T.C.
NECMETTİN ERBAKAN
ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

BİR BOYUTLU EVRİŞİMLİ SİNİR AĞLARI KULLANILARAK AĞ SALDIRI TESPİTİ

Zahide TOPBAŞ

YÜKSEK LİSANS TEZİ

Bilgisayar Mühendisliği Anabilim Dalı

Nisan-2024
KONYA

TEZ KABUL VE ONAYI

Zahide TOPBAŞ tarafından hazırlanan "Bir Boyutlu Evrişimli Sinir Ağları Kullanılarak Ağ Saldırı Tespiti" adlı tez çalışması 19/04/2024 tarihinde aşağıdaki jüri tarafından oy birliği ile Necmettin Erbakan Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Anabilim Dalı'nda YÜKSEK LİSANS TEZİ olarak kabul edilmiştir.

Jüri Üyeleri	**İmza**
Başkan Dr. Öğr. Üyesi Onur İNAN	……..
Danışman Doç. Dr. Şaban GÜLCÜ	……..
Üye Dr. Öğr. Üyesi Özlem ERDAŞ ÇİÇEK	……..

Fen Bilimleri Enstitüsü Yönetim Kurulu'nun …./…/20.. gün ve …….. sayılı kararıyla onaylanmıştır.

Prof. Dr. Şerife Yurdagül KUMCU
FBE Müdürü

TEZ BİLDİRİMİ

Bu tezdeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edildiğini ve tez yazım kurallarına uygun olarak hazırlanan bu çalışmada bana ait olmayan her türlü ifade ve bilginin kaynağına eksiksiz atıf yapıldığını bildiririm.

DECLARATION PAGE

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

İmza

Zahide TOPBAŞ

19.04.2024

ÖZET

YÜKSEK LİSANS TEZİ

BİR BOYUTLU EVRİŞİMLİ SİNİR AĞLARI KULLANILARAK AĞ SALDIRI TESPİTİ

Zahide TOPBAŞ

Necmettin Erbakan Üniversitesi Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı

Danışman: Doç.Dr. Şaban GÜLCÜ

2024, 99 Sayfa

Jüri
Doç.Dr. Şaban GÜLCÜ
Dr. Öğr. Üyesi Onur İNAN
Dr. Öğr. Üyesi Özlem ERDAŞ ÇİÇEK

Derin öğrenme; insandaki sinir sisteminden ilham alan yapay sinir ağları algoritmasının çok seviyeli, derin yaklaşımıdır. Derin öğrenme yöntemlerinin kullanımı ile ilgili olarak literatürde birçok farklı alanda başarılı örnekler mevcuttur. Bizim çalışmamızda ise ağ saldırılarının tespitine yönelik siber güvenlik alanındaki kullanımına bir örnek sunulmuştur. Teknolojinin gelişmesi ve tehdit alanının büyümesinden dolayı ağlar üzerinde siber güvenlik olaylarının tespiti ve iyileştirmesi çalışmaları eylem planlarına girmiş durumdadır. Saldırı tespitini imza tabanlı olarak gerçekleştiren sistemler mevcuttur. Burada ise ağ saldırı trafiği veri setinden öğrenme gerçekleştirilerek saldırı tespitinin otomatize olarak sağlanması amaçlanmıştır. Çalışmalarda K En Yakın Komşu (KNN) ve Bir Boyutlu Evrişimli Sinir Ağı (1DCNN) modellerinin CSE-CIC-IDS 2018 güncel veri seti ile eğitimi sağlanmıştır. Gerçekleştirilen testlerde veri kümesinde hibrit yöntemler olan SMOTETomek, SMOTEENN, Tek Taraflı Seçim (One Sided Selection) algoritmaları uygulanarak veri seti dağılımının performansa etkisi incelenmiştir. Gerçekleştirilen testlerde saldırı türüne göre DoS, DDoS, Bot saldırılarının tespitinde %99 ve üzerinde başarılı sonuç elde edilmiştir.

Anahtar Kelimeler: Bir Boyutlu Evrişimli Sinir Ağları (1DCNN), Derin Öğrenme, Evrişimli Sinir Ağları (ESA), K En Yakın Komşu (KNN), Saldırı Tespiti

ABSTRACT

MS THESIS

NETWORK INTRUSION DETECTION USING ONE DIMENSIONAL CONVOLUTIONAL NEURAL NETWORKS

Zahide TOPBAŞ

THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCE OF NECMETTİN ERBAKAN UNIVERSITY
THE DEGREE OF MASTER OF SCIENCE

Advisor: Assoc.Prof.Dr. Şaban GÜLCÜ

2024, 99 Pages

Jury
Assoc.Prof.Dr. Şaban GÜLCÜ
Asst.Prof.Dr. Onur İNAN
Asst.Prof.Dr. Özlem ERDAŞ ÇİÇEK

Deep learning; It is a multi-level, deep approach of artificial neural networks algorithm inspired by the human nervous system. There are successful examples in the literature regarding the use of deep learning methods in many different fields. In our study, an example of its use in the field of cyber security for the detection of network attacks is presented. Due to the development of technology and the growth of the threat area, efforts to detect and remediate cyber security incidents on networks have been included in action plans. There are systems that perform attack detection on a signature-based basis, where the aim is to provide automatic attack detection by learning from the network attack traffic data set. In the studies, K Nearest Neighbor (KNN) and One-Dimensional Convolutional Neural Network (1DCNN) models were trained with the CSE-CIC-IDS 2018 current data set. In the tests performed, the effect of data set distribution on performance was examined by applying hybrid methods SMOTETomek, SMOTEENN, One Sided Selection algorithms on the data set. In the tests performed, 99% or more successful results were achieved in detecting DoS, DDoS and Bot attacks, depending on the attack type.

Keywords: Convolutional Neural Networks (CNN), Deep Learning, Intrusion Detection, K Nearest Neighbor (KNN), One Dimensional Convolutional Neural Network (1DCNN)

ÖNSÖZ

“Bir Boyutlu Evrişimli Sinir Ağları Kullanılarak Ağ Saldırı Tespiti” adlı bu çalışmada yapay zekânın siber güvenlik alanındaki kullanımına dair bir örnek sunulmuştur. Hazırlamış olduğum bu çalışmanın ilgili araştırmacılar için faydalı olmasını temenni ederim.

Tez sürecimdeki desteği ve anlayışı için danışman hocam Doç. Dr. Şaban GÜLCÜ’ye, çalışmamda jüri üyeleri olan tezimi geliştirici önerilerde bulunan Dr. Öğr. Üyesi Onur İNAN hocama ve aynı zamanda çalışmalarımı yaparken motive olmamı sağlayan Dr. Öğr. Üyesi Özlem ERDAŞ ÇİÇEK hocama teşekkür ederim.

Bu süreçte desteklerini hissettiğim kıymetli arkadaşlarıma, hayatım boyunca desteklerini esirgemeyen her zaman yanımda olan sevgili aileme kalpten teşekkür ederim.

Zahide TOPBAŞ
KONYA-2024

İÇİNDEKİLER

SİMGELER VE KISALTMALAR

Kısaltmalar

CNN: Evrişimli Sinir Ağları *(Convolutional Neural Network)*
1DCNN: Bir Boyutlu Evrişimli Sinir Ağları *(One Dimensional Convolutional Neural Networks)*
KNN: K En Yakın Komşu *(K Nearest Neighbors)*
Adam: Uyarlanabilir Gerçek Zamanlı Tahmin *(Adaptive Moment Estimation)*
ReLU: Doğrultulmuş Lineer Birim *(Rectified Linear Unit)*
RMSProp: Kök Ortalama Kare Yayılımı *(Root Mean Square Propagation)*
IDS: Saldırı Tespit Sistemi *(Intrusion Detection System)*
XGBoost: Aşırı Gradyan Artırma *(eXtreme Gradient Boosting)*
OSS: Tek Taraflı Seçim *(One Sided Selection)*
SMOTE: Sentetik Azınlık Aşırı Örnekleme Tekniği *(Synthetic Minority Oversampling Technique)*

ŞEKİLLER LİSTESİ

ÇİZELGELER LİSTESİ

1. GİRİŞ

Yapay zekâ fikri ve makine öğrenimi alanındaki uygulamaları sayesinde yeni bir çağa adım atılmıştır. Tecrübe gerektiren birçok sorunun elimizde uygun veriler olduğunda modellenebilir olması tüm sektörlerde problemlerin çözümüne yardımcı olmaktadır. Bu yaklaşım, teknolojinin gelişmesiyle birlikte saldırı yüzeyinin artması neticesinde siber güvenlik dünyasında da yer bulmuştur.

Teknolojinin gelişmesi ve tehdit alanının büyümesinden dolayı ağlar üzerinde siber güvenlik olaylarının tespiti ve iyileştirmesi çalışmaları eylem planlarına girmiş durumdadır. Siber saldırıların sayısındaki artışlar düşünüldüğünde saldırı tespiti konusu oldukça önemli ve güncel bir konudur. Çalışmalar hem akademik olarak hem de sahada devam etmektedir.

Saldırı tespit sistemleri, ağ üzerindeki trafiği izleyerek güvenliği tehdit eden bir olay olup olmadığını tespit ederek bilgi üreten sistemlerdir. Anormallik tespiti, imza, kötüye kullanım, kural tabanlı yöntemleri kullanarak değerlendirme yapan saldırı tespit sistemleri mevcuttur (Baykara ve Daş, 2019). Güvenlik operasyon merkezlerinde değerlendirme süreçlerinde yapay zekâ ve makine öğrenimi yöntemleri entegrasyonunun sağlanması ile güvenlik sistemlerinin fazla miktarda üretmiş olduğu alarmların değerlendirilme sürecinin kolaylaştırılması, güvenlik analistlerinin iş yükünün ve yanlış pozitif durum tespitinin azaltılması amaçlanmaktadır.

Yapay zekâ ve makine öğreniminin siber güvenlik alanında kullanımına örnek olarak sunulan bu çalışmada saldırı türü ve anormallik durumu tespiti için “K En Yakın Komşu” ve derin öğrenme tekniklerinden “Bir Boyutlu Evrişimli Sinir Ağları” kullanılmıştır. Çalışma kapsamında CSE-CIC-IDS2018 veriseti kullanılmış, aşağıda tanımlanan saldırı türleri ile model eğitimleri sağlanmıştır (Kılınç ve Eyüpoğlu, 2023).

- DoS (Denial of Service – Servis Hizmet Reddi) saldırıları: Saldırgan tarafından kurban makinenin internet hizmetlerinin kesintiye uğratıldığı/yavaşlatıldığı, saldırının botnet şeklinde değil de tek bir cihaz ile makine ve sunucuların sorumluları tarafından erişilmez hale getirildiği saldırı tipidir. Kurban makineye karşılayabileceğinden çok fazla istek düştüğünde, makine normal trafiği karşılayamaz ve yavaşlar/hizmet kesintisi yaşanır.
- Botnet saldırısı: Saldırgan tarafından kullanıcılara ait kullanıcı adı ve parolaları ele geçirilmesi ile bir bot ağı oluşturularak komuta kontrol sunucusundan aynı anda DDoS, Brute Force gibi saldırıların gerçekleştirilmesidir.

- Sızma saldırısı (infiltration): Güvenli bir ağ üzerinden hassas bilgilerin ele geçirilmesini ifade eder. Saldırgan tarafından sistemdeki zafiyetler keşfedildikten sonra keşfedilen açıklıklar üzerinden ağdaki makinelerin tespit edilmesi, makineler üzerinde ayrıcalıklı hakların elde edilmesi ile çeşitli saldırıların gerçekleştirilmesidir.
- DDoS (Distrubuted Denial of Service – Dağıtılmış Hizmet Reddi) saldırıları: Ağ üzerinden sunulan bir hizmete, bir bot ağının oluşturulması ile taşıyabileceğinden daha fazla yük göndererek hizmet kesintisinin yaşatılmasıdır.
- Kaba Kuvvet (Brute Force) saldırıları: Sunulan hizmetlerde kullanıcı adı ve parolalarının elde edilmesi için olasılıkların denenmesi ile şahısların veya kurumların gizli bilgilerinin ele geçirilmesidir.
- SQL Injection saldırıları: Uygulamaların girdi alanlarında gerçekleştirilen sql sorgulamalar ile veritabanı sunucularına erişimin amaçlandığı saldırı türüdür.

Çalışma kapsamında aşağıdaki sorulara yanıt bulmaya çalışılmıştır:

- "K En Yakın Komşu" ve "Bir Boyutlu Evrişimli Sinir Ağları" modellerinin saldırı tespiti konusundaki başarısı nasıldır?
- Modeller hangi saldırı türlerinin tespit edilmesinde daha başarılıdır?
- Dengesiz verisetinin dengeli hale getirilmesi için kullanılan örneklem azaltma ve örneklem artırma tekniklerinin model başarısına olan etkisi nasıldır?
- Özellik seçiminin model başarısına olan etkisi nasıldır?

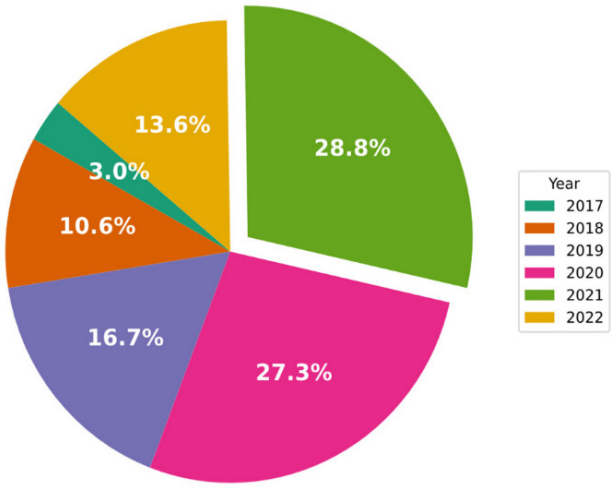
Sorular kapsamında yapmış olduğumuz bu çalışmada; 2. Bölüm "Kaynak Araştırması"nda, kullanmış olduğumuz CSE-CIC-IDS2018 veriseti ile modellere göre incelemesini yaptığımız literatür çalışması; 3. Bölüm "Materyal ve Yöntem" kısmında kullanılan metotlar hakkında açıklamalar yer almaktadır. 4. Bölüm "Araştırma Sonuçları ve Tartışma" başlığı altında çalışma ortamının hazırlanması, verisetinin hazırlanması, test edilecek modellerin oluşturulması ve performansının değerlendirilmesi işlemleri anlatılmıştır. Yapılan testler ile ilgili sonuçlar ise 5. Bölüm "Sonuçlar ve Öneriler" başlığı altında sunulmuştur.

2. KAYNAK ARAŞTIRMASI

Saldırı tespiti problemine çözüm bulabilmek için literatürde birçok çalışma bulunmaktadır. Bu bölümde derin öğrenme yöntemlerinin bir çeşidi olan evrişimli sinir ağları ve makine öğrenmesi tekniklerinden k en yakın komşu algoritmasının saldırı tespitinde kullanılması konusu ile ilgili literatür çalışması yer almaktadır.

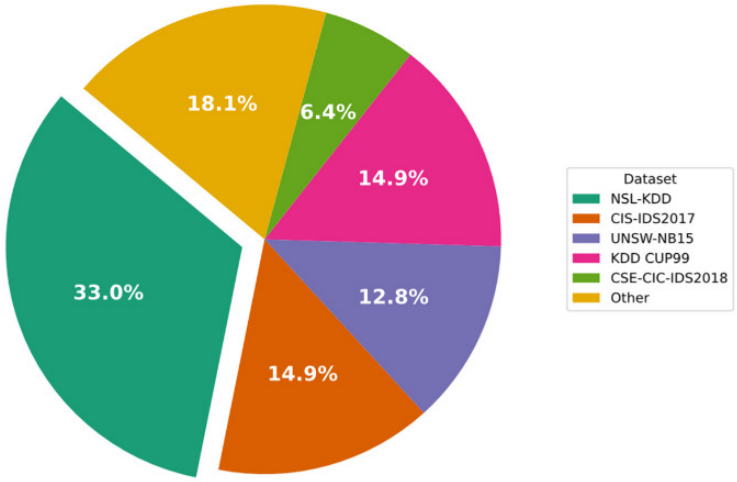
Osken ve ark. (2019) derin öğrenme ile saldırı tespit yöntemleri hakkında hazırlanan yayınların sistemli bir şekilde incelenebilmesi için sistematik haritalama çalışması yapılmıştır. Bu kapsamda son 10 yıla ait IEEE Explorer, ACM Digital Library, Science Direct, Scopus ve Wiley veritabanlarından 6088 makale incelenmiştir. Yayınlarda kullanılan derin öğrenme modelleri DNN, LSTM, RNN, DBN, CNN şeklinde sıralanmıştır.

Mohammadpour ve ark. (2022), bizim de konumuz olan saldırı tespitinde evrişimli sinir ağlarının kullanımı (CNN-IDS) hakkında bir araştırma makalesi hazırlamıştır. Makale kapsamında 2017 ve 2022 yılları arasındaki 66 adet CNN-IDS tabanlı makale incelenmiştir. Elde edilen sonuçlara göre saldırı tespit sistemlerinde evrişimli sinir ağları kullanımı Şekil 2.1'de sunulmuştur. Evrişimli sinir ağı modeli oluşturulurken giriş verisinin bir boyutlu veya iki boyutlu olarak tanımlanmasına göre hazırlanan çalışmaların dağılımında ise %36.2'sinin bir boyutlu, %63.2'sinin iki boyutlu olduğu görülmüştür.



Şekil 2.1. Saldırı tespit sistemlerinde CNN kullanımı yıllara göre dağılımı (Mohammadpour ve ark., 2022)

Evrişimli sinir ağı modeli oluşturulurken eğitim ve test aşamalarında kullanılan verisetine göre makalelerin dağılımı Şekil 2.2'de gösterilmiştir.



Şekil 2.2. Saldırı CNN-IDS yaklaşımında kullanılan veriseti dağılımı (Mohammadpour ve ark., 2022)

Gerçekleştirmiş olduğumuz çalışmada bir boyutlu evrişimli sinir ağı modeli CSE-CIC-IDS2018 veriseti kullanılarak eğitilip test edilmiştir. Benzer yöntemleri kullanarak yapılan çalışmalar incelendiğinde;

Lam (2021), CSE-CIC-IDS2018 verisetini kullanarak Bot, DoS ve HTTP saldırılarının tespitini sağladığı çalışmasında rastgele orman, çok katmanlı algılayıcı, bir boyutlu evrişimli sinir ağları modellerini test etmiştir. Evrişim katmanında 32, 39, 64 adet filtre, filtre boyutu 5, yığın boyutu 32 tanımlayarak 50 iterasyonda eğitimler gerçekleştirdiğinde 3 katmanlı model ile Conv1D(32,5), Conv1D(64,5), MaxPool(2), Conv1D(39,5), MaxPool(2), FC(), FC() katmanları ile sinir ağını oluşturarak accuracy, precision, racall, f1 değerlerini 99.986 olarak tahmin etmiştir.

Karataş (2020) tarafından gerçekleştirilen çalışmada BiRNN, BiLSTM, CNN, CNN-LSTM, DAE, GRU, LSTM ve RNN algoritmaları ile modeller oluşturulmuştur. Modeller CSE-CIC-IDS2018 veriseti kullanılarak eğitilmiştir. 1DCNN ile oluşturulan modelde filtre sayısı 128, çekirdek boyutu 3, aktivasyon fonksiyonu olarak ReLU kullanılmıştır. Conv1D, Maxpooling1D, Dropout ve 2 dense katmanı olmak üzere model oluşturulmuştur. Test doğruluğu 98.19, precision 99.54, recall 98.19, f1 değeri 98.86 olarak hesaplanmıştır. Saldırı türüne göre model doğruluğu incelendiğinde doğruluk oranları; benign 99.89, bot saldırıları 99.88, brute force 98.92, sızma saldırılarında 18.57 olarak tespit edilmiştir. Sentetik veri miktarı %16.2 oranında arttırıldıktan sonra örneklenmiş veri kümesiyle testler gerçekleştirildiğinde test doğruluğu 97.96, precision

99.26, recall 97.96, f1 değeri 98.61 olarak tahmin edilmiştir. Model doğruluğu benign 99.54, bot saldırıları 99.90, brute force 98.83, DoS 99.70, sızma saldırılarında 19.39, SQL injection saldırılarında 3.77 olarak saldırı tespiti sağlamıştır. Son olarak azınlık sınıfları aşırı örnekleme, çoğunluk sınıflarında boyut azaltma gerçekleştirilerek tüm sınıflar 286.191 olarak eşitlenmiştir. Bir boyutlu evrişimli sinir ağında test doğruluğu 89.89, precision 91.66, recall 89.89, f1 değeri 90.76 hesaplanmıştır. Model doğruluğu benign 70.26, bot saldırıları 99.89, brute force 98.25, DoS 99.06, sızma saldırılarında 91.27, SQL injection saldırılarında 100 olarak saldırı tespiti sağlamıştır. Diğer algoritmalarla kıyaslandığında BiLSTM en başarılı, DAE en başarısız sonucu vermiştir.

