Erciyes Üniversitesi
Fen Bilimleri Enstitüsü Dergisi
Cilt 40, Sayı 3, 2024

Erciyes University
Journal of Institue Of Science and Technology
Volume 40, Issue 3, 2024

Güvenlik Duvarlarındaki Ağ Trafiği Log Kayıtlarının Analizinde Açıklanabilir Yapay Zekâ ve Derin Sinir Ağlarının Kullanımı: Karşılaştırmalı Bir Analiz

Anıl Utku[*1]

[*1] Munzur Üniversitesi Mühendislik Fakültesi Bilgisayar Mühendisliği, TUNCELİ

(Alınış / Received: 10.10.2024, Kabul / Accepted: 19.11.2024, Online Yayınlanma / Published Online: 30.12.2024)

Anahtar Kelimeler
Açıklanabilir Yapay Zekâ,
Derin Sinir Ağları,
Makine Öğrenmesi,
Derin Öğrenme,
Ağ Trafiği Analizi

Öz: İnternet kullanımında yaşanan hızlı büyüme, siber tehditlerin çeşitlenmesine ve karmaşıklığının artmasına neden olmuştur. Bu durum, özellikle ağ güvenliği konusunda daha gelişmiş ve dinamik çözümler geliştirilmesini zorunlu kılmaktadır. Geleneksel güvenlik yöntemleri, özellikle güvenlik duvarları, belirli kurallar çerçevesinde çalıştığından, yeni nesil siber tehditlere karşı yetersiz kalabilmektedir. Güvenlik duvarları, ağ trafiğini izleyerek potansiyel tehditleri tespit eden ve engelleyen önemli araçlar olarak uzun yıllardır kullanılmaktadır. Ancak, statik kurallara dayanan geleneksel güvenlik duvarları, karmaşık ve dinamik tehditleri algılamada yetersiz kalmakta ve yüksek oranda yanlış pozitif/negatif sonuçlar üretmektedir. Bu durum, daha esnek ve kendini sürekli olarak geliştirebilen yapay zekâ tabanlı güvenlik çözümlerine olan ihtiyacı artırmıştır. Bu çalışmada, açıklanabilir yapay zekâ ve DNN tabanlı bir derin öğrenme modeli kullanarak güvenlik duvarlarındaki ağ trafiği loglarının analiz edilmesi hedeflenmiştir. Geliştirilen DNN tabanlı model, RF, kNN, SVM, LR ve XGBoost gibi popüler makine öğrenmesi algoritmalarıyla karşılaştırılarak performansı değerlendirilmiştir. Deneysel sonuçlar, geliştirilen DNN tabanlı modelin %99,87 doğruluk oranıyla karşılaştırılan modellerden ve literatürdeki çalışmalardan daha başarılı olduğunu göstermiştir.

Use of Explainable Artificial Intelligence and Deep Neural Networks in Analyzing Network Traffic Logs in Firewalls: A Comparative Analysis

Keywords
Explainable Artificial Intelligence,
Deep Neural Networks,
Machine Learning,
Network Traffic Analysis

Abstract: The rapid growth in internet usage has led to the diversification and complexity of cyber threats. This situation necessitates the development of more advanced and dynamic solutions, especially in network security. Since traditional security methods, especially firewalls, operate within the framework of certain rules, they may be insufficient against new generation cyber threats. Firewalls have been used for many years as important tools that detect and block potential threats by monitoring network traffic. However, traditional firewalls based on static rules are insufficient in detecting complex and dynamic threats and produce high rates of false positive/negative results. This situation has increased the need for more flexible and continuously improving artificial intelligence-based security solutions. In this study, it is aimed to analyze network traffic logs on firewalls using explainable artificial intelligence and a DNN-based deep learning model. The developed DNN-based model was compared with popular machine learning algorithms such as RF, kNN, SVM, LR and XGBoost and their performances were evaluated. Experimental results showed that the developed DNN-based model was more successful than the compared models and studies in the literature with 99.87% accuracy.

1. Giriş

Günümüzde artan bilgi miktarı ve bilgi teknolojileri kullanımının yaygınlaşmasıyla birlikte siber suçlarda artış görülmektedir. Bu durum, özellikle hassas verilerin korunması için daha güvenli ağ altyapılarına olan ihtiyacı artırmaktadır [1]. Güvenlik duvarları (firewall), iç ve dış ağlar arasında ağ geçidi görevi görerek kötü amaçlı yazılımların yayılmasını önlemeye, saldırı girişimlerini engellemeye ve diğer çevrimiçi tehditlere karşı korumaya yardımcı olur [3]. Güvenlik duvarları ağ trafiği verilerinin içeriğini, verilerin hangi güvenlik duvarı bağlantı noktalarını kullanmaya çalıştığı ve verilerin nereden kaynaklandığı açılarından analiz eder [3]. Farklı güvenlik duvarı türleri, potansiyel olarak kötü amaçlı kaynakları değerlendirmek için farklı yöntemler kullanır [4].

Güvenlik duvarları, bir ağın iç ve dış trafiği arasındaki veri paketlerini izler ve belirli bir dizi kural doğrultusunda bu trafiği engeller veya geçişine izin verir. Güvenlik duvarları, ağa izinsiz girişleri engelleyerek veri hırsızlığı veya yetkisiz işlemler gibi tehditlerin önüne geçer [5]. Gelen ve giden ağ trafiğini izleyerek anormal davranışları tespit eder [6]. Geleneksel güvenlik duvarları, statik kurallar seti kullanarak ağ trafiğini inceler. Ancak bu sistemler karmaşık saldırıları tespit etmede verimsiz kalmaları, dinamik tehditlere karşı esnek olmamaları ve yüksek yanlış pozitif/negatif oranı gibi sınırlılıklara sahiplerdir [2].

Güvenlik duvarları, ağ tabanlı saldırılara karşı ilk savunma hattı olarak kullanılan önemli bir güvenlik aracıdır. Gelen ve giden ağ trafiğini belirli kurallara göre izleyen ve kontrol eden güvenlik duvarları, ağ güvenliğini sağlamak için gereklidir [7]. Ancak, siber tehditlerin giderek daha karmaşık hale gelmesi ve geleneksel saldırı tespit yöntemlerinin sınırlamaları, güvenlik duvarlarının da evrim geçirmesini zorunlu kılmıştır [8]. Ağ güvenliğini sağlamak için kullanılan en önemli araçlardan biri olan güvenlik duvarları, artan tehditlerle başa çıkmak için daha gelişmiş ve dinamik çözümler gerektirmektedir. Geleneksel güvenlik duvarları, sabit kurallar üzerinden çalışarak belirli tehditleri tespit etse de, modern siber saldırıların karmaşıklığı karşısında yetersiz kalabilmektedir [9]. Bu nedenle, yapay zekâ tabanlı güvenlik duvarları siber tehditlere karşı daha etkili bir savunma mekanizması olarak öne çıkmaktadır.

Yapay zekâ tabanlı güvenlik duvarları, ağ trafiğini daha akıllı ve dinamik bir şekilde izleyerek tehditlere daha etkili bir şekilde yanıt verebilir. Yapay zeka tabanlı güvenlik duvarları dinamik öğrenme, anomali tespiti, büyük veri analizi ve sürekli öğrenme gibi avantajlara sahiptir [10]. Yapay zekâ tabanlı güvenlik duvarları ağ trafiği verilerini sürekli olarak analiz ederek yeni tehdit modellerini otomatik olarak öğrenebilir ve geleneksel güvenlik duvarlarında bir sınırlılık olarak nitelendirilebilecek dinamik tehdit algılama yeteneğine sahiptir. Ağ üzerindeki normal ve anormal davranışları öğrenebilir ve bu davranışların dışına çıkan hareketleri tespit ederek potansiyel saldırıları engelleyebilir. Büyük miktarda veri üzerinde eğitim yaparak ağ trafiğindeki tüm paketleri analiz edebilir ve gerçek zamanlı tehdit tespiti sağlayabilir.

Yapay zekâ yöntemlerinin sunduğu bu gibi avantajlar sebebiyle bu çalışmada Deep Neural Network (DNN) tabanlı bir model ile güvenlik duvarındaki ağ trafiği loğlarının analiz edilmesi amaçlanmıştır. Geliştirilen model, güvenlik duvarları üzerinden geçen veri trafiğine dair kayıtları ve özellikleri içeren bir veri seti kullanılarak Rastgele Orman (Random Forest-RF), k-En Yakın Komşu (k Nearest Neighbour-kNN), Destek Vektör Makinesi (Support Vector Machine-SVM), Lojistik Regresyon (Logistic regression-LR) ve Ekstrem Gradyan Arttırma (eXtreme Gradient Boosting-XGBoost) gibi popüler makine öğrenmesi algoritmalarıyla uygulamalı olarak karşılaştırılmıştır.

Bu çalışmanın literatüre olan katkıları aşağıdaki gibi özetlenebilir:

- Geliştirilen DNN tabanlı bir model ile ağ üzerindeki trafik loglarının sınıflandırılması amaçlanmıştır.
- Geliştirilen model literatürdeki popüler makine öğrenmesi algoritmalarıyla kapsamlı olarak karşılaştırılmıştır.
- Açıklanabilir Yapay Zekâ (Explainable Artificial Intelligence -XAI) yöntemleri kullanılarak yapılan sınıflandırma işlemlerinin şeffaflığı sunulmuştur.
- Bu çalışma ile XAI konusunda Türkçe literatüre katkıda bulunmak hedeflenmiştir.
- Geliştirilen DNN tabanlı derin öğrenme modeli, %99,87 sınıflandırma doğruluğuyla karşılaştırılan modellerden ve literatürdeki çalışmalardan daha başarılı deneysel sonuçlara sahip olmuştur.

2. Literatürdeki Çalışmalar

Bu bölümde, literatürdeki aynı veri setini kullanan çalışmaların kullandıkları modeller ve elde edilen başarı oranları incelenmiştir.

Ertam ve Kaya, bu çalışmada kullanılan veri setini kullanarak SVM'in bir uygulamasını sunmuştur [11]. Çalışmada linear, polynomial, RBF ve sigmoid SVM çekirdekleri kullanılarak karşılaştırmalı bir analiz sunulmuştur. Deneyler,

Sigmoid çekirdek kullanan SVM'in %98,5 duyarlılık, %60,3 kesinlik ve %74,8 F-skoru değerine sahip olduğunu göstermiştir.

Al-Haijaa ve Ishtaiwi, bu çalışmada kullanılan veri setini kullanarak Sığ Sinir Ağı (Shallow Neural Network-SNN) Optimize Edilebilir Karar Ağacı (Optimizable Decision Tree-ODT) yöntemlerini kullanarak bir sınıflandırma sistemi geliştirmiştir [12]. Deneysel sonuçlar, SNN'in %98,5, ODT'nin ise %99,8 doğruluk değerine ulaştığını göstermiştir.

Al-Behadili, bu çalışmada kullanılan veri setini kullanarak Karar Ağacı (Decision Tree-DT), SVM, One Rule (OneR), Yapay Sinir Ağı (Artificial Neural Network-ANN), Çok Sınıflı Sınıflandırıcı, Particle Swarm Optimization (PSO) ve Zero Rule (ZeroR) modellerinin karşılaştırmalı bir analizini sunmuştur [13]. Deneysel sonuçlar, DT'nin %99,83 sınıflandırma doğruluğuyla karşılaştırılan modellerden daha başarılı olduğunu göstermiştir.

Naryanto ve Delimayanti, bu çalışmada kullanılan veri setini kullanarak DT ve kNN modellerinin karşılaştırmalı bir analizini sunmuştur [14]. Rapidminer kullanılarak yapılan deneysel çalışmalar, DT'nin %94,84 sınıflandırma doğruluğuyla kNN'den daha başarılı olduğunu göstermiştir.

Aljabri ve ark., bu çalışmada kullanılan veri setini kullanarak NB, kNN, J48, RF ve ANN'in uygulamalı bir analizini sunmuştur [15]. Deneysel çalışmalar, RF'ın %99,64 doğrulukla karşılaştırılan modellerden daha başarılı olduğunu göstermiştir.

Rahman ve ark., bu çalışmada kullanılan veri setini kullanarak RF modelinin bir uygulamasını sunmuştur [16]. Deneyler, RF'ın %99 doğruluk değerine ulaştığını göstermiştir.

Al-Tarawneh ve Bani-Salameh, bu çalışmada kullanılan veri setini kullanarak kNN, RF ve DNN modellerinin karşılaştırmalı bir analizini sunmuştur [17]. Deneyler, kNN'in %99,38 sınıflandırma doğruluğuyla karşılaştırılan modellerden daha başarılı olduğunu göstermiştir.

