için tahminde bulunduğu saldırı türüne (örneğin XSS, SQLi, RCE vb.) ilişkin tahmin skoruna, web isteğinde yer alan her kelimenin etkisi ölçülerek, en çok etki eden ***M*** adet kelime belirlenir.

- En etkili ***M*** adet kelimenin ikili kelime kombinasyonları oluşturularak, karakutu WAF sistemi model tahmini gerçekleştirilir. İkili kelime vektörlerinin model tahminine, içerdiği bireysel kelimelerden daha çok etki yapması halinde bunlar gruplanır.

- En etkili gruplar sıralanarak kelime vektörleri oluşturulur, grupların içerdikleri bireysel kelimeler çıkarılır. Bireysel kelimelerin ikili vektörlerden daha çok etki yapması durumunda bireysel kelimeler de en etkili kelime vektörler listesine dâhil edilir.

- GAZI-HTTP web anomali veriseti kullanılarak eğitilen Word2vec kelime gömme modeli kullanılarak, en etkili kelime vektörlerinde yer alan kelimeler ile word2vec modeline göre bunlara en yakın ***K*** adet kelimeden aynı web isteği içinde yer alan kelimeler bulunarak, ikili kelime vektörleri oluşturulur. Bu ikili kelime vektörleri orijinal web isteğinden çıkarılarak, karakutu WAF sistemi saldırı türü sınıfına etkisi ölçülmektedir. İkili kelime vektörü, içermekte olduğu kelimelerden daha çok etki yapıyorsa, bu yeni ikili kelime vektörü de en etkili kelime vektörlerine dâhil edilir.

- Web isteklerinde yukarıda bahsedildiği üzere kullanıcı yükleri (payload) en sık saldırı paterni içeren kısımlar olup, yükler birden fazla kelimeden oluşabilmektedir. Bu durumda web isteği içindeki kelimelerin komşuları ile birlikte model tahminine etkisinin ölçülmesi de önemli hale gelmektedir. En etkili kelime vektörleri içinde yer alan kelimelerin sağındaki ve solundaki komşu kelimeler ile karakutu model tahminine etkisi ölçülmekte ve modelin tahmin ettiği sınıf etiketinin olasılık değerine

barındırdığı etkili kelimeden daha çok etki ettiği tespit edilirse bu ikili de en etkili vektörler arasına dâhil edilmektedir.

- Yukarıdaki adımlar sonucunda oluşturulan ikili ve/veya tek kelimelerden oluşan en etkili kelime vektörlerinin model tahminine etkilerinin standart sapması hesaplanmaktadır. Her sınıf etiketi için öğrenme kümesinde yer alan tüm girdiler için bu standart sapma değerleri bir kez hesaplanmakta ve ortalaması alınmaktadır.

- İlgilenilen girdi için hesaplanan etki değerlerinin standart sapması ve karakutu WAF sistemi tarafından tahmin edilen sınıf için önceden hesaplanan standart sapmaların ortalamasından hangisi büyük ise bu değer eşik değer olarak belirlenmektedir.

- En etkili kelime vektörlerinden etki değeri belirlenen bu eşik değerin altında kalan kelime vektörleri atılmaktadır. Kalan kelime vektörleri yeni en etkili kelime vektörlerini oluşturmaktadır.

- Yeni kelime vektörlerinde yer alan kelimelerin, çoklu kelime etkilerini ölçmek için ***(1,C)*** li kombinasyonlarının karakutu WAF sistemi model tahminine etkisi ölçülmekte, ayrıca model tahminini değiştirip değiştirmediği belirlenerek, gerçek sınıf etiketi, tahmin sınıf etiketi, normal ve saldırı tipleri arasındaki geçişe göre skorlar atanmaktadır.

- Model tahminin değiştirmeyen kombinasyonlar arasında yukarıdakine benzer şekilde kombinasyonların etki değerlerinin standart sapması ve tahmin sınıfına ilişkin hesaplanan etki değerlerinin standart sapmasının ortalamasından maksimum olanı eşik değer olarak belirlenerek, eşik değerin altında kalan kombinasyonlar göz ardı edilmektedir.

- Oluşturulan kombinasyonların WAF sistemi tahmin türünü değiştirip değiştirmediğine göre kelime kombinasyonlarına skorlar verilmektedir. Bu işlem gerçekleştirilirken, açıklanması istenilen girdinin gerçek sınıf etiketi belli ise, bu etiket de kelime kombinasyonlarının renklendirilerek etkilerinin vurgulanması (highlited) ve skorlarının belirlenmesinde kullanılmaktadır. Eğer gerçek sınıf etiketi belli değilse modelin girdi için tahmin sonucu gerçek sınıf etiketi olarak kabul edilmektedir.

- Kombinasyonlarda yer alan kelimelerin girdiden çıkarılması sonucunda model tahminleri değişiyorsa ve modelin bu girdi için yanlış sınıflandırma yaptığı belirlendi ise (girdi için açıklama modülüne gerçek sınıf etiketinin gönderilmesi durumunda), eğer yanlış tahminden doğru tahmine geçiş varsa bu kelime kombinasyonları ile model tahmini arasında negatif korelasyon olduğu için negatif değer ile skorlanmaktadır.

- Eğer WAF sistemi doğru sınıflandırma yaptı ve yanlış sınıfa geçiş varsa bu kelime kombinasyonları model tahmini ile pozitif korelasyon içerdiği için pozitif değerler ile skorlanmaktadır.

- Eğer doğru sınıf etiketinden yanlış sınıfa geçti ve bir saldırı türünden "Normal" türüne yani saldırı olmama durumuna geçiş varsa bu kombinasyon çok önemlidir, saldırı paternlerini içermektedir ve yüksek pozitif değer ile skorlanır.

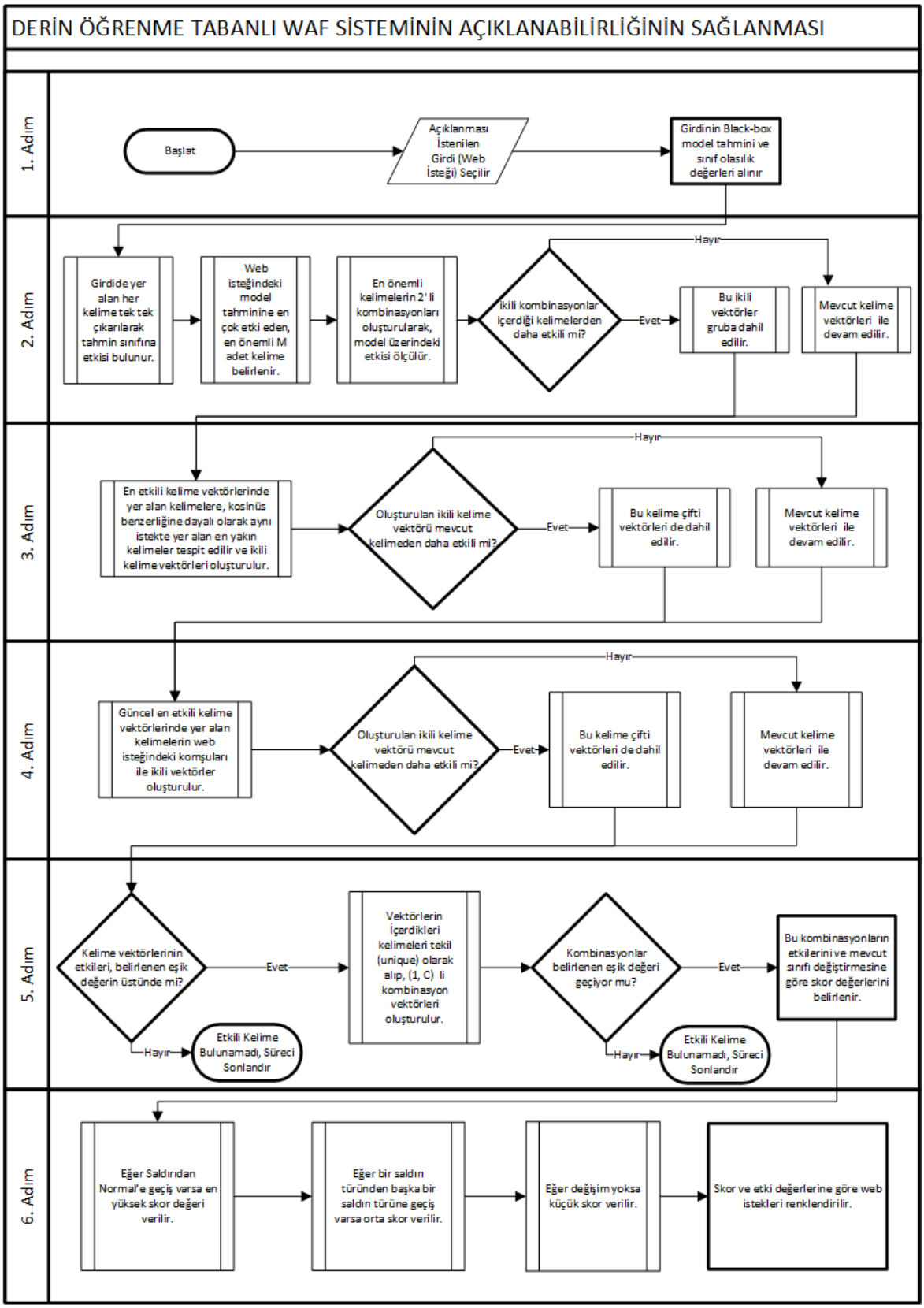
- Eğer yanlış sınıf türünden doğru sınıf türüne geçiş varsa, bu kelime kombinasyonu önemlidir, yanlış sınıflandırmaya neden olduğu için negatif orta değer ile skorlanmaktadır.

- Eğer bir saldırı türünden başka bir saldırı türüne geçiş varsa ve ilk tahminin doğru olduğu biliniyorsa pozitif orta değer, ilk tahminin yanlış olduğu biliniyorsa negatif orta değer ile skorlanır. Eğer yorumlama modeline gerçek sınıf etiketi gönderilmedi ise modelin ilk tahmini doğru kabul edileceği için pozitif orta değer ile skorlanmaktadır.

- Kelime kombinasyonlarının yukarıdaki adımlar sonucunda skor ve etki değerlerine bağlı olarak pozitif ve negatif skorlar için farklı renkler kullanılmak üzere etki değerleri ve skor derecelerine göre farklı renk tonları kullanarak web istekleri vurgulanmaktadır.

- Renkli olarak vurgulanan web isteği, XAI-WAF framework 'ün sağladığı web arayüzü aracılığıyla sistem yöneticisine sunulmaktadır.

Geliştirilen XAI-WAF açıklanabilirlik yaklaşımının iş akış şeması Şekil 6.12' de görülmektedir.



Şekil 6.12. XAI-WAF açıklanabilirlik yaklaşımı akış diyagramı

XAI-WAF ile SQLi olduğu bilinen bir web isteğinin, Word2vec ve komşuluğuna olarak dayalı sadece ikili kelime vektörlerinin oluşturulması sonucunda, model tahminine en çok etki eden tek ve ikili kelime vektörleri ve etki skor değerleri Şekil 6.13'de görülmektedir. Burada eşik değerleri aşan kelime grupları işleme alınmamıştır. Buradaki pozitif değerler model tahmini ile pozitif ilişkiyi gösterirken, negatif değerler tahmine olumsuz katkı sağlayan kelime vektörlerini göstermektedir.

	term_index	term	group_id	group_proba
0	[63]	[select]	0	0.000196
1	[68]	[or]	1	0.000008
2	[66]	[where]	2	0.000004
3	[52]	[origin]	3	0.000003
4	[48]	[with]	4	0.000003
5	[63, 68]	[select, or]	0	0.001953
6	[63, 66]	[select, where]	0	0.001581
7	[63, 52]	[select, origin]	0	0.000932
8	[63, 48]	[select, with]	0	0.000773
9	[68, 66]	[or, where]	1	0.000033
10	[68, 52]	[or, origin]	1	0.000010
11	[68, 48]	[or, with]	1	0.000008
12	[66, 52]	[where, origin]	2	0.000017
13	[66, 48]	[where, with]	2	0.000014
14	[63, 66]	[select, where]	0	0.001581
15	[63, 68]	[select, or]	0	0.001953
16	[63, 69]	[select, or]	0	0.001953
17	[68, 63]	[or, select]	1	0.001953
18	[68, 66]	[or, where]	1	0.000033
19	[66, 63]	[where, select]	2	0.001581
20	[66, 68]	[where, or]	2	0.000033
21	[66, 69]	[where, or]	2	0.000033
22	[68, 69]	[or, or]	1	0.000021

Şekil 6.13. Bir SQLi web isteğinin XAI-WAF yaklaşımına göre sınıflandırılmasında en etkili olan kelime vektörleri