Qazi ve arkadaşları (2022), 1DCNN modelini kullandığı saldırı tespit çalışmasında CSE-CIC-IDS2017 veriseti ile eğitim ve test işlemlerini sağlayarak DoS Hulk, DoS GoldenEye, DDoS, Portscan ve zararsız trafiğin sınıflandırılmasını sağlamıştır. Eğitim doğruluğu 99.32, test doğruluğu 98.96, kesinlik 98.7, recall 99.2, f1 değeri 98.94 olarak tahmin edilmiştir.

Kilichev ve Kim (2023) tarafından gerçekleştirilen çalışmada 1DCNN modeli genetik algoritma (GA) ve parçacık sürü optimizasyonu (PSA) ile iyileştirilmiştir. Oluşturulan model CSE-CIC-IDS2017 veriseti ile eğitilmiştir. GA ile iyileştirilen testte test doğruluğu 99.71, precision 100, recall 99, f1 değeri 99 olarak tahmin sağlanmıştır. PSA ile iyileştirilen testte test doğruluğu 99.74, precision 100, recall 99, f1 değeri 100 olarak tahmin sağlanmıştır.

Karaman (2020) tarafından gerçekleştirilen çalışmada CSE-CIC-IDS2018 saldırı veri seti üzerinde Yapay Sinir Ağları (YSA), K-En Yakın Komşu (KNN), Lojistik Regresyon (LR), Destek Vektör Makineleri (DVM), Karar Ağacı (KA), Naive Bayes (NB) ve Rassal Orman (RO) algoritmaları uygulanmıştır. Karar Ağacı yöntemi anomali tespitinde (%99,97) ve saldırı türünün algılanmasında (%99,81) en başarılı algoritma olarak tespit edilmiştir. Saldırı tespitinde en belirleyici özellikler saptanarak veri seti boyutu azaltılmış, eğitim ve test zaman maliyetlerinin düşürülmesine çalışılmıştır.

Karaman ve ark. (2020) anomali tabanlı bir saldırı tespit sistemi ATSTS geliştirme çalışması yapmıştır. ATSTS'de yöntem olarak YSA kullanılmıştır. Sistemin eğitim ve test işlemlerinde CSE-CIC-IDS2018 veri seti kullanılmıştır. Tehdit türü olarak Botnet, DDoS, DOS, Brute Force saldırıları ele alınmıştır. Paket tehdit midir sorusunun cevabı %99.11 ile doğru bulunmuştur. Botnet %93.23, DDoS %99.31, DoS %92.26, Brute Force %99,26 ile doğru bulunmuştur.

Altunay ve Albayrak (2021) saldırı veri seti olarak CSE-CIC-IDS2018'in kullanıldığı çalışmada Brute Force, Sql Injection, Botnet ve DoS saldırıları incelenmiştir. SMOTE ile sentetik veri artırımı gerçekleştirmiştir. CNN mimarisi ile derin öğrenme modeli oluşturularak saldırıların tahmin edilmesi gerçekleştirilmiştir. Sentetik veri artırımının başarıyı arttırdığı gözlemlenmiştir.

Gerçekleştirmiş olduğumuz çalışmada bir boyutlu evrişimli sinir ağı (1DCNN), k en yakın komşu (KNN) algoritması ile kıyaslanmıştır. Saldırı tespitinde KNN yöntemini kullanan araştırmalar incelendiğinde:

Karataş (2020), KNN yöntemini çalıştığı testlerinde CSE-CIC-IDS2018 orijinal verisetini kullanarak 98.52 test doğruluğu, 99.28 precision, 98.52 recall, 98.89 f1 değeri elde etmiştir. Saldırı türüne göre doğruluk değerleri benign 99.75, bot 99.97, brute force 70.76, DoS 99.97, sızma saldırıları 36.16, sql injection saldırıları 3.96 olarak hesaplanmıştır. Sentetik veri arttırımı gerçekleştirildikten sonra test doğruluğu 98.08, precision 97.92, recall 98.08, f1 değeri 98 olarak elde etmiştir. Saldırı türüne göre doğruluk değerleri benign 99.75, bot 99.97, brute force 70.76, DoS 99.97, sızma saldırıları 36.16, sql injection saldırıları 3.96 olarak hesaplanmıştır.

Tuğrul ve Ahmed (2022) CSE-CIC-IDS2017 veri setinin kullanıldığı çalışmada KNN ile DNN kıyaslamıştır. KNN yöntemi doğruluk değeri 90.913, recall değeri 91.283, precision 90.302 olarak hesaplanmıştır (Atefi ve ark., 2019). Tuğrul ve Ahmed (2022) tarafından gerçekleştirilen çalışmada CSE-CIC-IDS2017 veri seti alt örneklenerek benign, DoS Hulk, port tarama ve DDoS saldırıları sınıflandırılmıştır. KNN modelinde 10 kat çapraz doğrulama uygulanarak doğruluk 99.87; precision, recall, f1 değerleri 0.99 olarak tespit edilmiştir. PCA uygulanarak boyut küçültüldüğünde doğruluk 99.88 olarak tespit edilmiştir.

3. MATERYAL VE YÖNTEM

3.1. Makine Öğrenmesi

Yapay zeka, en genel anlamı ile insana ait düşünme, öğrenme, tanıma, karar verme, geçmişten ders çıkarma gibi yetilerin makinelere kazandırılmasını amaçlayan bilimdir. Bu bilimin alt kümesi olan makine öğrenmesi, makinelerin matematiksel modeller sayesinde veri analizleri gerçekleştirerek öğrenmesini sağlayan bir teknolojidir.

Makine öğrenmesi sayesinde matematiksel ve istatistiksel modeller oluşturularak verilerin sınıflandırılması ve veriler üzerinden çıkarımda bulunularak doğru tahminler üreten sistemler geliştirilmesi sağlanır. Makine öğrenmesinde problemin amacına göre regresyon, sınıflandırma, topluluk, ilişkilendirme, kümeleme algoritmaları kullanılmaktadır. Regresyon algoritmalarında bağımsız değişkenin (etken), bağımlı değişkeni (çıktı) ne kadar etkilediği sorgulanır. Sınıflandırma algoritmalarında kategorize edilmiş çıktılardan hareketle girdinin hangi sınıfa ait olduğu tahmin edilir.

Bizim problemimiz olan saldırı veri setinden anomali durumunun ve saldırı kategorisinin tahmin edilmesi durumunda ise makine öğrenmesi modellerinden olan sınıflama ve regresyon problemlerinde kullanılan K-En Yakın Komşu (KNN) ile insandaki sinir sistemini örnek alan derin öğrenme modellerinden olan Evrişimsel Sinir Ağları (ESA) algoritması kullanılmıştır.

3.1.1. K En Yakın Komşu (KNN-K Nearest Neighbors) Algoritması

K En Yakın Komşu (KNN) algoritması kümeleme ve sınıflandırma problemlerinde kullanılan makine öğrenmesi yöntemlerindendir. KNN yöntemi verinin hangi sınıfa ait olduğunun tahmin edilebilmesi için eğitim setindeki verilerin bağımsız değişkenlerinden oluşan vektöre en yakın konumda bulunduğu noktaya uzaklığının ölçülerek sınıfının belirlenmesidir. Uzaklık ölçülürken mesafe hesaplama yöntemi olarak genellikle "Öklid uzaklığı" yöntemi kullanılmakla birlikte Manhattan ve Minkowski uzaklık ölçüm formülleri de kullanılmaktadır (Hacıbeyoğlu ve ark., 2023).

KNN algoritmasında kullanılan "k" parametresi ise sınıfı bilinmeyen verinin sınıfı tahmin edilirken gruptaki kaç tane veri ile yakınlığının ölçüleceğini belirtmektedir. "k" parametresi için genellikle "3", "5" ve "7" değerleri tercih edilmektedir (Akyel ve

Seçkin, 2012). Bu nedenle çalışmamızda k değerini belirten "n_neighbors" parametresinde 3 ve 5 değerleri kullanılmıştır.

3.1.2. Aşırı Gradyan Artırma (XGBoost-Extreme Gradient Boosting) Algoritması

Çalışmamızda öznitelik seçimi ile önemli özellikler belirlenirken büyük veri setlerinde kullanım kolaylığı sebebiyle tercih edilen XGBoost algoritması kullanılmıştır. XGBoost algoritması ilk olarak "XGBoost: A Scalable Tree Boosting System" adlı makalede Chen ve Guastrin (2016) tarafından sunulmuştur (Chen ve Guestrin, 2016). Topluluk öğrenmesi (ensemble learning) yaklaşımları çerçevelerinden gradient boosting yaklaşımına dayanan algoritma karar ağacı yapısını temel almaktadır.

Topluluk öğrenmesi yaklaşımı eğitim verisinin tek bir öğrenici model ile değil de n adet öğrenici model ile gerçekleştirilmesi, son olarak sonuçların birlikte değerlendirilerek oylama ile tahminin gerçekleştirilmesidir. Stacking (istifleme, yığma), bagging (torbalama) ve boosting (güçlendirme) olmak üzere 3 mimariden oluşmaktadır. Yığma, karar ağacı, KNN gibi birden fazla modelden elde ettiği tahminleri kullanarak değerlendirme yapan topluluk öğrenmesi yaklaşımıdır. Torbalama, modellerin paralel olarak konumlandırıldığı, elde edilen tahminlerin değerlendirmesi yapılarak sonuç üretildiği yaklaşımdır. Güçlendirme (boosting) mantığında ise öğrenici model daha iyi tahmin gerçekleştirebilmek için hataya odaklanır, hatalardan faydalanarak her seçimde öğreniciyi geliştirerek daha iyi tahminin gerçekleştirilmesini sağlar. Güçlendirmede veri kümesi seri olarak işlenmektedir (Ataş ve Alhajahmad, 2023). Güçlendirme algoritmalarından olan gradient boosting algoritması da benzer şekilde hatayı minimize etmek için gradyan iniş (gradient descent) yöntemini kullanarak veri kümesini işlemektedir.

Gradient boosting algoritmasının gelişmiş bir hali olan XGBoost algoritmasında benzer şekilde karar ağacı sürekli geliştirilerek yeni ağaçlar oluşturulmakta ve en iyi tahmin seçilmektedir. Gradient boosting algoritmasından gelişmiş olarak XGBoost algoritmasında düzenlileştirme ile ağaç aşağı doğru çok fazla ilerleme gerçekleştirirse budama gerçekleştirilerek aşırı uyumun önlenmesi, paralelleştirme yapılarak hesaplama hızının yükseltilmesi, çapraz doğrulamanın gerçekleştirilerek en iyi performansın üretildiği yinelemenin tespit edilmesi, eksik değerler ile çalışabilmesi, out-of-core özelliği sayesinde hesaplama işlemlerini önbellekte gerçekleştirerek büyük veri setlerinde

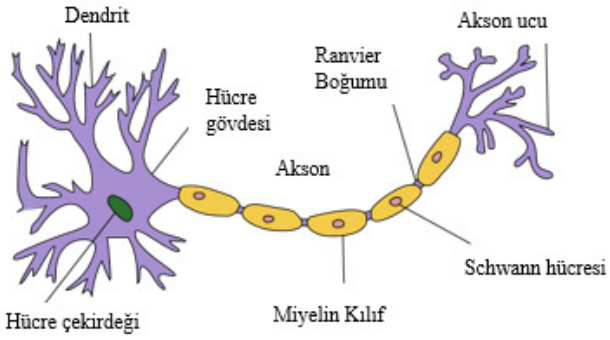
işlem hızı sağlaması gibi özellikleri mevcuttur (Chen ve Guestrin, 2016; Yu ve ark., 2019).

XGBoost algoritması veri değerlendirme işlemini veriyi parçalara ayırarak sağlamaktadır. XGBoost algoritması avantajlarına ek olarak; tahminin doğru olarak tespit edilebilmesi için performansın en iyi sağlandığı parçaların tespit edilmesi, parçalar içindeki özelliklerin sıralanması, oluşturulacak ağaçların belirlenmesi işlemlerinin verimli olarak sağlanması için "Sketch" algoritması kullanılarak özellik seçimi sağlanmaktadır (Chen ve Guestrin, 2016). Gerçekleştirmiş olduğumuz çalışmada XGBoost algoritması özellik seçimi için kullanılmıştır. Yu ve arkadaşlarının (2019) biyoinformatik alanında gerçekleştirdiği çalışmada da bu özellikten faydalanılarak 2325 adet olan özellik sayısı 117 özelliğe indirgenerek 0.797 olan doğruluk değeri 0.883 olarak tahmin edilmiştir. (Yu ve ark., 2019)

3.2. Derin Öğrenme

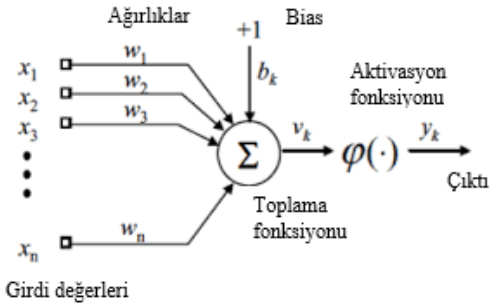
Çalışmamızda kullandığımız derin öğrenme yöntemi makine öğrenmesinin alt kümesidir ve insandaki sinir sisteminden ilham alan yapay sinir ağları algoritmasının çok seviyeli, derin yaklaşımıdır.

Biyolojik olarak sinir hücresinde bilgi transferi, dendrit adı verilen sinir uçlarının uyarıları algılaması ile başlar. Alınan sinyaller hücre gövdesinde toplanır. Hücre gövdesinde çekirdek ve hücrenin yaşamsal faaliyetlerinin yürütülmesini sağlayan mekanizma bulunur. Sinir hücresi gövdesindeki elektriksel uyarılar akson aracılığı ile akson uçlarına taşınır. Bir nöron ile diğer nöron arasında veri iletiminin olduğu kısım akson uçları ile diğer nöronun hücre gövdesi arasında birleşme yeri olan sinapslardır. Aksonun taşıdığı bu uyarılara aksiyon potansiyeli denir. Aksiyon potansiyeli bütün aksonlar için aynı ölçüde ve hızdadır. Aksiyon potansiyeli, yumru şeklindeki akson uçlarına geldiğinde nörotransmitter adı verilen kimyasallar salgılanır. Nörotransmitter, uyarıyı diğer nörona iletir veya engeller. Biyolojik olarak tipik bir sinir hücresinin yapısı Şekil 3.1'de verilmiştir.



Şekil 3.1. Tipik bir sinir hücresinin yapısı (Sinir hücresi - Vikipedi, t.y.)

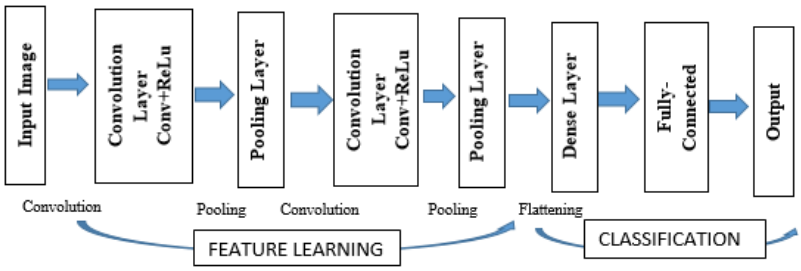
Biyolojik bir sinir hücresini (nöron) örnek alan yapay bir sinir hücresi (perceptron) modelinde ise aksondan nörona gelen sinyaller girdi değerleridir. Girdi değerleri sinapsis noktalarında dendritlerdeki ağırlıklarla çarpılarak hücre gövdesine gelir. Hücre gövdesinde toplama fonksiyonu ile ağırlıklı toplama işlemi gerçekleştirilir. Bias değeri eklendikten sonra aktivasyon fonksiyonuna girer ve bilgi çıkışı sağlanır. Bu değere skor fonksiyonu da denir. Aynı katmanda bulunan nöronlar arasında ilişki bulunmamaktadır. Farklı katmanlar arasındaki nöronlar ise birbirini aktivasyon fonksiyonu ile etkiler ve nörondan elde edilen çıktılar standardize edilir. Aktivasyon fonksiyonu olmazsa yapay sinir ağlarına gerçek dünya nitelikleri aktarılmaz ve çıkış fonksiyonu basit doğrusal bir fonksiyon olur. Yapay bir sinir hücresinin matematiksel modeli Şekil 3.2'de sunulmuştur. Sinir ağının modellenmesinde etkili olan işlem ağırlık ve bias değerlerinin doğru şekilde tanımlanmasıdır. Ağırlık değerleri ilk kez tanımlanabileceği gibi daha önceden eğitilmiş bir modele ait ağırlıkları da kullanmak mümkündür. Modelin performansını etkileyen diğer bir unsur kayıp fonksiyonudur (loss function). Kayıp fonksiyonunun çeşitli optimizasyon teknikleri ile 0'a yaklaşması beklenir. Bu değer eğitim ve test gruplarının karşılaştırılması ile elde edilir (Kızrak, 2018; Gulsen, 2021).



Şekil 3.2. Bir yapay sinir hücresinin matematiksel modeli

3.2.1. Evrişimli Sinir Ağları / Convolutional Neural Networks (ESA / CNN)

Evrişimli sinir ağları, çok katmanlı yapay sinir ağlarının bir türüdür. Derin öğrenme bilimine ait temel mimari olarak kabul edilir. CNN'de giriş verileri alındıktan sonra katman katman eğitim gerçekleştirilir. Üretilen sonuç ile istenen sonuç arasındaki fark yani hata değeri geri yayılım algoritması (backpropagation) ile bütün ağırlıklar üzerinde dağıtılır. Her epoch işleminde ağırlıklar azaltılarak hata giderilmeye çalışılır (İnik ve Ülker, 2017; Sert, 2020). Tipik bir evrişimli sinir ağı modeli Şekil 3.3'te gösterilmiştir.



Şekil 3.3. Tipik bir CNN mimarisi örneği

Evrişim (convolution) katmanında resim üzerindeki düşük ve yüksek seviyeli özelliklerin çıkarılabilmesi için çok boyutlu filtreler uygulanarak özellik haritası (feature map) çıkarılır. Evrişim işleminin ardından ReLU aktivasyon fonksiyonu uygulanarak görüntü matrisi üzerindeki negatif değerler sıfırlanır. Burada farklı bir aktivasyon fonksiyonu da tercih edilebilir fakat performans olarak ReLu daha iyi sonuç verdiği için bu yöntem tercih edilmektedir. Pooling (downsampling) katmanı convolution katmanları arasına çoğunluk olarak eklenen bir katmandır. Bu katman sinir ağının doğru olarak karar verebilmesi için boyutları azaltarak önemli bilginin edinilmesini sağlar. Filtre kayma

işleminde kapsadığı alan içindeki en büyük sayıyı aldığında max pooling, ortalamasını aldığında average pooling işlemi gerçekleştirilir.

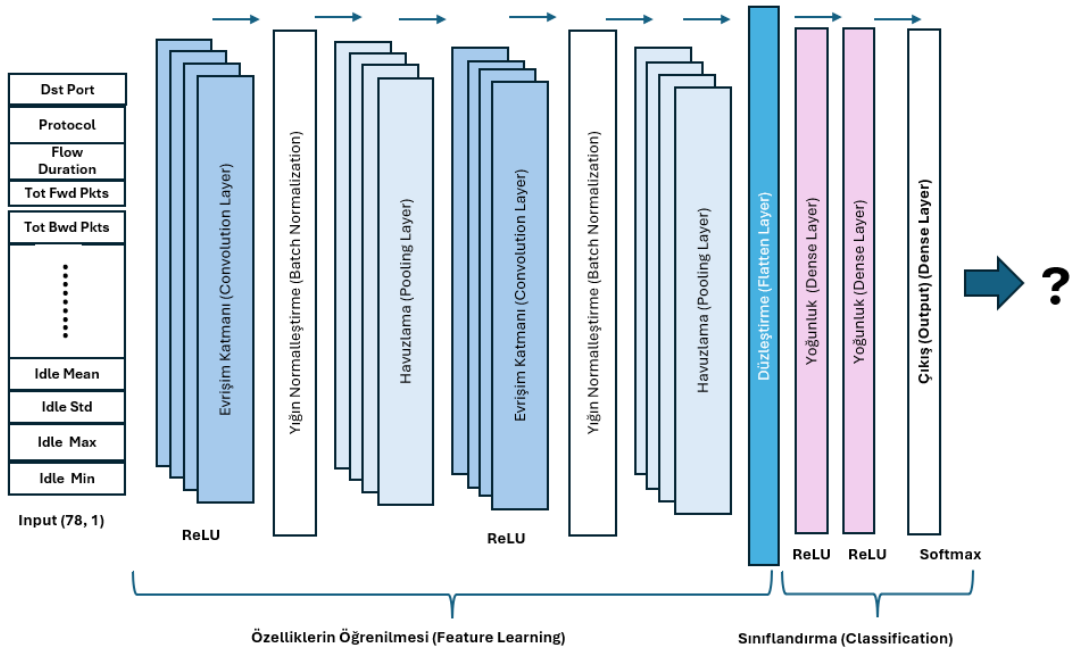
Özelliklerin çıkarılması işleminden sonra sınıflandırma işleminin gerçekleştirilmesi için convolution ve pooling katmanından gelen matrisler flattening (düzleştirme) katmanına gelir ve tek boyutlu dizi haline dönüştürülür.