Efeoğlu ve Tuna, bu çalışmada kullanılan veri setini kullanarak Decision Stump, Simple Cart ve NB Tree algoritmalarının karşılaştırmalı bir analizini sunmuştur [18]. Deneysel sonuçlar, Simple Cart ve NB Tree'nin %99,84, Decision Stump'ın ise %79,68 sınıflandırma doğruluğuna sahip olduğunu göstermiştir.

Lee ve ark., bu çalışmada kullanılan veri setini kullanarak NB, kNN, OneR, J48, SVM, LR ve ANN'in uygulamalı bir analizini sunmuştur []. Deneysel sonuçlar, kNN ve ANN'in %97 doğrulukla karşılaştırılan modellerden daha başarılı olduğunu göstermiştir.

Tablo 1'de literatürdeki çalışmaların kullandıkları modeller ve deneysel sonuçları sunulmuştur.

Tablo 1. Literatürdeki çalışmaların deneysel sonuçları ve kullandıkları modeller

Referans	Kullanılan model(ler)	Başarı oranı
11	linear, polynomial, RBF ve sigmoid SVM	%98,5 duyarlılık, %60,3 kesinlik ve %74,8 F-skoru (Sigmoid SVM)
12	SSN ve ODT	%98,5 doğruluk (SNN), %99,8 doğruluk (ODT)
13	DT, SVM, OneR, ANN, Çok Sınıflı Sınıflandırıcı, PSO ve ZeroR	%99,83 (DT)
14	DT ve kNN	%94,84 doğruluk (DT)
15	NB, kNN, J48, RF ve ANN	%99,64 doğruluk (RF)
16	RF	%99 doğruluk (RF)
17	kNN, RF ve DNN	%99,38 doğruluk (kNN)
18	Decision Stump, Simple Cart ve NB Tree	%99,84 doğruluk (Simple Cart ve NB Tree)
19	NB, kNN, OneR, J48, SVM, LR ve ANN	%97 doğruluk (kNN ve ANN)
Mevcut çalışma	RF, kNN, SVM, LR, XGBoost ve DNN	%99,87 doğruluk, %99,86 kesinlik, %99,89 duyarlılık ve %99,87 F-skoru

Tablo 1'de görüldüğü gibi geliştirilen DNN tabanlı derin öğrenme modeli, %99,87 doğruluk, %99,86 kesinlik, %99,89 duyarlılık ve %99,87 F-skoru değerleriyle karşılaştırılan literatürdeki çalışmalardan daha başarılı olmuştur. DNN'in başarılı olmasının sebepleri arasında geliştirilen modelin mimari yapısı ve yapılan hiper-parametre optimizasyonu gösterilebilir.

3. Açıklanabilir Yapay Zekâ

XAI, karmaşık yapay zekâ tabanlı modellerin çıktılarını daha şeffaf ve anlaşılır hale getirerek modelin çıktılarına güven duyulmasını sağlayan yöntemler bütünüdür [20]. XAI model güvenliğini sağlamada önemli bir rol oynar. Özellikle karmaşık yapay zekâ sistemlerinin iç süreçlerini anlamak ve güvenilir kararlar almak için önemli bir araçtır. XAI'nin önemi, özellikle yüksek doğruluk oranlarına sahip olsa bile kara kutu olarak bilinen modellerin nasıl çalıştığını anlamanın zor olduğu senaryolarda artmaktadır [21]. Şeffaflık ve güven, hesap verebilirlik, etik ve yasal yükümlülükler, önyargı, sürekli iyileştirme, sağlamlık, gizlilik ve hata ayıklama XAI'nin bileşenleri olarak değerlendirilebilir [22].

Yapay zekâ modelleri karar verme sürecinde genellikle insanlar açısından anlaşılması karmaşık algoritmalar ve büyük veri kullanmaktadır. Şeffaflık ve güven, XAI ile model tarafından kararların nasıl alındığının sunulması ile şeffaflığının bu sayede son kullanıcıların güveninin arttırılmasını ifade etmektedir [23]. Hesap verebilirlik, sistemlerinin düzenleyici gereklilikleri karşılamak için olmazsa olmaz olan kararları için açık ve anlaşılır nedenler sunmasını sağlar. Örneğin, finans sektöründe düzenlemeler genellikle kredi onayları veya kredi puanlama gibi kararların şeffaf olmasını gerektirir [24]. XAI, belirli bir kararın neden alındığına dair ayrıntılı iç görüler sunarak sürecin şeffaf olmasını ve düzenleyiciler tarafından denetlenebilmesini sağlar. Etik ve yasal yükümlülükler, özellikle sağlık, finans ve hukuk gibi kritik alanlarda ön plana çıkmaktadır. Bu alanlarda XAI, sistemlerin neden belirli bir karara vardığını anlamayı sağlar ve etik sorumluluklar için destek sunar [25]. Önyargı, yapay zekâ tabanlı sistemlerin ırk, cinsiyet veya diğer korunan özelliklere dayalı olarak bireylere haksız muameleye yol açabilecek önyargılardan arınmış olmasını ifade etmektedir [26]. XAI, karar alma sürecini şeffaf hale getirerek önyargıları belirlemeye ve azaltmaya yardımcı olur. Kuruluşlar, bu sayede ayrımcılık karşıtı yasalara ve düzenlemelere uyduklarını gösterebilirler. Sürekli iyileştirme, geliştiricilerin ve veri bilimcilerin, yapay zekâ modellerinin neden hatalı çıktılar verdiğini XAI yöntemleriyle anlayarak modeli geliştirmelerini ifade etmektedir [27]. Hangi özelliklerin ve veri noktalarının modele nasıl etki ettiğini bilmek, daha doğru ve etkin modeller geliştirmeye yardımcı olur. Sağlamlık, model hiper-parametrelerindeki değişimlere ve beklenmeyen senaryolara karşı tutarlı olmayı ifade etmektedir [28]. Gizlilik, hassas kullanıcı bilgilerinin korunmasını garanti altına almayı ifade etmektedir. Hata ayıklama ise XAI'nin, bir sorunun belirli bölümlerini ve sistemin mantığındaki veya eğitim verilerindeki hataları belirlemeye yardımcı olmasını ifade etmektedir [29].

Yapay zekâ modellerin verdiği kararlar neticesinde kara kutu paradoksu ön plana çıkmaktadır. Yapay zekâ sistemlerinin sonuçlara nasıl ulaştığı konusunda daha fazla netliğe ihtiyaç duyulması sebebiyle, bu süreçler için yorumlayıcı yöntemler geliştirilmiştir [30]. Bu yöntemler, yapay zekânın belirsiz işleyişi ile insanın kavrama ve güven ihtiyacı arasında ilişki kurmayı amaçlamaktadır. Özellik önem analizi, her bir girdi değişkeninin modelin tahminleri üzerindeki etkisini inceleyen bir yöntemdir. Hangi özelliklerin algoritmanın kararlarını en çok etkilediğini vurgulayarak, insanların akıl yürütme örüntüleri hakkında net bir çerçeve oluşturmayı amaçlamaktadır [31].

SHapley Additive exPlanations (SHAP) gibi teknikler, yapay zekânın karmaşıklığını daha anlaşılabilir bir yapıya dönüştürmektedir. Modelin tahminlerini bireysel düzeyde parçalara ayırarak belirli durumlarda kullanılan mantığın anlık görüntüsünü sunmaktadır. SHAP, her bir özelliğin tahmine katkısını hesaplayarak bir örneğin tahminini açıklamayı amaçlamaktadır [32]. SHAP, modelden bağımsız bir yöntemdir ve yalnızca bireysel örnekler için açıklamalar sağladığı anlamına gelen yerel bir yöntemdir. Bir veri örneğinin özellik değerleri, bir koalisyondaki oyuncular gibi davranır [33]. Shapley değeri, tüm olası koalisyonlar arasında bir özellik değerinin ortalama marjinal katkısıdır. SHAP, bir modelin girdilerinin her birinin çıktıya katkısını hesaplar. Bu katkı, her bir girdinin karar üzerindeki etkisini gösterir ve modellerin daha anlaşılır hale gelmesini sağlar [34].

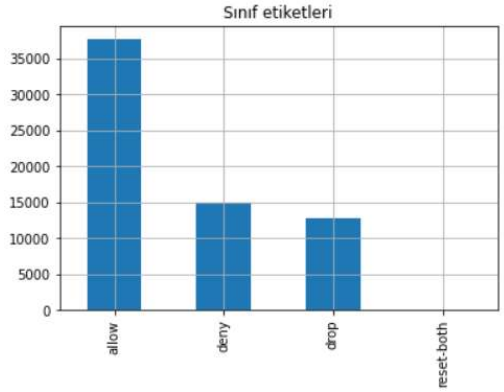
4. Materyal ve Metot

Bu çalışmada, ağ trafiği analizi ve güvenlik duvarı etkinliğinin değerlendirilmesi amacıyla, XAI ve DNN yöntemleri kullanılmıştır. İlk olarak, Kaggle üzerinden genel erişime açık olarak sunulan bir ağ trafiği veri seti analiz edilmiştir. Kullanılan veri seti, farklı protokollere ve trafiğin sınıflandırılmasına yönelik çok sayıda özelliği içermektedir. DNN, ağ trafiğindeki zararlı aktivitelerin tespitinde kullanılmış, model performansını artırmak için hiper-parametre optimizasyonu gerçekleştirilmiştir. Ayrıca, XAI teknikleriyle modellerin karar süreçleri analiz edilerek, güvenlik duvarlarının ağ trafiği üzerindeki etkinliğini şeffaf bir şekilde ortaya koymak amaçlanmıştır. Bu bölümde kullanılan veri seti ve karşılaştırılan modellere ilişkin detaylar sunulmaktadır.

4.1. Veriseti

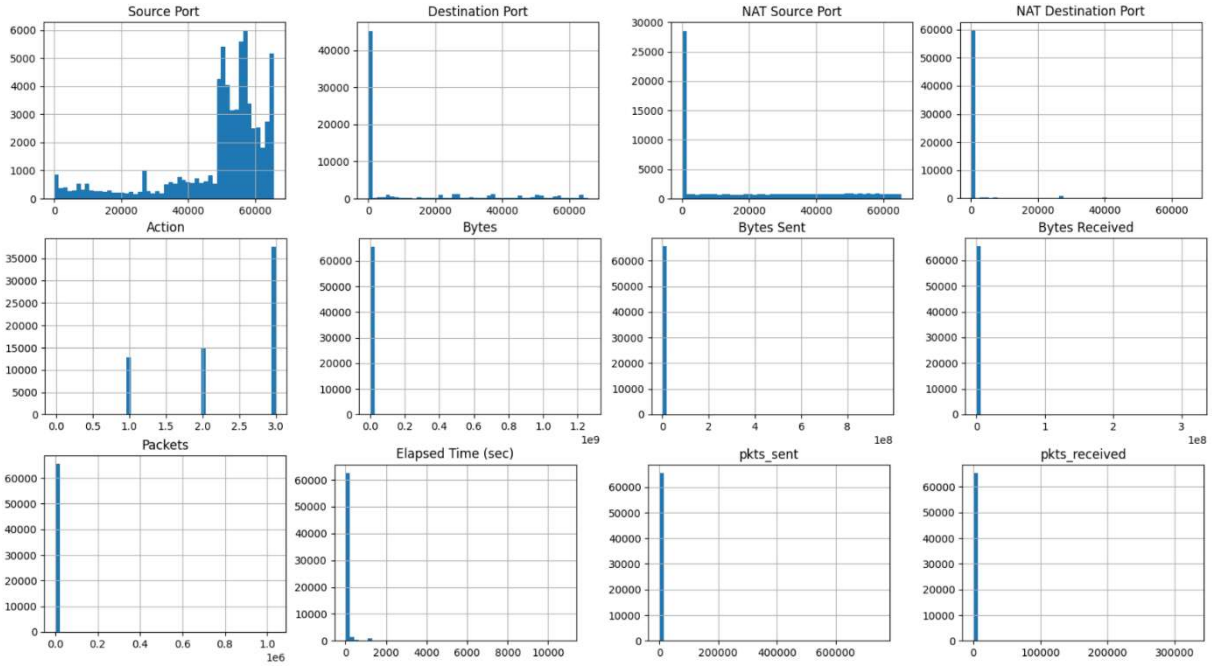
Bu çalışmada, ağ üzerindeki trafiği izleyen bir güvenlik duvarının (firewall) log verileri kullanılmıştır [35]. Güvenlik duvarı, genellikle bir ağın giriş ve çıkışında bulunan, zararlı veya istenmeyen trafiklerin engellenmesini sağlayan bir güvenlik sistemi olduğundan, bu veriler siber güvenlik ve ağ güvenliği alanında kullanılabilecek potansiyel verilerdir. Veri seti, güvenlik duvarları üzerinden geçen veri trafiğine dair kayıtları ve özellikleri içermektedir. Özellikle siber saldırı girişimleri veya şüpheli faaliyetler gibi anormal davranışları tespit etmek amacıyla önemlidir.