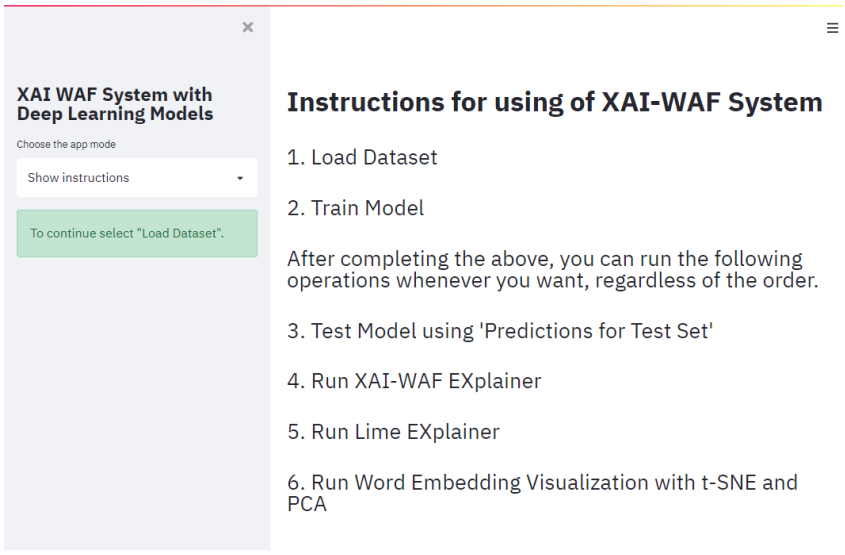
Yukarıda tek ve ikili vektörleri verilen SQLi isteği için en etkili kelime vektörleri oluşturulduktan sonra kelime kombinasyonlarının oluşturulması ve etki değerleri eşik değerin üzerinde kalan kombinasyonlar için skorların belirlenmesi sonucunda oluşan açıklanabilirlik matrisi Şekil 6.14 ‘de görülmektedir.

	term_index	term	proba_effect	pred	alteration	score
0	(63, 68, 66)	[select, or, where]	0.043845	SQLi	False	1
1	(63, 68, 66, 52)	[select, or, where, origin]	0.037137	SQLi	False	1
2	(63, 68, 66, 48)	[select, or, where, with]	0.034598	SQLi	False	1
3	(63, 68, 52)	[select, or, origin]	0.010246	SQLi	False	1
4	(63, 68, 48)	[select, or, with]	0.009043	SQLi	False	1

Şekil 6.14. Eşik değeri geçen kelime vektörlerinin etki değerleri, skorları ve yorumlanabilirlik matrisi

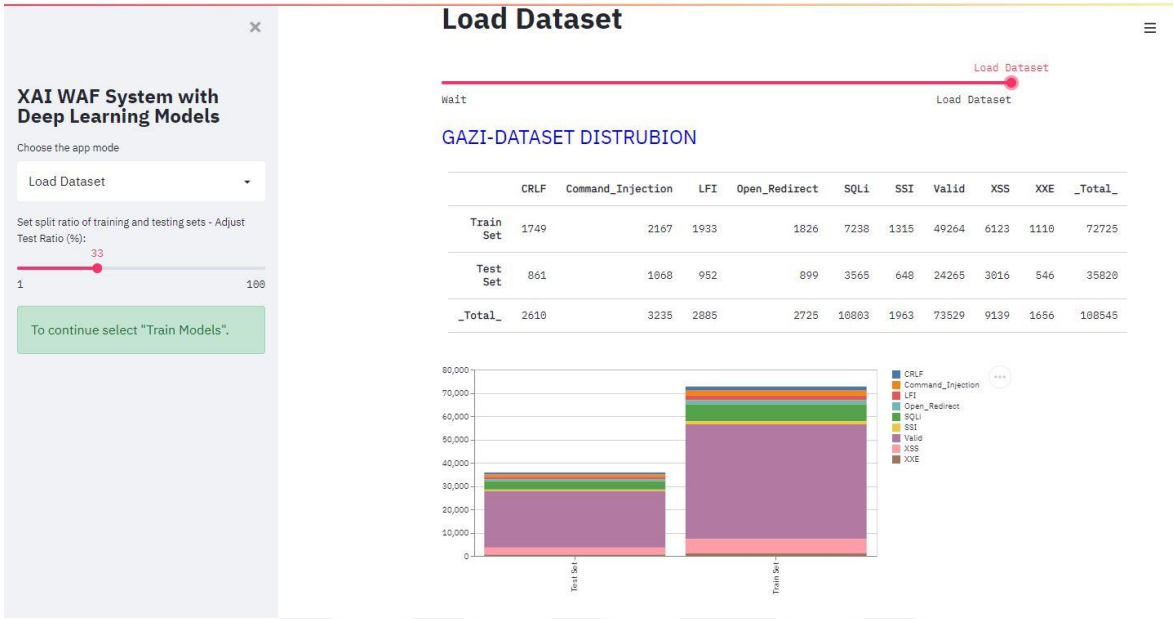
6.3. XAI-WAF Açıklanabilirlik Framework Arayüz ve Açıklanabilirlik Çıktıları

Tez kapsamında geliştirilen XAI-WAF framework web tabanlı olarak geliştirilmiştir. Web arayüzene ilk girildiğinde adım adım XAI-WAF sisteminin nasıl kullanılacağını açıklayan arayüz Şekil 6.15’ de görülmektedir. Öncelikle kullanılacak web anomali verisetinin yüklenmesi ve model eğitiminin gerçekleştirilmesi gerekmektedir. Önceden eğitilmiş bir modelde solda yer alan “Load Model” sekmesi seçilerek model parametreleri ve ağırlık değerini içeren dosyalardan doğrudan yüklenebilmektedir. Veriseti yükleme ve model oluşturma / yükleme işlemleri tamamladıktan sonra istenilen adımlardan devam edilebilmektedir.



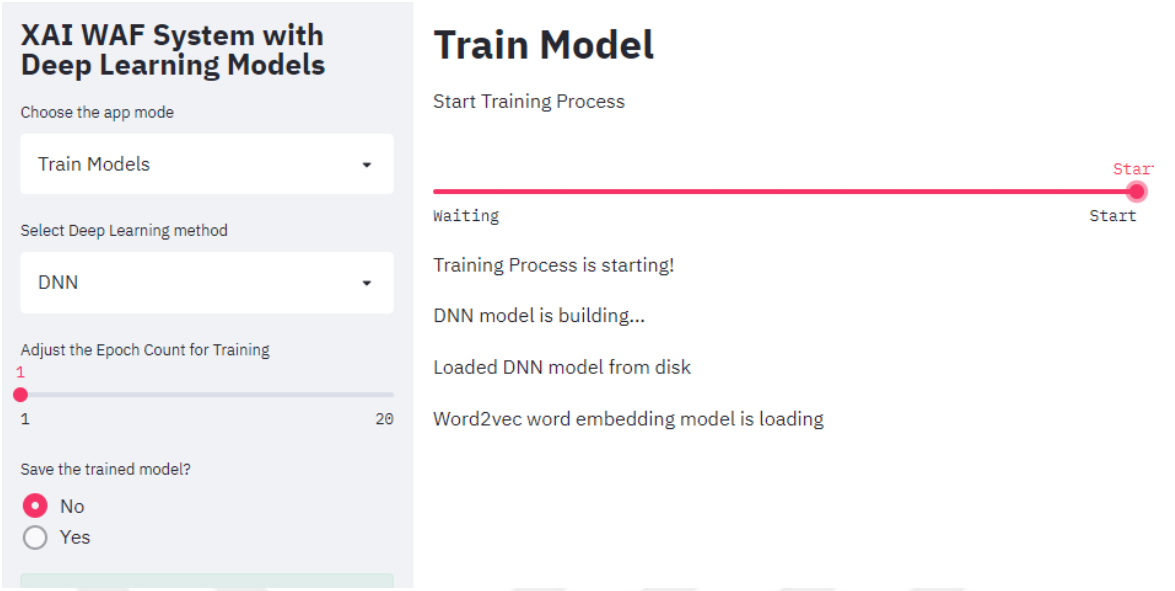
Şekil 6.15. XAI-WAF framework ‘ün giriş arayüzü

Şekil 6.16’ da görüldüğü üzere XAI-WAF web arayüzü kullanılarak veriseti yüklenebilmekte, yüzde bölmeye dayalı olarak istenilen oranda veriseti bölme işlemi gerçekleştirilebilmektedir. Bölme işlemi belirlenen orana göre rastgele olarak yapılmaktadır.



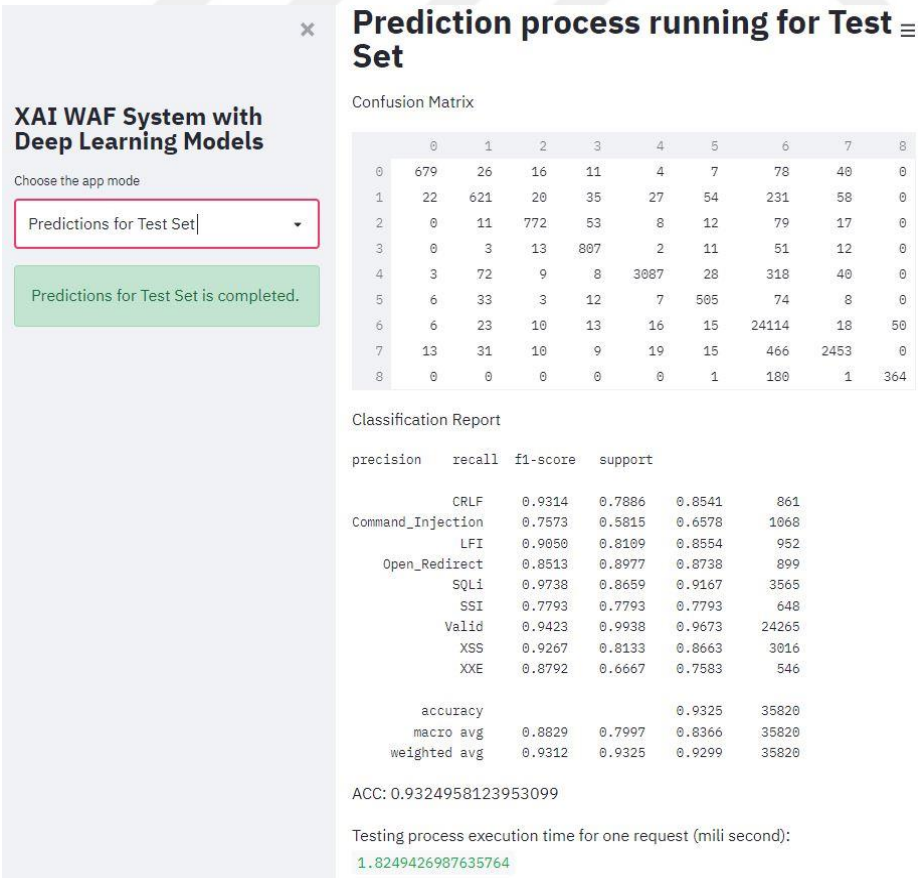
Şekil 6.16. XAI-WAF framework’ün veriseti yükleme arayüzü

XAI-WAF framework ‘ün web arayüzü aracılığıyla, istenilen derin öğrenme algoritması ve mimarisi seçilerek, gerekli eğitim parametreleri ve optimizayon işlemleri web arayüzünden yapıldıktan sonra daha önce yüklenen veriseti ile istenilen döngü sayısında model eğitimi gerçekleştirilebilmektedir. Dilenirse daha önce eğitilmiş ve kaydedilmiş derin öğrenme model parametreleri ve ağırlık değerli yüklenerek, doğrudan kullanılabilmekte ya da istenilen döngü sayısında yüklenen veriseti ile yüklenen ağırlık değerlerinde başlatılarak tekrar model eğitimi gerçekleştirilebilmektedir. Geliştirilen XAI-WAF framework’ü tüm derin öğrenme algoritma ve mimarilerini destekleyecek şekilde modüler olarak geliştirilmiştir. Derin öğrenme modellerinin eğitildiği web arayüzü Şekil 6.17’ de görülmektedir.