Fully-connected katmanı öğrenmenin gerçekleştiği katmandır. Çok katmanlı sinir ağı yapısına benzerdir. Çıktılar sınıflandırmanın sağlanması için aktivasyon fonksiyonuna verilir. Nesnelerin sınıflandırılması sağlanır (Amidi ve Amidi; Şeker ve ark., 2017; Prabhu, 2018).

Evrişimli sinir ağları görüntü işleme ve tanıma alanında kullanımıyla bilinir. Buraya kadar anlattığımız kısımda evrişimli sinir ağlarının görüntü kümelerinin sınıflandırılmasında yani iki boyutlu verilerde kullanımına değindik. Evrişimli sinir ağlarının bir boyutlu verilerde ve üç boyutlu verilerde kullanımı da mümkündür. Üç boyutlu evrişimli sinir ağları (3DCNN), video görüntülerinden nesnelerin üç boyutlu olarak algılanmasını sağlar, nesne eşleştirme gibi robotik uygulamalarda, tıbbi görüntülemede tercih edilir. Bir boyutlu evrişimli sinir ağları (1DCNN) ise zaman serilerinin sınıflandırılması, sinyal analizi, doğal dil işleme, ses ve metin gibi sıralı verilerin analizinde tercih edilmektedir. (Kiranyaz ve ark., 2021)

3.2.1.1. Bir Boyutlu Evrişimli Sinir Ağları (1DCNN)

Saldırı veriseti özniteliklerinden elde edilen özellik vektörünün bir boyutlu matris görüntüsünde olduğu ve zamansal olarak sıralı özellik vektörlerinden oluştuğu düşünülerek bir boyutlu evrişimli sinir ağları kullanılmıştır. Kullanmış olduğumuz 1DCNN yapısı Şekil 3.4'te sunulmuştur.



Şekil 3.4. Çalışmada kullanılan bir boyutlu evrişimli sinir ağı modeli

Yapmış olduğumuz çalışmada sinir ağı oluşturulurken katmanların sıralı olarak eklenebilmesi için Keras'ın "Sequential" modeli eklenmiştir. Özelliklerin çıkarılması aşamasında evrişim katmanında metin girdisine uygun olması için "Conv1D" katmanı eklenmiştir, evrişimde kullanılacak filtre boyutu, kaç adet filtre kullanılacağı, "input_shape" denilen girdi boyutu tanımlanmıştır. "input_shape" tanımlanırken (time_steps, input_dim) şeklinde girdi oluşturulur; burada time_steps eğitim verisindeki özelliklerin sayısı, input_dim girdinin boyut bilgisidir. Çıktıların normalleştirilmesi ve ağın daha hızlı eğitilebilmesi için "BatchNormalization", boyut azaltarak hesaplama kolaylığı sağlaması için "MaxPooling1D" katmanı eklenmiştir. Modele "Flatten" katmanı uygulanmıştır. Ardından Fully-Connected katmanı için Keras'ın "Dense" yoğunluk katmanları eklenerek model oluşturulmuştur. Her bir katmana Keras'ın "activition" özelliği eklenerek kullanacağımız aktivasyon fonksiyonları belirtilmiştir. (Şekil 3.4) (Şekil 3.5)

```
Layer (type)                 Output Shape              Param #
=================================================================
conv1d (Conv1D)              (None, 78, 64)            448

batch_normalization (BatchN  (None, 78, 64)            256
ormalization)

max_pooling1d (MaxPooling1D  (None, 39, 64)            0
)

conv1d_1 (Conv1D)            (None, 39, 64)            24640

batch_normalization_1 (Batc  (None, 39, 64)            256
hNormalization)

max_pooling1d_1 (MaxPooling  (None, 20, 64)            0
1D)

flatten (Flatten)            (None, 1280)              0

dense (Dense)                (None, 128)               163968

dense_1 (Dense)              (None, 64)                8256

dense_2 (Dense)              (None, 5)                 325

=================================================================
Total params: 198,149
Trainable params: 197,893
Non-trainable params: 256
_________________________________________________________________
```

Şekil 3.5. Eğitim için hazırlanan model katmanları örneği

Bir boyutlu evrişimli sinir ağları modeli oluşturulurken model başarısının arttırılması için hiperparametre ayarlamaları gerçekleştirilir. Kilichev ve Kim (2023) tarafından gerçekleştirilen 1DCNN tabanlı saldırı tespit sisteminde GA (Genetik Algoritma) ve PSO (Parçacık Sürü Optimizasyonu) kullanılarak hiperparametre optimizasyonunun sağlanması ile ilgili makalede önerilen yöntemde iyileştirme ile ilgili her iki yöntemde de ortak parametreler elde edilmiştir (Kilichev ve Kim, 2023). (Çizelge 3.1)

Çizelge 3.1. Önerilen yöntemde sunulan hiperparametre ayarları (Kilichev ve Kim, 2023)

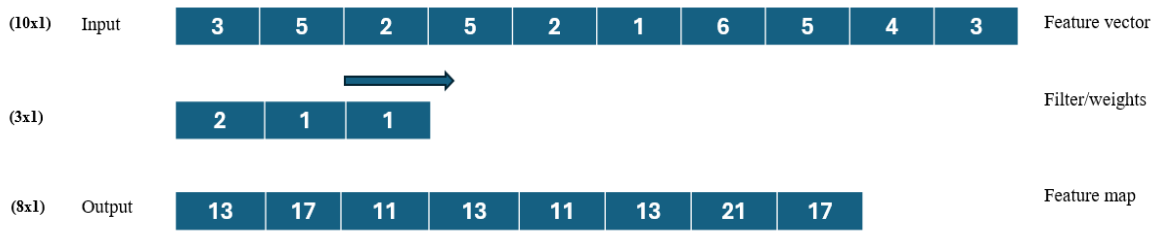
Yazar	Model	Yöntem	Hiperparametre
Kim ve Kilichev	1DCNN	Genetik Algoritma ve Parçacık Sürü Optimizasyonu	Batch size
			Dropout rate
			Kernel size
			Learning rate
			Number of dense layers
			Number of epochs
			Number of filters
			Number of neurons in dense layers
			Pooling size

1DCNN yapısında kullanılan hiperparametrelerin açıklaması aşağıda sunulmuştur:

- Katman sayısı (Number of layers): Sinir ağının derinliğini yani kaç katmandan oluştuğu bilgisidir.
- Filtre sayısı (Number of filters): Evrişim (konvolüsyon) katmanının kaç filtreden oluştuğu bilgisidir.
- Adım sayısı (Stride number): Evrişim (konvolüsyon) işlemi esnasında kaç adımda bir evrişimin gerçekleşeceğini belirtir.
- Piksel ekleme (Padding): Dolgu, evrişim işleminden sonra girdi ve çıktı matrisinin eşit olmasını sağlayarak bilginin kaybolmamasını sağlayan parametredir. Padding parametresi valid ve same değerlerini alır. "valid" olduğunda varsayılan olarak dolgu olamdığını ifade eder. "same" olduğunda ise girdi ve çıktının eşit olması için boşlukları 0 ile doldurur.
- Çekirdek boyutu (Kernel size): Evrişim işleminde kullanılan filtrenin uzunluğunu ifade eder.
- Havuz boyutu (Pool size): Havuzlama katmanında tanımlanır, önemli değerleri alarak boyut azaltmayı sağlar.

Evrişim işleminde özellik vektörü ve filtrenin adım adım ilerletilmesi ile konvolüsyon işlemi gerçekleştirilerek özellik haritası çıkartılır. (Şekil 3.6) Girdi ve çıktı arasındaki ilişki 3.1'de gösterilmiştir. Formülde W, özellik vektörü girdi boyutunu, F filtre, çekirdek boyutunu (kernel size), P eklenecek piksel boyutunu (padding), S filtrenin kaydırılması işleminde adım sayısını, p_s havuz boyutunu, O çıkış boyutunu ifade etmektedir.

$$O = \frac{W - F + 2P}{S} + 1 \tag{3.1}$$

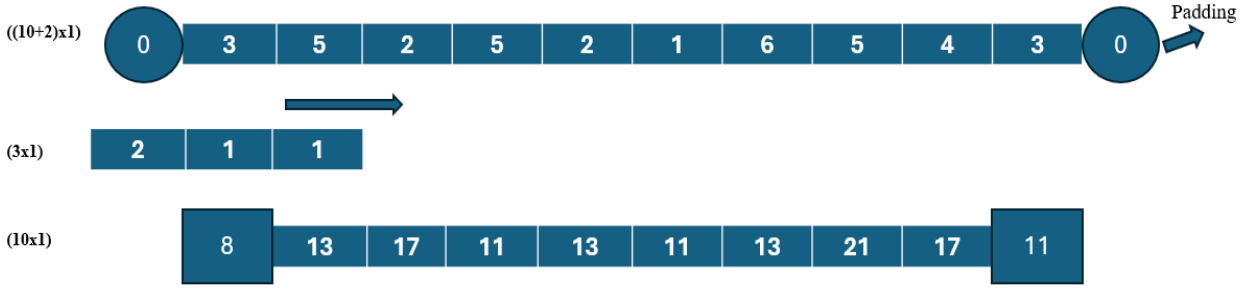


Şekil 3.6. Evrişim (convolution) işleminin gerçekleştirilmesi

Şekil 3.6'da 10 boyutlu girdi vektörü ile 3 boyutlu filtre evrişiminde 8 boyutlu çıkış vektörü elde edilerek boyutun azaldığı görülmektedir. Şekil 3.7'de padding eklenerek özelliklerin kaybolmaması, evrişim işlemi sonuç vektörünün girdi vektörü ile eşit boyutlu olması sağlanmıştır. Piksel ekleme özelliği "same" olarak tanımlandığında çıkış boyutu denklem 3.2'de olduğu gibi belirlenir. "valid" olarak tanımlandığında ise denklem 3.3'te olduğu gibi belirlenir.

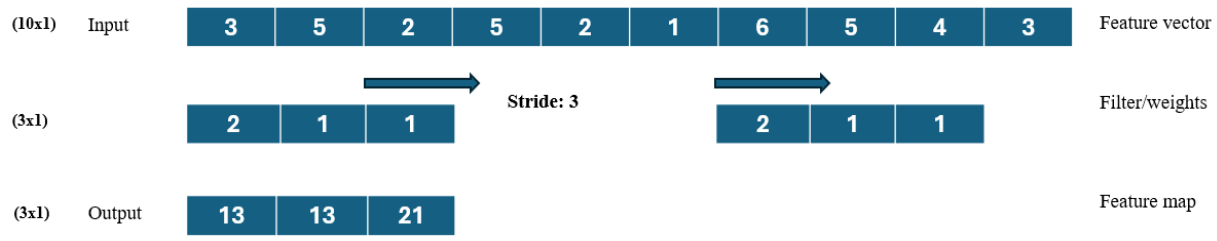
$$O = \frac{W - 1}{S} + 1 \tag{3.2}$$

$$O = \frac{W - (p\_s)}{S} + 1 \tag{3.3}$$



Şekil 3.7. Evrişim işleminde piksel ekleme, dolgu (padding) kavramı

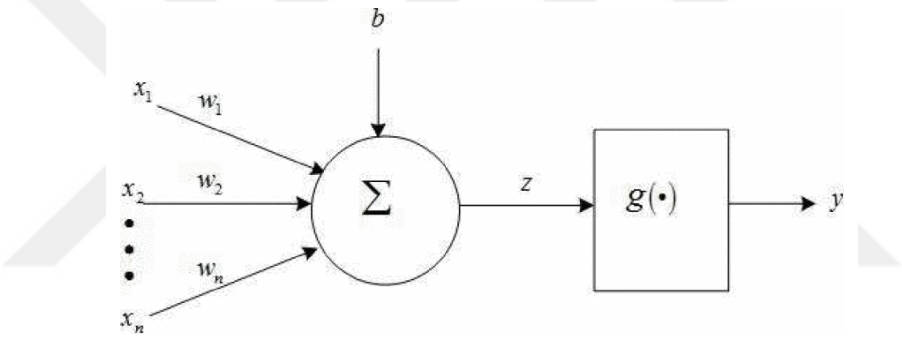
Şekil 3.8'de filtrenin adım sayısı (stride) 3 olarak evrişim gerçekleştirildiğinde özellik haritası boyutunun azaldığı görülmektedir. Adım sayısının arttırılması ile evrişim işlemi hızlanır ancak bu işlem özelliklerin kaybolmasına neden olur.



Şekil 3.8. Evrişim işleminde adım sayısı (stride) kavramı

3.2.2. Aktivasyon Fonksiyonları

Bir sinir ağının verilerdeki karmaşık yapıları öğrenebilmesi için ağın uyarılması, etkinleştirme işleminin gerçekleştirilmesi gereklidir. Biyolojik olarak örneklersek, nöronlar arasında bilgi iletiminde nörona hangi bilginin ateşleneceğini belirlemekten aktivasyon fonksiyonu sorumludur. Bir yapay sinir ağı modelinde nöronların giriş değerleri ile ağırlıklar çarpılır ve bias eklenerek çıktı elde edilir. Aktivasyon fonksiyonları üretilen çıktıların doğrusal yapıdan uzaklaştırılarak ve gerçeğe yakın olarak nasıl standardize edilebileceğini, nasıl bir değişimden geçmesi gerektiğini belirler. Aktivasyon fonksiyonu olmadığında ise sinir ağı modeli linear regression gibi davranır, basit durumlar için öğrenme yetisi olsa da video, görüntü, ses gibi karmaşık dünya verilerini öğrenme konusunda yetersiz kalır. İleri beslenme adımında fonksiyonun kendisi ile tahmini sağlarken geri yayılımda fonksiyonun türevi ile öğrenme sağlanır (Karakuş). Şekil 3.9'da g(.) işlevi aktivasyon fonksiyonunu ifade etmektedir.



Şekil 3.9. Aktivasyon fonksiyonu modelinin gösterilmesi (Ding et al., 2018)

Genel olarak derin öğrenmede kullanılan aktivasyon fonksiyonu çeşitleri basamak (step), doğrusal (linear), sigmoid, hiperbolik tanjant (tanh), softsign, relu (rectified linear unit), elu (exponential linear unit), leaky relu, swish (a self-gated/kendinden geçitli) ve softmax fonksiyonu olarak bilinmektedir.

Modelimizde ReLU ve softmax aktivasyon fonksiyonları kullanılmıştır. Kullanılacak aktivasyon fonksiyonunun belirlenmesinde doğrusal olması, türevlenebilir olması, alt ve üst sınırlarının olması, monoton olarak artan ve azalan olması ve orijin noktasında fonksiyonun kendisine yakınsaması nitelikleri olumlu olarak etkilidir. Kullanacağımız fonksiyonları belirlerken bu kriterlere dikkat ederek seçim yapılmıştır.

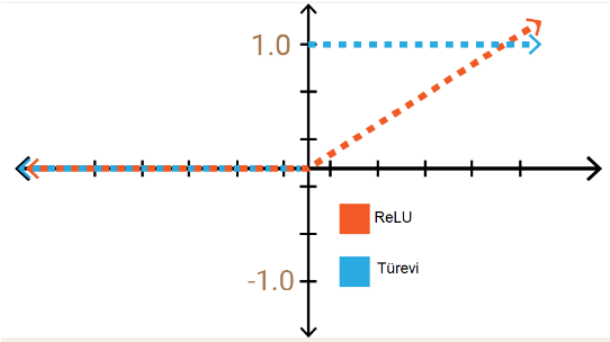
Fonksiyon türevi "0" olduğunda, geri yayılım esnasında parametreler güncellenmediği için öğrenme gerçekleşmez. Bu problem dying ReLU olarak

adlandırılır. Bu sorunla karşılaşmamak için Leaky ReLU fonksiyonu geliştirilmiştir. Öğrenmenin tüm değerler için gerçekleşmesi sağlanmıştır (Kurt ve Efe, 2018).

Derin öğrenme modelimizde çıkış fonksiyonu olarak "softmax" kullanılmıştır. Bu aktivasyon fonksiyonu sayesinde "0" ile "1" aralığında olasılıksal olarak loss (kayıp) değeri üretilerek girdilerin hangi sınıfa ait olduğu belirlenir. Softmax, loss değerini cross entropy olarak hesaplar (Kızrak, 2019).

3.2.2.1. ReLU (Rectified Linear Unit) Fonksiyonu

"Doğrultulmuş Lineer Birim" fonksiyonu, ilk olarak 2012 yılında literature girmiştir ve performans olarak iyi sonuç verdiği için yaygın olarak kullanımı devam etmektedir (Kılıçarslan ve Adem, 2021). Fonksiyon negatif girişler için sıfır, pozitif değerler için doğrudan çıkış değerini üretir. Bu eşikleme işlemi fonksiyona hız kazandırır ancak çıktının sıfır olduğu bölgelerde türev de sıfır olduğu için gradyanların ölmesi durumu ReLU için de geçerlidir (Karakuş). ReLU fonksiyonu Şekil 3.10'da gösterilmiştir. Fonksiyona ait denklemler 3.4, 3.5 ve 3.6'da verilmiştir (Ding ve ark., 2018).



Şekil 3.10. ReLU fonksiyonu (Kızrak, 2019)

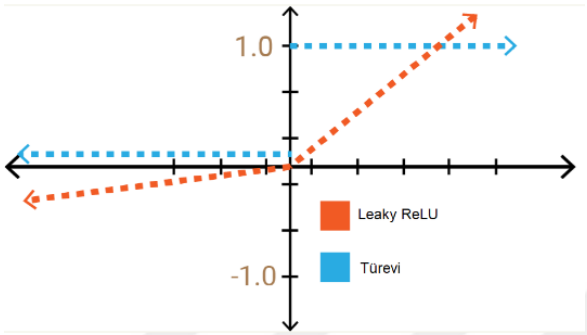
$$R(z) = \begin{cases} 0, & z \leq 0 \\ z, & z > 0 \end{cases} \quad (3.4)$$

$$R(\acute{z}) = \begin{cases} 0, & z \leq 0 \\ 1, & z > 0 \end{cases} \quad (3.5)$$

$$\acute{R}(z) = \begin{cases} a.(e^z), & z \leq 0 \\ 1, & z > 0 \end{cases} \quad (3.6)$$

3.2.2.2. Leaky ReLU Fonksiyonu

"Sızıntı Doğrultulmuş Lineer Birim" fonksiyonu ReLU fonksiyonu çeşitlerindendir. Bu fonksiyon ReLU'da sıfır çıktı veren negatif bölgelerde sıfırdan farklı sıfıra yakın bir çıktı üreterek gradyanların kaybolması probleminin önüne geçer. Leaky ReLU fonksiyonu Şekil 3.11'de verilmiştir. Fonksiyona ait denklemler (3.7) ve (3.8)'de verilmiştir.



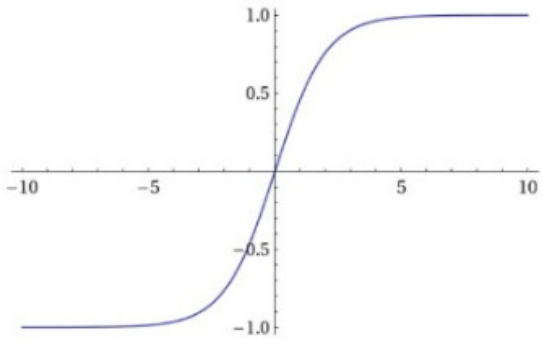
Şekil 3.11. Leaky ReLU fonksiyonu (Kızrak, 2019)

$$R(z) = \begin{cases} az, & z \leq 0 \\ z, & z > 0 \end{cases} \quad (3.7)$$

$$R(\acute{z}) = \begin{cases} a, & z \leq 0 \\ 1, & z > 0 \end{cases} \quad (3.8)$$

3.2.2.3. Softmax Fonksiyonu

Softmax fonksiyonu, genellikle sınıflandırma amacıyla sinir ağı modelinin çıkış katmanında kullanılır. Sigmoid fonksiyonu ile benzer yapıya sahiptir. Sigmoid fonksiyonu iki sınıflı değerlendirme yaparken softmax çok sınıflı modellerde 0 ile 1 arası olasılık değerleri üreterek girdilerin ait olabileceği sınıfın tahmin edilmesini sağlar (Kızrak, 2019). Softmax fonksiyonu Şekil 3.12'de verilmiştir. Fonksiyon denklemi (3.9)'dadır.



Şekil 3.12. Softmax fonksiyonu (Ayten, 2021)

$$\sigma(z)_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \quad i=1,\ldots, K \quad z=z_1, z_2 \ldots \ldots z_K \tag{3.9}$$

Çıkış değerleri denklem (3.9)'da belirtilen denkleme tabi tutularak olasılık değerlerim belirlenir.