Kullanılan veri seti toplamda 12 öznitelik ve izin ver (allow), engelle (deny), reddet (drop) ve sıfırla (reset-both) olmak üzere 4 sınıf olmak üzere 65532 satır veriden oluşmaktadır. İzin ver etiketi, güvenlik duvarının gelen veya giden trafiğe izin verdiğini, veri paketinin güvenlik duvarından geçip hedefe ulaştığını ifade etmektedir. Engelle etiketi, güvenlik duvarının kaynağa ya da hedefe herhangi bir bildirimde bulunmadan veri paketlerini engellemesini, paketleri yok saymasını ifade eder. Reddet etiketi, güvenlik duvarının belirli bir veri paketinin hedefe ulaşmasını aktif olarak engellemesini ifade eder. Reddet işlemi, engelle işlemine benzerdir ancak kaynağa bağlantının reddedildiğine yönelik bir bildirim gönderilmektedir ve yetkisiz veya zararlı trafiği durdurmak için kullanılır. Sıfırla etiketi ise güvenlik duvarının bağlantının sıfırlandığına ya da sonlandırıldığına yönelik kaynağa ve hedefe bir bildirim gönderilmesini ifade etmektedir. Şekil 1'de veri setinde bulunan sınıflar ve dağılımları görülmektedir.



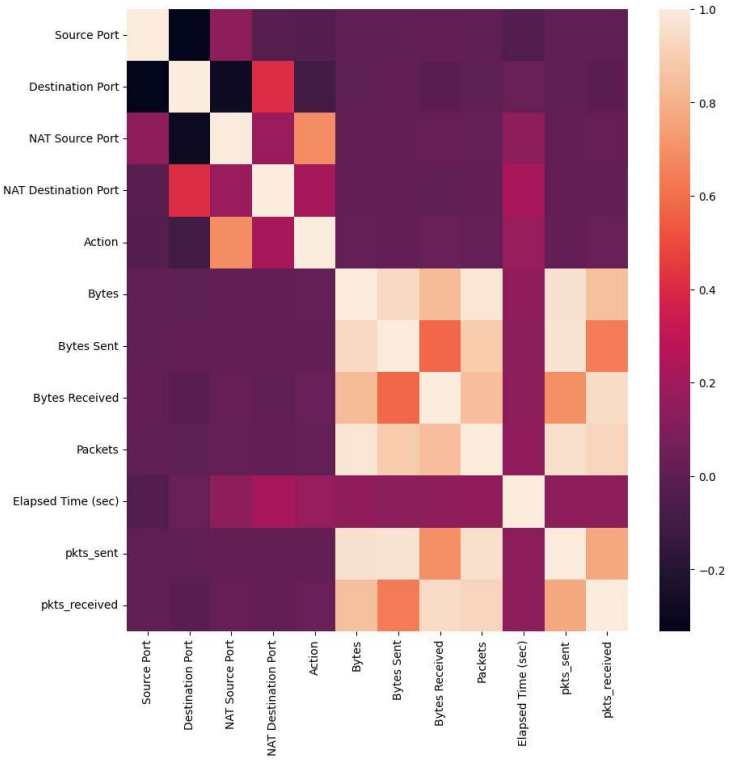
Şekil 1. Veri setinde bulunan sınıfların dağılımı

Şekil 1'de görüldüğü gibi veri setinde izin ver (allow) sınıfına ait 37640 örnek, engelle (deny) sınıfına ait 14987 örnek, reddet (drop) sınıfına ait 12851 örnek ve sıfırla (reset-both) sınıfına ait 54 örnek bulunmaktadır. Veri setinde bulunan öznitelikler hedef portu (destination port), kaynak portu (source port), NAT hedef portu (NAT destination port), NAT kaynak portu (NAT source port), aksiyon (action), gönderilen baytlar (bytes sent), baytlar (bytes), alınan baytlar (bytes received), geçen süre (elapsed time), paketler (packets), alınan paket (pkts_received) ve gönderilen pakettir (pkts_sent). Şekil 2'de veri setinde bulunan özniteliklerin histogram grafikleri görülmektedir.



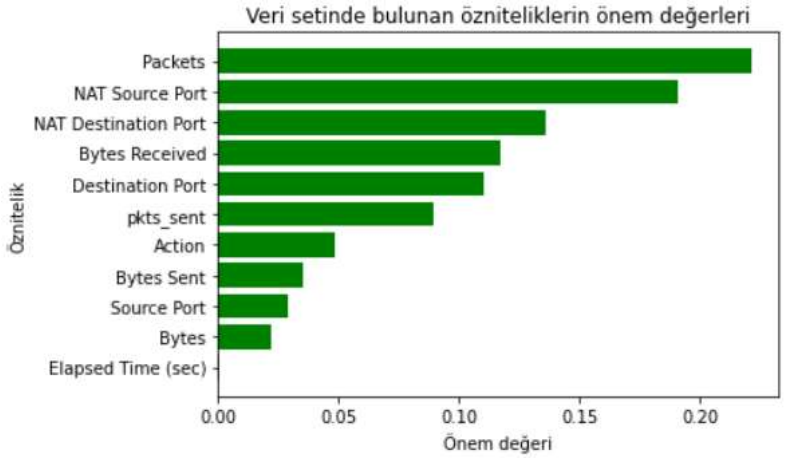
Şekil 2. Veri setinde bulunan özniteliklerin histogram grafikleri

Kaynak portu, veriyi gönderen kaynak cihazın kullandığı port numarasını ifade eder. Hedef portu, verinin gönderilmiş olduğu hedef port numarasını göstermektedir. NAT kaynak portu, Network Address Translation (NAT) sırasında kaynak portun değiştirilmiş halini yani verinin dış ağa çıktığında aldığı port numarasını ifade eder. NAT hedef portu, NAT sonrası hedef portun değiştirilmiş halini yani dış ağdaki hedef portun yerel ağa döndüğünde aldığı numarayı ifade eder. Aksiyon, güvenlik duvarının veri paketine karşı aldığı izin verildi veya engellendi şeklindeki aksiyonu ifade eder. Baytlar, bağlantı sırasında gönderilen ve alınan toplam veri miktarını byte cinsinden ifade eder. Gönderilen baytlar, kaynak cihazdan hedef cihaza gönderilen veri miktarını byte cinsinden ifade eder. Alınan baytlar, hedef cihazdan kaynak cihaza alınan veri miktarını byte cinsinden ifade eder. Paketler, toplam paket sayısını göstermektedir. Geçen süre, bağlantının başlangıcından bitişine kadar geçen süreyi saniye cinsinden ifade eder. Gönderilen paket, kaynak cihaz tarafından gönderilen toplam paket sayısını, alınan paket ise hedef cihaz tarafından alınan toplam paket sayısını ifade eder. Şekil 3'te veri setinde bulunan öznitelikler arasındaki ilişkiler görülmektedir.



Şekil 3. Veri setinde bulunan öznitelikler arasındaki ilişkiler

Şekil 3'te görüldüğü gibi Bytes ile Bytes Sent, Bytes Received, Packets, ve Elapsed Time öznitelikleri arasında güçlü pozitif korelasyonlar olduğu, daha fazla veri gönderilip alındığında, daha fazla paket ve zaman harcandığı görülmektedir. Packets ile pkts_sent ve pkts_received arasında olan güçlü pozitif ilişki, paket sayısının artmasıyla gönderilen ve alınan paketlerin de arttığını göstermektedir. Veri setinde bulunan özelliklerin önem değerleri Şekil 4'te görülmektedir.



Şekil 4. Veri setinde bulunan özniteliklerin önem değerleri

Şekil 4'te her bir özniteliğin modelin tahmin performansına olan katkısı görülmektedir. Packets özniteliği, en yüksek önem değerine sahip özniteliktir ve modelin karar verme sürecinde en etkili özniteliktir. NAT destination port ve NAT source port, NAT üzerinden yönlendirilen bağlantılarda hedef ve kaynak portlar modelin doğru kararlar vermesi için önemlidir. Gönderilen ve alınan veri miktarlarını ifade eden bytes sent ve bytes received öznitelikleri, paket sayıları kadar yüksek öneme sahip değildir. Destination port, source port ve action öznitelikleri, göreceli olarak daha az öneme sahiptir. Elapsed time özniteliği ise en az öneme sahip özniteliklerden biri olarak görülmektedir.

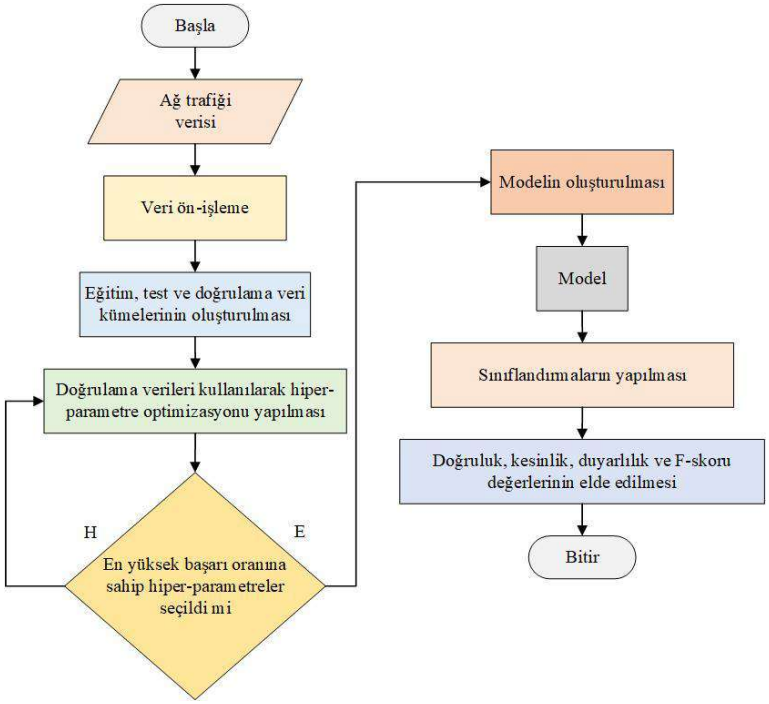
Veri ön-işleme aşamasında kategorik-nümerik veri dönüşümü yapılmıştır. Allow sınıf etiketi 3, deny sınıf etiketi 2, drop sınıf etiketi 1 ve reset-both sınıf etiketi ise 0'a dönüştürülmüştür. Yapılan deneysel çalışmalar neticesinde elde edilen başarı oranları karşılaştırılarak verisetinin %80'i eğitim, %20'si ise test için ayrılmıştır. Model hiper-parametrelerinin belirlenmesi amacıyla eğitim verisetinin %10'u doğrulama için kullanılmıştır. Uygulanan her bir modelin en başarılı sonuçlara sahip olabilmesi amacıyla ızgara arama (grid search) kullanılarak hiper-parametre analizleri yapılmıştır. Izgara arama, makine öğrenmesi ve derin öğrenme modellerinde en iyi hiperparametreleri belirlemek için kullanılan bir optimizasyon yöntemidir. Grid Search, belirli hiperparametrelerin her bir olası kombinasyonu için modelin performansını değerlendirmektedir. İlk adımda, hangi hiperparametrelerin ayarlanacağını ve bu parametreler için hangi değer aralıklarının denenmesi gerektiğini belirlenmektedir. Parametrelerin her olası kombinasyonunu bir ızgara şeklinde oluşturur. Her bir kombinasyon için model eğitilir ve belirlenen bir değerlendirme metriğine göre modelin performansını ölçer. Modeller uygulanırken 5-katlı çapraz doğrulama tekniği kullanılmıştır. Çapraz doğrulama, makine öğrenmesi modellerinin doğruluğunu ve genelleme yeteneğini daha iyi değerlendirmek için kullanılan bir tekniktir. Çapraz doğrulama, veri setini birden fazla alt küme (katman) olarak böler ve her bir alt kümeyi modelin doğruluğunu değerlendirmek için kullanır. Bu sayede modelin farklı veri bölümlerindeki performansı ölçülür ve overfitting (aşırı uyum) gibi sorunlar tespit edilebilir. Çapraz doğrulama, modelin eğitim ve test verisi üzerindeki performansını daha iyi değerlendirmek ve modelin yalnızca belirli bir veri kümesine değil, genel veri yapısına ne kadar iyi uyduğunu görmek için kullanılır.