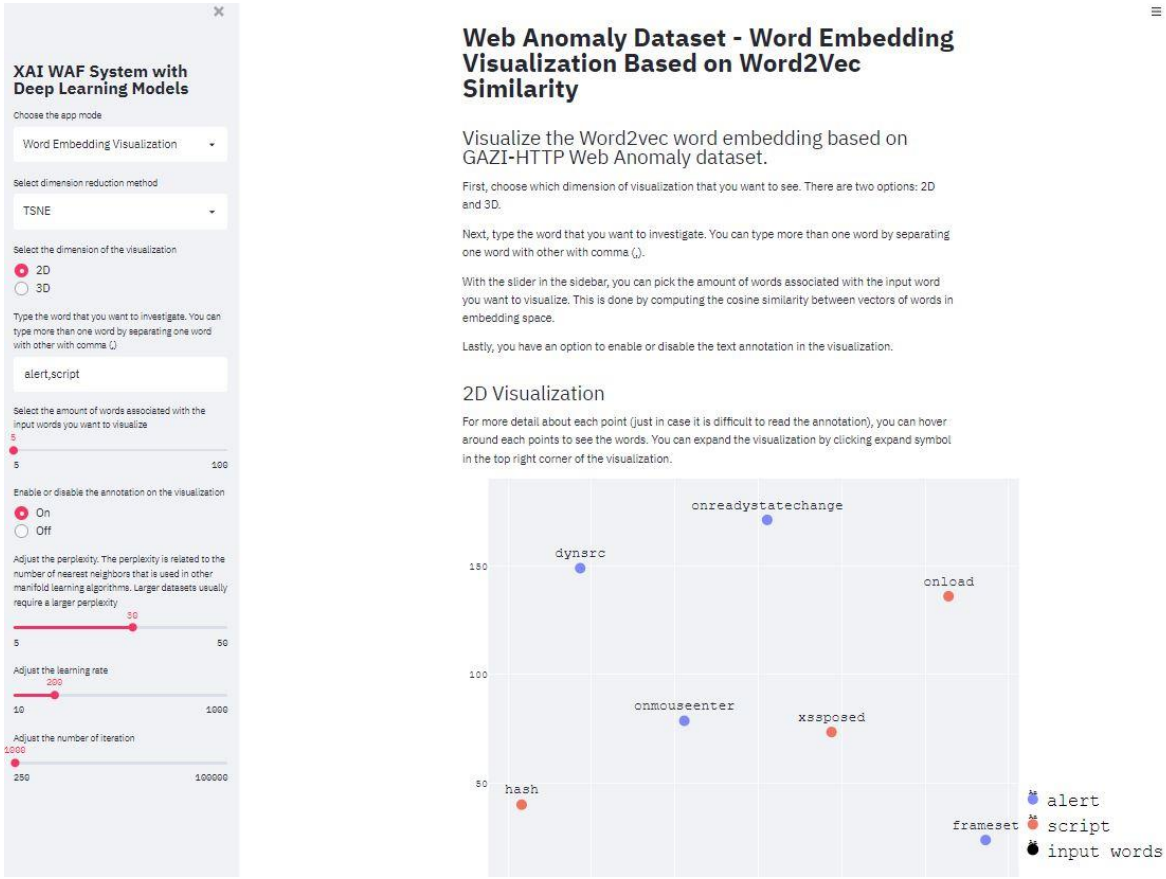
Şekil 6.17. XAI-WAF framework' ün derin öğrenme modeli eğitim arayüzü

XAI-WAF framework 'ün web arayüzü aracılığıyla, Şekil 6.18' de görüldüğü üzere; eğitilen ya da daha önce eğitilip yüklenmiş olan derin öğrenme modeli veriseti yükleme aşamasında bölünen test kümesi ile test edilerek, model tahminlere göre oluşturulan karşıtlık matrisi ve model performans değerleri görüntülenebilmektedir.



Şekil 6.18. Eğitimi tamamlanan DL modelinin test edilmesi ve performans değerleri

XAI-WAF framework ’ün web arayüzü aracılığıyla, Şekil 6.19 ve Şekil 6.20’ de görüldüğü üzere; yüklenen web anomali verisetinde bulunan isteklerin içerdikleri kelimelere dayalı olarak eğitilen Word2vec modellinin benzerlik vektörlerine üzerinde kelime birliktelik analizleri ve görselleştirilmesi gerçekleştirilebilmektedir. XAI-WAF framework ’ün de yer alan veri görselleştirme arayüzü ile bu analiz işlemi gerçekleştirilmektedir. Word2vec vektörlerinin görselleştirilebilmesi için yukarıda açıklandığı üzere veri indirgeme ve özellik seçme işlemlerinin yapılması gerekmektedir. XAI-WAF arayüzü üzerinden hem PCA hem de t-SNE ile 2 boyutlu ve 3 boyutlu veri görselleştirme ve kelime benzerlik listelerinin çıkarılması işlemleri yapılabilmektedir.



Şekil 6.19. XAI-WAF framework’ ün veri görselleştirme arayüzü



Şekil 6.20. XAI-WAF framework' ün kelime benzerlik analizi arayüzü

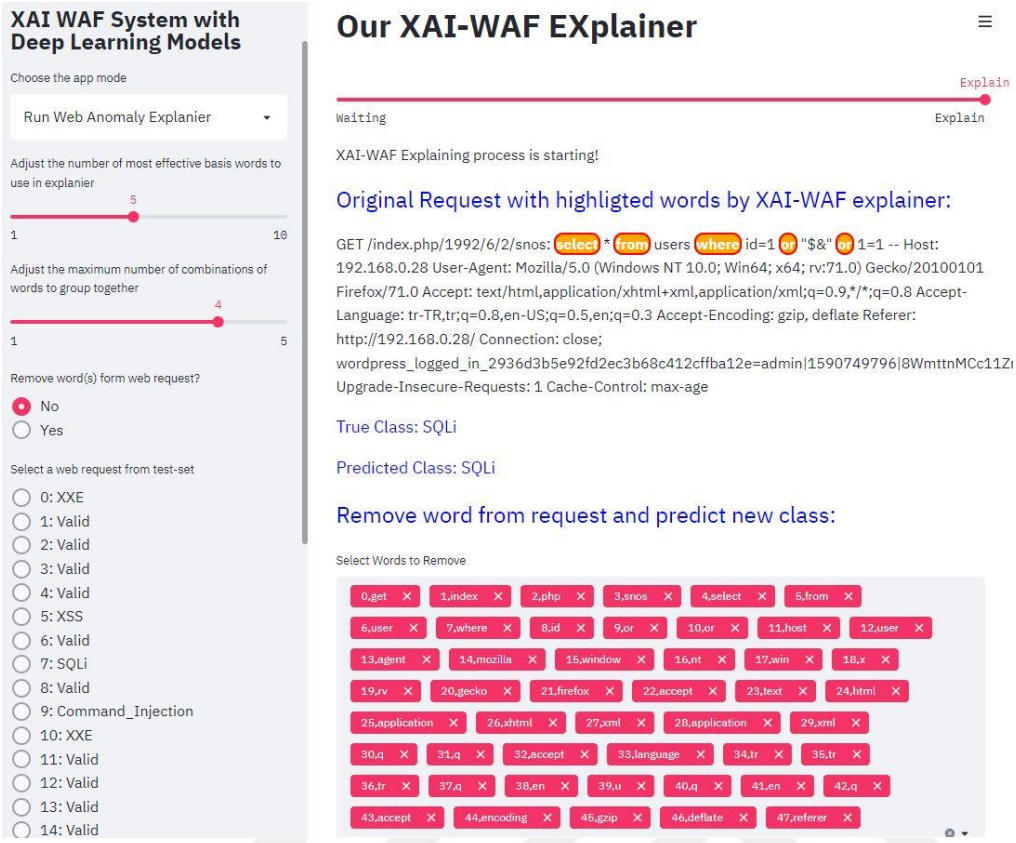
XAI-WAF framework' ü ile geliştirilen açıklanabilirlik modülü ile istenilen web isteğinin açıklaması ve en etkili kelime vektörleri oluşturulabilmektedir. Çalışma kapsamında geliştirilen açıklanabilirlik yaklaşımı için kullanışlı bir açıklama arayüzü oluşturulmuştur. Öncelikle, arayüz üzerinden tahmin sonucu açıklanmak istenilen web isteği seçilmekte ve açıklamaya en etkili kaç kelime ile başlanacağı, kelime birliktelik vektörleri oluşturmak için gereken maksimum vektör uzunluğu arayüz üzerinden seçilmektedir. Daha sonra, geliştirilen açıklanabilirlik yaklaşıma ile seçilen web isteğinin karakutu model tahminine dayalı olarak en etkili kelime ve kelime vektörleri ve model tahminine etkileri yukarıda açıklandığı üzere hesaplanmaktadır. Oluşan etkili kelime, kelime vektörleri ve skor değerlerine göre girdi web isteği üzerinde renklendirme ile yerel açıklanabilirlik sağlanmaktadır. Ayrıca etkili kelime ve kelime vektörleri için etki değerleri en önemliden başlamak üzere tablo halinde sunulmaktadır. Web arayüzü üzerinden sağlanan bir liste ile seçilen web isteğinde yer alan tüm kelimeler sırası ile etiketlenmiş şekilde web ekranında gösterilmekte ve istenilen kelimelerin ilgili web isteğinden çıkarılabilmesi, yeniden karakutu model tahmini üretilmesi ve bu yeni türetilen isteğin yeni karakutu model

tahminine dayalı olarak açıklanabilirliği sunulabilmektedir. Böylece istenilen kelime ya da kelimeler web isteğinden çıkarılarak, bu kelime/kelimelerin model tahminine etkisi doğrudan ölçülebilmektedir. Aşağıda farklı saldırı türlerine ilişkin XAI-WAF üzerinden oluşturulan açıklamalar ve etki değeri sunulmuştur.

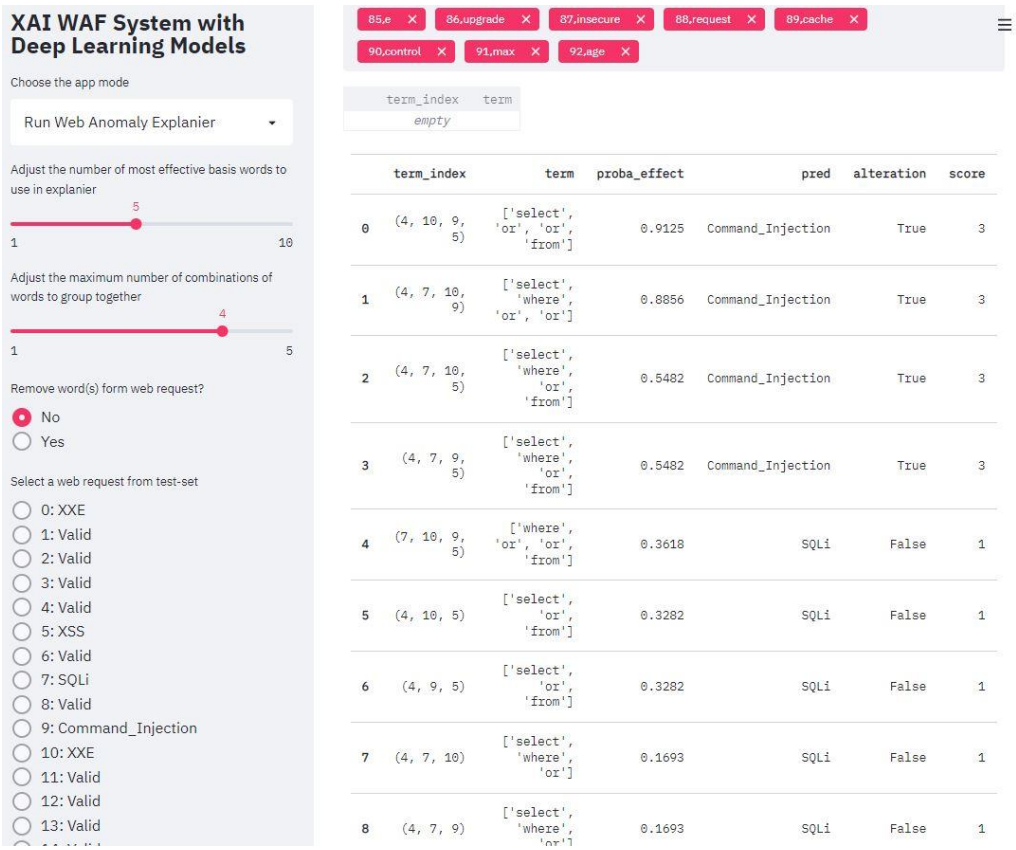
Test kümesinden seçilen ve SQLi sınıfında olduğu bilinen bir web isteğinin XAI-WAF açıklamaları ve en etkili kelime vektörleri ile etki değerleri, sırasıyla Şekil 6.21 ve Şekil 6.22'de görülmektedir. Şekil 6.21' de görüleceği üzere SQLi olarak tahmin edilen bir web isteği girdisinin modelin bu tahminine en çok katkı yapan ve diğer kelimeler ile birlikte hesaplanan eşik değeri aşan "select", "from","where","or","or" kelimeleri XAI-WAF tarafından yeşil renk ile işaretlenmiştir. Yeşil renk model tahminine pozitif katkı sağlandığını, gerçek sınıf etiketinin XAI-WAF sistemine girildiği durumlarda; eğer model yanlış tahminde bulundu ise kırmızı renkli kelimeler modelin yanlış tahminine neden olan özellikler olarak vurgulanmaktadır. Bir web isteğinde aynı kelime birden çok yerde geçiyorsa kelimenin kendisinden bağımsız olarak konumu da dikkate alınmaktadır. Bu örnekte iki adet "or" kelimesi vurgulanmış olup, bunların etki değerleri ayrı ayrı oluşturulmuş ve kelimenin konumu, komşu kelimeler ile olan ilişkisi de dikkate alınmıştır. Bu durum diğer açıklanabilirlik yöntemlerinin hiçbirisinde mevcut olmayıp, önerilen açıklanabilirlik yaklaşımının güçlü özeliklerinden birisidir. Şekil 6.22 incelendiğinde model tahminine en çok etki eden kelime ya da kelime grupları (term_index), tahmine etki skorları (score), tahmini değiştirme miktarı (proba_effect), bu kelime / kelime grupları çıkarıldığı durumda tahminin değişip değişmediği (alteration) ile web isteğinin yeni model tahmini (pred) sırasıyla görülmektedir. Buna göre bu istek için model tahminine en çok etki eden kelime vektörü ["select","or","or", "from"] olup, bu kelime grubu içerdiği tüm alt kümelerden daha fazla etki etmekte ve sorgu sonucunu değiştirmektedir. Bu kelimeler birlikte çıkarıldığında modelin yeni tahmini "Command_Injection" olacaktır.