3.2.3. Kayıp (Maliyet-Loss) Fonksiyonları

Loss fonksiyonu tasarlanan modelin hata oranını aynı zamanda başarımını ölçen fonksiyondur. Kayıp fonksiyonları gerçek değer ile modelin tahmin ettiği değer arasındaki farkı ifade eder. Kayıp ne kadar az olursa tahmin o kadar doğru gerçekleşmiş olur. Bu sayede hatalardan öğrenme işlemi gerçekleştirilir. Problem tipine göre kullanılacak fonksiyona karar verilir. Sinir ağı modelinde hatayı belirtmek için kullanılan fonksiyonlar Cross-Entropy, Hinge, Kullback-Leibler (KL), Iraksama (Göreceli Entropi), Ortalama Mutlak Hata (MAE - L1 Loss) ve Ortalama Kare Hata (MSE) (Quadratic Loss-L2 Loss) şeklinde bilinmektedir (Koşan ve ark., 2019). Hazırlamış olduğumuz modelde saldırıların sınıflandırılabilmesi için "categorical cross entropy" kullanılmıştır.

3.2.3.1. Cross-Entropy (Çapraz Entropi)

Bilgi teorisinde entropi, bir olasılık dağılımından rasgele bir olayın olma olasılığının hesaplanabilmesi için bilgi ölçümüdür ve ölçümü bit sayısı olarak ifade eder. Bir olayın olma olasılığı yüksek ise daha az şaşırtıcıdır ve düşük bilgiye sahiptir, olay herşeyin eşit olarak dağılımı gibi düşük olasılıklı bir olay ise çok şaşırtıcıdır ve yüksek bilgiye sahiptir. Sonuç olarak gerçekleşmesi beklenmeyen nadir olaylar daha

bilgilendiricidir ve rasgele bir x değeri için bilginin hesaplanması, x değişkeninine ait olayların olasılık dağılımını hesaplamak ile aynıdır (Brownlee, 2019b).

Çapraz entropi ise entropi temeline dayanan belirli bir rasgele değişken ya da olay kümesi için iki olasılık dağılımı arasındaki farkın ölçülmesini sağlayan yöntemlerden biridir. KL sapması iki olasılık dağılımı arasındaki göreceli entropiyi hesaplarken, çapraz entropi dağılımlar arasındaki toplam entropiyi hesaplar. KL sapması ile entropi değeri toplandığında çapraz entropi değeri elde edilir. Çapraz entropi bir olayı P yerine Q ile temsil etmek istersek gereken ortalama bit sayısını ifade eder (Brownlee, 2019a).

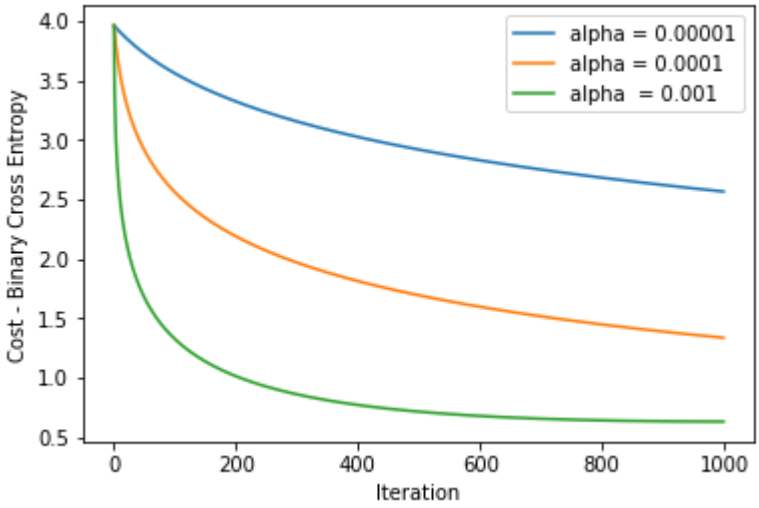
H(P, Q) çapraz entropiyi ifade eder, burada P hedef dağılım, gerçek dağılımı ifade ederken Q hedef dağılımın yaklaşık değeridir yani tahmin değeridir. Tahmin değeri, gerçek değerinden uzaklaştıkça çapraz entropi değeri artar (Brownlee, 2019a).

Aşağıdaki denklemde (3.16) P durumunda x olayının olma olasılığı P(x), Q durumunda x olayının olma olasılığı Q(x) olarak ifade edilir. Sonuçların bit cinsinden ifade edilmesi için Q(x)'in 2 tabanlı logaritması alınır. Makine öğrenmesinde sınıflandırma problemlerinde ise çapraz entropi hesaplanırken taban-e veya doğal logaritma kullanılır yani sonuç bit olarak ifade edilmez, "nats" adı verilen birimler şeklindedir (Brownlee, 2019a). Ayrıca denklemde (3.10) bulunan "-" işareti sayesinde sonuç her zaman pozitiftir.

$$H(P, Q) = -\sum_{x \in X} P(x) \log Q(x) \tag{3.10}$$

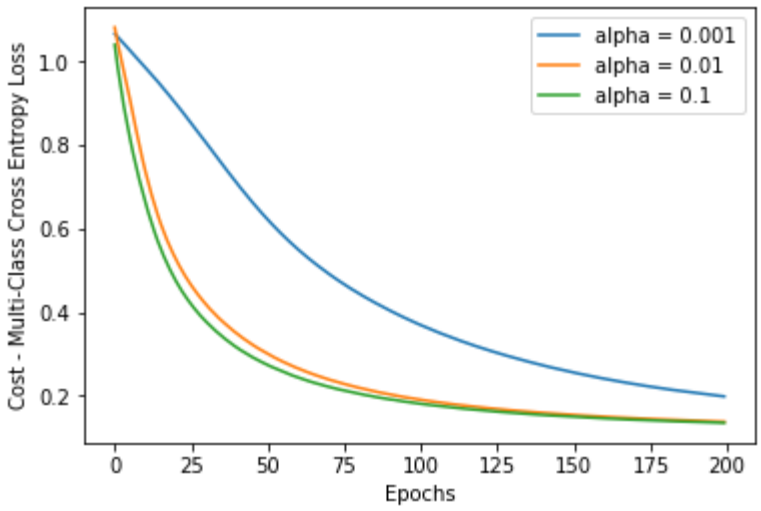
Makine öğrenmesinde kayıp fonksiyonu olarak çapraz entropi kullanıldığında herbir noktaya ait olasılıklar için kayıp değerleri hesaplanır. Kayıp değerlerinin ortalaması alınarak 0 ile 1 arasında çıktı üretilir bu sayede sınıflandırma modelinin performansı hesaplanır (Brownlee, 2019a). Sınıflandırma problemlerinin çeşidine göre iki tür çapraz entropi kullanımı vardır.

- İkili çapraz entropi (binary cross entropy): Verilen bir örneğin iki etiket sınıfından hangisine ait olduğunun tahmin edilmesi amacıyla kullanılan kayıp fonksiyonudur (Brownlee, 2019a). Aşağıdaki şekilde öğrenme katsayısına bağlı olarak 1000 iterasyonda kayıp değerleri sunulmuştur. (Şekil 3.13)



Şekil 3.13. İkili (binary) çapraz entropi (Mahendru, 2019)

- Kategorik çapraz entropi (categorical cross entropy): Verilen bir örneğin ikiden fazla sınıf etiketinden hangisine ait olduğunun tahmin edilmesi amacıyla kullanılan kayıp fonksiyonudur. Şekil 3.14'te 200 iterasyonda öğrenme katsayısına bağlı olarak kayıp değerleri sunulmuştur. (Brownlee, 2019a)



Şekil 3.14. Çok sınıflı çapraz entropi (Mahendru, 2019)

Sınıflandırma modeli değerlendirilirken cross entropi değerinin düşük olması hedeflenir. İyi bir çapraz entropi değeri için sonuç 0.2'den küçük ise durumun iyi olduğu değerlendirilir. Sonuç 1' den büyük ise modelin iyileştirme için tekrar değerlendirilmesi tavsiye edilmektedir (Versloot, 2019).

Çapraz entropi hata fonksiyonu kullanıldığında hata oranı arttıkça eğitim hızı yükselir. Öğrenme yavaşlığı probleminden korunmak için çapraz entropi kullanılmaktadır.

3.2.4. Optimizasyon Fonksiyonları

Sinir ağında öğrenmenin en iyi olabileceği modelin belirlenmesi bir optimizasyon problemidir. Sinir ağı modelimizi oluştururken amacımız tahmin işleminin doğru olarak gerçekleştirilerek kayıp değerinin en düşük olduğu noktanın bulunmasıdır. Sinir ağında bu işlem geri yayılım algoritması (backpropagation) ile ağırlık değerlerinin önceki katmanlara yayılımı ile gerçekleştirilir. Her iterasyonda hatanın giderilmesi için yapılan bu işlemde ağırlıklar ve bias değerleri gibi model parametreleri optimizasyon fonksiyonları ile değiştirilir. Optimize edici, öğrenme oranı (learning rate) parametresi ile öğrenme oranını belirler. Optimizasyon algoritmalarının başarısını etkileyen ilgili hiperparametreler mini batch-size, momentum ve beta seçimidir (Altun ve Talu, 2020). Sinir ağı modellerinde kullanılan optimizasyon algoritmaları Stokastik Bayır/Gradyan İniş (Stochastic Gradient Descent-SGD), Adagrad Algoritması (Adaptive Gradyan Descent), RMSProp (Root Mean Square) Algoritması, Adadelta Algoritması ve Adam (Adaptive Moment Estimation) olarak bilinmektedir (Çarkacı, 2018; Ser ve Bati, 2019).

Optimizasyon algoritmaları arasında performans olarak farklılıklar vardır. Yapmış olduğumuz çalışmada daha az salınımlı, daha hızlı ve tutarlı bir optimizasyon algoritması kullanılması tercih edildiği için Adam (Adaptive Moment Estimation) optimizasyon tekniği kullanılmıştır.

3.2.4.1 RMSProp (Root Mean Square) Algoritması

2012 yılında Geoffrey Hinton tarafından önerilmiş fakat yayınlanmamış gradyan tabanlı adaptif bir optimizasyon algoritmasıdır. RMSProp Adagrad algoritmasında sürekli azalan öğrenme oranı sorununu ortadan kaldırır. Adagrad algoritmasından farklı olarak burada gradyanların kareleri değil momentumlu gradyanların karesi alınır (Seyyarer ve ark., 2020).

3.2.4.2. Adam (Adaptive Moment Estimation)

RMS ve momentum yöntemlerini bir araya getiren gradyan düşüm algoritmasıdır. RMSProp ile birlikte derin öğrenme modellerinde en iyi performans veren optimizasyon algoritmalarındandır. Parametreleri öğrenme oranı, momentum parametresi (önerilen 0.9), RMSProp parametresi (önerilen 0.999) ve epsilon sabiti (10^-8) olarak verilir (Seyyarer ve ark., 2020).

3.2.5. Sinir Ağı Modelinin Bileşenleri ve Hiperparametreler

Modelin eğitim sürecini oluşturmamız için (compile) tanımlamamız gereken parametreler mevcuttur. Bu parametreler kayıp fonksiyonunun belirlenmesi, optimizasyon algoritmasının belirlenmesi ve metrik değeridir.

3.2.5.1. Kayıp (Loss) Fonksiyonları

Sinir ağındaki son katmanda kayıp fonksiyonu tanımlanır. Kayıp fonksiyonları, sinir ağı modelini bir bütün olarak değerlendirerek her iterasyonda sonuç olarak bir kayıp (loss) değeri sunar. Kayıp değerinin büyüklüğüne göre eğitim sonucu elde edilen ağırlık ve bias parametrelerinin modelimiz için ne kadar uygun olduğu hakkında bilgi sahibi oluruz ve çalışmalarımıza yön veririz. Kayıp fonksiyonu problem tipine göre belirlenir.

Sınıflandırma problemleri için "kategorik çapraz entropi (categorical_cross entropy)" veya "ikili çapraz entropi (binary_crossentropy)" fonksiyonları seçilebilir.

3.2.5.2. Optimizasyon Algoritmaları (optimizer)

Sinir ağı modelimizi oluştururken amacımız tahmin işleminin doğru olarak gerçekleştirilerek kayıp değerinin en düşük olduğu noktanın bulunmasıdır. Sinir ağında bu işlem geri yayılım algoritması (backpropagation) ile ağırlık değerlerinin önceki katmanlara yayılımı ile gerçekleştirilir. Her iterasyonda hatanın giderilmesi için yapılan bu işlemde ağırlıklar optimizasyon fonksiyonları ile değiştirilir. Optimize edici, öğrenme oranı (learning rate) parametresi ile öğrenme oranını belirler.

Yapmış olduğumuz çalışmada daha az salınımlı, daha hızlı ve tutarlı bir optimizasyon algoritması kullanılması tercih edildiği için Adam (Adaptive Moment Estimation) optimizasyon tekniği kullanılmıştır (Çarkacı, 2018; Ser ve Bati, 2019).

3.2.5.3. Metrikler (metrics)

Modelimizin eğitim sırasında nasıl bir performans gösterdiğini anlayabilmek ve her bir iterasyonda doğruluk ve kayıp değerlerini görebilmek için ‘accuracy’ parametresi tanımlanmıştır.

3.2.5.4. Mini-Batch Boyutu (batch_size)

Sinir ağı modelinin aynı anda ne kadar veriyi işleyebileceğinin belirlendiği hiperparametredir. Hata oranının düşürülmesi için her iterasyonda geriye dönük olarak gradyan (gradient descent) hesabı ile ağırlıklar güncellenmektedir. Veri sayısı miktarına göre hesaplama süresi değişim göstermektedir. batch_size parametresini en küçük 1 olarak girdiğimizde modelimizin gürültüyü de öğrenmesine sebep oluruz. Bu şekilde “stochastic gradient descent” optimizasyon işlemini gerçekleştirmiş oluruz. Tersi olarak boyutumuz büyük olduğunda ise “overfitting” dediğimiz aşırı öğrenme durumu oluşur.

3.2.5.5. Eğitim Adımı Sayısı (epoch-iterasyon)

Öğrenme işleminin gerçekleşebilmesi için eğitimin tekrarlanarak hata değerinin düşürülmesi gerekmektedir. Tüm eğitim verisinin işlendiği bu tekrarlara epoch denilir. Doğruluğun en yüksek olduğu yerde eğitim tamamlanacak şekilde epoch hiperparametre değeri ayarlanır (Çarkacı, 2018).

Modelin eğitim sürecini oluşturmamız için tanımlamamız gereken parametreler mevcuttur. Bu parametreler kayıp fonksiyonunun belirlenmesi, optimizasyon algoritmasının belirlenmesi ve metrik değeridir.

3.3. Model Performansının Değerlendirilmesi

Ağ saldırılarının tespit edilmesi için oluşturmuş olduğumuz en yakın komşu ve bir boyutlu evrişimli sinir ağı modelinin performansının değerlendirilmesi amacıyla karışıklık matrisi (confusion matrix), doğruluk (accuracy), duyarlılık (recall), kesinlik (precision), f-skor (f-score) metrikleri kullanılmıştır.

a) Karışıklık/Hata Matrisi (Confusion Matrix)

Karışıklık matrisi gerçek ve tahmin edilen sonuçların karşılaştırılabilmesi için kullanılır. (Çizelge 3.2) Sonuçların karşılaştırılabilmesi için TP (True-Pozitif), TN (True-Negatif), FP (False-Pozitif), FN (False-Negatif) değerlerini kullanır.

Çizelge 3.2. Karışıklık matrisi gösterimi

Tahmin /Gerçek	+	-
+	Doğru Pozitif (TP)	Yanlış Pozitif (FP)
-	Yanlış Negatif (FN)	Doğru Negatif (TN)

- Doğru Pozitif: Gerçekte doğru bir değer model tarafından da doğru tespit edildiğinde TP,
- Yanlış Pozitif: Gerçekte yanlış olan bir değer model tarafından doğru tespit edildiğinde FP,
- Yanlış Negatif: Gerçekte pozitif olan bir durum model tarafından negatif olarak tespit edildiğinde FN,
- Doğru Negatif: Gerçekte negatif olan bir durumun model tarafından da negatif tespit edilmesi ile TN durumu oluşur.

b) Doğruluk (Accuracy)

Doğru tahmin edilen örneklerin sayısının tüm örneklerin sayısına oranıdır. Doğruluk metriğinin denklemi 3.11'de verilmiştir.

$$Doğruluk = {TP + TN}/{TP + FP + FN + TN} \tag{3.11}$$

c) Duyarlılık (Recall)

Doğru tahmin edilen pozitif örneklerin sayısının gerçekteki pozitif örneklerin sayısına oranıdır. Duyarlılık metriğinin denklemi 3.12'de verilmiştir.

$$Duyarlılık = {TP}/{TP + FN} \tag{3.12}$$

d) Kesinlik (Precision)

Doğru olarak tahmin edilen pozitif örneklerin sayısının tüm pozitif olarak tahmin edilen örneklerin sayısına oranıdır. Kesinlik değerinin yüksek olması modelin

doğru tahmin ettiği bir değerin gerçekten pozitif olduğunu göstermektedir. Kesinlik metriğinin denklemi 3.13'te verilmiştir.

$$Duyarlılık = {}^{TP}/_{TP + FP} \tag{3.13}$$

e) F-Skor (F-Score)

Duyarlılık ve kesinlik değerlerinin harmonik ortalamasıdır. Modelin kesinlik ve duyarlılığının ölçüsüdür ve en az 0, en fazla 1 değerlerini alabilir. F-Skor metriğinin denklemi 3.14'te verilmiştir.

$$F - Score = \frac{2xDuyarlılıkxKesinlik}{Duyarlılık + Kesinlik} \tag{3.14}$$

4. ARAŞTIRMA SONUÇLARI VE TARTIŞMA

4.1. Çalışma Ortamının Hazırlanması

Çalışmalarda Windows 10 işletim sistemi ve RTX 3050 ekran kartına sahip bir makine kullanılmıştır. Çalışma ortamı olarak için ise Anaconda kurulumunu gerçekleştirilmiştir. "tf" isimli bir conda ortamı oluşturulmuş ve phyton 3.10 sürümü yüklenmiştir.

Derin öğrenme çalışmalarında tensorflow kütüphanesinin kullanılabilmesi için GPU ve CPU destekli olarak kurulum yapılabiliyor. Ekran kartı özelliklerine karşılık düşen "compute capability" değeri 3.5 ve üzeri bir değerdeyse GPU kurulumu destekleniyor. Derin öğrenme ortamını hazırlarken bilgisayar özellikleri desteklediğinden dolayı tensorflow-gpu kurulumu tercih edilmiştir. Windows makinede GPU çalıştırabilmek için donanım gereksinimleri olarak CUDA Toolkit ve CuDNN kurulumlarının sağlanması gerekiyor. Windows makinede GPU desteği 2.10 ve önceki sürümlerde mevcuttur. Uyumluluk tablosundan incelenerek Tensorflow_gpu 2.10 ve Python 3.10 sürümleri için CUDA toolkit 11.2 ve CuDNN 8.1.0 kurulumları tamamlanarak çalışma ortamı hazırlanmıştır (Tensorflow).

4.2. Veri Seti İşlemleri

4.2.1. Veri Setinin Tanımlanması

Yapmış olduğumuz çalışmada veri seti olarak CSE-CIC-IDS2018 kullanılmıştır. Veri, İletişim Güvenliği Kuruluşu (Communications Security Establishment (CSE)) ve Kanada Siber Güvenlik Enstitüsü (Canadian Institute for Cybersecurity (CIC)) tarafından hazırlanmıştır. CSE-CIC-IDS2018 saldırı veriseti AWS platformunda oluşturulmuştur; Brute-force, Heartbleed, Botnet, DoS, DDoS, Web saldırıları ve ağa içeriden sızma şeklinde 7 farklı saldırı senaryosu içermektedir. Saldırı senaryolarının gerçekleştirilmesi için test ortamı 50 adet saldırgan, 420 bilgisayar ve 30 sunucu olmak üzere kurgulanmıştır. Ağ trafiği CICFlowMeter-V3 aracı ile yakalanarak 80 sütundan oluşan veri seti oluşturulmuştur (UNB; Leevy ve Khoshgoftaar, 2020). Veri seti özellik adları ve açıklamaları Çizelge 4.1'de sunulmuştur.