Tablo 2'de modellerin hiper-parametre optimizasyonu sonucunda elde edilen parametreleri görülmektedir.

Tablo 2. Modeller ve sahip oldukları hiper-parametreler

Model	Hiper-parametreler
kNN	n_neighbors: 5, weights: distance, algorithm: auto, p = 2
LR	penalty: l2, C: 1.0, solver: lbfgs
RF	n_estimators: 100, max_depth: 20, min_samples_split: 2, min_samples_leaf: 1, max_features: sqrt
SVM	C: 1.0, kernel: rbf, gamma: scale
XGBoost	n_estimators: 100, max_depth: 6, learning_rate: 0.1, subsample: 0.8, colsample_bytree: 0.8, gamma: 0
DNN	Gizli katman sayısı: 4, öğrenme oranı: 0.01, epoch: 50, batch boyutu: 32, optimizer: Adam, dropout oranı: 0.2

Geliştirilen sistemin akış diyagramı Şekil 5'te görülmektedir.



Şekil 5. Geliştirilen sistemin akış diyagramı

Şekil 5'te görüldüğü gibi geliştirilen sistemde öncelikle ağ trafiği verileri üzerinde veri ön-işleme yapılmaktadır. Veri ön-işlemenin ardından eğitim, test ve doğrulama veri kümeleri oluşturulmaktadır ve doğrulama verileri üzerinde modellerin hiper-parametreleri optimize edilmektedir. Izgara arama yöntemi kullanılarak belirlenen en yüksek başarı oranına sahip hiper-parametreler ile modeller oluşturulmaktadır. Model sınıflandırma işlemleri neticesinde doğruluk, kesinlik, duyarlılık ve F-skoru metriklerine göre deneysel sonuçlar elde edilmektedir.

4.2. Sınıflandırma Modelleri

RF, birden fazla karar ağacını (decision tree) kullanarak çalışır ve sonuçları birleştirir. Ana prensibi, veriyi küçük parçalara bölerek farklı ağaçlar oluşturmak ve her ağacın ayrı bir tahmin yapmasına olanak sağlamaktır [36]. RF, birden fazla karar ağacının sonuçlarını birleştirerek daha doğru ve dengeli tahminler elde eder. Sınıflandırma problemlerinde, her bir ağacın yaptığı tahminlerin çoğunluğu alınarak sonuç belirlenir [37]. Verilerden rastgele örnekler seçilerek her ağaç eğitilir. Bu sayede, aşırı öğrenme riskini azaltır. Her karar ağacı için özelliklerin sadece bir alt kümesi kullanılır. Bu sayede karar ağaçlarının birbirinden farklı olmasını ve çeşitli tahminler üretmesini sağlar [38].

kNN, bir veri noktasının benzerliğini, eğitim veri kümesiyle karşılaştırmak için yakınlığı kullanan bir makine öğrenme algoritmasıdır. kNN, benzer noktaların birbirine yakın bulunabileceği varsayımıyla çalışır [39]. kNN, komşuları içindeki çoğunluk kümesine yeni bir veri noktası atamaktadır. kNN, veri dağılımı hakkında herhangi bir varsayım gerektirmez. Belirli bir veri kümesindeki veri noktalarının benzerliğine dayalı tahminler yapan parametrik olmayan bir yöntemdir [40]. kNN, diğer algoritmalara kıyasla aykırı değerlere karşı daha az hassastır. kNN, öklid mesafesi gibi bir mesafe metriğine dayanarak belirli bir veri noktasına en yakın k komşuyu bularak çalışır [41].

SVM'nin temel amacı, iki farklı sınıfı mümkün olan en geniş sınırla (maksimum marjin) ayıran bir hiper düzlem (decision boundary) bulmaktır [42]. SVM, verileri iki sınıfa ayırmak için bir hiper düzlem tanımlar. İki boyutlu bir problemde bu hiper düzlem bir doğru olabilir, ancak daha yüksek boyutlarda düzlem ya da daha karmaşık bir yüzey olabilir [43]. SVM, iki sınıf arasında mümkün olan en geniş mesafeyi (marjin) sağlayan bir hiper düzlem seçmeye çalışır. Sınıflar arasındaki en yakın veri noktalarıyla (destek vektörleri) hiper düzlem arasındaki mesafe en büyük olacak şekilde bir ayrım yapılır. Sınıfları ayıran hiper düzleme en yakın veri noktalarına destek vektörleri adı verilir. Bu noktalar, hiper düzlemi belirlemede kritik bir rol oynar [44].

LR, bir veri noktasının iki farklı sınıftan birine ait olma olasılığını tahmin etmek için kullanılır. Temel amacı, bir olayın olasılığını hesaplayarak sınıflandırma yapmaktır [45]. LR, doğrusal regresyonda olduğu gibi her bir girdinin (özelliğin) ağırlıklandırıldığı doğrusal bir fonksiyonla başlar. doğrusal modelin çıktısını sigmoid fonksiyonu adı verilen bir fonksiyona uygular. Sigmoid fonksiyonu, tahmin edilen değeri 0 ile 1 arasında sıkıştırır [46]. Çok sınıflı LR, her bir sınıf için bir olasılık tahmin eder ve bu olasılıkların toplamı 1 olacak şekilde normalizasyon sağlar. Her sınıf için bir olasılık hesaplanır ve Softmax fonksiyonu kullanılarak olasılıkların toplamının 1 olmasını sağlayacak şekilde normalize edilir. Bu, hangi sınıfa ait olduğuna karar verirken her sınıfa bir olasılık atanmasını sağlar [47].

XGBoost, gradyan artırma (gradient boosting) metoduna dayalı bir yöntemdir. Gradyan artırma, bir dizi zayıf tahminleyici karar ağacını ardışık olarak eğiterek tahmin hatalarını minimize etmeyi hedefler [48]. Her yeni ağaç, önceki modelin hatalarını öğrenir ve böylece genel tahmin performansını iyileştirir. Model, hatalı tahminlerin daha fazla dikkate alındığı ve hataların minimize edilmeye çalışıldığı iteratif bir süreç kullanır. Hataları minimize etmek için model, gradyan tabanlı optimizasyon tekniklerini kullanarak ağırlıklarını günceller [49]. XGBoost, optimizasyon amacıyla bir kayıp fonksiyonu kullanır. XGBoost, modelin aşırı öğrenme riskini azaltmak için L1 (Lasso) ve L2 (Ridge) regularizasyon yöntemlerini kullanır. L1 regularizasyonu, özelliklerin bazılarını sıfıra düşürerek modelin basitleştirilmesine yardımcı olur. L2 regularizasyonu, ağırlıkları küçültmeye yardımcı olur ve böylece modelin daha iyi genellemesini sağlar [50].

4.3. Performans Değerlendirme Metrikleri

Sınıflandırma modellerinin değerlendirilmesinde temel olarak karışıklık matrisi kullanılarak elde edilen doğruluk (accuracy), kesinlik (precision), duyarlılık (recall) ve F-skoru (F-score) metrikleri kullanılmaktadır. Şekil 6'da çok sınıflı sınıflandırma problemleri için karışıklık matrisi görülmektedir.

		Tahmin edilen değerler			
		Deny	Drop	Allow	Reset-both
Gerçek değerler	Deny	DN	YP	DN	DN
	Drop	YN	DP	YN	YN
	Allow	DN	YP	DN	DN
	Reset-both	DN	YP	DN	DN

Şekil 6. Çok sınıflı sınıflandırma problemleri için karışıklık matrisi

Şekil 6'da, örnek olarak drop sınıfı için Doğru pozitif (DP), Yanlış Negatif (YN), Doğru Negatif (DN) ve Yanlış Pozitif (YP) değerlerinin nasıl seçileceği renkli biçimde sunulmuştur. Drop olduğu tahmin edilen ancak aslında başka bir sınıfa ait olan örnekler (YP) turuncu renkte, gerçekte drop sınıfına ait olması gereken ancak farklı bir sınıfa ait olduğu tahmin edilen örnekler (YN) ise sarı renkte gösterilmiştir. Drop sınıfı için tahmin edilen ve gerçek değerlerin kesiştiği hücre (DP) yeşil renkte gösterilmiştir. Doğru şekilde tahmin edilen farklı sınıflara ait örnekler (DN) mavi renkte gösterilmiştir.

Doğruluk, Eş. 1'de görüldüğü gibi, model tarafından doğru şekilde sınıflandırılan örnek sayısının toplam örnek sayısına bölünmesiyle elde edilir [51].

$$\text{Doğruluk} = \frac{DP+DN}{DP+YN+DN+YP} \tag{1}$$

Kesinlik, Eş. 2'de görüldüğü gibi, drop olarak sınıflandırılan örneklerden kaçının aslında drop sınıfına ait olduğunu gösterir [52].

$$\text{Kesinlik} = \frac{DP}{DP+YP} \tag{2}$$

Duyarlılık, Eş. 3'te görüldüğü gibi drop olarak sınıflandırılması gereken örneklerden kaçının doğru şekilde sınıflandırıldığını gösterir [53].

$$\text{Duyarlılık} = \frac{DP}{DP+YN} \tag{3}$$

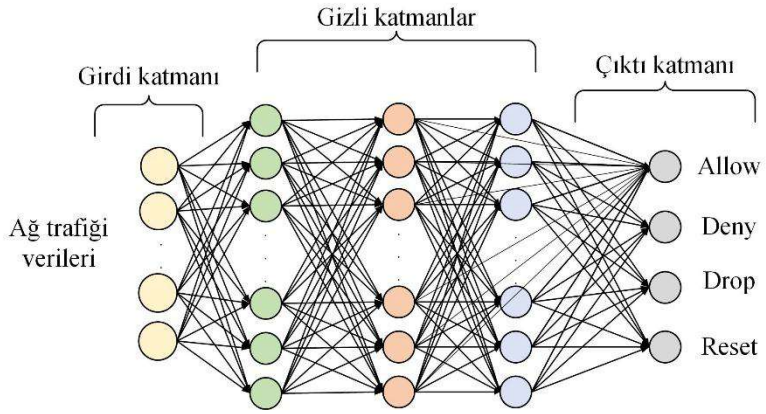
F-skoru, kesinlik ve duyarlılık değerlerinin etkisini korumak için bu değerlerin harmonik ortalaması alınarak Eş. 4'te görüldüğü gibi hesaplanmaktadır [54].

$$\text{F-skoru} = \frac{2*\text{Kesinlik}*\text{Duyarlılık}}{\text{Kesinlik}+\text{Duyarlılık}} \quad (4)$$

4.4. Geliştirilen DNN Modeli

Bu çalışmada, popüler makine öğrenme algoritmaları ile geliştirilen DNN tabanlı modelin güvenlik duvarlarındaki ağ trafiğinin analizi problemine yönelik karşılaştırmalı bir analizi sunulmuştur. DNN, t emelde, insan beynindeki nöronların çalışma prensibini taklit eden bir yapıya sahiptir. DNN'de bulunan gizli katmanlar, dış dünyaya doğrudan bağlı olmayan, hesaplamalar yapan ve giriş düğümlerinden çıkış düğümlerine bilgi ileten yapılardır. DNN, birden fazla gizli katman içerebilir. Her gizli katman, önceki katmandan aldığı verileri işler ve daha karmaşık özellikler öğrenir. Çıktı katmanı ise bilgi işleme ve bilgiyi ağdan dış dünyaya aktarmaktan sorumludur. Çıktı katmanı, modelin ne tür sonuçlar üreteceğini belirler. Geliştirilen DNN modeli, eğitim veri setindeki ağ trafiği verilerini girdi olarak alır ve test veri setindeki ağ trafiğine yönelik verilecek kararı tahmin eder. Her bir nöron, kendisine gelen girişleri belirli bir ağırlık ile çarpar ve ardından bir aktivasyon fonksiyonu uygular. Aktivasyon fonksiyonları, gizli katmanlardaki nöronların çıkışlarını belirleyen fonksiyonlardır. Geriye yayılım ile hesaplanan hata, modelin ağırlıklarının güncellenmesi için geri yayılır. Geri yayılım, DNN'nin öğrenme sürecinde en kritik adımdır.

Geliştirilen DNN modelinin mimarisi Şekil 7'de görülmektedir.