Test kümesinden seçilen ve XSS sınıfında olduğu bilinen bir web isteğinin XAI-WAF açıklamaları ve en etkili kelime vektörleri ile etki değerleri, sırasıyla Şekil 6.23 ve Şekil 6.24'de görülmektedir.

Test kümesinden seçilen ve LFI sınıfında olduğu bilinen bir web isteğinin XAI-WAF açıklamaları ve en etkili kelime vektörleri ile etki değerleri, sırasıyla Şekil 6.25 ve Şekil 6.26'de görülmektedir.

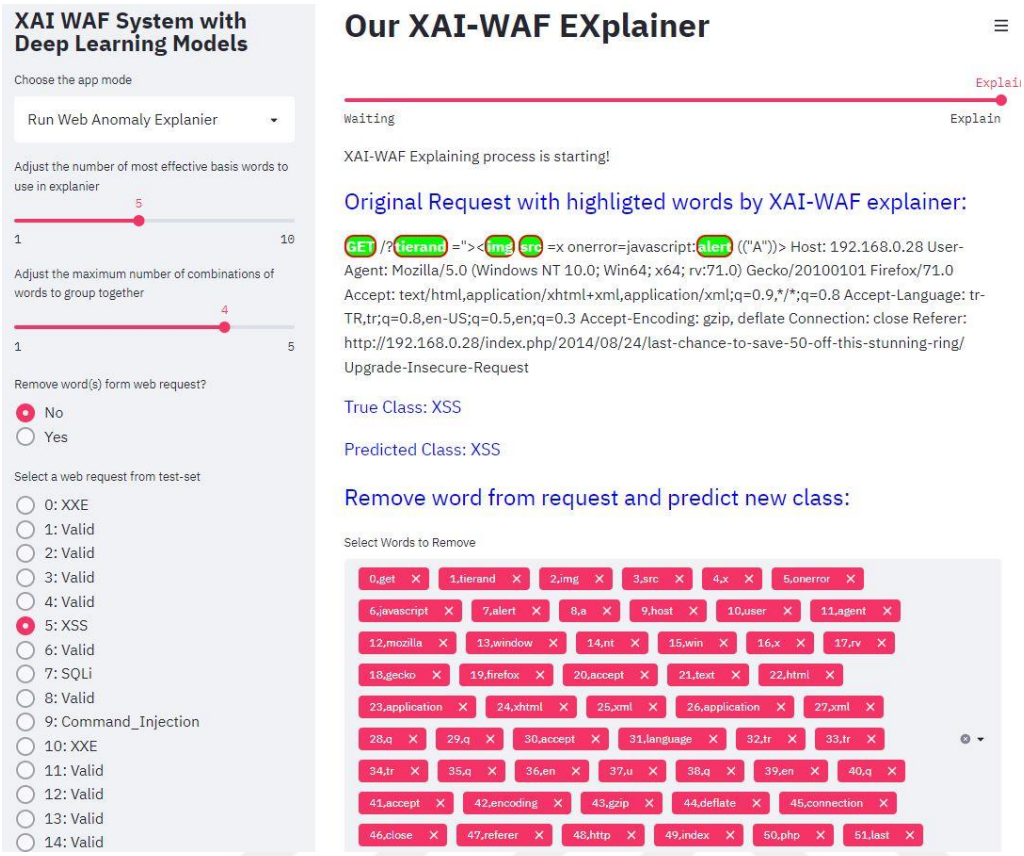


Şekil 6.21. SQLi isteğinin XAI-WAF modülü ile açıklanabilirliğin sağlanması

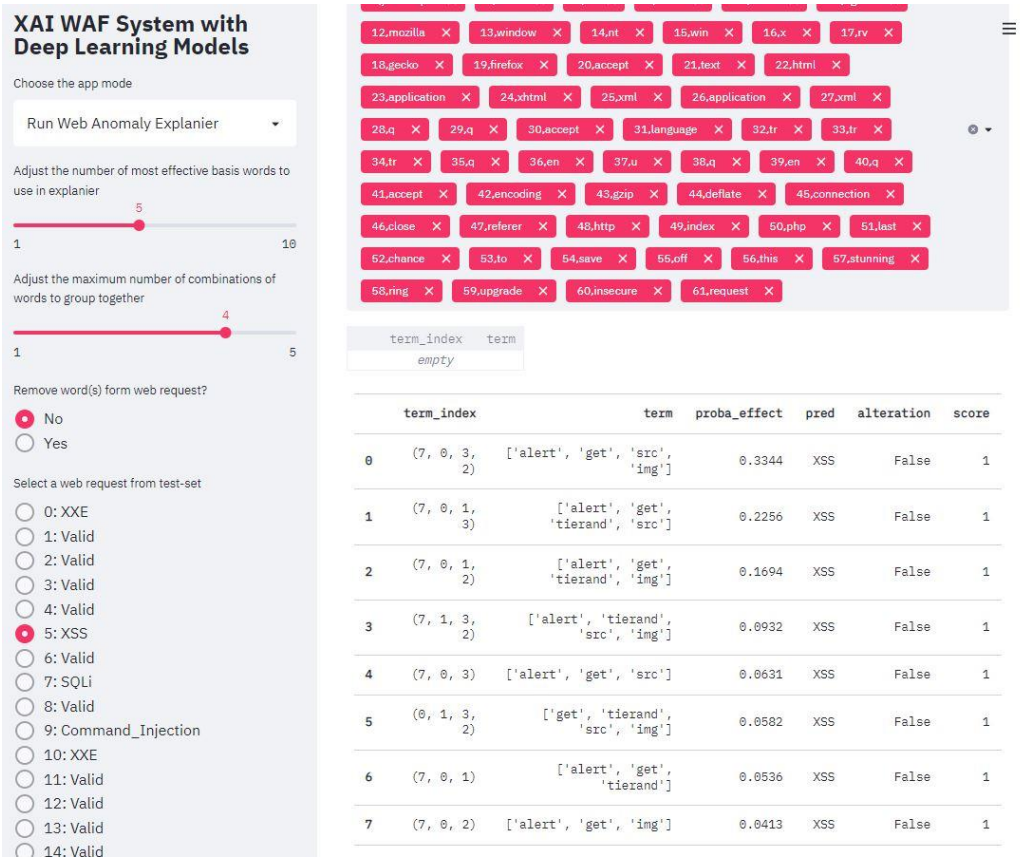


	term_index	term	proba_effect	pred	alteration	score
0	(4, 10, 9, 5)	['select', 'or', 'or', 'from']	0.9125	Command_Injection	True	3
1	(4, 7, 10, 9)	['select', 'where', 'or', 'or']	0.8856	Command_Injection	True	3
2	(4, 7, 10, 5)	['select', 'where', 'or', 'from']	0.5482	Command_Injection	True	3
3	(4, 7, 9, 5)	['select', 'where', 'or', 'from']	0.5482	Command_Injection	True	3
4	(7, 10, 9, 5)	['where', 'or', 'or', 'from']	0.3618	SQLi	False	1
5	(4, 10, 5)	['select', 'or', 'from']	0.3282	SQLi	False	1
6	(4, 9, 5)	['select', 'or', 'from']	0.3282	SQLi	False	1
7	(4, 7, 10)	['select', 'where', 'or']	0.1693	SQLi	False	1
8	(4, 7, 9)	['select', 'where',	0.1693	SQLi	False	1