Çizelge 4.1. Veri seti öznitelikleri ve açıklamaları

No	Özellik Adı	Özellik Tanımı	Dtype (I)	Dtype (II)
1	Dst Port	Hedef port bilgisi	object	int64
2	Protocol	Bağlantı sırasında kullanılan protokol	object	int64
3	Timestamp	İletimin sağlandığı zaman	object	int64
4	Flow Duration	Mikro saniyedeki akış süresi	object	int64
5	Tot Fwd Pkts	İleri yönde gönderilen toplam paket sayısı	object	int64
6	Tot Bwd Pkts	Geri yönde gönderilen toplam paket sayısı	object	int64
7	TotLen Fwd Pkts	İleri yönde iletilen paketlerin toplam uzunluğu	object	int64
8	TotLen Bwd Pkts	Geri yönde iletilen paketlerin toplam uzunluğu	object	int64
9	Fwd Pkt Len Max	İleri yönde iletilen maksimum paket boyutu	object	int64
10	Fwd Pkt Len Min	İleri yönde iletilen minimum paket boyutu	object	int64
11	Fwd Pkt Len Mean	İleri yönde iletilen paketin ortalama boyutu	object	float64
12	Fwd Pkt Len Std	İleri yönde iletilen paketin standart sapma boyutu	object	float64
13	Bwd Pkt Len Max	Geri yönde iletilen maksimum paket boyutu	object	int64
14	Bwd Pkt Len Min	Geri yönde iletilen minimum paket boyutu	object	int64
15	Bwd Pkt Len Mean	Geri yönde iletilen paketin ortalama boyutu	object	float64
16	Bwd Pkt Len Std	Geri yönde iletilen paketin standart sapma boyutu	object	float64
17	Flow Byts/s	Saniyede akan byte sayısı	object	float64
18	Flow Pkts/s	Saniyede akan paket sayısı	object	float64
19	Flow IAT Mean	İki akış arasındaki ortalama süre	object	float64
20	Flow IAT Std	İki akış arasındaki standart sapma süresi	object	float64
21	Flow IAT Max	İki akış arasındaki maksimum süre	object	float64
22	Flow IAT Min	İki akış arasındaki minimum süre	object	float64
23	Fwd IAT Tot	İleri yönde gönderilen iki paket arasındaki toplam süre	object	float64
24	Fwd IAT Mean	İleri yönde gönderilen iki paket arasındaki ortalama süre	object	float64
25	Fwd IAT Std	İleri yönde gönderilen iki paket arasındaki standart sapma süresi	object	float64
26	Fwd IAT Max	İleri yönde gönderilen iki paket arasındaki maksimum süre	object	int64
27	Fwd IAT Min	İleri yönde gönderilen iki paket arasındaki minimum süre	object	int64

28	Bwd IAT Tot	Geri yönde gönderilen iki paket arasındaki toplam süre	object	int64
29	Bwd IAT Mean	Geri yönde gönderilen iki paket arasındaki ortalama süre	object	float64
30	Bwd IAT Std	Geri yönde gönderilen iki paket arasındaki standart sapma süresi	object	float64
31	Bwd IAT Max	Geri yönde gönderilen iki paket arasındaki maksimum süre	object	int64
32	Bwd IAT Min	Geri yönde gönderilen iki paket arasındaki minimum süre	object	int64
33	Fwd PSH Flags	İleri yönde iletilen dolaşımdaki paketlerde PSH bayrağının ayarlandığı zamanların sayısı (UDP: 0)	object	int64
34	Bwd PSH Flags	Geri yönde iletilen dolaşımdaki paketlerde PSH bayrağının ayarlandığı zamanların sayısı (UDP: 0)	object	int64
35	Fwd URG Flags	İleri yönde iletilen dolaşımdaki paketlerde URG bayrağının ayarlandığı zamanların sayısı (UDP: 0)	object	int64
36	Bwd URG Flags	Geri yönde iletilen dolaşımdaki paketlerde URG bayrağının ayarlandığı zamanların sayısı (UDP: 0)	object	int64
37	Fwd Header Len	İleri yönde iletilen başlıklar için kullanılan toplam başlık boyutu	object	int64
38	Bwd Header Len	Geri yönde iletilen başlıklar için kullanılan toplam başlık boyutu	object	int64
39	Fwd Pkts/s	Saniyede gönderilen ileri yöndeki paket sayısı	object	int64
40	Bwd Pkts/s	Saniyede gönderilen geri yöndeki paket sayısı	object	int64
41	Pkt Len Min	Bir akışın minimum uzunluğu	object	int64
42	Pkt Len Max	Bir akışın maksimum uzunluğu	object	int64
43	Pkt Len Mean	Bir akışın ortalama uzunluğu	object	float64
44	Pkt Len Std	Bir akışın standart sapma uzunluğu	object	float64
45	Pkt Len Var	Ardışık olarak iletilen paket arasındaki minimum süre	object	float64
46	FIN Flag Cnt	FIN içeren paket bilgisi	object	int64
47	SYN Flag Cnt	SYN içeren paket bilgisi	object	int64
48	RST Flag Cnt	RST içeren paket bilgisi	object	int64
49	PSH Flag Cnt	PSH içeren paket bilgisi	object	int64
50	ACK Flag Cnt	ACK içeren paket bilgisi	object	int64
51	URG Flag Cnt	URG içeren paket bilgisi	object	int64

52	CWE Flag Count	CWE içeren paket bilgisi	object	int64
53	ECE Flag Cnt	ECE içeren paket bilgisi	object	int64
54	Down/Up Ratio	İndirme ve yükleme oranı	object	int64
55	Pkt Size Avg	Ortalama paket boyutu	object	float64
56	Fwd Seg Size Avg	İleri yönde gözlemlenen ortalama boyut	object	float64
57	Bwd Seg Size Avg	Geri yönde gözlemlenen ortalama boyut	object	float64
58	Fwd Byts/b Avg	İleri yönde iletilen kütle oranı ortalama byte sayısı	object	int64
59	Fwd Pkts/b Avg	İleri yönde iletilen kütle oranı ortalama paket sayısı	object	int64
60	Fwd Blk Rate Avg	İleri yönde iletilen kütle oranının ortalama sayısı	object	int64
61	Bwd Byts/b Avg	Geri yönde iletilen kütle oranı ortalama byte sayısı	object	int64
62	Bwd Pkts/b Avg	Geri yönde iletilen kütle oranı ortalama paket sayısı	object	int64
63	Bwd Blk Rate Avg	Geri yönde iletilen kütle oranının ortalama sayısı	object	int64
64	Subflow Fwd Pkts	İleri yönde iletilen bir alt akıştaki ortalama paket sayısı	object	int64
65	Subflow Fwd Byts	İleri yönde iletilen bir alt akıştaki ortalama byte sayısı	object	int64
66	Subflow Bwd Pkts	Geri yönde iletilen bir alt akıştaki ortalama paket sayısı	object	int64
67	Subflow Bwd Byts	Geri yönde iletilen bir alt akıştaki ortalama byte sayısı	object	int64
68	Init Fwd Win Byts	İleri yönde iletilen ilk pencere içinde gönderilen byte sayısı	object	int64
69	Init Bwd Win Byts	Geri yönde iletilen ilk pencere içinde gönderilen byte sayısı	object	int64
70	Fwd Act Data Pkts	İleri yönde iletilen en az 1 byte TCP veri yüküne sahip paket sayısı	object	int64
71	Fwd Seg Size Min	İleri yönde gözlemlenen minimum segment boyutu	object	int64
72	Active Mean	Bir akışın boşta kalmadan önce aktif olduğu ortalama süre	object	float64
73	Active Std	Bir akışın boşta kalmadan önce aktif olduğu standart sapma süresi	object	float64
74	Active Max	Bir akışın boşta kalmadan önce aktif olduğu maksimum süre	object	int64
75	Active Min	Bir akışın boşta kalmadan önce aktif olduğu minimum süre	object	int64
76	Idle Mean	Bir akışın aktif hale gelmeden boşta kaldığı ortalama süre	object	float64

77	Idle Std	Bir akışın aktif hale gelmeden boşta kaldığı standart sapma süresi	object	float64
78	Idle Max	Bir akışın aktif hale gelmeden boşta kaldığı maksimum süre	object	int64
79	Idle Min	Bir akışın aktif hale gelmeden boşta kaldığı minimum süre	object	int64
80	Label	Etiket	object	int64

Çalışmanın ilk kısmında DoS saldırıları incelenmiştir. Yakalanan veri benign, Hulk, GoldenEye, Slowloris, SlowHTTPTest etiketli verilerden oluşmaktadır. Çalışmanın devamında bu saldırı veriseti "Senaryo 1" olarak adlandırılacaktır. Veri setinde uygulanan saldırılara ilişkin bilgiler Çizelge 4.2'de sunulmuştur.

Çizelge 4.2. CSE-CIC-IDS 2018 düzenlenen DoS saldırı senaryoları (UNB)

Saldırgan	**Kurban**	**Saldırı Adı**	**Tarih**	**Saldırı Başlama Saati**	**Saldırı Bitiş Saati**
172.31.70.46 (Valid IP:18.219.211.138)	18.217.21.148-172.31.69.25	DoS-GoldenEye	Perşembe, 15-02-2018	09:26	10:09
172.31.70.8 (Valid IP:18.217.165.70)	18.217.21.148-172.31.69.25	DoS-Slowloris	Perşembe, 15-02-2018	10:59	11:40
172.31.70.23 (Valid IP: 13.59.126.31)	18.217.21.148-172.31.69.25	DoS-SlowHTTPTest	Cuma, 16-02-2018	10:12	11:08
172.31.70.16 (Valid IP:18.219.193.20)	18.217.21.148-172.31.69.25	DoS-Hulk	Cuma, 16-02-2018	13:45	14:19

Çalışmanın ikinci kısmında sızma ve bot saldırıları incelenmiştir. Yakalanan veri benign, infiltration, bot etiketli verilerden oluşmaktadır. Çalışmanın devamında bu saldırı veriseti "Senaryo 2" olarak adlandırılacaktır. Veri setinde uygulanan saldırılara ilişkin bilgiler Çizelge 4.3'te sunulmuştur.

Çizelge 4.3. CSE-CIC-IDS 2018 düzenlenen sızma ve bot saldırı senaryoları (UNB)

Saldırgan	Kurban	Saldırı Adı	Tarih	Saldırı Başlama Saati	Saldırı Bitiş Saati
13.58.225.34	18.221.148.137-172.31.69.24	Infiltration	Çarşamba, 28-02-2018	10:50	12:05
13.58.225.34	18.221.148.137-172.31.69.24	Infiltration	Çarşamba, 28-02-2018	13:42	14:40
13.58.225.34	18.216.254.154-172.31.69.13	Infiltration	Perşembe, 01-03-2018	09:57	10:55
13.58.225.34	18.216.254.154-172.31.69.13	Infiltration	Perşembe, 01-03-2018	14:00	15:37
13.58.225.34	18.216.254.154-172.31.69.13	Infiltration	Perşembe, 01-03-2018	14:00	15:37
18.219.211.138	18.217.218.111-172.31.69.23 18.222.10.237-172.31.69.17 18.222.86.193-172.31.69.14 18.222.62.221-172.31.69.12 13.59.9.106-172.31.69.10 18.222.102.2-172.31.69.8 18.219.212.0-172.31.69.6 18.216.105.13-172.31.69.26 18.219.163.126-172.31.69.29 18.216.164.12-172.31.69.30	Bot	Cuma, 02-03-2018	10:11	11:34
18.219.211.138	18.217.218.111-172.31.69.23 18.222.10.237-172.31.69.17 18.222.86.193-172.31.69.14 18.222.62.221-172.31.69.12 13.59.9.106-172.31.69.10 18.222.102.2-172.31.69.8 18.219.212.0-172.31.69.6 18.216.105.13-172.31.69.26 18.219.163.126-172.31.69.29 18.216.164.12-172.31.69.30	Bot	Cuma, 02-03-2018	14:24	15:55

4.2.2. Veri Analizi ve Ön İşlemenin Gerçekleştirilmesi

1. Veri setinin yüklenmesi ve veri dönüşümünün gerçekleştirilmesi: Veri seti değişken dönüşümü Çizelge 4.1'de sunulmuştur. Çizelge 4.1 incelendiğinde başlangıçta "DType (I)"de "object" olan veri tiplerinin, "DType (II)"de olduğu gibi veri dönüşümü gerçekleştirilmiştir.

Senaryo 1'de veri dönüşümü gerçekleştirilirken çok sınıflı saldırı trafiği için "Benign":0, "DoS attacks-Hulk":1, "DoS attacks-Slowloris":2, "DoS attacks-GoldenEye":3, "DoS attacks-SlowHTTPTest":4 olarak etiketlenmiştir. İki sınıflı saldırı veri setinde ise "Benign":0, "Attack":1 olarak etiketlenmiştir.

Senaryo 2'de veri dönüşümü gerçekleştirilirken çok sınıflı saldırı trafiği için "Benign":0, "Bot":1, "Infiltration":2 olarak etiketlenmiştir. İki sınıflı saldırı veri setinde ise "Benign":0, "Attack":1 olarak etiketlenmiştir.

2. Veri setindeki infinive değerlerin (np.inf, -np.inf) NaN değerlere dönüştürülerek diğer NaN değerleri ile birlikte temizlenmesi: Çizelge 4.4'te veri setinin ilk durumu I'de, temizlik sonrası veri sayısı II'de gösterilmiştir. Çizelge 4.5, Çizelge 4.6, Çizelge 4.7, Çizelge 4.8'de alt gruplara göre veri sayıları verilmiştir.

Çizelge 4.4. CSE-CIC-IDS 2018 temizlik sonrası veri sayısı

Madde	I	II
Senaryo 1	2097150	2089122
Senaryo 2	1379700	1372706

Çizelge 4.5. "Senaryo 1" veri seti saldırı tipine göre etiket verileri

Veri Etiketi	Veri Sayısı
Benign	1434822
DoS attacks-Hulk	461912
DoS attacks-SlowHTTPTest	139890
DoS attacks-GoldenEye	41508
DoS attacks-SlowLoris	10990

Çizelge 4.6. "Senaryo 1" veri seti anomali durumuna göre etiket verileri

Veri Etiketi	Veri Sayısı
Benign	1434822
Attack	654300

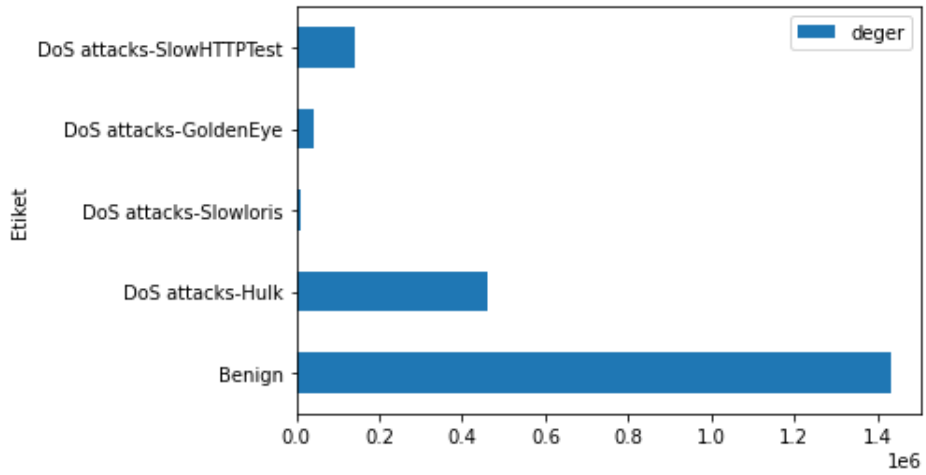
Çizelge 4.7. "Senaryo 2" veriseti saldırı tipine göre etiket verileri

Veri Etiketi	Veri Sayısı
Benign	994112
Bot	286191
Infiltration	92403

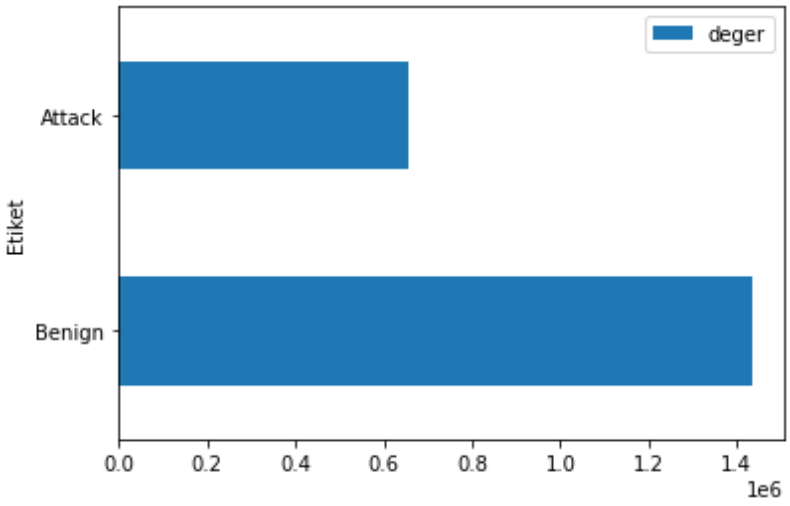
Çizelge 4.8. "Senaryo 2" veri seti anomali durumuna göre etiket verileri

Veri Etiketi	Veri Sayısı
Benign	994112
Attack	378594

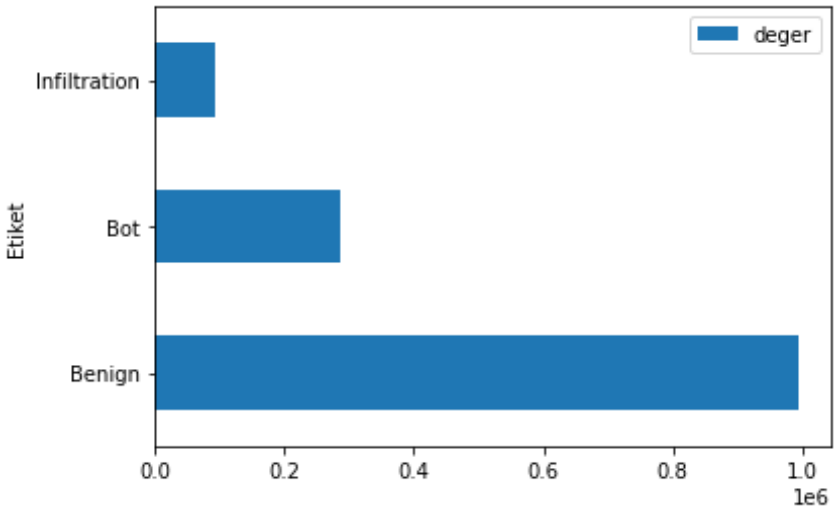
3. Veri görselleştirme: "import matplotlib.pyplot as plt" kütüphanesi kullanılarak "Senaryo 1" için Şekil 4.1, Şekil 4.2; "Senaryo 2" için Şekil 4.3, Şekil 4.4 saldırı tipi ve anomali durumuna göre etiket dağılımını gösteren grafikler çizdirilmiştir.



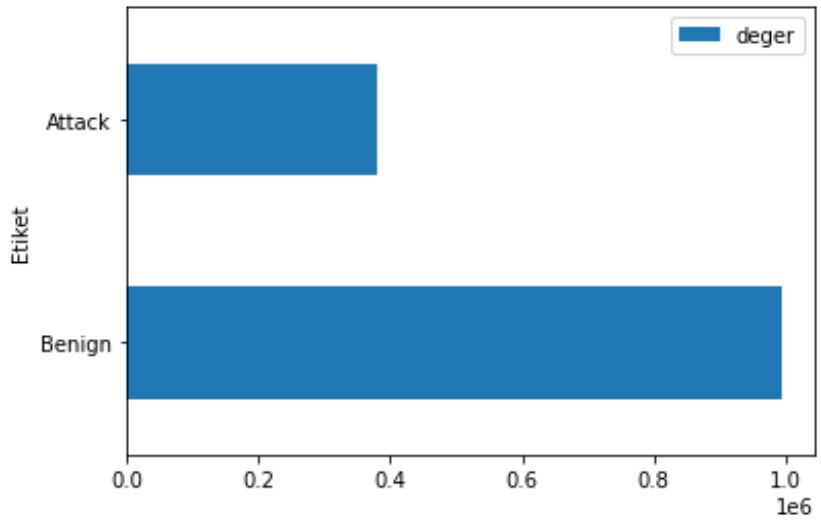
Şekil 4.1. "Senaryo 1" veri seti etiketine göre saldırı çeşidi dağılımı



Şekil 4.2. "Senaryo 1" veri seti etiketine göre anomali durumu



Şekil 4.3. "Senaryo 2" veri seti etiketine göre saldırı çeşidi dağılımı



Şekil 4.4. "Senaryo 2" veri seti etiketine göre anomali durumu

4.2.3. Dengesiz veri setinin dengeli hale getirilmesi

Senaryo 1 ve Senaryo 2'de veri seti dağılımı incelendiğinde sınıf dağılımlarının birbirine yakın olmadığı, dengesiz olduğu görülmüştür. Veri setini dengeli hale getirnek için örneklem azaltma (undersampling), örneklem artırma (oversampling) teknikleri kullanılmıştır.

Senaryo 1, toplam 2,089,122 veriden oluşmaktadır. Veri sayısının çok fazla olması, artırma yöntemi kullanıldığında işlemin yaklaşık olarak 12 saat sürmesi ve mevcut kapasite ile elde edilen verinin verimli olarak kullanılamayacağı gerekçesiyle örmeklem azaltma (undersampling) yöntemi tercih edilmiştir. Örneklem azaltma tekniklerinden TomekLinks, NearMiss, CondensedNearestNeighbor (CNN), EditedNearestNeighbors (ENN) ve OneSidedSelection (OSS) yöntemleri veri setine uygulandığında elde edilen sonuçlar Çizelge 4.9'da sunulmuştur.