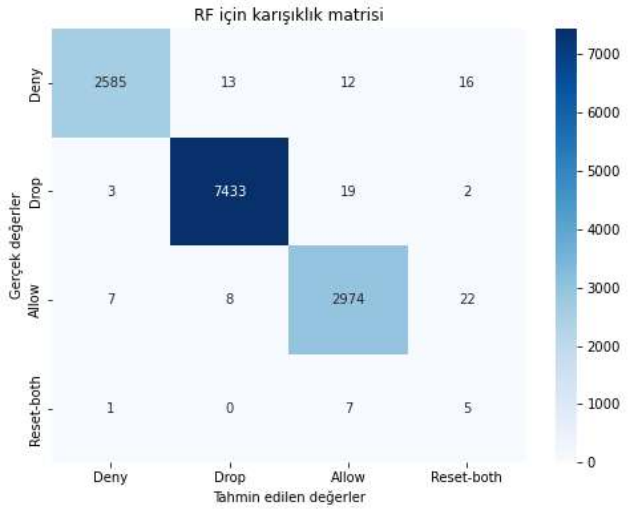
Şekil 7. Geliştirilen DNN modelinin mimarisi

Şekil 7'de görüldüğü gibi DNN modeli girdi ve çıktı katmanları ile 3 gizli katmandan oluşmaktadır. Model, ağ trafiği verilerini alarak bu verileri sınıflandırmayı amaçlamaktadır. Girdi katmanı modelin ağ trafiği verilerini aldığı ilk katmandır. Her düğüm, ağ trafiği verilerinin bir özelliğini veya değişkenini temsil eder. Ağ trafiği verileri, modelin işleyebileceği sayısal formata dönüştürülerek bu katmanda işlenmeye başlar. Gizli katmanlar, modelin ağ trafiği verilerinden özellikleri öğrenmesini sağlar. Her gizli katman, önceki katmandan aldığı bilgileri işleyerek bir sonraki katmana iletir. Her düğüm, önceden tanımlanmış bir aktivasyon fonksiyonunu kullanarak verileri işler ve ileri doğru yayılım yapar. Gizli katmanlar 64 nörondan oluşmaktadır ve Rectified Linear Unit (ReLU) aktivasyon fonksiyonundan oluşmaktadır. Çıktı katmanı, modelin sınıflandırma sonuçlarını verir. Her düğüm, modelin sınıflandıracağı bir sınıfı temsil eder. Çıktı katmanı ise allow, deny, drop ve reset-both sınıflarını ifade etmek üzere 4 nörondan oluşmaktadır ve çoklu sınıflandırma için softmax aktivasyon fonksiyonunu kullanmaktadır. Modelin öğrenme oranı 0.01, epoch sayısı 50, batch boyutu 32, optimizasyon algoritması Adam ve dropout oranı 0.2'dir. Her bir düğüm, kendisinden sonraki tüm düğümlerle bağlantılıdır. Bu, tam bağlantılı (fully connected) bir sinir ağı yapısını ifade eder. Girdi katmanından gelen veriler, gizli katmanlarda işlenir ve ardından çıktı katmanında nihai sınıflandırma yapılır.

5. Deneysel Sonuçlar

Bu çalışmada geliştirilen DNN tabanlı derin öğrenme modeli, RF, kNN, SVM, LR ve XGBoost ile uygulamalı olarak karşılaştırılmıştır. Her bir model ve sınıf için doğruluk, kesinlik, duyarlılık ve F-skoru metriklerine göre deneysel sonuçlar elde edilmiştir. Ayrıca ağırlıklı ortalama ve makro ortalamalar alınarak sınıf tahminlerinin etkileri ölçülmüştür. Test veri kümesinde deny sınıfına ait 2626 örnek, drop sınıfına ait 7457 örnek, allow sınıfına ait 3011 örnek ve reset-both sınıfına ait 13 örnek bulunmaktadır.

Şekil 8'de RF için karışıklık matrisi görülmektedir.



Şekil 8. RF için karışıklık matrisi

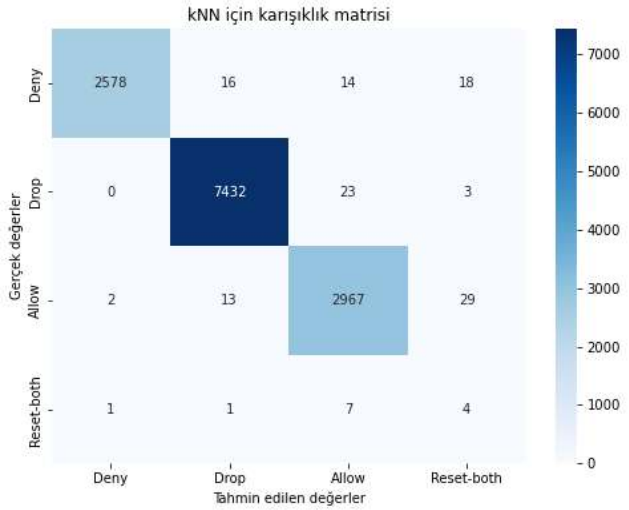
Şekil 8'de görüldüğü gibi RF deny sınıfına ait 2626 örnekten 2585'ini, drop sınıfına ait 7457 örnekten 7433'ünü, allow sınıfına ait 3011 örnekten 2974'ünü ve reset-both sınıfına ait 13 örnekten 5'ini doğru bir şekilde sınıflandırmıştır. RF, 13107 örneğin 12997'sini doğru, 110'unu ise yanlış sınıflandırarak %99,16 doğruluğa ulaşmıştır.

Tablo 3'te RF için deneysel sonuçlar görülmektedir.

Tablo 3. RF için deneysel sonuçlar

Sınıf	Doğruluk	Kesinlik	Duyarlılık	F-skoru
Deny	%99,16	%99,57	%98,43	99,00%
Drop		%99,71	%99,67	99,69%
Allow		%98,73	%98,77	98,75%
Reset-both		%11,11	%38,46	17,24%
Makro ortalama		%77,28	83,83%	78,67%
Ağırlıklı ortalama		%99,36	%99,15	%99,25

Şekil 9'da kNN için karışıklık matrisi görülmektedir.



Şekil 9. kNN için karışıklık matrisi

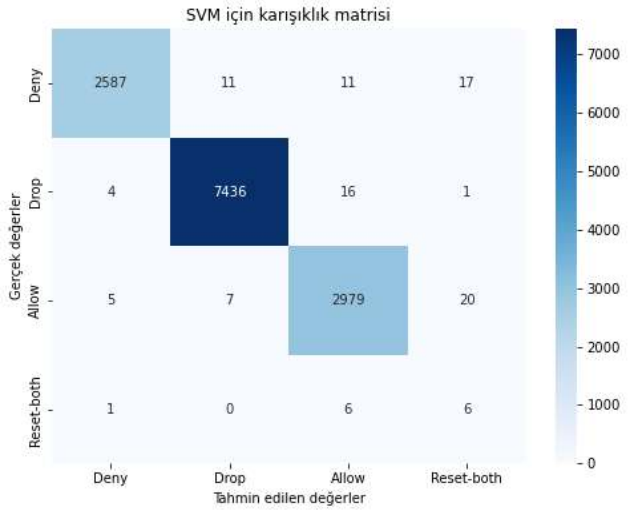
Şekil 9'da görüldüğü gibi kNN deny sınıfına ait 2626 örnekten 2578'ini, drop sınıfına ait 7457 örnekten 7432'sini, allow sınıfına ait 3011 örnekten 2967'sini ve reset-both sınıfına ait 13 örnekten 4'ünü doğru bir şekilde sınıflandırmıştır. kNN, 13107 örneğin 12981'ini doğru, 126'sını ise yanlış sınıflandırarak %99,03 doğruluğa ulaşmıştır.

Tablo 4'te kNN için deneysel sonuçlar görülmektedir.

Tablo 4. kNN için deneysel sonuçlar

Sınıf	Doğruluk	Kesinlik	Duyarlılık	F-skoru
Deny	%99,03	%99,88	%98,17	%99,02
Drop		%99,59	%99,65	%99,62
Allow		%98,53	%98,53	%98,53
Reset-both		%8,00	%30,76	%12,70
Makro ortalama		%76,50	%81,78	%77,47
Ağırlıklı ortalama		%99,31	%99,02	%99,16

Şekil 10'da SVM için karışıklık matrisi görülmektedir.



Şekil 10. SVM için karışıklık matrisi

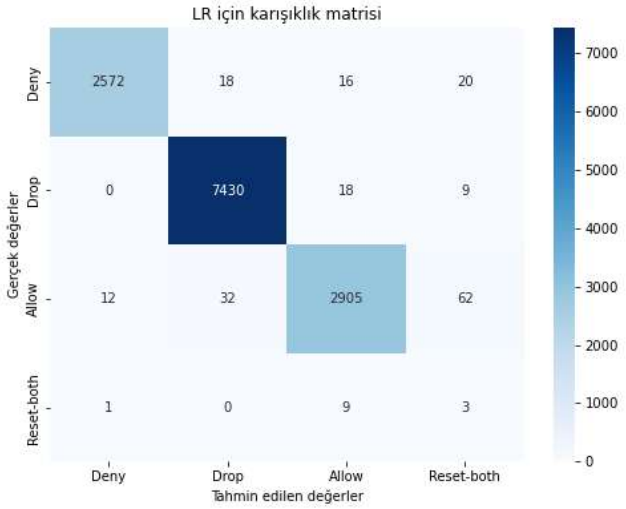
Şekil 10'da görüldüğü gibi SVM deny sınıfına ait 2626 örnekten 2587'sini, drop sınıfına ait 7457 örnekten 7436'sını, allow sınıfına ait 3011 örnekten 2979'unu ve reset-both sınıfına ait 13 örnekten 6'sını doğru bir şekilde sınıflandırmıştır. SVM, 13107 örneğin 13008'ini doğru, 99'unu ise yanlış sınıflandırarak %99,24 doğruluğa ulaşmıştır.

Tablo 5'te SVM için deneysel sonuçlar görülmektedir.

Tablo 5. SVM için deneysel sonuçlar

Sınıf	Doğruluk	Kesinlik	Duyarlılık	F-skoru
Deny	%99,24	%99,61	%98,51	%99,06
Drop		%99,75	%99,71	%99,73
Allow		%98,90	%98,93	%98,91
Reset-both		%13,63	%46,15	%21,04
Makro ortalama		%77,97	%85,83	%79,69
Ağırlıklı ortalama		%99,44	%99,23	%99,32

Şekil 11'de LR için karışıklık matrisi görülmektedir.



Şekil 11. LR için karışıklık matrisi

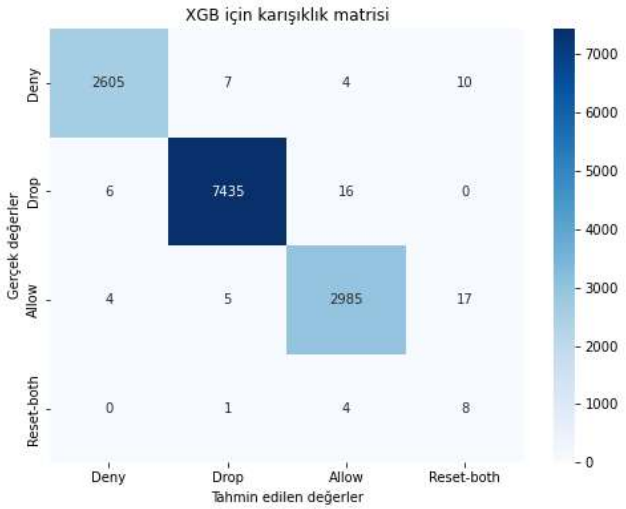
Şekil 11'de görüldüğü gibi LR deny sınıfına ait 2626 örnekten 2572'sini, drop sınıfına ait 7457 örnekten 7430'unu, allow sınıfına ait 3011 örnekten 2905'ini ve reset-both sınıfına ait 13 örnekten 3'ünü doğru bir şekilde sınıflandırmıştır. LR, 13107 örneğin 12910'unu doğru, 197'sini ise yanlış sınıflandırarak %98,49 doğruluğa ulaşmıştır.

Tablo 6'da LR için deneysel sonuçlar görülmektedir.

Tablo 6. LR için deneysel sonuçlar

Sınıf	Doğruluk	Kesinlik	Duyarlılık	F-skoru
Deny	%98,49	%99,49	%97,94	%98,71
Drop		%99,33	%99,63	%99,48
Allow		%98,54	%96,47	%97,49
Reset-both		%3,19	%23,07	%5,60
Makro ortalama		%75,14	%79,28	%75,32
Ağırlıklı ortalama		%99,08	%98,48	%98,77

Şekil 12'de XGBoost için karışıklık matrisi görülmektedir.