Şekil 6.22. SQLi isteğinde yer alan en etkili kelimeler ve etki değerleri

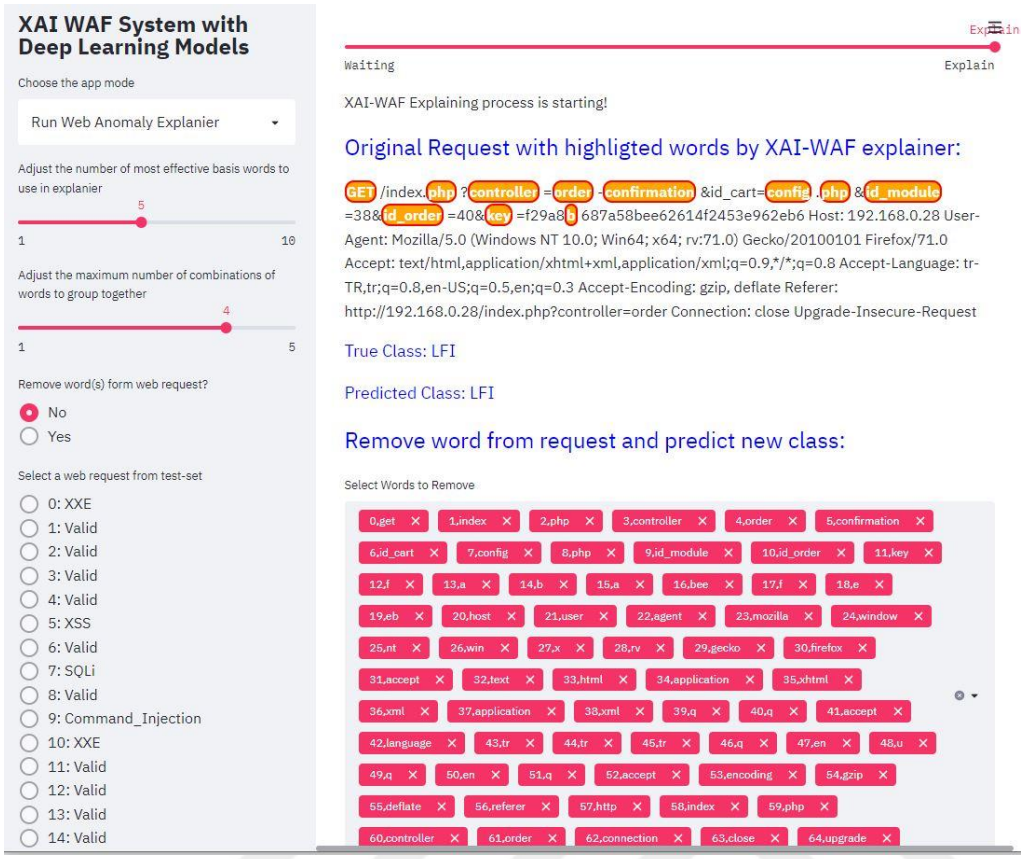


Şekil 6.23. SQLi isteğinin XAI-WAF modülü ile açıklanabilirliğin sağlanması

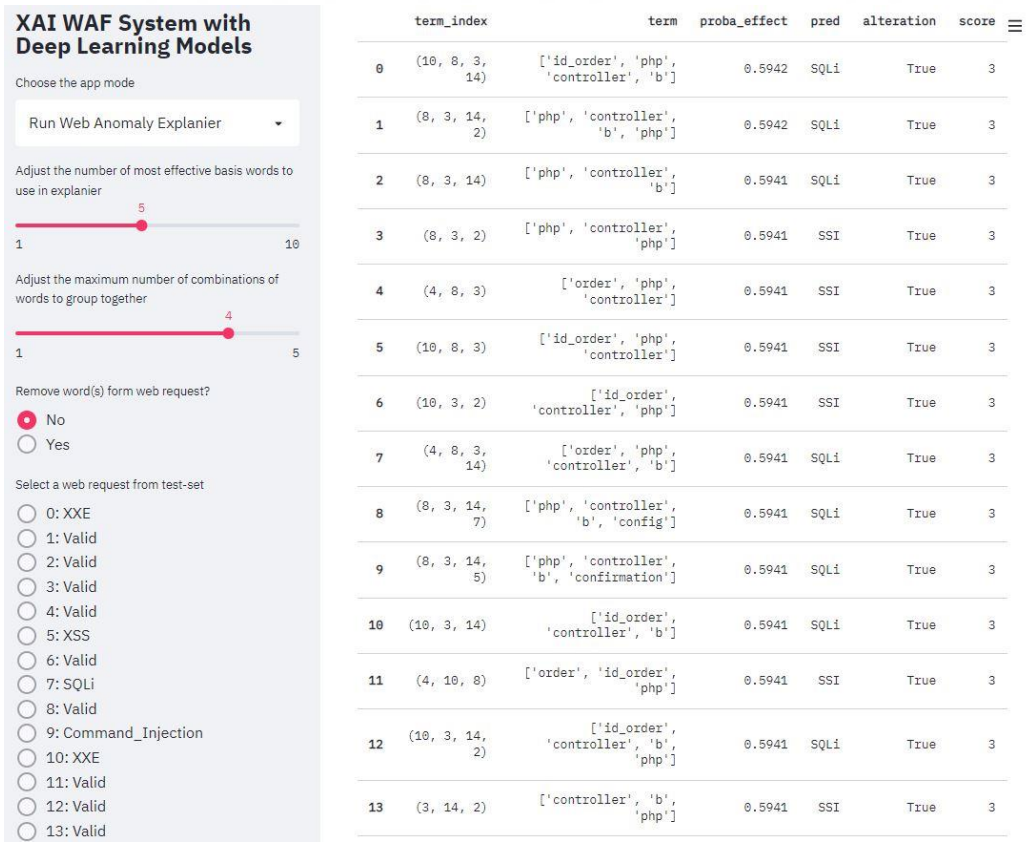


	term_index	term	proba_effect	pred	alteration	score
0	(7, 0, 3, 2)	['alert', 'get', 'src', 'img']	0.3344	XSS	False	1
1	(7, 0, 1, 3)	['alert', 'get', 'tierand', 'src']	0.2256	XSS	False	1
2	(7, 0, 1, 2)	['alert', 'get', 'tierand', 'img']	0.1694	XSS	False	1
3	(7, 1, 3, 2)	['alert', 'tierand', 'src', 'img']	0.0932	XSS	False	1
4	(7, 0, 3)	['alert', 'get', 'src']	0.0631	XSS	False	1
5	(0, 1, 3, 2)	['get', 'tierand', 'src', 'img']	0.0582	XSS	False	1
6	(7, 0, 1)	['alert', 'get', 'tierand']	0.0536	XSS	False	1
7	(7, 0, 2)	['alert', 'get', 'img']	0.0413	XSS	False	1

Şekil 6.24. SQLi isteğinde yer alan en etkili kelimeler ve etki değerleri



Şekil 6.25. LFI isteğinin XAI-WAF modülü ile açıklanabilirliğin sağlanması



	term_index	term	proba_effect	pred	alteration	score
0	(10, 8, 3, 14)	['id_order', 'php', 'controller', 'b']	0.5942	SQLi	True	3
1	(8, 3, 14, 2)	['php', 'controller', 'b', 'php']	0.5942	SQLi	True	3
2	(8, 3, 14)	['php', 'controller', 'b']	0.5941	SQLi	True	3
3	(8, 3, 2)	['php', 'controller', 'php']	0.5941	SSI	True	3
4	(4, 8, 3)	['order', 'php', 'controller']	0.5941	SSI	True	3
5	(10, 8, 3)	['id_order', 'php', 'controller']	0.5941	SSI	True	3
6	(10, 3, 2)	['id_order', 'controller', 'php']	0.5941	SSI	True	3
7	(4, 8, 3, 14)	['order', 'php', 'controller', 'b']	0.5941	SQLi	True	3
8	(8, 3, 14, 7)	['php', 'controller', 'b', 'config']	0.5941	SQLi	True	3
9	(8, 3, 14, 5)	['php', 'controller', 'b', 'confirmation']	0.5941	SQLi	True	3
10	(10, 3, 14)	['id_order', 'controller', 'b']	0.5941	SQLi	True	3
11	(4, 10, 8)	['order', 'id_order', 'php']	0.5941	SSI	True	3
12	(10, 3, 14, 2)	['id_order', 'controller', 'b', 'php']	0.5941	SQLi	True	3
13	(3, 14, 2)	['controller', 'b', 'php']	0.5941	SSI	True	3

Şekil 6.26. LFI isteğinde yer alan en etkili kelimeler ve etki değerleri

6.4. Çalışmanın Sonuç Bölümü

Anomali tabanlı saldırı tespit sistemlerine güven açısından modellerin şeffaf olması ve nasıl karar aldığının bilinmesi önemlidir. Bu çalışma kapsamında derin öğrenme tabanlı WAF modellerinin açıklanabilirliğinin sağlaması amacıyla yeni bir XAI-WAF framework'ü geliştirilmiştir. Çalışma kapsamında web güvenliği alanına özgü, yeni ve güçlü bir açıklanabilirlik yaklaşımı geliştirilmiştir. Çalışma kapsamında geliştirilen açıklanabilirlik yaklaşımı tüm derin öğrenme algoritmalarına dayalı WAF modellerinin bireysel tahminlerini, web isteğinde yer alan kelime ve kelime gruplarına bağlı olarak açıklamaktadır. Geliştirilen açıklanabilirlik yaklaşımının, mevcut yaklaşımlardan en önemli farkları; web güvenliği alanına özgü olması, web isteklerinde yer alan kelimelerin birliktelik ve korelasyonlarını dikkate alması, kelime gömme vektörlerindeki benzerlik ve korelasyonun dikkate alması, güçlü ve kullanılabilir bir arayüz ile açıklanabilirlik ve analizlerin gerçekleştirilmesine imkân sağlanmasıdır. Çalışma kapsamında ayrıca mevcut açıklanabilirlik yaklaşımları ve derin öğrenme tabanlı saldırı tespit sistemlerine uygulanması ele alınmıştır. Bu kapsamda DNN ve CNN tabanlı geliştirilen IDS modellerinin global ve yerel açıklanabilirliğinin sağlanması amacı ile SHAP ve LIME tabanlı bir yapı oluşturulmuştur. Modellere etki eden en önemli özellikler ortalama Shapley değerlerine dayalı olarak açıklanırken, bireysel açıklamalarda Shapley değerline ek olarak LIME tarafından oluşturulan ağırlık değerleri kullanılmıştır.

7. SONUÇ

Web uygulamalarının güvenliğinin sağlanması devletler, kurumlar, kuruluşlar ve şahıslar açısından çok kritik olup, web uygulamalarında bulunan açıklıkların sömürülmesi durumunda hassas veriler sızabilmekte, önemli ekonomik kayıplar oluşabilmekte, kullanıcılar ve sistemler telafisi mümkün olmayan zararlar görebilmektedir. Web uygulama güvenliğinin sağlanmasında çözülmesi gereken çok önemli zorluklar mevcuttur. Gerçek zamanlı olarak kullanılabilecek, hızlı, başarılı ve güvenilir web uygulama güvenlik duvarlarının geliştirilmesi bu zorlukların başında gelmektedir.

Bu tez kapsamında web uygulama güvenliğinin sağlanmasında karşılaşılan farklı problemler ele alınmış ve derin öğrenme tabanlı anomali tespit yaklaşımına dayalı olarak çözümler geliştirilmiştir. Bu kapsamda ele alınan ilk problem güçlü ve genelleştirebilme yeteneği olan, kapsamlı ve güncel bir verisetinin oluşturulmasıdır.

- Web anomali verisetlerinin sayısı çok kısıtlı olup, güncel saldırı tiplerini ve saldırı paternlerini içeren kapsamlı bir veriseti bilgimiz dâhilinde mevcut değildir. Anomali tabanlı web güvenliği çalışmalarının gerçekleştirilebilmesi amacı ile GAZI-HTTP isimli yeni bir web anomali veriseti oluşturulmuştur.

- GAZI-HTTP veriseti iki farklı yöntem ile oluşturulmuş olup, ilk yöntemde internette herkesin erişimine açık şekilde konumlandırılan farklı web uygulamalarına gelen tüm web istekleri kayıt altına alınmış ve imza tabanlı ModSecurity WAF sistemi ile etiketlenmiştir. İkinci olarak bu web uygulamaları yerel sunuculara kurulmuştur. Web uygulamalarının aldığı web istekleri ve parametreler taranarak belirlenmiştir. Bu isteklerdeki parametrelere uygun olarak, çok kapsamlı bir tarama işlemi ile internette ve saldırı araçlarında yer alan saldırı yükleri toplanarak farklı saldırı tiplerini için saldırı yük listeleri oluşturulmuştur. Ayrıca web istek parametrelerine uygun olarak, dünya da sık kullanılan terimlere dayalı normal web istek yük listeleri oluşturulmuştur. Bu yük listeleri kullanılarak, otomatize araçlar ile web trafikleri oluşturulmuş ve kaydedilmiştir.

- GAZI-HTTP veriseti biri normal olmak üzere 9 farklı etikete sahiptir. Oluşturulan veriseti bilgimiz dâhilinde en güncel ve en kapsamlı web anomali verisetidir. GAZI-HTTP veriseti bu tezin önemli çıktılarından birisi olup, herkese açık olarak

paylaşılması ve bundan sonra gerçekleştirilecek çalışmalara önemli bir katkı sağlanması planlanmaktadır.

Bu tez kapsamında ikinci olarak web anomali verisetlerinin dengesiz (imbalanced) dağılmasından kaynaklanan sınıflandırma ve hassasiyet problemleri çözmek amacı ile web anomali verisetlerine özgü DA-SANA isimli yeni bir veri artırma (data augmention) yöntemi geliştirilmiştir. Ayrıca bu çalışma kapsamında yüksek başarım ve hassasiyete sahip Bi-LSTM tabanlı WAF modelleri geliştirilmiştir.

- DA-SANA veri artırım yöntemi, web isteklerinin uzunluklarına dayalı adaptif gürültü eklemesi ile oluşturulan yeni web isteklerinin mevcut verisetine eklenmesi ile veri artırım işlemi gerçekleştirmektedir. Çalışma kapsamında yapılan analizler ve daha önceki çalışmalar göz önünde bulundurularak saldırı web isteklerinin bir web uygulamasının aldığı web isteklerinin uzunluklarının varyansını önemli oranda değiştirdiği ve web saldırı tespiti için web istek uzunluklarının önemli olduğundan yola çıkılarak geliştirilmiştir. Veri artırımı işlemi web isteklerinin mevcut temsillerini bozmadan, web isteklerinin uzunluğu ile doğru orantılı olarak web isteğinde bulunmayan nesnelerin gürültü olarak eklenmesi ile gerçekleştirilmektedir.

- Çalışma kapsamında geliştirilen DA-SANA ve Bi-LSTM derin öğrenme mimarisine dayalı geliştirilen WAF modelleri ile mevcut çalışmalar ile karşılaştırıldığında bilgimiz dâhilinde hem iki sınıflı hem de çok sınıflı web anomali tespitinde en yüksek başarıma sahip modellerdir. Farklı derin öğrenme mimarilerine dayalı geliştirilen WAF modelleri ile DA-SANA yönteminin etkinliği ölçülmüştür. Geliştirilen DA-SANA yönteminin çok sınıflı web anomali tespitinde önemli bir iyileştirme sağladığı ve GAZI-HTTP ve ECML-PKDD verisetleri için alanda kabul gören (state-of-the-art) beş farklı DL algoritmasının sınıflandırma doğruluğunda ortalama olarak %6.52 artış sağlandığı görülmüştür. GAZI-HTTP veri seti için iki sınıflı sınıflandırmada %97,5, çok sınıflı sınıflandırmada ise %96,6 gibi yüksek bir başarı oranları elde edilmiştir. Çalışma kapsamında ayrıca, ECML-PKDD veri setinde iki sınıflı sınıflandırmada %99,47, çok sınıflı sınıflandırmada ise %93,91; CSIC-HTTP veriseti için ise iki sınıflı sınıflandırma da %99,08 ile literatürdeki en yüksek tespit oranlarına ulaşılmıştır.

- Geliştirilen DA-SANA yöntemi ile düşük dağılıma sahip saldırı tiplerinin sınıflandırma hassasiyetlerin artırıldığı gösterilmiştir.