Çizelge 4.9. "Senaryo 1" örneklem azaltma sonucu veri sayıları

Etiket	İlk Durum	TomekLinks	NearMiss	CNN	ENN	OSS
0	1434822	975632	7512	7512	975568	8998
1	461912	313974	7512	141	313970	8012
2	139890	95331	7512	20	95331	8000
3	41508	28153	7512	6	28133	8000
4	10990	7512	7512	2	7512	7512

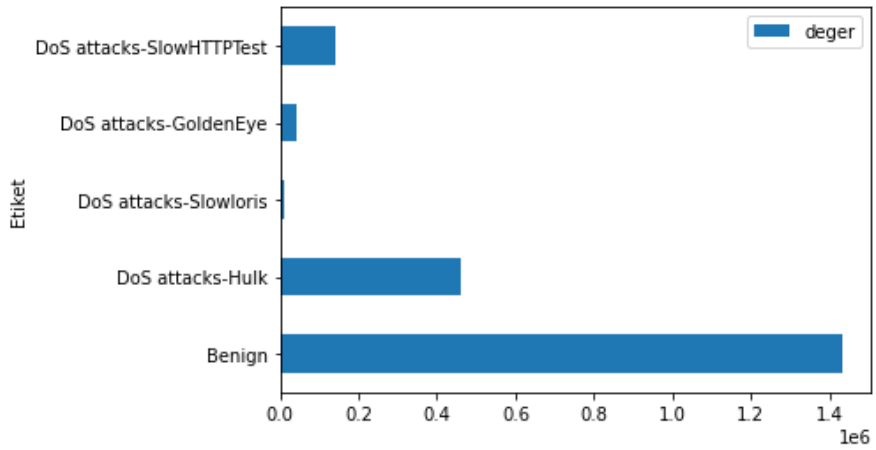
Yöntem olarak hibrit bir yöntem olması, gürültülü veri örneklerini ortadan kaldırması ve aynı zamanda çoğunluk sınıfı veri örneklerini veri setinden kaldırdığı için One Side Selection (OSS) yöntemi tercih edilmiştir.

Senaryo 2 toplam 1,372,706 veriden oluşmaktadır. Senaryo 2'de veri dengelemenin sağlanabilmesi için azınlık sınıfını örnekleyerek sentetik veri artışını sağlayan SMOTEENN (SMOTE Edited Nearest Neighbors) ve SMOTETomek yöntemleri ve örneklem sayısının azaltıldığı One Side Selection (OSS) yöntemi kullanılmıştır. İşlemler sonrası veri dağılımı Çizelge 4.10'da sunulmuştur.

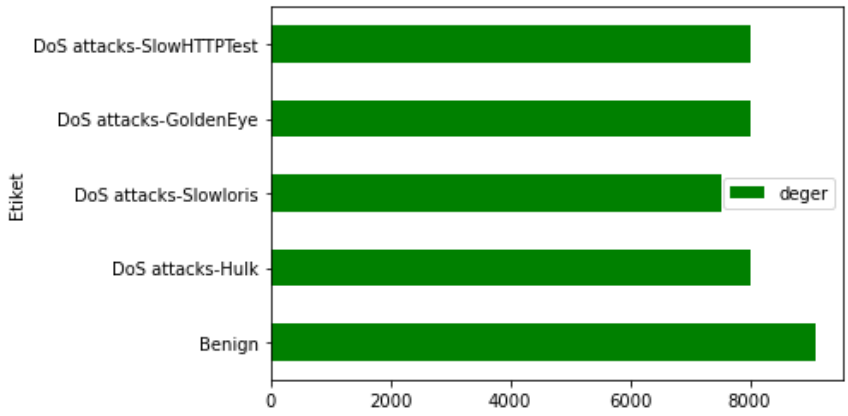
Çizelge 4.10. "Senaryo 2" veri dengeleme işlemi sonrası veri sayıları

Etiket	İlk Durum	SMOTEENN	SMOTETomek	OSS
0	994112	675658	675670	232877
1	286191	501296	646248	62016
2	92403	481492	646244	62732

Gerçekleştirilen testlerde Senaryo 1 grubunda veriseti büyüklüğünden dolayı OSS yöntemi tercih edilmiştir. (Şekil 4.5) İşlem sonrası veriseti dağılımı Şekil 4.6'deki gibidir.

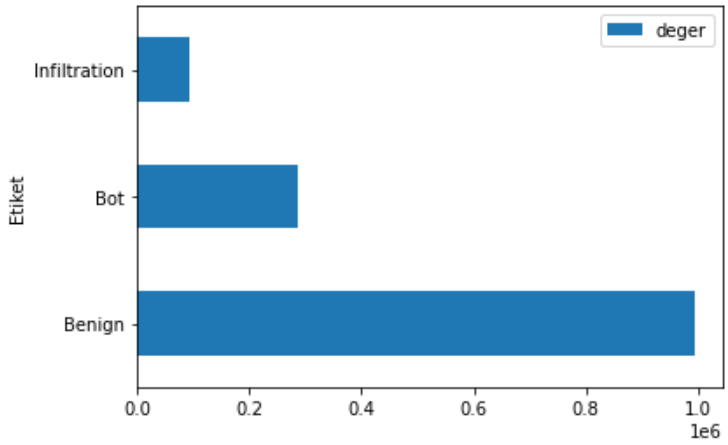


Şekil 4.5. "Senaryo 1" veri seti ilk hali

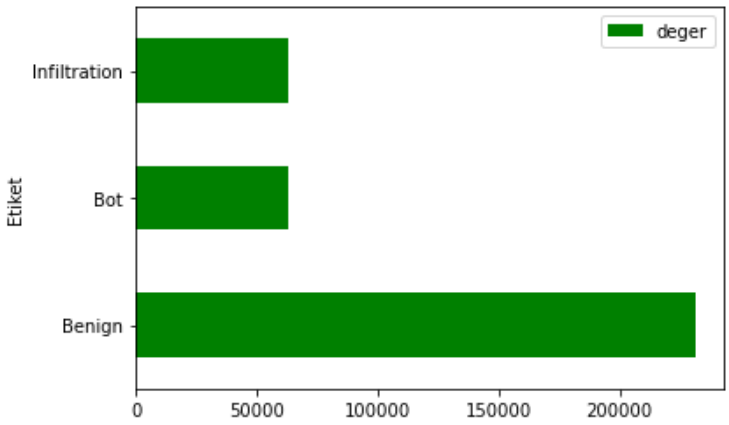


Şekil 4.6. "Senaryo 1" veri setinin OSS ile dengelenmesi

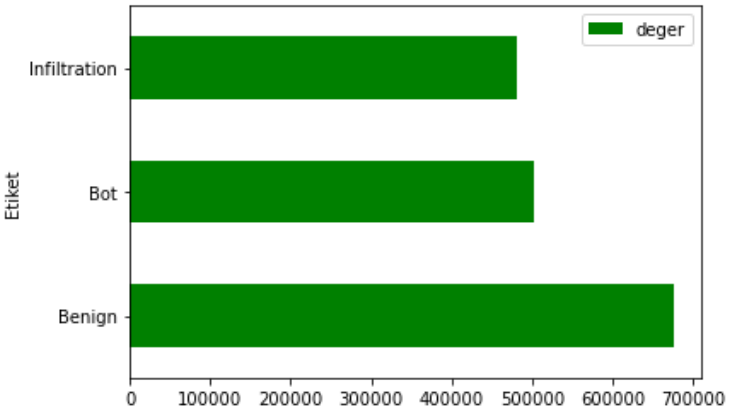
Senaryo 2 grubunda ise OSS, SMOTEENN, SMOTETomek yöntemi tercih edilmiştir. Şekil 4.7'de veri seti ilk hali sunulmuştur. İşlem sonrası veriseti dağılımı Şekil 4.8, Şekil 4.9, Şekil 4.10'daki gibidir.



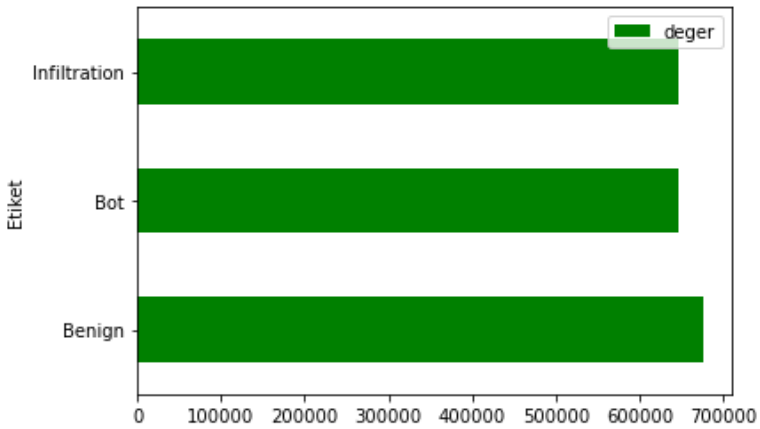
Şekil 4.7. "Senaryo 2" veri seti ilk hali



Şekil 4.8. "Senaryo 2" veri setinin OSS ile dengelenmesi

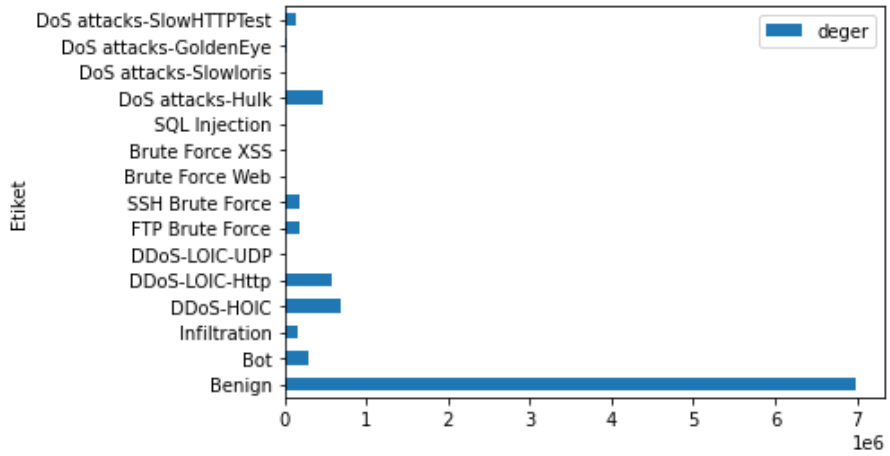


Şekil 4.9. "Senaryo 2" veri setinin SMOTEENN ile dengelenmesi

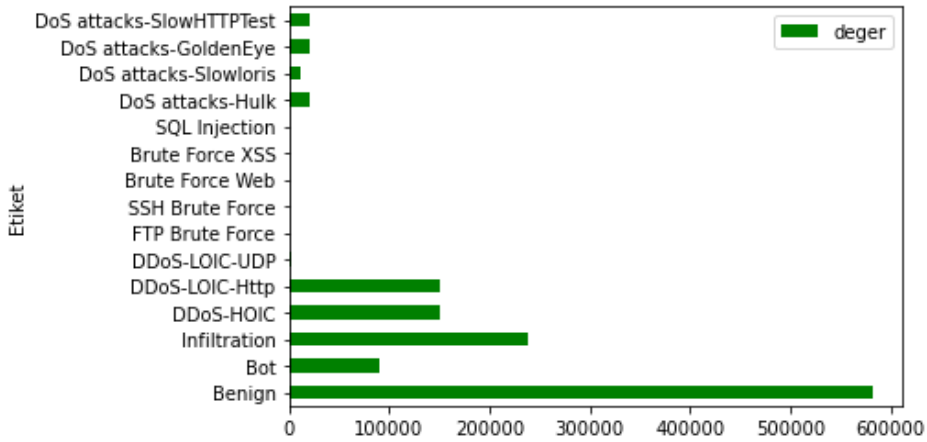


Şekil 4.10. "Senaryo 2" veri setinin SMOTETomek ile dengelenmesi

Son olarak Senaryo 3 grubunda tüm saldırı veri seti grup gruplar halinde alınarak birleştirilmiştir, Şekil 4.11'de veri seti ilk dağılımı sunulmuştur. Şekil 4.12'de gösterildiği üzere veri seti, büyüklüğünden dolayı OSS yöntemi ile azaltılarak kullanılabilir hale getirilmiştir.



Şekil 4.11. "Senaryo 3" veri seti ilk hali



Şekil 4.12. "Senaryo 3" veri setinin OSS ile dengelenmesi

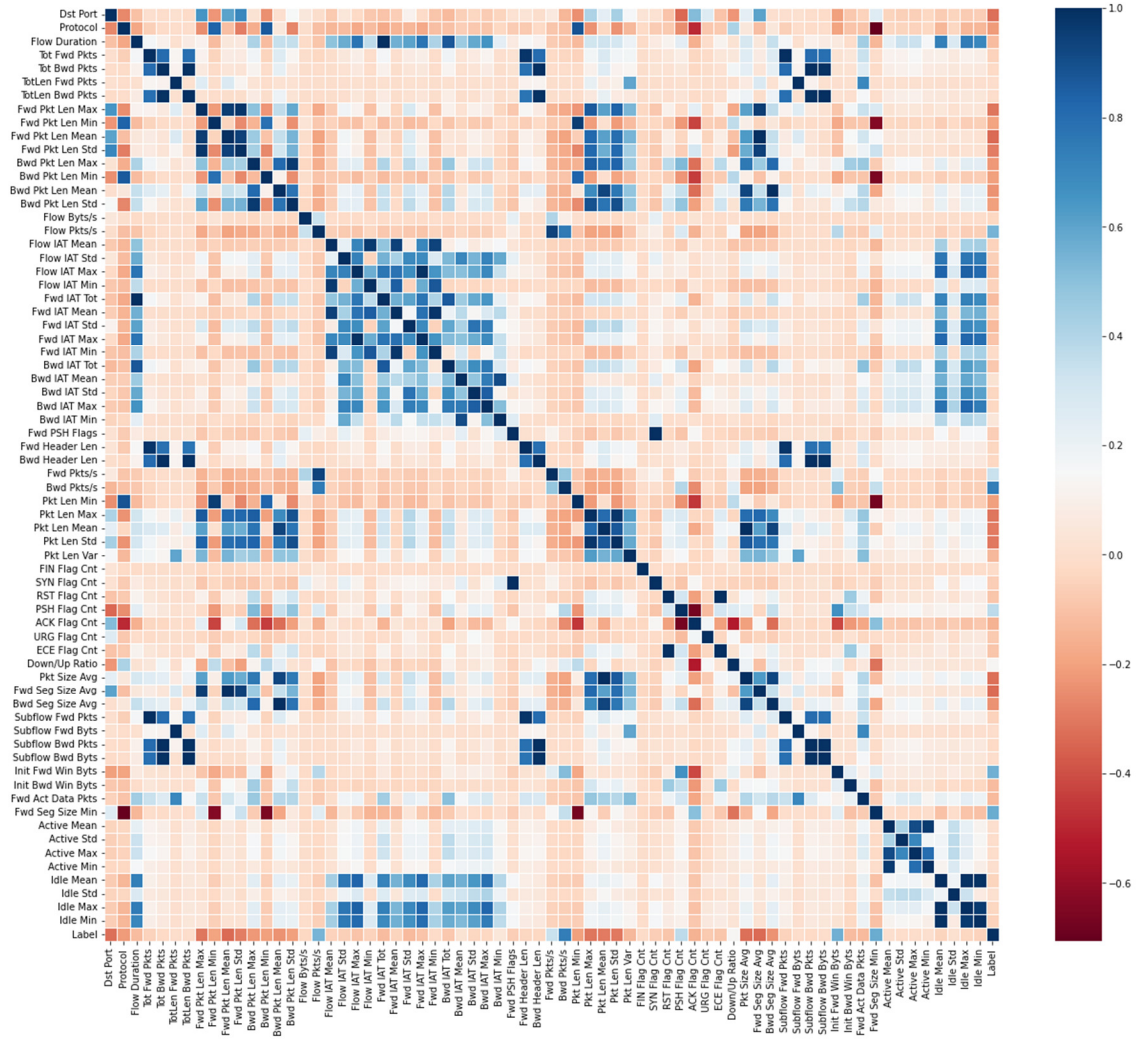
4.2.4. Özellik Seçiminin Gerçekleştirilmesi

4.2.4.1. Isı Haritası ile Korelasyon Matrisi

Çalışmamızın bu kısmında hedef değişkenle özellikler arası ilişkilerin tespit edilebilmesi için "import seaborn as sns" kütüphanesi dahil edilerek ısı haritası (heatmap) oluşturulmuştur.

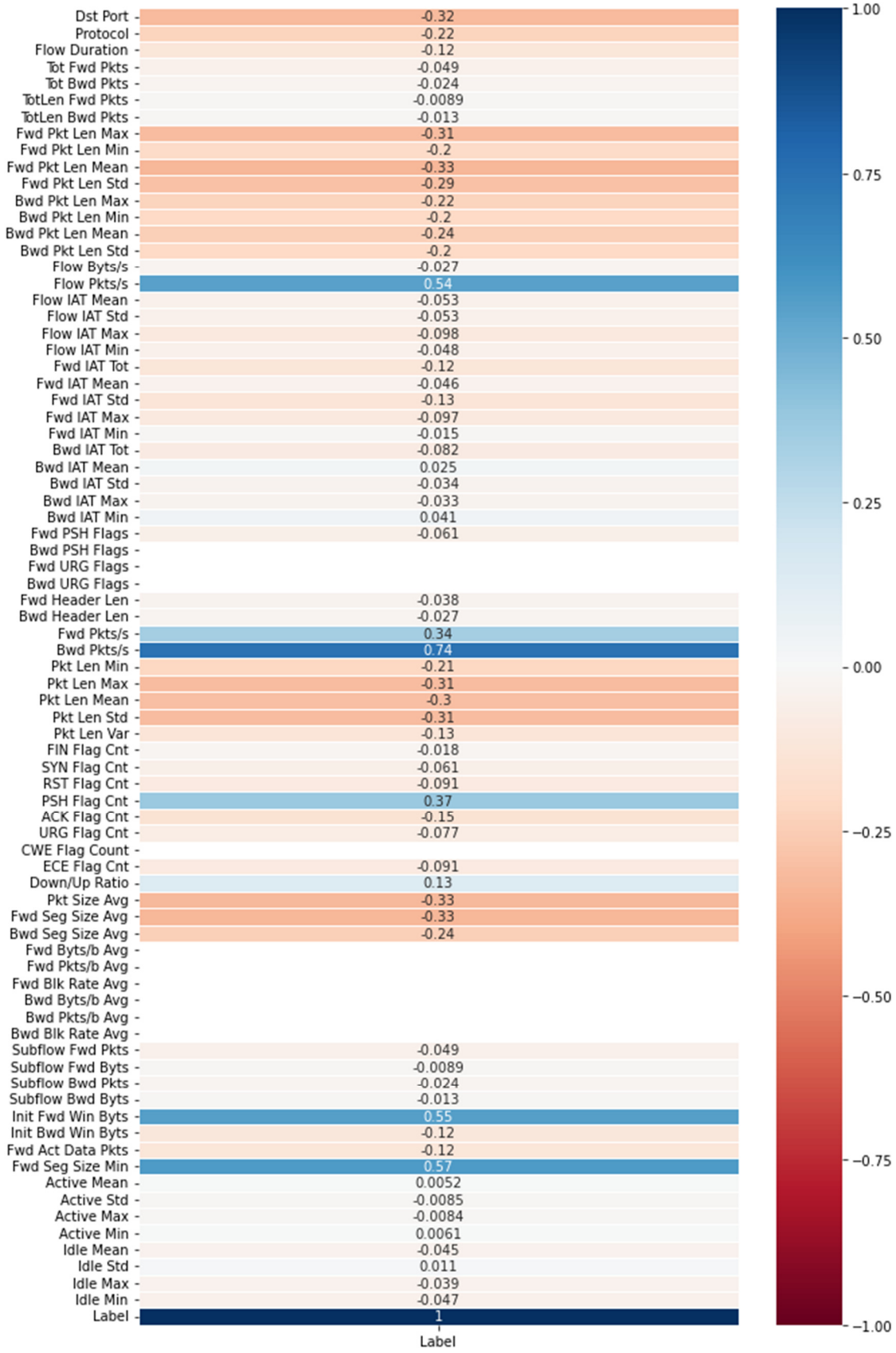
Senaryo 1 için ısı haritaları oluşturulurken bağımsız değişkenlerin saldırı çeşidine olan etkisinin incelenebilmesi için 5 sınıf olarak kategorilendirilen veri seti kullanılarak Şekil 4.13 ve Şekil 4.14'te ısı haritası ve değişkenler arası korelasyon gösterilmiştir. Şekil 4.15 ve Şekil 4.16'da ise bağımsız değişkenlerin anomali durumuna olan etkisinin

incelenebilmesi için ise 2 sınıf olarak kategorilendirilen veri seti kullanılarak ısı haritası ve değişkenler arası korelasyon gösterilmiştir.