Şekil 12. XGBoost için karışıklık matrisi

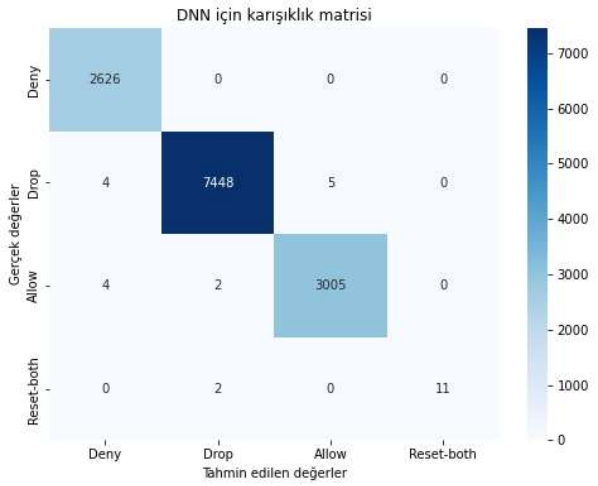
Şekil 12'de görüldüğü gibi XGBoost deny sınıfına ait 2626 örnekten 2605'ini, drop sınıfına ait 7457 örnekten 7435'ini, allow sınıfına ait 3011 örnekten 2985'ini ve reset-both sınıfına ait 13 örnekten 8'ini doğru bir şekilde sınıflandırmıştır. XGBoost, 13107 örneğin 13033'ünü doğru, 74'ünü ise yanlış sınıflandırarak %99,43 doğruluğa ulaşmıştır.

Tablo 7'de XGBoost için deneysel sonuçlar görülmektedir.

Tablo 7. XGBoost için deneysel sonuçlar

Sınıf	Doğruluk	Kesinlik	Duyarlılık	F-skoru
Deny	%99,43	%99,61	%99,20	%99,40
Drop		%99,82	%99,70	%99,76
Allow		%99,20	%99,13	%99,16
Reset-both		%22,85	%61,53	%33,32
Makro ortalama		%80,37	%89,89	%82,91
Ağırlıklı ortalama		%99,55	%99,43	%99,48

Şekil 13'te DNN için karışıklık matrisi görülmektedir.



Şekil 13. DNN için karışıklık matrisi

Şekil 13'te görüldüğü gibi DNN deny sınıfına ait 2626 örnekten 2626'sını, drop sınıfına ait 7457 örnekten 7436'sını, allow sınıfına ait 3011 örnekten 2993'ünü ve reset-both sınıfına ait 13 örnekten 10'unu doğru bir şekilde sınıflandırmıştır. DNN, 13107 örneğin 13065'ini doğru, 42'sini ise yanlış sınıflandırarak %99,67 doğruluğa ulaşmıştır.

Tablo 8'de DNN için deneysel sonuçlar görülmektedir.

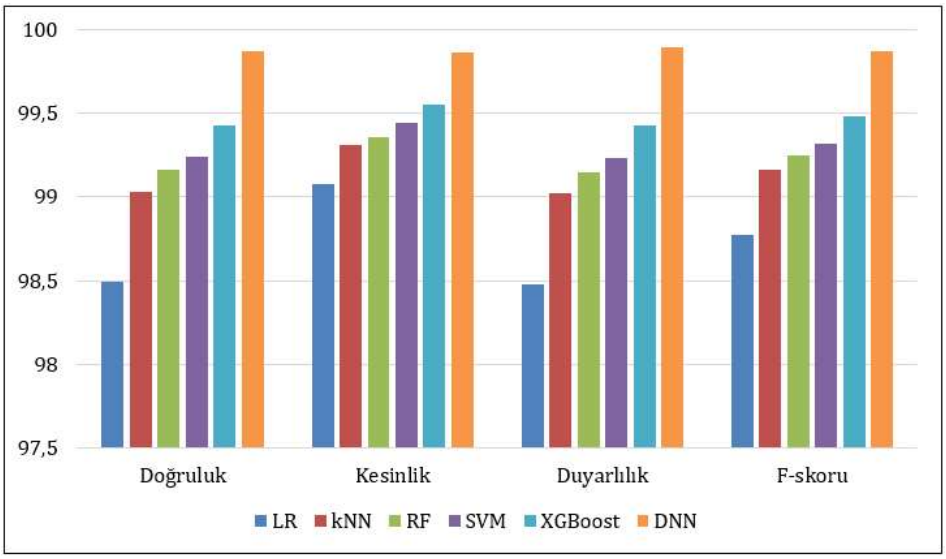
Tablo 8. DNN için deneysel sonuçlar

Sınıf	Doğruluk	Kesinlik	Duyarlılık	F-skoru
Deny	%99,87	%99,69	%100,00	%99,84
Drop		%99,94	%99,87	%99,90
Allow		%99,83	%99,93	%99,87
Reset-both		%100,00	%84,61	%91,66
Makro ortalama		%99,87	%96,10	%97,82
Ağırlıklı ortalama		%99,86	%99,89	%99,87

Tablo 9 ve Şekil 14'te ağırlıklı ortalamaya göre karşılaştırmalı deneysel sonuçlar görülmektedir.

Tablo 9. Ağırlıklı ortalamaya göre karşılaştırmalı deneysel sonuçlar

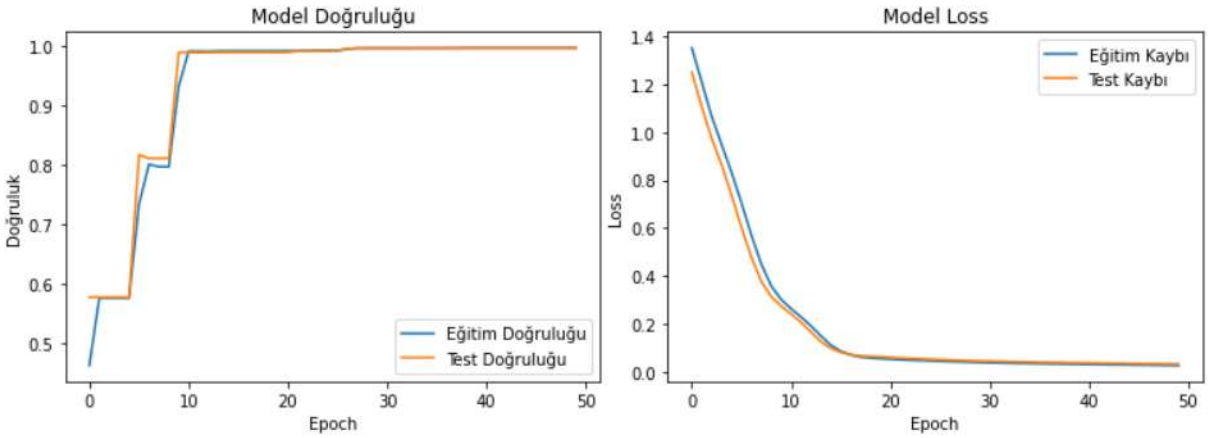
Model	Doğruluk	Kesinlik	Duyarlılık	F-skoru
LR	%98,49	%99,08	%98,48	%98,77
kNN	%99,03	%99,31	%99,02	%99,16
RF	%99,16	%99,36	%99,15	%99,25
SVM	%99,24	%99,44	%99,23	%99,32
XGBoost	%99,43	%99,55	%99,43	%99,48
DNN	%99,87	%99,86	%99,89	%99,87



Şekil 14. Ağırlıklı ortalamaya göre karşılaştırmalı deneysel sonuçlar

Tablo 9 ve Şekil 14'te güvenlik duvarı üzerinden geçen ağ trafiğinin sınıflandırılması problemine yönelik uygulanan modellerin performans karşılaştırmaları sunulmuştur. LR, %98,49 doğruluk değerine sahip olmuştur. LR için tahmin edilen doğru pozitiflerin oranı yüksektir yani model yanlış pozitif oranını düşük tutabilmiştir ancak LR gerçek pozitiflerin %98,48'ini yakalayabilmiştir. kNN, %99,03 doğrulukla LR'ye kıyasla daha yüksek doğruluk sunmaktadır. kNN, hem yanlış pozitifleri azaltmada hem de gerçek pozitifleri bulmada dengelidir. RF, %99,16 doğrulukla kNN'e kıyasla daha doğru tahminler yapmaktadır. %99,36 kesinlik ve %99,15 duyarlılık değeriyle, doğru sınıflamaları yapmada başarılıdır. SVM genellikle doğrusal olmayan ayrım çizgileri gerektiren verilerde iyi çalışır ve bu problemde %99,24 doğrulukla RF ve kNN'e göre daha iyi sonuç vermiştir. XGBoost, genellikle büyük veri setleri ve karmaşık sınıflandırma problemleri için çok güçlü bir modeldir. Diğer modellere göre daha karmaşık bir optimizasyon tekniği kullandığı için bu problemde oldukça yüksek performans göstermiştir. DNN, en yüksek doğruluğa ve dengeli performansa sahip model olarak öne çıkmaktadır. XGBoost ve SVM de oldukça yüksek doğruluk değerine sahip olmuş ve duyarlılık/kesinlik oranlarında oldukça başarılıdır. Deneysel sonuçlar güvenlik duvarı üzerinden geçen ağ trafiğinin sınıflandırılmasında DNN'in karşılaştırılan modellerden daha başarılı olduğunu, XGBoost ve SVM'in DNN'in ardından en başarılı modeller olduğunu göstermiştir.

DNN'in epoch/loss grafikleri Şekil 15'te görülmektedir.



Şekil 15. DNN'in epoch/loss grafikleri

Şekil 15'te görülen model doğruluğu grafiğinde mavi çizgi eğitim doğruluğunu, turuncu çizgi ise test doğruluğunu ifade etmektedir. Modelin doğruluğunun başlangıçta düşük olduğu ancak yaklaşık 10. epoch'tan sonra hızla arttığını ve ardından 1'e yakın sabit bir değere ulaştığını gösteriyor. Model doğruluğu grafiği, modelin eğitim ve test doğruluğunun yüksek olduğunu ve dolayısıyla eğitimin başarılı bir şekilde ilerlediğini göstermektedir. Model loss grafiğinde ise ilk epoch'larda kayıp değeri yüksek olsa da, model eğitim aldıkça kayıp değeri düşmektedir. Yaklaşık 20. epoch'tan sonra kayıp değerleri stabilize olmuş, eğitim ve test kaybı birbirine yakın seyretmektedir. Şekil 15, modelin eğitim sürecinin başarılı bir şekilde ilerlediğini gösteriyor. Model doğruluğunun hızla yükselip sabit kalması ve kayıp değerinin düşerek az bir farkla seyretmesi, modelin iyi bir şekilde genelleme yapabildiğini ve aşırı uyum (overfitting) olmadığını gösterir.

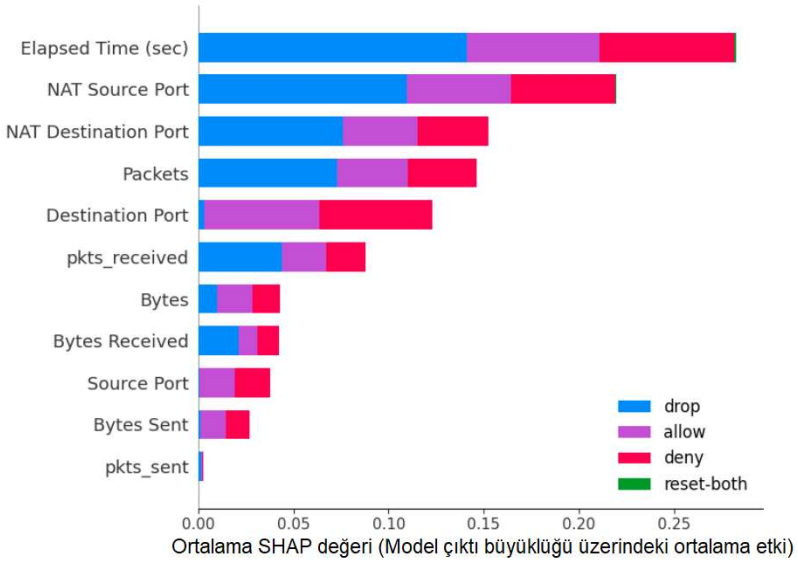
5.1. XAI ile Elde Edilen Sonuçların İncelenmesi

SHAP, makine öğrenmesi ve derin öğrenme modellerinin tahminlerini açıklamak için kullanılan güçlü bir XAI yöntemidir. Modelin karar verme sürecini şeffaflaştırır ve bir modelin nasıl çalıştığını anlamayı sağlar. Shapley değeri, oyun teorisinde her oyuncunun oyun sonucuna ne kadar katkıda bulunduğunu hesaplamak için kullanılır. SHAP, bu matematiksel çerçeveyi makine öğrenimi modellerine uyarlayarak, her bir özellik için modelin tahminine olan katkıyı belirler. SHAP değeri, bir özelliğin modelin tahminine olan katkısını ölçen değerdir. Modelin ürettiği tahmin, tüm özelliklerin toplu etkisi ile elde edilmektedir. Her bir tahminin, tüm özelliklerin katkılarının (pozitif veya negatif) toplamı ile elde edildiği varsayılmaktadır. Modelin tahmin ettiği değer, Eş. 5 kullanılarak hesaplanmaktadır.

$$f(x) = \phi_0 + \sum_{i=1}^{n} \phi_i \tag{5}$$

ϕ_0 Modelin ortalama tahmini, ϕ_i ise i. özelliğin tahmin üzerindeki katkısını (SHAP değeri) ifade etmektedir.