Bu tez kapsamında üçüncü olarak web saldırı tespitinden kaynaklı gecikmelerin önüne geçmek ve gerçek zamanlı kullanılabilir bir WAF sistemi oluşturmak amacı ile iki aşamalı, katmanlı ensemble öğrenmeye dayalı derin öğrenme temelli bir WAF sistemi geliştirilmiştir. Bu çalışma ile normal web kullanıcılarının saldırı türü tespitinden kaynaklı sofistike sistem gereksinimi ve zaman gecikmesinin önlemek amacı ile iki aşamalı çalışan bir model oluşturulmuştur. Geliştirilen WAF modelinin ilk aşamasında iki sınıflı saldırı tespiti gerçekleştirilmektedir. Birinci aşamada normal olarak sınıflandırılan web istekleri web uygulamasına gönderilirken, saldırı olduğu tespit edilen istekler ağdan düşülmekte ve ikinci aşama da yer alan katmanlı ensemble modele iletilerek saldırı türü yüksek hassasiyet ile belirlenmektedir.

- İlk aşamada hızlı ve yüksek başarıma sahip bir DNN modeli kullanılmaktadır. Böylece, gerçek zamanlı kullanılabilecek hızda bir WAF modeli oluşturulmuştur.

- İkinci aşamada CNN ve DNN temel öğrenci modelleri ve bu modellerin çıktılarını birleştirmede kullanılan DNN meta öğrenici model yer almaktadır. Geliştirilen ensemble modelin genelleştirme gücünün yüksek olması ve eğitim kümesi dışındaki web uygulamalarında güvenliğinin sağlanmasında kullanılması amaçlanmıştır. Bu kapsamda farklı istatistiksel analiz yöntemleri ile benzerlik ve çeşitlilik analizleri yapılarak en başarılı ve çeşitliliğe sahip temel modellerin bir arada kullanılması sağlanmıştır.

- Geliştirilen iki aşamalı WAF modelinin başarımını kıyaslamak amacıyla farklı DL algoritmaları ile tek algoritmaya dayalı modeller geliştirilmiştir. Önerilen model ile GAZI-HTTP ve ECML-PKDD için çok sınıflı sınıflandırma da diğer modellere göre sırasıyla %1,27 ve %4,57 oranında iyileştirme sağlanmış ve sırasıyla %97,43 ve %94,77 tespit oranlarına ulaşılmıştır. Önerilen modelin, bireysel DL algoritmalarına ve önceki çalışmalara göre önemli bir performans artışına sahip olduğu görülmüştür.

Bu tez kapsamında gerçekleştirilen dördüncü çalışmada model şeffaflığı ve açıklanabilirliği bulunmayan derin öğrenme tabanlı karakutu WAF modellerinin açıklanabilirliğinin sağlanması amaçlanmıştır. Anomali tabanlı WAF modellerinin performansının yüksek olması gerçek dünya da kullanılabilmesi için tek başına yeterli değildir. Modellerin şeffaf

olması ve tahminlerinin yorumlanabilmesi anomali tabanlı saldırı tespit sistemlerine güven açısından önemlidir. Bu kapsamda derin öğrenme tabanlı tüm WAF modellerinin tahminlerini model bağımsız açıklayabilen web uygulama güvenliği alanına özel XAI-WAF isimli yeni bir açıklanabilirlik yöntemi ve web tabanlı yazılım çerçevesi (framework) geliştirilmiştir.

- Derin öğrenme temelli WAF modelleri yüksek tespit oranlarına sahip olmak ile birlikte modellerin ve model tahminlerinin kendiliğinden yorumlanabilmesi mümkün olmayıp karakutu modeller olarak adlandırılmaktadır. Bu çalışma ile WAF sistemi olarak geliştirilen tüm derin öğrenme tabanlı modellerin tahminlerinin yerel olarak açıklanmasını sağlayan bir açıklanabilirlik yaklaşımı geliştirilmiştir.

- Geliştirilen XAI-WAF açıklanabilirlik yönteminde, bir web isteğine ilişkin karakutu model tahmininin açıklanması, temelde web isteğinde yer alan kelimelerin yokluğuna dayalı oluşturulan alt veri kümesinin karakutu model tahminlerinin istatistiksel olarak ağırlıklandırılması ile gerçekleştirilmektedir.

- XAI-WAF yönteminde mevcut açıklanabilirlik yöntemlerinin çoğundan farklı olarak lineer regresyon ya da karar ağacı gibi kendiliğinde yorumlanabilir bir vekil model kullanılmamakta, tüm süreç karakutu model tahminleri, kelime gömme vektörlerinin içerdiği benzerlik ve web isteğindeki kelime komşulukları ile gerçekleştirilmektedir. Böylece diğer mevcut açıklanabilirlik yaklaşımlarımda var olan kolay manipüle edilme sorunu XAI-WAF için ortadan kalkmaktadır.

- Mevcut açıklanabilirlik yaklaşımlarında kelime gömme süreçlerinin oluşturduğu vektörler ve içerdiği kelime benzerlik bilgileri dikkate alınmamaktadır. XAI-WAF yaklaşımında Word2vec modeline dayalı olarak bir web isteğinde yer alan kelimelerin birbiri ile benzerlik matrisi model tahminlerinin yorumlanmasında kullanılacak kelimeler ve kelime gruplarının belirlenmesi aşamalarında aktif olarak kullanılmaktadır.

- Ayrıca XAI-WAF yaklaşımı ile bir web isteğinde yer alan kelimelerin birlikte model tahminine etkisi ölçülmekte ve içerdiği alt kümelerden daha çok etkiye sahip olan kelime grupları açıklanabilirlikte kullanılmaktadır.

- XAI-WAF yaklaşımında, bir web isteğinde yer alan kelimelerin sağında ve solundaki komşu kelimeler ile birlikteliklerinin etkisi dikkate alınan başka bir unsurdur. Böylece web isteğindeki kelimelerin konum bilgisi de açıklanabilirlik sürecine dâhil edilmiştir. Ayrıca diğer açıklanabilirlik yaklaşımlarında bulunmayan, bir web isteğinde içinde aynı kelimenin birden fazla kez bulunması durumununda bu kelimelerin birbirinden bağımsız özellikler olarak değerlendirilmesi, XAI-WAF yaklaşımın önemli avantajlarından birisidir.

- Geliştirilen XAI-WAF yöntemi ile bir tahmine etki eden kelime ve kelime grupları belirlenmekte, web isteğinin karakutu model tahminini doğru ya da yanlış şekilde değiştirme / değiştirmeme durumlarına dayalı olarak, model tahminin normal ya da hangi saldırı türüne ait olduğuna göre ağırlıklandırılmakta ve renklendirilerek açıklanabilirlik süreci tamamlanmaktadır.

- Geliştirilen XAI-WAF yaklaşımı için herkesin kullanabileceği kolay ve kullanışlı bir web tabanlı framework geliştirilmiştir. Yaklaşım ve framework tüm derin öğrenme algoritmalarını ve modellerini destekleyecek şekilde modüler olarak geliştirilmiştir.

- XAI-WAF yaklaşımı bilgimiz dâhilinde web güvenliği alanına odaklı ilk açıklanabilirlik yaklaşımı olup, mevcut genel yöntemlere göre daha stabil ve başarılı çalıştığı görülmüştür.

Tez kapsamında gerçekleştirilen yukarıdaki çalışmalar doğrultusunda yapılan bilimsel çalışmalar ve bilime sağlanan katkılardan bazıları aşağıda sunulmaktadır.

- Tez çalışması kapsamında 2 adet SCI-E makale yayımlanmıştır. XAI-WAF çalışmasına ilişkin 1 adet SCI-E makalenin hazırlanması süreci devam etmekte olup, yakın zamanda alandaki önemli dergilerinden birisine yüklenmesi planlanmaktadır.

- Tez çalışması kapsamında 1 adet patent başvurusu gerçekleştirilmiş olup, inceleme ve araştırma süreci Türk Patent ve Marka Kurumu nezdinde devam etmektedir.

- Tez çalışması kapsamında 1 adet kitap bölümü kabulü alınmış olup, yayın süreci devam etmektedir.

- Tez çalışması kapsamında toplam 4 adet uluslararası konferans bildirisi sunulmuş ve yayımlanmıştır.

Web uygulama güvenliği alanındaki önemli problemlerin çözümünün hedeflendiği bu tez çalışması ile alan açısından önemli katkılarının sunulduğunu düşünmekteyiz. Bununla birlikte web uygulama alanında her geçen gün değişen ve gelişen teknolojik altyapı ve uygulama geliştirme çözümleri ile yeni açıklıklar ve web güvenliği alanında çözülmesi gereken yeni ve önemli problemler ortaya çıkmaktadır. Buna ek olarak, gerçek zamanlı çalışan, modern web teknolojilerini içeren, yoğun saldırı trafiği alan, farklı web uygulamalarından gerçek zamanlı bir web anomali verilerin toplanması ve buna dayalı modellerin oluşturulması, mevcut çalışmaları bir adım daha öteye taşıyacaktır. Ek olarak WAF geliştirme sürecinde web isteklerinin yanı sıra, web sunucu cevapları ve web sunucu log kayıtlarının da web anomali geliştirme sürecine dâhil edilmesi önemlidir. Ayrıca web uygulamalarının güvenliği ağ katmanı veya uygulama katmanı güvenliğinin ayrı ayrı ele alınması ile yeterli derecede çözülemeyecektir. Gelecekte gerçekleştirilecek çalışmalarda, bu iki katmana ek olarak derin paket analizini sürecinide dâhil ederek tüm güvenlik yapısını bir bütün olarak ele alan modüler güvenlik sistemlerinin geliştirilmesi önemlidir.

KAYNAKLAR

1. İnternet: Cisco. (2016). *IP Next-Generation Network Security for Service Providers*. Cisco Public Information White Paper. URL: https://www.cisco.com/c/dam/en/us/solutions/collateral/service-provider/secure-infrastructure/net_implementation_white_paper0900aecd803fcbbe.pdf Son Erişim Tarihi 03.02.2019

2. İnternet: Symantec. (2016) *Internet Security Threat Report*. URL: https://www.symantec.com/content/dam/symantec/docs/reports/istr-22-2017-en.pdf Son Erişim Tarihi 03.02.2019

3. İnternet: EdgeScan. (2019) *Vulnerability Statistics Report*. 4-17. URL: https://www.edgescan.com/wp-content/uploads/2019/02/edgescan-Vulnerability-Stats-Report-2019.pdf. Son Erişim Tarihi 14.05.2021

4. İnternet: OWASP. *Top Ten Application Security Risks*. 2017; URL: https://owasp.org/www-project-top-ten/2017. Son Erişim Tarihi 17.11.2019

5. Halfond, W. G., Viegas, J., & Orso, A. (2006, March). A classification of SQL-injection attacks and countermeasures. In Proceedings of *the IEEE international symposium on secure software engineering* (vol. 1, pp. 13-15). IEEE.

6. Huluka, D., & Popov, O. (2012, June). Root cause analysis of session management and broken authentication vulnerabilities. In *World Congress on Internet Security (WorldCIS-2012)* ,82-86, IEEE.

7. Shu, X., Yao D., and Bertino E. (2015). Privacy-preserving detection of sensitive data exposure. *IEEE transactions on information forensics and security*, 10(5),1092-1103.

8. Barth, A., Jackson, C., Reis, C., and TGC Team. (2008). The security architecture of the chromium browser. *Technical report*. Stanford University.

9. Alalfi, M. H., Cordy, J. R., & Dean, T. R. (May, 2009). A verification framework for access control in dynamic web applications. In *Proceedings of the 2nd Canadian Conference on Computer Science and Software Engineering*. 109-113.

10. Katkar Anjali, S., & Kulkarni Raj, B. (2012). Web vulnerability detection and security mechanism. *International Journal of Soft Computing and Engineering (IJSCE)*, ISSN, 223, 1-2307.

11. Eshete, B., Villafiorita, A., & Weldemariam, K. (2011). *Early detection of security misconfiguration vulnerabilities in web applications.* Paper presented at the 2011 Sixth International Conference on Availability, Reliability and Security.

12. Bates, D., Barth, A., & Jackson, C. (2010). *Regular expressions considered harmful in client-side XSS filters.* Paper presented at the Proceedings of the 19th international conference on World wide web.

13. Fonseca, J., Vieira, M., & Madeira, H. (2007). *Testing and comparing web vulnerability scanning tools for SQL injection and XSS attacks.* 13th Pacific Rim international symposium on dependable computing (PRDC 2007).

14. Lai, C. (2008). Java insecurity: Accounting for subtleties that can compromise code. *IEEE software, 25*(1), 13-19.

15. Balzarotti, D., Monga, M., & Sicari, S. (2006). Assessing the risk of using vulnerable components. In *Quality of Protection* (pp. 65-77): Springer.

16. Neuhaus, S., Zimmermann, T., Holler, C., & Zeller, A. (2007). *Predicting vulnerable software components.* Paper presented at the Proceedings of the 14th ACM Conference on Computer and Communications Security.