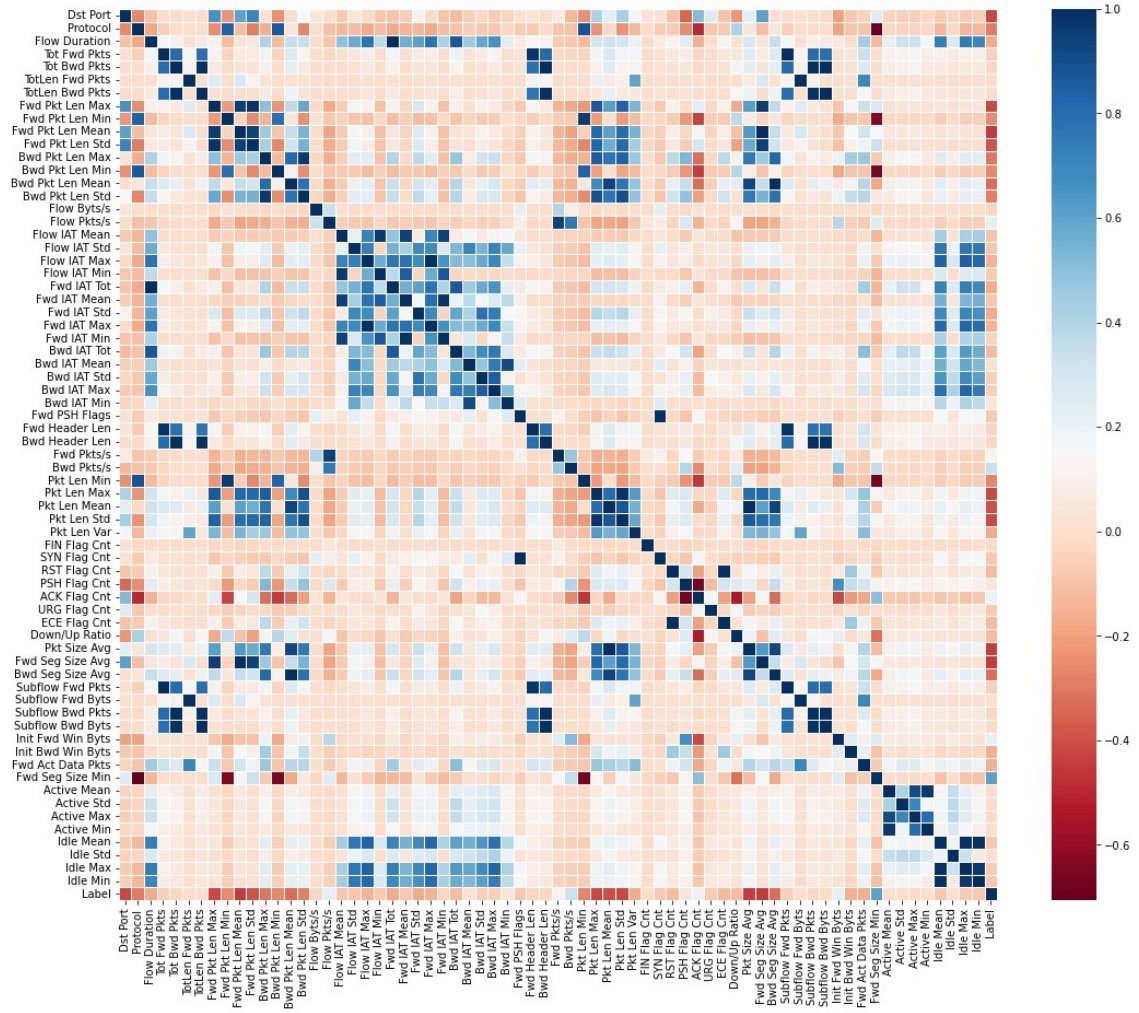


Şekil 4.13. "Senaryo 1" saldırı çeşidine göre oluşturulan ısı haritası

Isı haritasını incelediğimizde saldırı çeşitlerini gösteren "Label" sütunu ile diğer sütunlar arası ilişkilerde [-0,3, +0,3] değerleri aralığının dışındaki değerleri sınır değer olarak kabul edersek ise "Label" sütununu belirleyen etkenler olarak "Dst Port, Fwd Pkt Len Max, Fwd Pkt Len Min, Fwd Pkt Len Mean, Flow Pkts/s, Fwd Pkts/s, Bwd Pkts/s, Pkt Len Mean, Pkt Len Max, Pkt Len Std, PSH Flag Cnt, Pkt Size Avg, Fwd Seg Size Avg, Init Fwd Win Byts, Fwd Seg Size Min" değerlerinin ilişkisinin daha belirgin olduğu görünmektedir. En fazla etkili olan parametrenin ise "Bwd Pkts/s" olduğu görülmüştür.

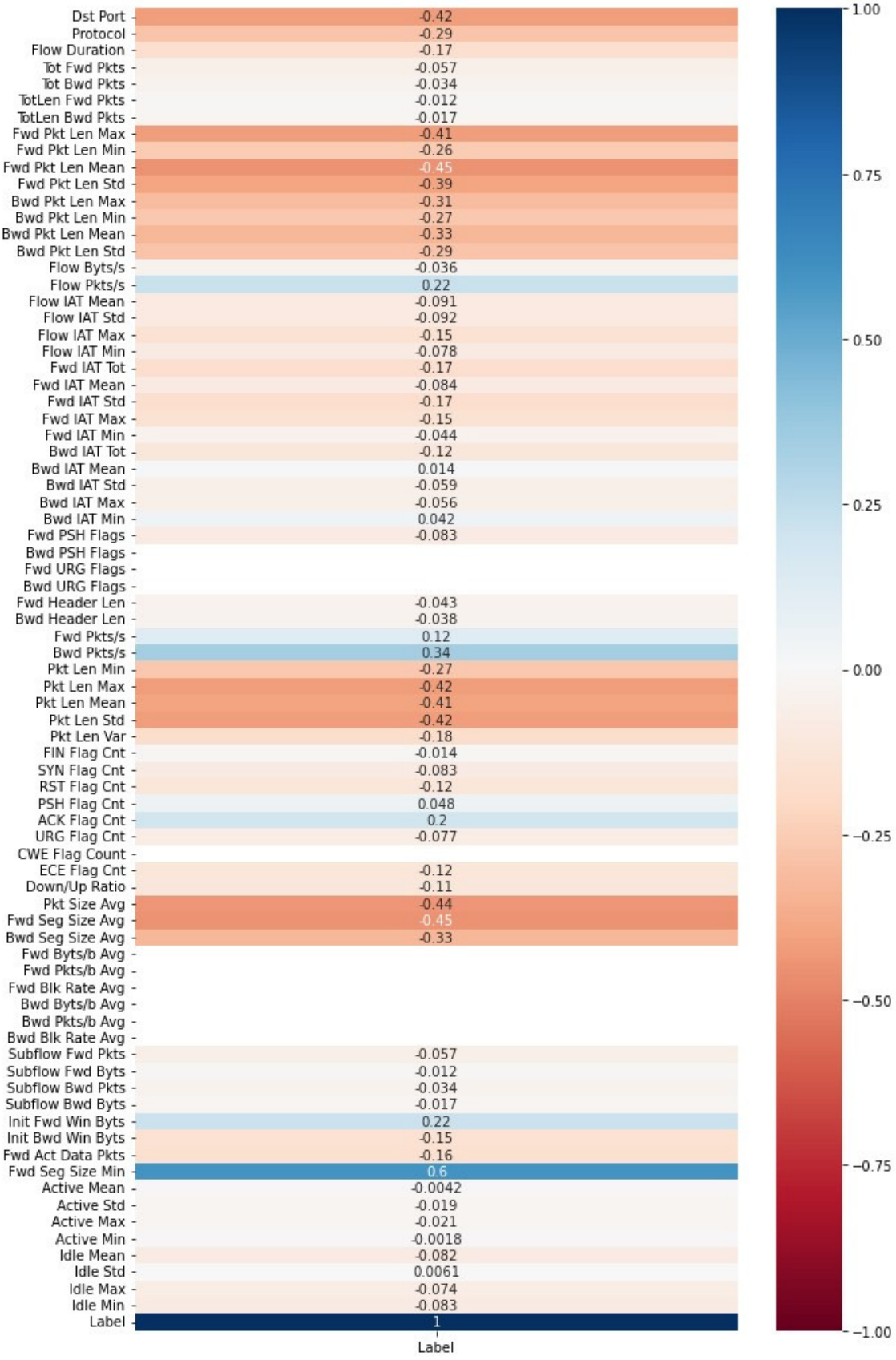


Şekil 4.14. "Senaryo 1" saldırı çeşidine göre oluşturulan "Label" ve değişkenler arası korelasyon



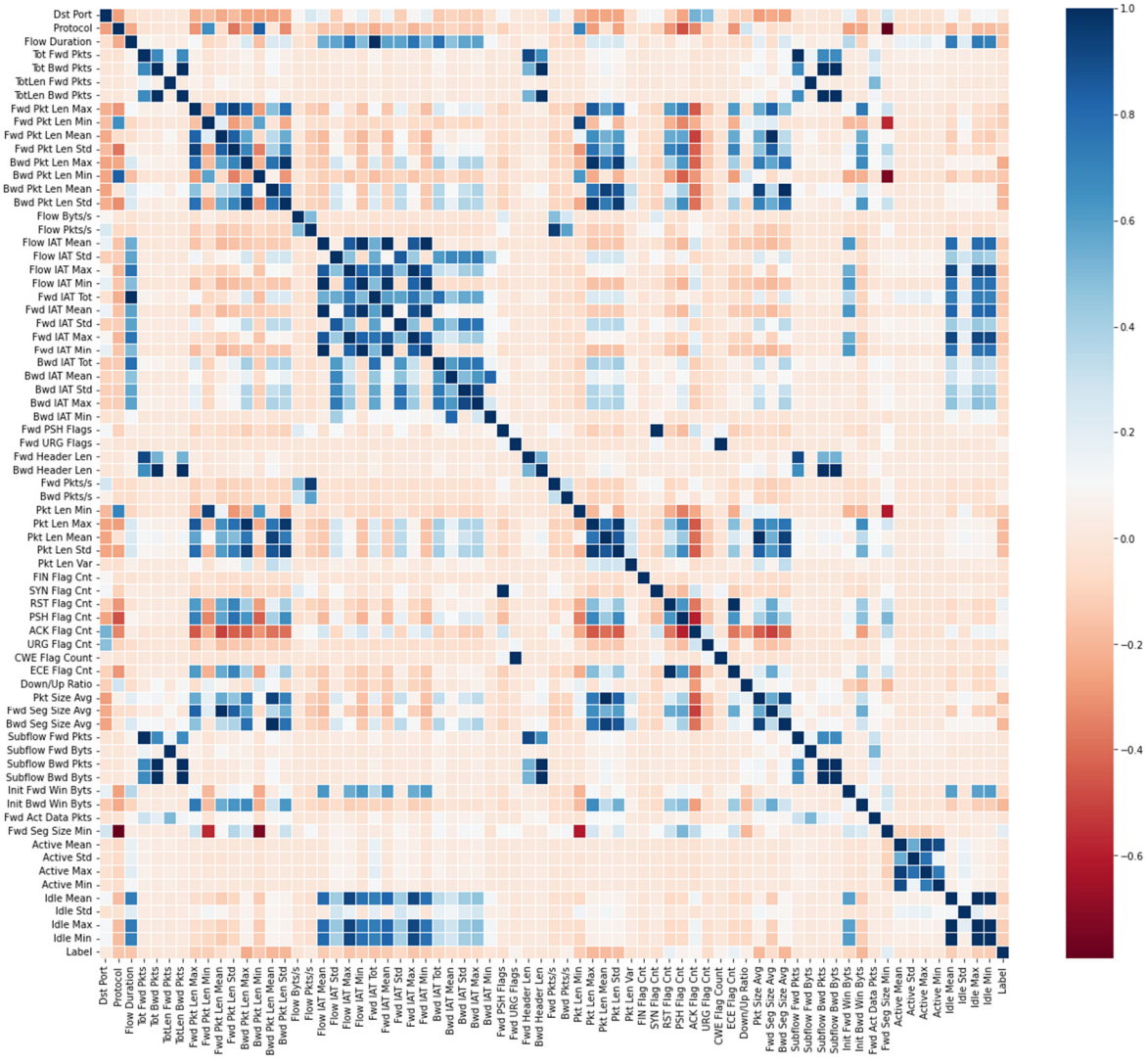
Şekil 4.15. "Senaryo 1" anomali durumuna göre oluşturulan ısı haritası

Isı haritasını incelediğimizde anomali durumunu ifade eden "Label" sütunu ile diğer sütunlar arası ilişkilerde [-0,3, +0,3] değerleri aralığının dışındaki değerleri sınır değer olarak kabul edersek "Dst Port, Fwd Pkt Len Max, Fwd Pkt Len Mean, Fwd Pkt Len Std, Bwd Pkt Len Max, Bwd Pkt Len Mean, Bwd Pkts/s, Pkt Len Max, Pkt Len Mean, Pkt Len Std, Pkt Size Avg, Fwd Seg Size Avg, Bwd Seg Size Avg, Fwd Seg Size Min" değerlerinin ilişkilerinin daha belirgin olduğu görünmektedir. En fazla etkili olan parametrenin ise "Fwd Seg Size Min" olduğu görülmüştür.

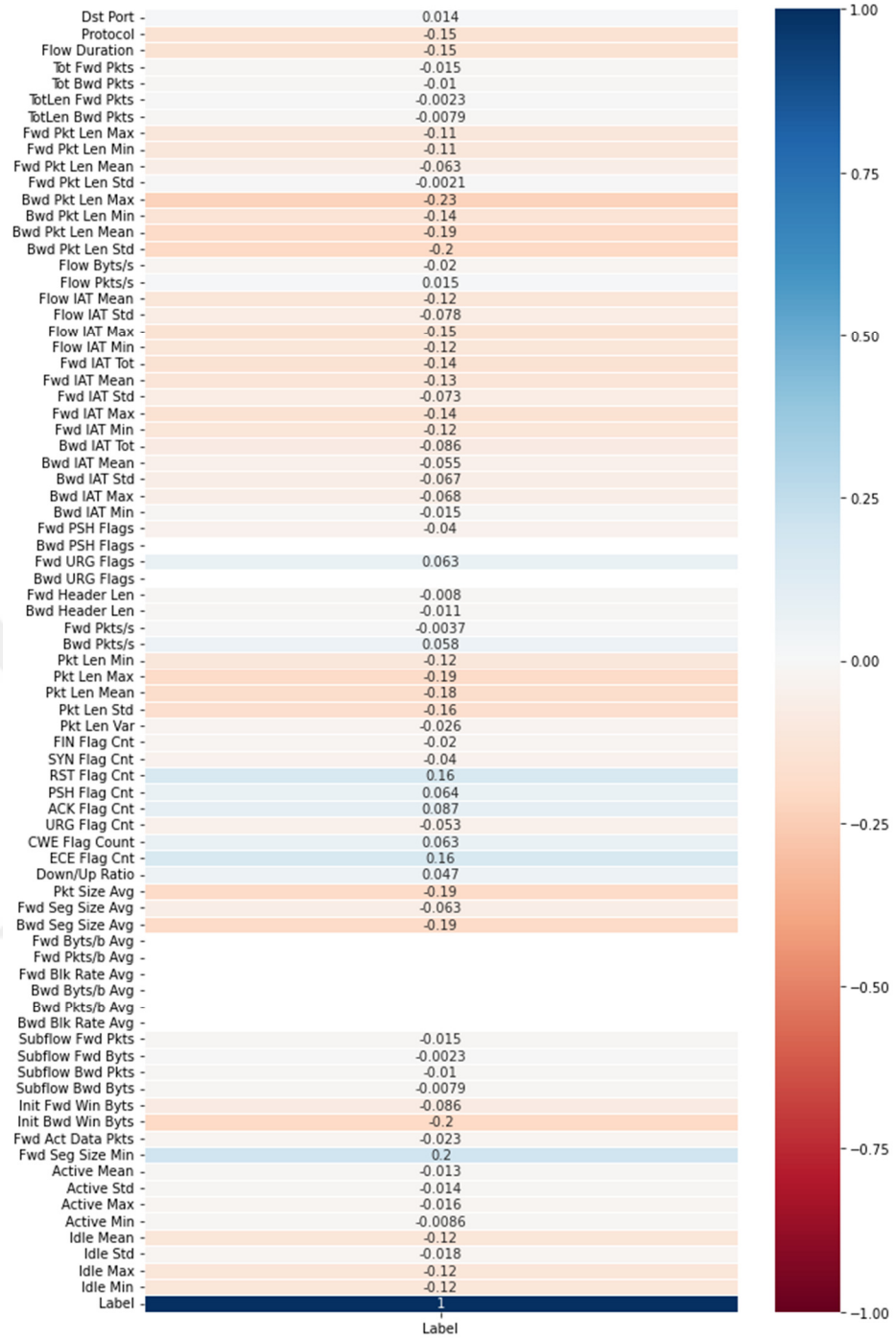


Şekil 4.16. "Senaryo 1" anomali durumuna göre oluşturulan "Label" ve değişkenler arası korelasyon

Senaryo 2 için ısı haritaları oluşturulurken bağımsız değişkenlerin saldırı çeşidine olan etkisinin incelenebilmesi için 3 sınıf olarak kategorilendirilen veri seti kullanılarak Şekil 4.17 ve Şekil 4.18'de ısı haritası ve değişkenler arası korelasyon gösterilmiştir. Şekil 4.19 ve Şekil 4.20'de ise bağımsız değişkenlerin anomali durumuna olan etkisinin incelenebilmesi için ise 2 sınıf olarak kategorilendirilen veri seti kullanılarak ısı haritası ve değişkenler arası korelasyon gösterilmiştir.