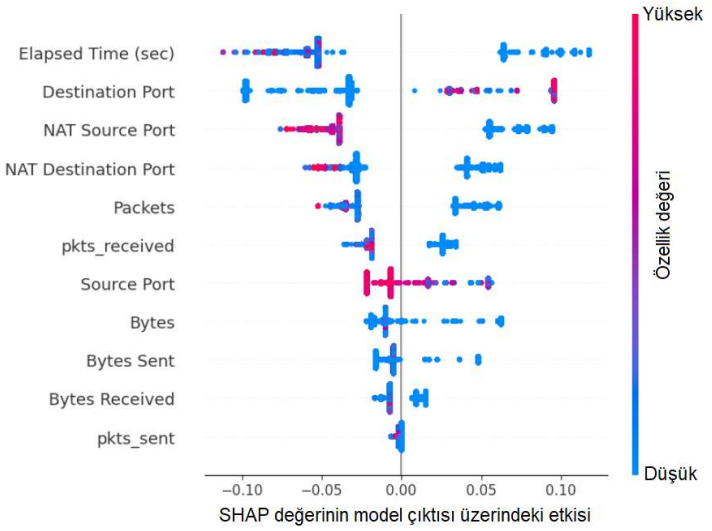
Bu bölümde, DNN ile yapılan tahminlerin SHAP kullanılarak XAI ile incelenmesine yönelik deneysel sonuçlar sunulmuştur. Şekil 16'da ortalama SHAP değeri görülmektedir.



Şekil 16. Ortalama SHAP değeri

Şekil 16, modelin tahmininde hangi özelliklerin ne kadar etkili olduğunu göstermektedir. Özellikler, SHAP değerlerinin ortalama etkisine göre sıralanmıştır. En üstteki Elapsed Time (sec) özniteliği, modelin tahminlerinde en büyük etkiye sahipken, pkts_sent en az etkili özelliktir. Özelliklerin rengine göre drop, allow, deny ve reset-both sınıfları için nasıl katkıda bulunduğu görülmektedir. Modelin kararlarını en çok etkileyen özelliklerin başında Elapsed Time (sec) ve NAT port özellikleri gelmektedir.

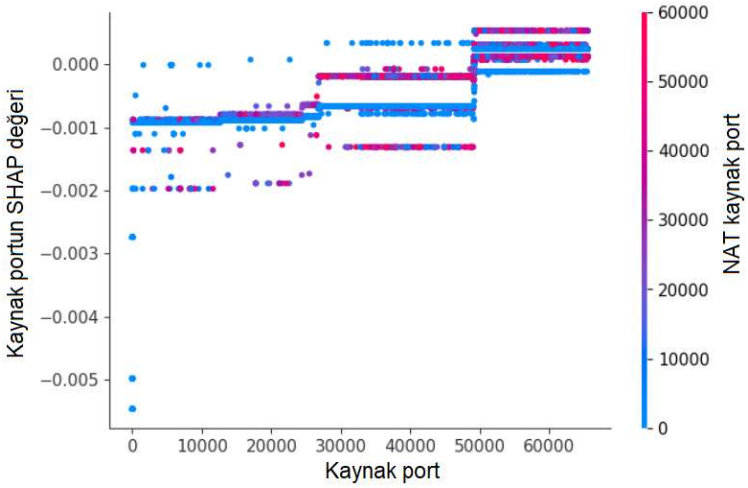
Şekil 17'de SHAP değerinin model çıktısı üzerindeki etkisi görülmektedir.



Şekil 17. SHAP değerinin model çıktısı üzerindeki etkisi

Şekil 17, her bir veri noktasındaki SHAP değerlerinin model çıktısı üzerindeki etkisini göstermektedir. Yüksek SHAP değerlerine sahip veri noktaları, modelin tahminlerini pozitif yönde etkilerken, düşük SHAP değerlerine sahip olanlar ise negatif etkiler. Renk skalası, ilgili özelliğin değeriyle SHAP değerini ilişkilendirmektedir. Örneğin, Elapsed Time (sec) özniteliğindeki yüksek değerlere sahip veri noktaları modelin çıktılarını pozitif etkilemektedir.

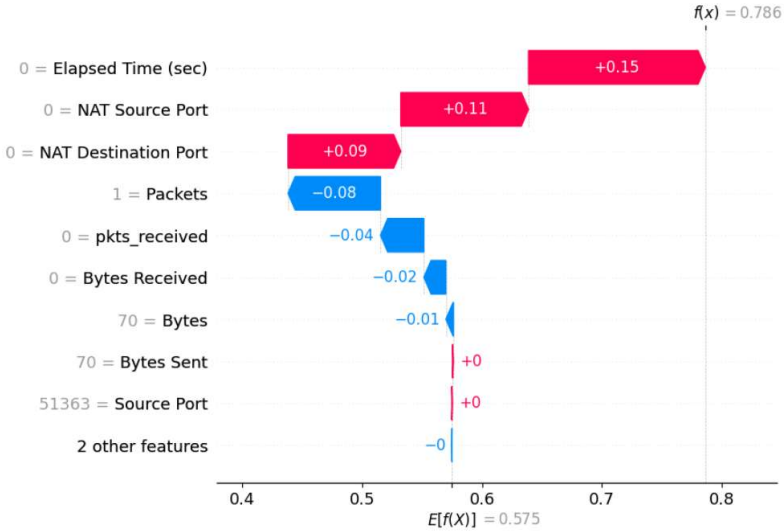
Şekil 18'de NAT kaynak portu ile kaynak portu arasındaki ilişki görülmektedir.



Şekil 18. Kaynak portu ile NAT kaynak portu arasındaki ilişki

Şekil 18, kaynak portu ile NAT kaynak portu arasındaki ilişkiyi göstermektedir ve noktaların rengi NAT kaynak portu değerlerine göre değişmektedir. Şekil 18, özellikle bu iki özelliğin model tahminlerini nasıl etkilediğini anlamak için kullanılmaktadır. Belirli bir NAT kaynak portu aralığında SHAP değerlerinin nasıl dağıldığını ve model çıktısına olan etkisini gözlemlemeyi mümkün kılmaktadır.

Şekil 19'da tek bir gözlem için sunulan SHAP değerleri görülmektedir.



Şekil 19. Tek bir gözlem için SHAP değerleri

Şekil 19'da, her bir özellik, modelin tahminini nasıl etkilediğine göre pozitif (kırmızı) veya negatif (mavi) renkte gösterilmiştir. Örneğin, bu gözlemde Elapsed Time (sec) özniteliği tahmini büyük ölçüde pozitif etkilerken, Packets ve pkts_received gibi özellikler tahmini negatif yönde etkilemektedir. Gözlemin genel tahminine en büyük katkıyı sağlayanlar kırmızı renkle, en büyük olumsuz etkileri sağlayanlar ise mavi renkle gösterilmiştir.

6. Tartışma

Veri setindeki sınıfların dengesizliği, LR'nin başarılı bir sınıflandırma yapamamasına neden olmuştur. kNN, basit bir sınıflandırma algoritmasıdır ve herhangi bir parametre ayarı gerektirmeden doğrudan veri üzerinde çalışabilir. kNN, veri noktalarının yerel yapısını kullanarak sınıflandırma yapar, bu da sınıflar arasında belirgin ayrımlar olduğunda etkili olabilir. RF, birden fazla karar ağacını bir araya getirerek, aşırı öğrenme riskini azaltmıştır ve daha güçlü bir genel performans sağlamıştır. SVM, yüksek boyutlu alanlarda oldukça iyi çalışır ve karmaşık sınıflandırma problemlerinde başarılıdır. XGBoost'un, aşırı öğrenme riskini azaltmak için L1 düzenleme kullanılmıştır ve hiper-parametre optimizasyonu ile bellek kullanımının verimli bir hale getirilmesi sağlanmıştır. DNN, derin katmanları sayesinde karmaşık ilişkileri öğrenerek yüksek boyutlu bu veri setinde etkili olmuştur. DNN'ler, çok sayıda katmandan oluşur ve her bir katman, verinin farklı özelliklerini öğrenir. Bu yapı, modelin karmaşık ilişkileri ve örüntüleri yakalamasına olanak tanır. Özellikle büyük ve karmaşık veri kümeleri için etkili bir şekilde öğrenebilir. DNN, ReLU ve Sigmoid gibi aktivasyon fonksiyonları kullanarak karmaşık doğrusal olmayan ilişkileri modelleyebilmiştir.

6. Sonuçlar

Küresel çapta internet kullanımındaki artış, yalnızca bireylerin ve işletmelerin çevrimiçi etkinliklerini hızlandırmakla kalmamış, aynı zamanda siber tehditlerin de daha karmaşık hale gelmesine neden olmuştur. Artan kullanıcı etkinlikleri ve çevrimiçi hizmetlere yönelik talepler, internet trafiğini giderek daha yoğun ve karmaşık bir yapıya dönüştürmüştür. Geleneksel ağ güvenliği yöntemleri ve özellikle güvenlik duvarları gibi çözümler, belirli kurallar setine dayandıkları için dinamik tehditler karşısında yetersiz kalabilmektedir. Güvenlik duvarları, ağlar arasındaki trafiği denetleyerek tehditlerin tespitine ve engellenmesine olanak sağlasa da, modern siber saldırıların karmaşıklığı karşısında bu yöntemler verimsiz hale gelmektedir. Bu noktada, yapay zekâ tabanlı çözümler, özellikle güvenlik duvarlarındaki log kayıtlarının daha etkin bir şekilde analiz edilmesi açısından yeni ve güçlü bir savunma mekanizması olarak öne çıkmaktadır. Ağ trafiği üzerindeki olası tehditlerin gerçek zamanlı olarak tespit edilebilmesi ve anormalliklerin fark edilmesi, yalnızca verilerin korunmasını sağlamakla kalmayıp, aynı zamanda önleyici tedbirlerin alınmasına da katkıda bulunmaktadır. Giderek daha karmaşık hale gelen saldırı türlerine karşı dinamik öğrenme yeteneği, yapay zekâ tabanlı sistemlerin daha esnek ve uyarlanabilir olmasına olanak tanır. Bu çalışmada, güvenlik duvarları üzerinden geçen ağ trafiği loglarının analizine yönelik DNN tabanlı bir derin öğrenme modeli geliştirilmiştir. Geliştirilen model ile verimli ve etkili bir analiz yöntemi sunarak geleneksel yöntemlerin sınırlılıklarını aşmak hedeflenmiştir. DNN, RF, kNN, SVM, LR ve XGBoost gibi popüler makine öğrenmesi algoritmalarıyla karşılaştırılarak kapsamlı bir değerlendirme sunulmuştur. Bu çalışma, sadece ağ güvenliği için değil, aynı zamanda yapay zekâ tabanlı güvenlik yaklaşımlarının açıklanabilirliğine de katkıda bulunmayı amaçlamaktadır. Güvenlik sistemlerinin karar alma süreçlerini şeffaflaştıran SHAP XAI yöntemi kullanılarak, yapılan sınıflandırma işlemlerinin anlaşılır ve izlenebilir olması sağlanmıştır. Çalışmanın sonuçları,

geliştirilen DNN tabanlı modelin %99,87 doğruluk oranıyla karşılaştırılan modellerden ve literatürdeki çalışmalardan daha yüksek bir performans sergilediğini ortaya koymuştur.

Kaynakça

[1] Diwan, T. D. 2021. An investigation and analysis of cyber security information systems: latest trends and future suggestion. Information Technology in Industry, 9(2), 477-492.