17. Grobauer, B., Walloschek, T., & Stocker, E. (2010). Understanding cloud computing vulnerabilities. *IEEE Security & privacy, 9*(2), 50-57.

18. BGA. *Web Application Firewall Tercih Rehberi*. 2021 URL: https://www.bgasecurity.com/makale/web-application-firewall-tercih-rehberi/. Son Erişim Tarihi:12.11.202];

19. Park, S., Kim, M., & Lee, S. (2018). Anomaly detection for HTTP using convolutional autoencoders. *IEEE Access, 6*, 70884-70901.

20. Lupták, P. (2011). *Bypassing web application firewalls.* Paper presented at the Proceedings of 6th International Scientific Conference on Security and Protection of Information.

21. C. Raissi, J. Brissaud, G. Dray, P. Poncelet, M. Roche, and M. Teisseire, (September., 2007). "Web analyzing traffic challenge: description and results," in Proc. *18nd International Conference ECML/PKDD*, Warsaw, Poland, 47-52.

22. M. Exbrayat, (September., 2007). ECML/PKDD challenge: analyzing web traffic a boundaries signature approach. in Proc. *18nd Int. Conf. ECML/PKDD*, Warsaw, Poland, 53-64.

23. Gallagher, B., & Eliassi-Rad, T. (2009). *Classification of http attacks: a study on the ECML/PKDD 2007 discovery challenge*.

24. Kruegel C. and Vigna G. (October 2003). Anomaly detection of web-based attacks. in *Proc. The 10th ACM Conference on Computer and Communications Security*, pp. 251-261.

25. Nguyen H.T. and Franke K., (December, 2012). Adaptive Intrusion Detection System via online machine learning. in *Proc. 12th International Conference on Hybrid Intelligent Systems (HIS)*, Pune, India, pp. 271-277.

26. Nguyen, H. T., Torrano-Gimenez, C., Alvarez, G., Franke, K., & Petrović, S. (2013). Enhancing the effectiveness of web application firewalls by generic feature selection. *Logic Journal of IGPL, 21*(4), 560-570.

27. Odumuyiwa, V., & Chibueze, A. (2020). Automatic Detection of HTTP Injection Attacks using Convolutional Neural Network and Deep Neural Network. *Journal of Cyber Security and Mobility,* 9(4), 489-514.

28. Pachopoulos, K., Valsamou, D., Mavroeidis, D., & Vazirgiannis, M. (September., 2007). *Feature extraction from web traffic data for the application of data mining algorithms in attack identification.* Paper presented at the Proceedings of the ECML/PKDD. pp. 65-70

29. Vartouni, A. M., Teshnehlab, M., & Kashi, S. S. (2019). Leveraging deep neural networks for anomaly-based web application firewall. *IET Information Security*, *13*(4), 352-361.

30. Kagal, L., Finin, T., & Joshi, A. (October, 2003). *A policy based approach to security for the semantic web.* Paper presented at the International semantic web conference. 402-418.

31. A. Tekerek, C. G., and O.F. Bay. (2016). Design and implementation of a web-based intrusion prevention system: a new hybrid model. *J. of the Faculty of Engineering and Architecture of Gazi University, 31*(3), 646-655. doi:10.17341/gummfd.63355

32. Hwang, C., Kim, D., & Lee, T. (2020). Semi-supervised based unknown attack detection in EDR environment. *KSII Transactions on Internet and Information Systems (TIIS), 14*(12), 4909-4926.

33. Liu, C., Yang, J., & Wu, J. (2020). Web intrusion detection system combined with feature analysis and SVM optimization. *EURASIP Journal on Wireless Communications and Networking, 2020*(1), 1-9.

34. Thang, N. M. (2020). Improving efficiency of web application firewall to detect code injection attacks with random forest method and analysis attributes HTTP request. *Programming and Computer Software, 46*(5), 351-361.

35. Zhou, Y., & Wang, P. (2019). An ensemble learning approach for XSS attack detection with domain knowledge and threat intelligence. *Computers & Security, 82*, 261-269.

36. Kar, D., Panigrahi, S., and Sundararajan, S. (2016). SQLiGoT: Detecting SQL injection attacks using graph of tokens and SVM. *Computers & Security, 60*, 206-225.

37. Tama, B. A., Nkenyereye, L., Islam, S. R., & Kwak, K.S. (2020). An enhanced anomaly detection in web traffic using a stack of classifier ensemble. *IEEE Access, 8*, 24120-24134.

38. Luo, Y., Cheng, S., Liu, C., and Jiang, F. (October, 2018). *PU learning in payload-based Web anomaly detection.* Paper presented at the 2018 Third International Conference on Security of Smart Cities, Industrial Control System and Communications (SSIC). 1-5.

39. Khan, F.A., A. Derhab, A. G. and Hussain, A. (2019). A novel two-stage deep learning model for efficient network intrusion detection. *IEEE Access, 7*, 30373-30385.

40. Sevri, M., & Karacan, H. (April, 2019). *Deep learning based web application security.* Paper presented at the 2nd International Conference on Advanced Technologies, Computer Engineering and Science (ICATCES), Antalya, Turkey. 349-354.

41. Ferrag, M. A., Maglaras, L., Moschoyiannis, S., and Janicke, H. (2020). Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study. *Journal of Information Security and Applications, 50*, 102419.

42. Zhang, M., Xu, B., Bai, S., Lu, S., & Lin, Z. (November, 2017). *A deep learning method to detect web attacks using a specially designed CNN.* Paper presented at the International Conference on Neural Information Processing. 828-836.

43. Tekerek, A. (2021). A novel architecture for web-based attack detection using convolutional neural network. *Computers & Security, 100*, 102096.

44. Hao, S., Long, J., & Yang, Y. (April, 2019). *BL-IDS: Detecting Web Attacks Using Bi-LSTM Model Based on Deep Learning.* Paper presented at the International Conference on Security and Privacy in New Computing Environments. 551-563.

45. Liang, J., Zhao, W., & Ye, W. (December, 2017). *Anomaly-based web attack detection: a deep learning approach.* Paper presented at the Proceedings of the 2017 VI International Conference on Network, Communication and Computing. 80-85.

46. Wang, M., Zheng, K., Yang, Y., & Wang, X. (2020). An explainable machine learning framework for intrusion detection systems. *IEEE Access, 8*, 73127-73141.

47. Sarhan, M., Layeghy, S., & Portmann, M. (2021). An Explainable Machine Learning-Based Network Intrusion Detection System for Enabling Generalisability in Securing IoT Networks. *arXiv e-prints*, arXiv: 2104.07183.

48. Le, T.-T.-H., Kim, H., Kang, H., & Kim, H. (2022). Classification and Explanation for Intrusion Detection System Based on Ensemble Trees and SHAP Method. *Sensors, 22*(3), 1154.

49. Mahbooba, B., Timilsina, M., Sahal, R., & Serrano, M. (2021). Explainable Artificial Intelligence (XAI) To Enhance Trust Management In Intrusion Detection Systems Using Decision Tree Model. *Complexity*.

50. Islam, S. R., Eberle, W., Ghafoor, S. K., Siraj, A., & Rogers, M. (2019). Domain knowledge aided explainable artificial intelligence for intrusion detection and response. *arXiv preprint arXiv:1911.09853*.

51. Ahn, S., Kim, J., Park, S. Y., & Cho, S. (2020). Explaining deep learning-based traffic classification using a genetic algorithm. *IEEE Access, 9*, 4738-4751.

52. Marino, D. L., Wickramasinghe, C. S., & Manic, M. (2018). An Adversarial Approach for Explainable AI in Intrusion Detection Systems. *IECON 2018 - 44th Annual Conference of the IEEE Industrial Electronics Society*. 3237-3243.

53. Ahmad, S., Lavin, A., Purdy, S., & Agha, Z. (2017). Unsupervised real-time anomaly detection for streaming data. *Neurocomputing, 262*, 134-147.

54. Torrano-Giménez, C., Perez-Villegas, A., & Alvarez Maranón, G. (2010). An anomaly-based approach for intrusion detection in web traffic. *Journal of Information Assurance and Security, 5*(4), 446-454.

55. Sağıroğlu, Ş., Yolaçan, E., & Yavanoğlu, U. (2011). ZEKİ SALDIRI TESPİT SİSTEMİ TASARIMI ve GERÇEKLEŞTİRİLMESİ. *Gazi Üniversitesi Mühendislik Mimarlık Fakültesi Dergisi, 26*(2), 325-340.

56. McHugh, J. (2000). Testing intrusion detection systems: a critique of the 1998 and 1999 darpa intrusion detection system evaluations as performed by lincoln laboratory. *ACM Transactions on Information and System Security (TISSEC), 3*(4), 262-294.

57. Tavallaee, M., Bagheri, E., Lu, W., & Ghorbani, A. A. (2009). *A detailed analysis of the KDD CUP 99 data set.* Paper presented at the 2009 IEEE symposium on computational intelligence for security and defense applications.

58. Lippmann, R., Haines, J. W., Fried, D. J., Korba, J., & Das, K. (2000). The 1999 DARPA off-line intrusion detection evaluation. *Computer networks, 34*(4), 579-595.

59. Xin, Y., Kong, L., Liu, Z., Chen, Y., Li, Y., Zhu, H., Wang, C. (2018). Machine learning and deep learning methods for cybersecurity. *IEEE Access, 6*, 35365-35381.

60. Fernandes, G., Rodrigues, J. J., Carvalho, L. F., Al-Muhtadi, J. F., & Proença, M. L. (2019). A comprehensive survey on network anomaly detection. *Telecommunication Systems, 70*(3), 447-489.

61. Bengio, Y., Courville, A., & Vincent, P. (2013). Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis snd Machine Intelligence, 35*(8), 1798-1828.

62. Mehta, P., Bukov, M., Wang, C.-H., Day, A. G., Richardson, C., Fisher, C. K., & Schwab, D. J. (2019). A high-bias, low-variance introduction to machine learning for physicists. *Physics reports, 810*, 1-124.

63. Pan, Y., Sun, F., Teng, Z., White, J., Schmidt, D. C., Staples, J., & Krause, L. (2019). Detecting web attacks with end-to-end deep learning. *Journal of Internet Services and Applications, 10*(1), 1-22.

64. Khan, F. A., Gumaei, A., Derhab, A., & Hussain, A. (2019). A novel two-stage deep learning model for efficient network intrusion detection. *IEEE Access, 7*, 30373-30385.

65. Berman, D. S., Buczak, A. L., Chavis, J. S., & Corbett, C. L. (2019). A survey of deep learning methods for cyber security. *Information, 10*(4), 122.

66. Liu, H., Lang, B., Liu, M., & Yan, H. (2019). CNN and RNN based payload classification methods for attack detection. *Knowledge-Based Systems, 163*, 332-341.

67. İnternet: Giménez, C. T., Villegas, A. P., & Marañón, G. Á. (2010). HTTP DATASET CSIC 2010. URL:http://www.isi.csic.es/dataset/, Son Erişim Tarihi: 03.02.2019

68. ECML/PKDD. (September, 2007). *HTTP Dataset Analyzing Web Traffic ECML/PKDD 2007 Discovery Challenge.* Paper presented at the The 11th European Conference On Principles And Practice Of Knowledge Discovery In Databases, Warsaw, Poland.

69. McWhirter, P. R., Kifayat, K., Shi, Q., & Askwith, B. (2018). SQL injection attack classification through the feature extraction of SQL query strings using a gap-weighted string subsequence kernel. *Journal of Information Security and Applications, 40*, 199-216.

70. Vinayakumar, R., Alazab, M., Soman, K., Poornachandran, P., Al-Nemrat, A., & Venkatraman, S. (2019). Deep learning approach for intelligent intrusion detection system. *IEEE Access, 7*, 41525-41550.

71. Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation, 9*(8), 1735-1780.

72. Graves, A., & Schmidhuber, J. (2005). Framewise phoneme classification with bidirectional LSTM networks. *Neural networks*, 18(5-6). 602-610.

73. Schuster, M., & Paliwal, K. (1997). Bidirectional recurrent neural networks. *IEEE Transaction on Signal Processing, 45(11),* 2673-2681.

74. Medeiros, I., Neves, N., & Correia, M. (2015). Detecting and removing web application vulnerabilities with static analysis and data mining. *IEEE Transactions on Reliability, 65*(1), 54-69.

75. Goodfellow, I., Bengio, Y., Courville, A., & Bengio, Y. (2016). Deep learning. vol. 1. MIT press Cambridge.

76. İnternet: Hettich, S. and Bay, S.D. The UCI KDD Archive URL: http://kdd.ics.uci.edu Irvine, CA: University of California, Department of Information and Computer Science, 1999. Erişim Tarihi: 12.04.2021

77. Paula, B., Torgo, L., & Ribeiro, R. (2016). A survey of predictive modeling on imbalanceddomains. *ACM Computing Surveys (CSUR).* 49(2). 1-50.

78. Sun, Y., Wong, A. K., & Kamel, M. S. (2009). Classification of imbalanced data: A review. *International journal of pattern recognition and artificial intelligence, 23*(04), 687-719.

79. Kozik, R., Choraś, M., Renk, R., & Hołubowicz, W. (2015). *A proposal of algorithm for web applications cyber attack detection.* Paper presented at the IFIP International Conference on Computer Information Systems and Industrial Management. 680-687.

80. Mac, H., Truong, D., Nguyen, L., Nguyen, H., Tran, H. A., & Tran, D. (December, 2018). *Detecting attacks on web applications using autoencoder*. in Proc. The 9th International Symposium on Information and Communication Technology. 416-421

81. Ito, M., & Iyatomi, H. (March, 2018). *Web application firewall using character-level convolutional neural network.* in *Proc.* IEEE 14th International Colloquium on Signal Processing & Its Applications (CSPA). 103-106

82. Yu, L., Chen, L., Dong, J., Li, M., Liu, L., Zhao, B., & Zhang, C. (July, 2020). *Detecting malicious web requests using an enhanced textcnn.* ," in *Proc.* IEEE 44th Annual Computers, Software, and Applications Conf. (COMPSAC). 768-777.

83. Karacan, H., & Sevri, M. (2021). A Novel Data Augmentation Technique and Deep Learning Model for Web Application Security. *IEEE Access, 9*, 150781-150797.

84. Al-Alyan, A., & Al-Ahmadi, S. (2020). Robust URL Phishing Detection Based on Deep Learning. *KSII Transactions on Internet and Information Systems (TIIS), 14*(7), 2752-2768.

85. Surowiecki, J. (2005). *The wisdom of crowds*: Anchor.

86. Guzel, K., & Bilgin, G. (October, 2019). *Classification of nuclei in colon cancer images using ensemble of deep learned features.* Paper presented at the 2019 Medical Technologies Congress (TIPTEKNO). 1-4.

87. Dietterich, T. G. (2000). An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization. *Machine learning, 40*(2), 139-157.

88. Wolpert, D. H. (1992). Stacked generalization. *Neural networks, 5*(2), 241-259.

89. Zhang, C., & Ma, Y. (2012). *Ensemble machine learning: methods and applications*. *Springer Science & Business Media*

90. Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural networks, 61*, 85-117.

91. Bengio, Y. (2009). *Learning deep architectures for AI*: Now Publishers Inc.

92. Guo, Y., Liu, Y., Oerlemans, A., Lao, S., Wu, S., & Lew, M. S. (2016). Deep learning for visual understanding: A review. *Neurocomputing, 187*, 27-48.

93. Bi, Y. (2012). The impact of diversity on the accuracy of evidential classifier ensembles. *International Journal of Approximate Reasoning, 53*(4), 584-607.

94. Choi, S.-S., Cha, S.-H., & Tappert, C. C. (2010). A survey of binary similarity and distance measures. *Journal of systemics, cybernetics and informatics, 8*(1), 43-48.

95. Sevri, M., & Karacan, H. (2022). Two Stage Deep Learning Based Stacked Ensemble Model for Web Application Security. *KSII Transactions on Internet and Information Systems (TIIS), 16*(2), 632-657.

96. Serpen, G., & Gao, Z. (2014). Complexity analysis of multilayer perceptron neural network embedded into a wireless sensor network. *Procedia Computer Science, 36*, 192-197.

97. Bienstock, D., Muñoz, G., & Pokutta, S. (2018). Principled deep neural network training through linear programming. *arXiv preprint arXiv:1810.03218*.

98. İnternet: Turek, M. (2018). *Explainable artificial intelligence (XAI)*. Defense Advanced Research Projects Agency. URL: https://www.darpa.mil/program/explainable-artificial-intelligence Son Erişim Tarihi: 20.01.2022

99. Yang, G., Ye, Q., & Xia, J. (2022). Unbox the black-box for the medical explainable ai via multi-modal and multi-centre data fusion: A mini-review, two showcases and beyond. *Information fusion, 77*, 29-52.

100. Yu, Z., Zhao, C., Wang, Z., Qin, Y., Su, Z., Li, X., . . . Zhao, G. (2020). *Searching central difference convolutional networks for face anti-spoofing.* Paper presented at the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 5295-5305.

101. Jagielski, M., Oprea, A., Biggio, B., Liu, C., Nita-Rotaru, C., & Li, B. (2018, May). *Manipulating machine learning: Poisoning attacks and countermeasures for regression learning.* Paper presented at the 2018 IEEE Symposium on Security and Privacy (SP). 19-3.

102. Arrieta, A. B., Díaz-Rodríguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A., Benjamins, R. (2020). Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Information fusion*, 58, 82-115.

103. Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). *" Why should i trust you?" Explaining the predictions of any classifier.* Paper presented at the Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining.

104. İnternet: Molnar, C. (2020). *Interpretable machine learning*. URL: https://christophm.github.io/interpretable-ml-book/ Son Erişim Tarihi: 20.01.2022

105. Alvarez-Melis, D., & Jaakkola, T. S. (2018). On the robustness of interpretability methods. *arXiv preprint arXiv:1806.08049*.

106. Slack, D., Hilgard, S., Jia, E., Singh, S., & Lakkaraju, H. (February, 2020). *Fooling lime and shap: Adversarial attacks on post hoc explanation methods.* Paper presented at the Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society. 180-186.

107. Lundberg, S. M., & Lee, S.I. (2017). A unified approach to interpreting model predictions. *Advances in neural information processing systems, 30*.

108. Janzing, D., Minorics, L., & Blöbaum, P. (2020). *Feature relevance quantification in explainable AI: A causal problem.* Paper presented at the International Conference on artificial intelligence and statistics. 2907-2916.

ÖZGEÇMİŞ

Kişisel Bilgiler

Soyadı, adı	: SEVRİ, Mehmet
Uyruğu	: T.C.
Doğum tarihi ve yeri	: -
Medeni hali	: -
Telefon	: -
Faks	: -
e-mail	: -

Eğitim

Derece	Eğitim Birimi	Mezuniyet Tarihi
Doktora	Gazi Üniversitesi / Bilişim Sistemleri	Devam ediyor
Yüksek lisans	Gazi Üniversitesi / Bilişim Sistemleri	2016
Lisans	Karadeniz Teknik Üniversitesi / Bilgisayar Mühendisliği	2011
Lise	Oğuzeli Çok Programlı Lisesi	2005

İş Deneyimi

Yıl	Yer	Görev
2013-Halen	Gazi Üniversitesi	Araştırma Görevlisi
2013-2013	Recep Tayyip Erdoğan Üniversitesi	Araştırma Görevlisi
2011-2013	Kilis 7 Aralık Üniversitesi	Bilgisayar Mühendisi

Yabancı Dil

İngilizce

Yayınlar

1. Hacer Karacan and **Mehmet Sevri**, "A Novel Data Augmentation Technique and Deep Learning Model for Web Application Security," in ***IEEE Access***, vol. 9, pp. 150781-150797, 2021, doi: 10.1109/ACCESS.2021.3125785. (**Tezden üretilmiştir.**)

2. **Mehmet Sevri**, Hacer Karacan, "Two Stage Deep Learning Based Stacked Ensemble Model for Web Application Security", ***KSII Transactions on Internet and Information Systems,*** vol 16, no. 2, pp. 632-657, 2022, doi: 10.3837/tiis.2022.02.014 (**Tezden üretilmiştir.**)

3. **Mehmet Sevri**, Hacer Karacan, "Açıklanabilir Derin Öğrenme Tabanlı Web Uygulama Güvelik Duvarı Yöntemi ve Sistemi", Patent Başvurusu, ***Türk Patent ve Marka Kurumu,*** Ankara, Türkiye (**Araştırma / İnceleme aşamasında, Tezden üretilmiştir.**)

4. **Mehmet Sevri**, Hacer Karacan, "Explainable Artificial Intelligence (XAI) for Deep Learning Based Intrusion Detection Systems", ***Engineering Cyber-Physical Systems and Critical Infrastructures,*** Springer, Kitap Bölümü (**Kabul alınmıştır, Tezden üretilmiştir.**)

5. **Mehmet SEVRİ,** Hacer KARACAN and M. Ali AKCAYOL, "Crime Analysis Based on Association Rules Using Apriori Algorithm", *International Journal of Information and Electronics Engineering (IJIEE)*, vol. 7, no. 3, pp. 99-102, 2017, doi: 10.18178/ijiee.2017.7.3.669.

6. **Mehmet Sevri**, Hacer Karacan, (May, 2022), "Explainable Artificial Intelligence (XAI) for Deep Learning Based Intrusion Detection Systems", 4th International Conference on Artificial Intelligence and Applied Mathematics in Engineering (ICAIAME 2022), Bakü, Azerbaycan (**Tezden üretilmiştir.**)

7. **Mehmet Sevri**, Hacer Karacan, (October, 2019), "Bidirectional LSTM based Intrusion Detection System for Web Applications", International Conference on All Aspects of Cyber Security (A2CS'19), pp. 53-57, Adana (**Tezden üretilmiştir.**)

8. **Mehmet Sevri,** Hacer Karacan, (April, 2019), "Deep Learning Based Web Application Security", 2nd International Conference on Advanced Technologies, Computer Engineering and Science (ICATCES), pp. 349-354, Alanya (**Tezden üretilmiştir.**)

9. **Mehmet SEVRİ**, M. Ali KOŞAN, Hacer KARACAN, "Applying Deep Learning to Cybersecurity Field and Its Advantages", International Advanced Researches and Engineering Congress (IAREC), Osmaniye, November, 2017. (**Tezden üretilmiştir.**)

10. **Mehmet SEVRİ**, M. Ali KOŞAN, Hacer KARACAN. (2018). "Secure Multiparty Computation and Blockchain-Smart Contract Based Election System", International Conference on Applied Mathematics in Engineering (ICAME), Balıkesir, June, 2018.

11. M. Ali KOŞAN, **Mehmet SEVRİ**, Hacer KARACAN. (2018). "Comparison of Classification Algorithms used in Intrusion Detection Systems", International Conference on Applied Mathematics in Engineering (ICAME), Balıkesir, June, 2018.

12. **Mehmet SEVRİ**, Hacer KARACAN, M. Ali AKÇAYOL, "Crime Analysis Based on Association Rules Using Apriori Algorithm", *4th IEEE International Conference on Electrical and Electronics Engineering (ICEEE 2017),* Ankara, April, 2017.

13. **Mehmet SEVRİ**, Hacer KARACAN, "Using Data Mining Algorithms For Identification Of Gifted/Talented Students And Performance Comparisons of Classification Models", *International Conference on the Talented and Gifted (INTAG 2017),* Ankara, May, 2017.

14. **Mehmet SEVRİ**, Nurettin TOPALOĞLU, "An Infrastructure Model to Detect and Prevent Web Attacks", *Global Conference on Applied Computing in Science and Engineering,* Rome, Italy, July, 2016.

15. **Mehmet SEVRİ**, Nurettin TOPALOĞLU, "Classification of Web Attacks Wıth Machine Learning Algorithms: An Application Example", *International Conference on Research in Education and Science (ICRES),* Bodrum, pp. 489-494, May, 2016

16. **Mehmet SEVRİ**, Nurettin TOPALOĞLU, "Hacking Last Trend: Ransomware Malware And Possible Future", *International Conference on Research in Education and Science (ICRES)*, Bodrum, pp. 1347-1353, May, 2016

17. **Mehmet SEVRİ**, Nurettin TOPALOĞLU, "Cyber Security Education In Turkey", *International Conference on Education in Mathematics, Science & Technology (ICEMST)*, Bodrum, pp. 923-928, May, 2016

18. İlker KOCATEPE, **Mehmet SEVRİ**, Hacer KARACAN, "Utilizing Linear Feedback Shift Registers for Selective-Random Memory Addressing", *International Advanced Research and Engineering Congress (IAREC)*, Osmaniye, November, 2017.

19. M. Ali KOŞAN, **Mehmet SEVRİ**, Hacer KARACAN, "Spy Attacks and Protection Methods in Android Smart Phones", *International Advanced Research and Engineering Congress (IAREC)*, Osmaniye, November, 2017

20. **Mehmet SEVRİ**, Nurettin TOPALOĞLU, "Unconscious DNS Changing and Risks", *XVII. Akademik Bilişim Konferansı*, Eskişehir, Feb, 2015.

21. Merve ORAKÇI, Bünyamin CİYLAN, İbrahim KÖK, **Mehmet SEVRİ**, "Using Data Mining Techniques and Machine Learning Algorithms in Crime Analysis", *XVIII. Akademik Bilişim Konferans*ı, Aydın, 2016.



GAZİLİ OLMAK AYRICALIKTIR.