[2] Alsaqour, R., Motmi, A., Abdelhaq, M. 2021. A systematic study of network firewall and its implementation. International Journal of Computer Science & Network Security, 21(4), 199-208.

[3] Islam, M. S., Uddin, M. A., Hossain, D. M. D., Ahmed, D. M. S., Moazzam, D. M. G. 2023. Analysis and evaluation of network and application security based on next generation firewall. International Journal of Computing and Digital Systems, 13(1), 193-202.

[4] Anwar, R. W., Abdullah, T., Pastore, F. 2021. Firewall best practices for securing smart healthcare environment: A review. Applied Sciences, 11(19).

[5] Varzaneh, M. R., Habbal, A., Dakkak, O. 2024. Firewalls and Internet of Things Security: A Survey. Current Trends in Computing, 1(1), 22-43.

[6] Lyu, M., Gharakheili, H. H., Russell, C., Sivaraman, V. 2021. Hierarchical anomaly-based detection of distributed DNS attacks on enterprise networks. IEEE Transactions on Network and Service Management, 18(1), 1031-1048.

[7] Lamdakkar, O., Ameur, I., Eleyatt, M. M., Carlier, F., Ibourek, L. A. 2024. Toward a modern secure network based on next-generation firewalls: recommendations and best practices. Procedia Computer Science, 238, 1029-1035.

[8] Malhotra, P., Singh, Y., Anand, P., Bangotra, D. K., Singh, P. K., Hong, W. C. 2021. Internet of things: Evolution, concerns and security challenges. Sensors, 21(5).

[9] Aslan, Ö., Aktuğ, S. S., Ozkan-Okay, M., Yilmaz, A. A., Akin, E. 2023. A comprehensive review of cyber security vulnerabilities, threats, attacks, and solutions. Electronics, 12(6).

[10] Ahmadi, S. 2023. Next Generation AI-Based Firewalls: A Comparative Study. International Journal of Computer (IJC), 49(1), 245-262.

[11] Ertam, F., Kaya, M. 2018. Classification of firewall log files with multiclass support vector machine. In 2018 6th International symposium on digital forensic and security (ISDFS), 22-25 Mart, Antalya, 1-4.

[12] Al-Haijaa, Q. A., Ishtaiwia, A. 2021. Machine learning based model to identify firewall decisions to improve cyber-defense. International Journal on Advanced Science, Engineering and Information Technology, 11(4), 1688-1695.

[13] Al-Behadili, H. N. K. 2021. Decision tree for multiclass classification of firewall access. International Journal of Intelligent Engineering and Systems, 14(3), 294-302.

[14] Naryanto, R. F., Delimayanti, M. K. (2022, November). Machine Learning Technique for Classification of Internet Firewall Data Using RapidMiner. 2022 6th International Conference on Electrical, Telecommunication and Computer Engineering (ELTICOM), 22-23 Kasım, Medan, 155-159.

[15] Aljabri, M., Alahmadi, A. A., Mohammad, R. M. A., Aboulnour, M., Alomari, D. M., Almotiri, S. H. 2022. Classification of firewall log data using multiclass machine learning models. Electronics, 11(12), 1851.

[16] Rahman, M. H., Islam, T., Rana, M. M., Tasnim, R., Mona, T. R., Sakib, M. M. 2023. Machine Learning Approach on Multiclass Classification of Internet Firewall Log Files. 2023 International Conference on Computational Intelligence and Sustainable Engineering Solutions (CISES), 28-30 Nisan, Delhi, 358-364.

[17] Al-Tarawneh, B. A., Bani-Salameh, H. 2023. Classification of firewall logs actions using machine learning techniques and deep neural network. AIP Proceedings, 2979(1).

[18] Efeoğlu, E., Tuna, G. 2024. Classification of Firewall Log Files with Different Algorithms and Performance Analysis of These Algorithms. Journal of Web Engineering, 23(4), 561-593.

[19] Lee, J. K., Hong, T., Lee, G. 2024. AI-Based Approach to Firewall Rule Refinement on High-Performance Computing Service Network. Applied Sciences, 14(11).

[20] Haque, A. B., Islam, A. N., Mikalef, P. 2023. Explainable Artificial Intelligence (XAI) from a user perspective: A synthesis of prior literature and problematizing avenues for future research. Technological Forecasting and Social Change, 186.

[21] Hassija, V., Chamola, V., Mahapatra, A., Singal, A., Goel, D., Huang, K., Hussain, A. 2024. Interpreting black-box models: a review on explainable artificial intelligence. Cognitive Computation, 16(1), 45-74.

[22] Díaz-Rodríguez, N., Del Ser, J., Coeckelbergh, M., de Prado, M. L., Herrera-Viedma, E., Herrera, F. 2023. Connecting the dots in trustworthy Artificial Intelligence: From AI principles, ethics, and key requirements to responsible AI systems and regulation. Information Fusion, 99.

[23] Haque, A. B., Islam, A. N., Mikalef, P. 2023. Explainable Artificial Intelligence (XAI) from a user perspective: A synthesis of prior literature and problematizing avenues for future research. Technological Forecasting and Social Change, 186.

[24] Novelli, C., Taddeo, M., Floridi, L. 2024. Accountability in artificial intelligence: what it is and how it works. Ai & Society, 39(4), 1871-1882.
[25] McDermid, J. A., Jia, Y., Porter, Z., Habli, I. 2021. Artificial intelligence explainability: the technical and ethical dimensions. Philosophical Transactions of the Royal Society A, 379(2207).
[26] Hagendorff, T., Bossert, L. N., Tse, Y. F., Singer, P. 2023. Speciesist bias in AI: how AI applications perpetuate discrimination and unfair outcomes against animals. AI and Ethics, 3(3), 717-734.
[27] Goldman, C. V., Baltaxe, M., Chakraborty, D., Arinez, J., Diaz, C. E. 2023. Interpreting learning models in manufacturing processes: towards explainable AI methods to improve trust in classifier predictions. Journal of Industrial Information Integration, 33.
[28] Tripathi, S., Muhr, D., Brunner, M., Jodlbauer, H., Dehmer, M., Emmert-Streib, F. 2021. Ensuring the robustness and reliability of data-driven knowledge discovery models in production and manufacturing. Frontiers in artificial intelligence, 4.
[29] Dwivedi, R., Dave, D., Naik, H., Singhal, S., Omer, R., Patel, P., Ranjan, R. 2023. Explainable AI (XAI): Core ideas, techniques, and solutions. ACM Computing Surveys, 55(9), 1-33.
[30] Maruthi, S., Dodda, S. B., Yellu, R. R., Thuniki, P., Reddy, S. R. B. 2022. Toward a Hermeneutics of Explainability: Unraveling the Inner Workings of AI Systems. Journal of Artificial Intelligence Research and Applications, 2(2), 27-44.
[31] El-Khawaga, G., Abu-Elkheir, M., Reichert, M. 2022. Xai in the context of predictive process monitoring: An empirical analysis framework. Algorithms, 15(6).
[32] Feng, D. C., Wang, W. J., Mangalathu, S., Taciroglu, E. 2021. Interpretable XGBoost-SHAP machine-learning model for shear strength prediction of squat RC walls. Journal of Structural Engineering, 147(11).
[33] Sahakyan, M., Aung, Z., Rahwan, T. 2021. Explainable artificial intelligence for tabular data: A survey. IEEE Access, 9, 135392-135422.
[34] Sim, T., Choi, S., Kim, Y., Youn, S. H., Jang, D. J., Lee, S., Chun, C. J. 2022. eXplainable AI (XAI)-based input variable selection methodology for forecasting energy consumption. Electronics, 11(18).
[35] Tunguz, B. 2021. Internet Firewall Data Set. https://www.kaggle.com/datasets/tunguz/internet-firewall-data-set/data (Erişim Tarihi: 01.09.2024).
[36] Mosavi, A., Sajedi Hosseini, F., Choubin, B., Goodarzi, M., Dineva, A. A., Rafiei Sardooi, E. 2021. Ensemble boosting and bagging based machine learning models for groundwater potential prediction. Water Resources Management, 35, 23-37.
[37] Aria, M., Cuccurullo, C., Gnasso, A. 2021. A comparison among interpretative proposals for Random Forests. Machine Learning with Applications, 6.
[38] Subbiah, S., Anbananthen, K. S. M., Thangaraj, S., Kannan, S., Chelliah, D. 2022. Intrusion detection technique in wireless sensor network using grid search random forest with Boruta feature selection algorithm. Journal of Communications and Networks, 24(2), 264-273.
[39] Zhang, S., Li, J., Li, Y. 2022. Reachable distance function for KNN classification. IEEE Transactions on Knowledge and Data Engineering, 35(7), 7382-7396.
[40] Dong, Y., Ma, X., Fu, T. 2021. Electrical load forecasting: A deep learning approach based on K-nearest neighbors. Applied Soft Computing, 99.
[41] Uddin, S., Haque, I., Lu, H., Moni, M. A., Gide, E. 2022. Comparative performance analysis of K-nearest neighbour (KNN) algorithm and its different variants for disease prediction. Scientific Reports, 12(1).
[42] Wisanwanichthan, T., Thammawichai, M. 2021. A double-layered hybrid approach for network intrusion detection system using combined naive bayes and SVM. Ieee Access, 9, 138432-138450.
[43] Ke, T., Ge, X., Yin, F., Zhang, L., Zheng, Y., Zhang, C., Wang, W. 2024. A general maximal margin hyper-sphere SVM for multi-class classification. Expert Systems with Applications, 237, 121647.
[44] Anyanwu, G. O., Nwakanma, C. I., Lee, J. M., Kim, D. S. 2022. Optimization of RBF-SVM kernel using grid search algorithm for DDoS attack detection in SDN-based VANET. IEEE Internet of Things Journal, 10(10), 8477-8490.
[45] Khalifa, R. M., Yacout, S., Bassetto, S. 2021. Developing machine-learning regression model with Logical Analysis of Data (LAD). Computers & Industrial Engineering, 151.
[46] Huang, K., Zhang, H. 2022. Classification and regression machine learning models for predicting aerobic ready and inherent biodegradation of organic chemicals in water. Environmental Science & Technology, 56(17), 12755-12764.
[47] Theissler, A., Thomas, M., Burch, M., Gerschner, F. 2022. ConfusionVis: Comparative evaluation and selection of multi-class classifiers based on confusion matrices. Knowledge-Based Systems, 247.
[48] Kiangala, S. K., Wang, Z. 2021. An effective adaptive customization framework for small manufacturing plants using extreme gradient boosting-XGBoost and random forest ensemble learning algorithms in an Industry 4.0 environment. Machine Learning with Applications, 4.
[49] Mohiuddin, G., Lin, Z., Zheng, J., Wu, J., Li, W., Fang, Y., Zeng, X. 2023. Intrusion detection using hybridized meta-heuristic techniques with Weighted XGBoost Classifier. Expert Systems with Applications, 232.

[50] Hajihosseinlou, M., Maghsoudi, A., Ghezelbash, R. 2024. Regularization in machine learning models for MVT Pb-Zn prospectivity mapping: applying lasso and elastic-net algorithms. Earth Science Informatics, 1-15.
[51] Vujović, Ž. 2021. Classification model evaluation metrics. International Journal of Advanced Computer Science and Applications, 12(6), 599-606.
[52] Mubarak, A. A., Cao, H., Hezam, I. M. 2021. Deep analytic model for student dropout prediction in massive open online courses. Computers & Electrical Engineering, 93.
[53] Lee, W., Seo, K. 2022. Downsampling for binary classification with a highly imbalanced dataset using active learning. Big Data Research, 28.
[54] Agarwal, N., Tayal, D. K. 2022. FFT based ensembled model to predict ranks of higher educational institutions. Multimedia Tools and Applications, 81(23), 34129-34162.

Ekler

Ek A. Ek başlığı

Bu başlık zorunlu değildir. Metin içerisindeki şekil, grafik, tablo veya resim gibi görseller hakkında uzun ek bilgilere gerek duyulması durumunda bu kısımda verilmelidir. Bölüm başlığı "Cambria" fontunda 10 punto ve kalın olarak yazılmalıdır. Birden fazla ek kullanılacak olması durumunda alt başlıklar **"Ek A., Ek B., vb."** şeklinde "Cambria" fontunda 9 punto ve kalın olarak yazılmalıdır. Eğer çalışmanızı İngilizce sunmak istiyorsanız bölüm başlığını "**Appendices**", alt başlıkları ise **"Appendix A., Appendix B., etc"** olarak değiştiriniz.