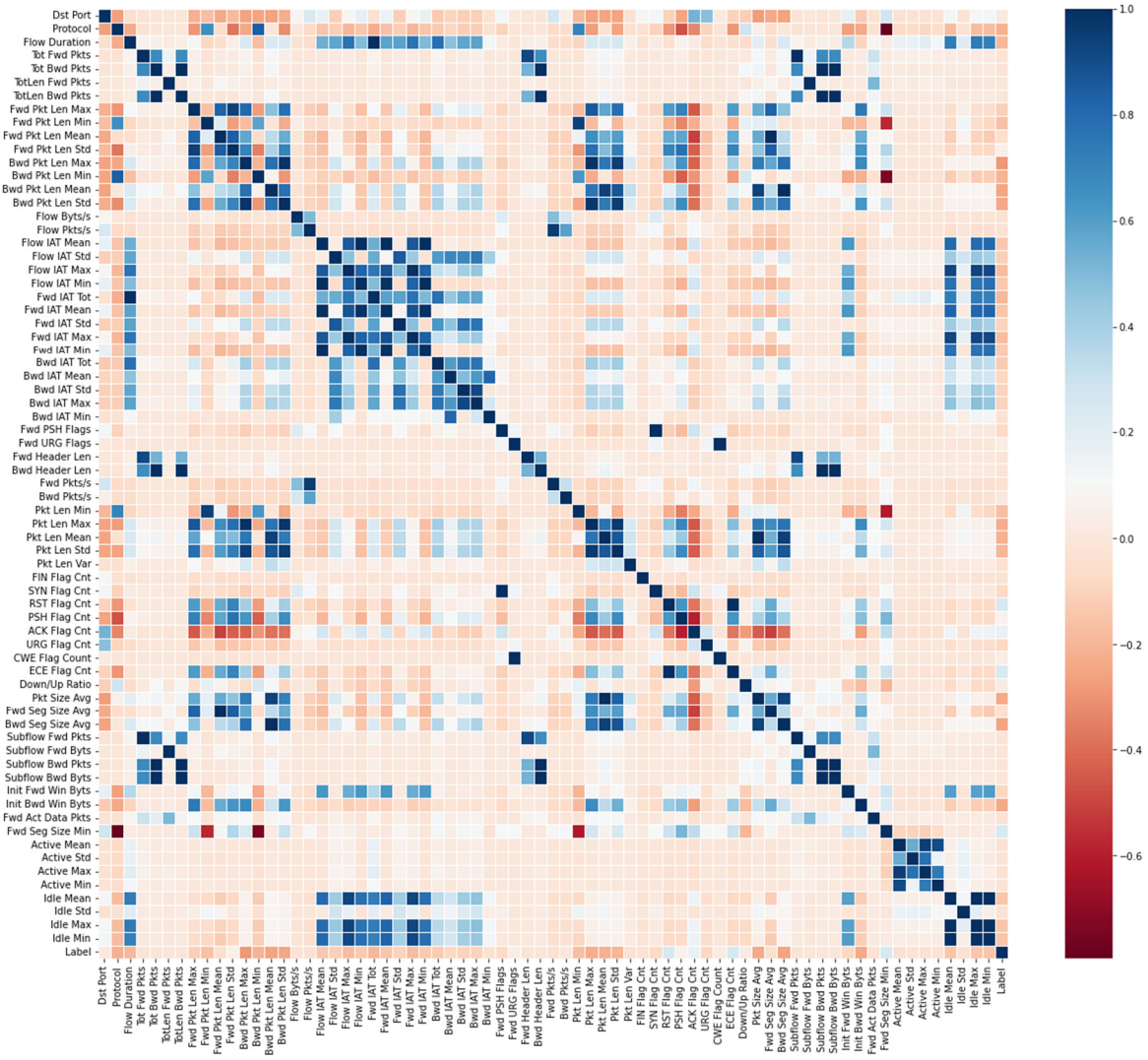


Şekil 4.17. "Senaryo 2" saldırı çeşidine göre oluşturulan ısı haritası



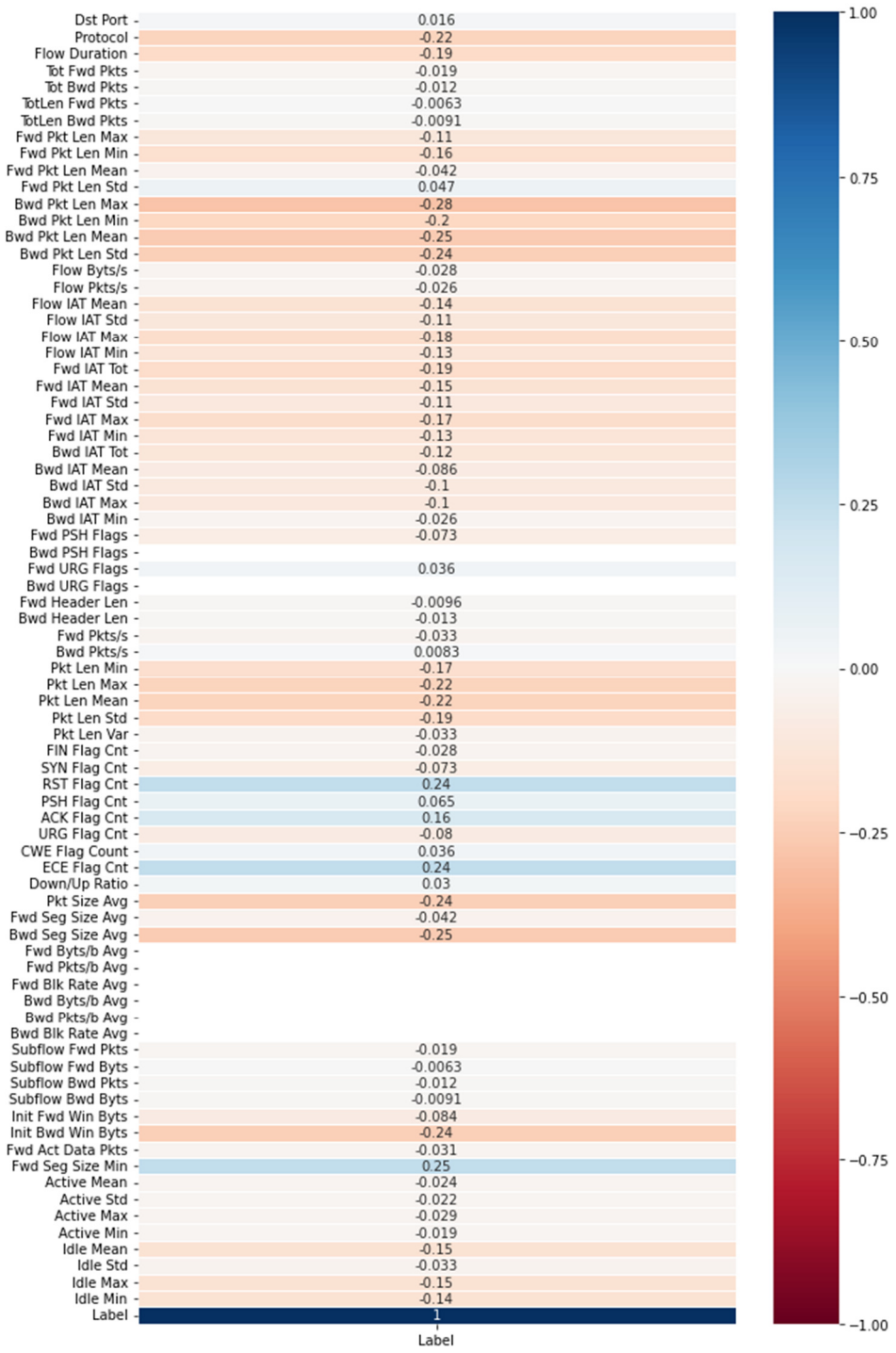
Şekil 4.18. "Senaryo 2" saldırı çeşidine göre oluşturulan "Label" ve değişkenler arası korelasyon

Senaryo 2 için saldırı çeşitlerini gösteren ısı haritasını incelediğimizde "Label" sütunu ile diğer sütunlar arası korelasyon çok düşük olduğu için önemli özellikler ısı haritası ile belirlenmemiştir.



Şekil 4.19. "Senaryo 2" anomali durumuna göre oluşturulan ısı haritası

Senaryo 2 için anomali durumuna göre oluşturulan ısı haritasını incelediğimizde "Label" sütunu ile diğer sütunlar arası korelasyon çok düşük olduğu için önemli özellikler ısı haritası ile belirlenmemiştir.

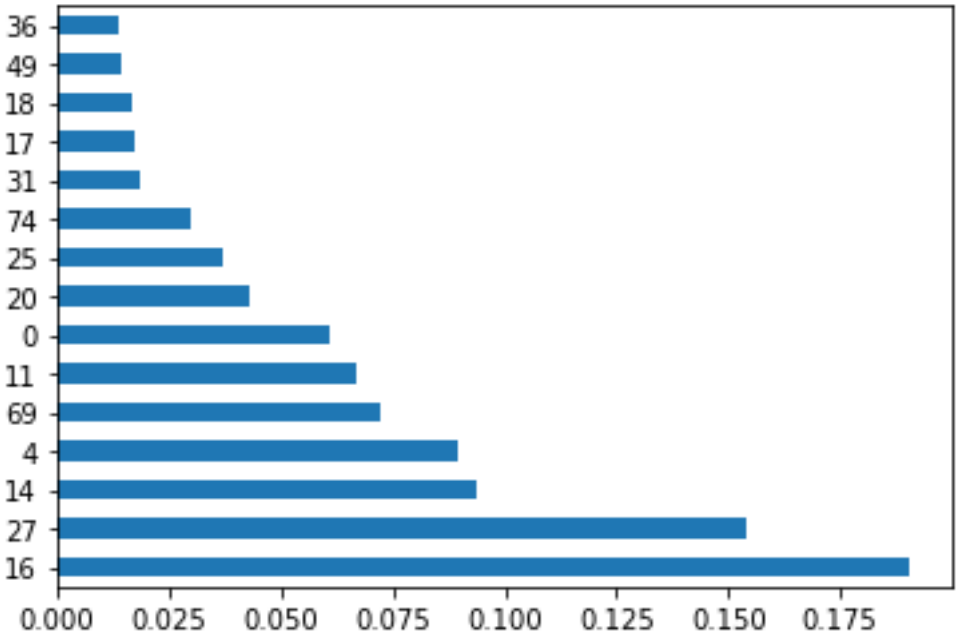


Şekil 4.20. "Senaryo 2" anomali durumuna göre oluşturulan "Label" ve değişkenler arası korelasyon

4.2.4.2. XGBoost Algoritması ile Özellik Seçimi

Özellik seçimi ile önemli özellikler belirlenirken büyük veri setlerinde kullanım kolaylığı sebebiyle tercih edilen XGBoost algoritması kullanılmıştır.

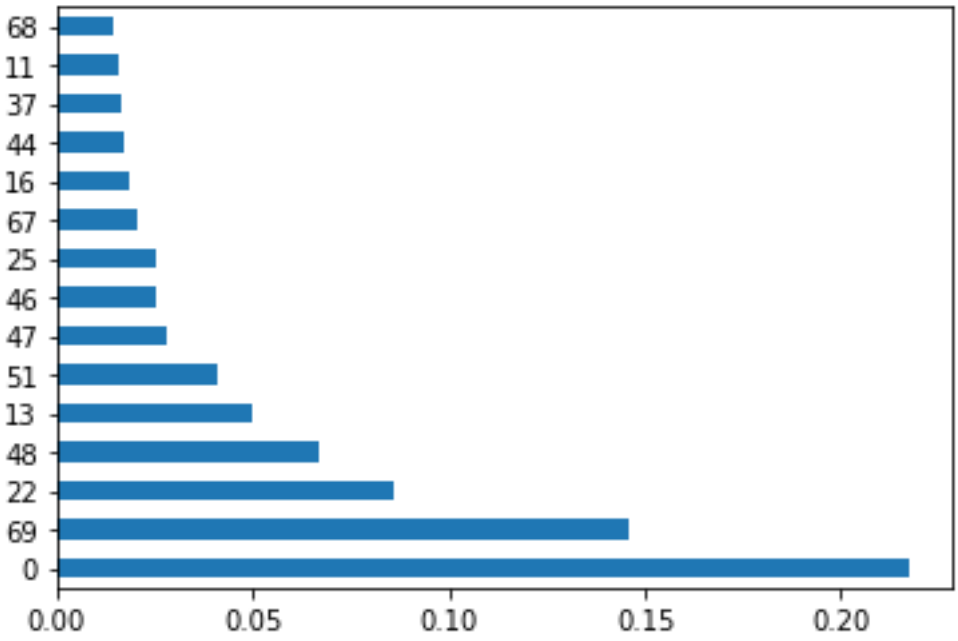
Senaryo 1'de OneSidedSelection yöntemi ile dengeli hale getirilmiş olan veri seti kullanılmıştır. XGB sınıflandırıcı tanımlanırken öğrenme oranı (learning_rate) 0.1, oluşturulacak ağaçların sayısı (n_estimators) 100, maksimum derinlik (max_depth) 6 olarak tanımlanmıştır. Önemli 15 özellik çizdirildiğinde Şekil 4.21 elde edilmiştir.



Şekil 4.21. "Senaryo 1" için XGBoost ile tespit edilen önemli özellikler (feature_importances_)

Burada tespit edilen dizi numaraları Flow Pkts/s, Bwd IAT Mean, Bwd Pkt Len Std, Tot Bwd Pkts, Fwd Seg Size Min, Bwd Pkt Len Max, Dst Port, Flow IAT Min, Fwd IAT Min, Idle Mean, Fwd PSH Flags, Flow IAT Mean, Flow IAT Std, URG Flag Cnt, Bwd Header Len sütunlarına denk düşmektedir.

Senaryo 2'de SMOTETomek yöntemi ile dengeli hale getirilmiş olan veri seti kullanılmıştır. XGB sınıflandırıcı tanımlanırken öğrenme oranı (learning_rate) 0.1, oluşturulacak ağaçların sayısı (n_estimators) 100, maksimum derinlik (max_depth) 6 olarak tanımlanmıştır. Önemli 15 özellik çizdirildiğinde Şekil 4.22 elde edilmiştir.

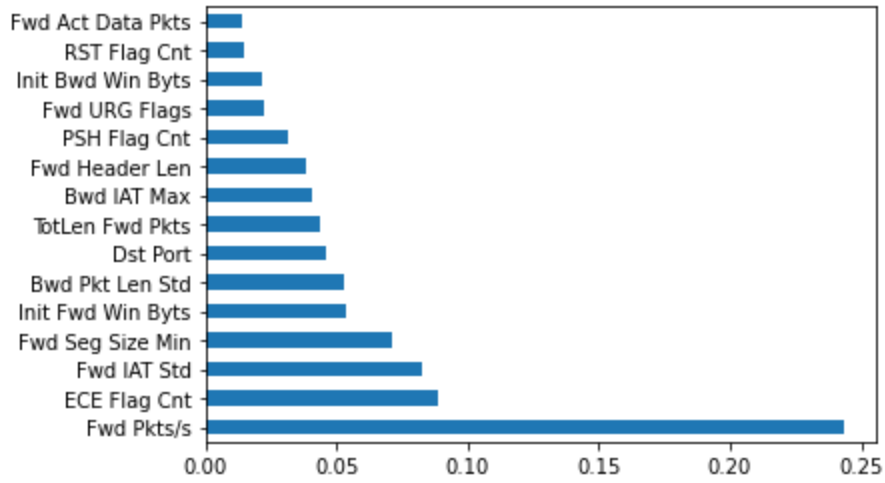


Şekil 4.22. "Senaryo 2" için XGBoost ile tespit edilen önemli özellikler (feature_importances_)

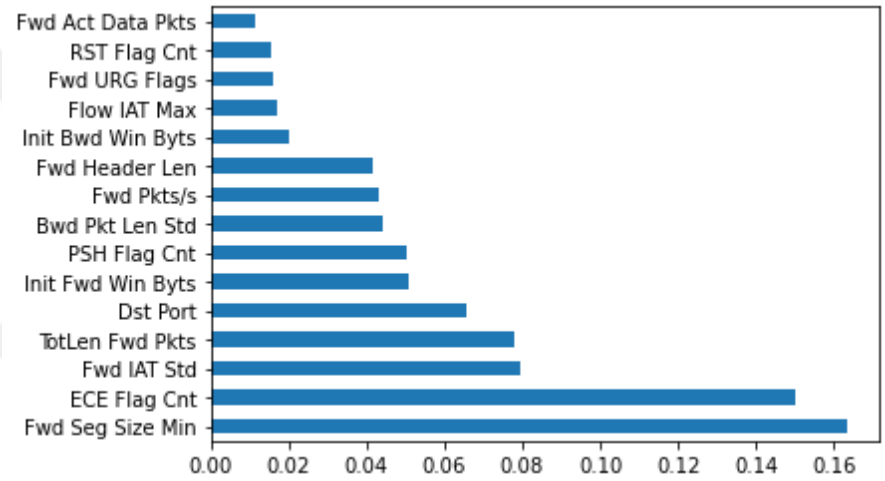
Burada tespit edilen dizi numaraları Dst Port, Fwd Seg Size Min, Fwd IAT Mean, ACK Flag Cnt, Bwd Pkt Len Mean, ECE Flag Cnt, PSH Flag Cnt, RST Flag Cnt, Fwd IAT Min, Init Bwd Win Byts, Flow Pkts/s, FIN Flag Cnt, Fwd Pkts/s, Bwd Pkt Len Max, Fwd Act Data Pkts sütunlarına denk düşmektedir.

4.2.5. Veri Setinin Görselleştirilmesi

CSE-CIC-IDS2018 veri setinde saldırı türünün belirlenmesinde etkili olan özelliklerin gözlemlenebilmesi için OSS yöntemi ile azaltılan veri setinin tamamı kullanılarak dağılım grafikleri çizdirilmiştir. XGBoost algoritması ile özellik seçimi sağlanarak etiketin belirlenmesinde etkili olan 15 özellik incelenmiştir. Saldırı çeşidine göre önemli özellikler tespit edildiğinde Şekil 4.23, saldırı grubuna göre kategorilendirme yapıldığında (Benign, Bot, Infiltration, DDoS, Brute Force, SQL Injection, DoS) Şekil 4.24'deki özellikler tespit edilmiştir.

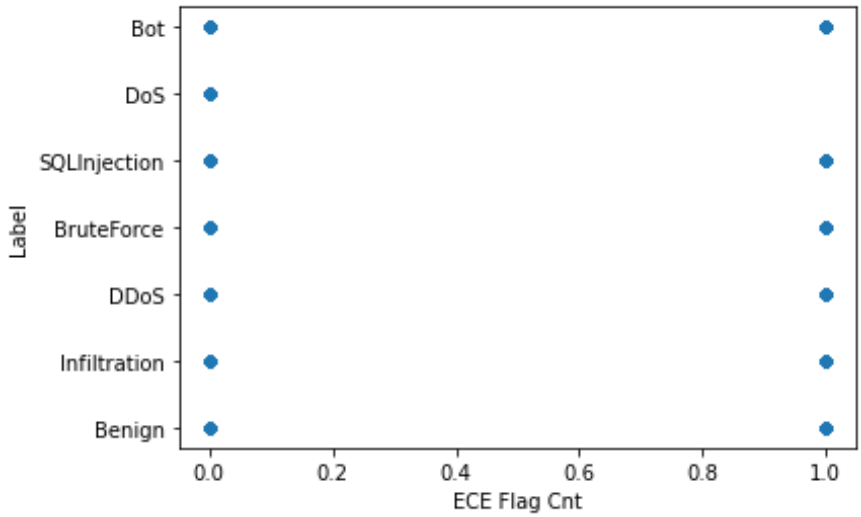


Şekil 4.23. "Senaryo 3" için XGBoost ile tespit edilen önemli özellikler (feature_importances_)



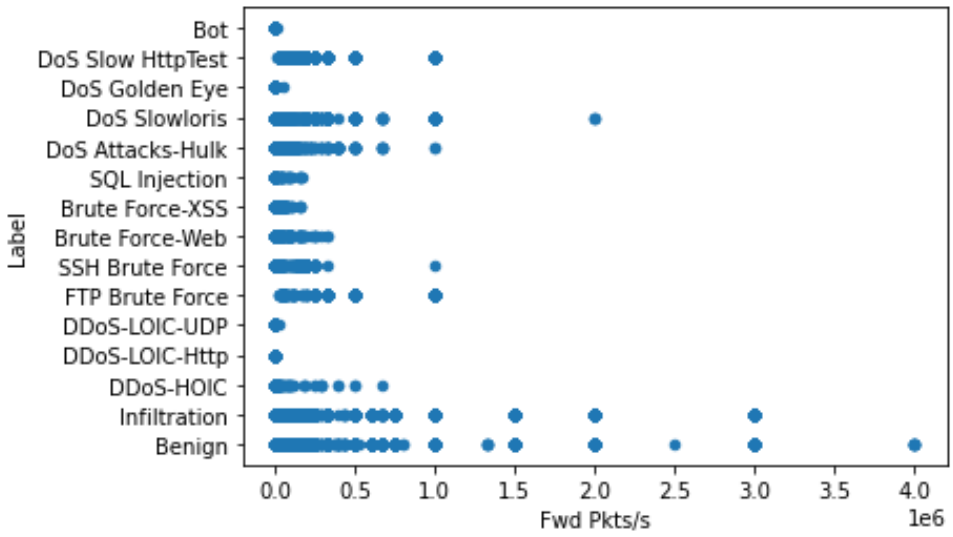
Şekil 4.24. "Senaryo 3" için saldırı gruplarına göre XGBoost ile tespit edilen önemli özellikler (feature_importances_)

Özellik seçimine bakılarak gruplarına göre saldırı türünün belirlenmesinde etkili olan niteliklerden "ECE Flag Cnt" özelliğinin etikete olan etkisi Şekil 4.25'te gösterilmiştir. ECE (Explicit Congestion Notification-Echo) bayrağı TCP segment başlığında yer almakta ve TCP uç noktalarının ECN uyumlu olduğunu göstermektedir. TCP protokolü gecikmeden ziyade paket iletiminin kayıpsız olarak gerçekleştirilmesini, güvenli iletişimi önemsemektedir. Açık tıkanıklık bildirimi, TCP üçlü el sıkışmanın düşük gecikmeli olarak gerçekleştirilebilmesi için gönderici ve alıcı arasında ortalama paket boyutu belli bir seviyenin üzerine çıktığında paket kaybı yaşanmadan işaretleyerek alıcıya ağ tıkanıklığının bildirilmesini sağlamaktadır (Luo ve ark., 2017).



Şekil 4.25. "ECE Flag Cnt" ve "Label" ilişkisi

"Fwd Pkts/s" niteliği saniyede gönderilen ileri yöndeki paket sayısını göstermektedir. Etikete göre dağılımı Şekil 4.26'da gösterilmiştir.

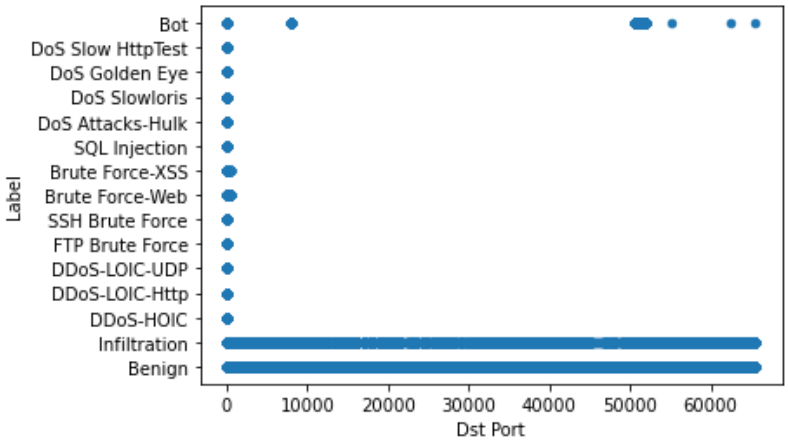


Şekil 4.26. "Fwd Pkts/s" ve "Label" ilişkisi

CSE-CIC-IDS2018 veri setinde hedef portlara göre saldırı türleri incelendiğinde (Şekil 4.27);

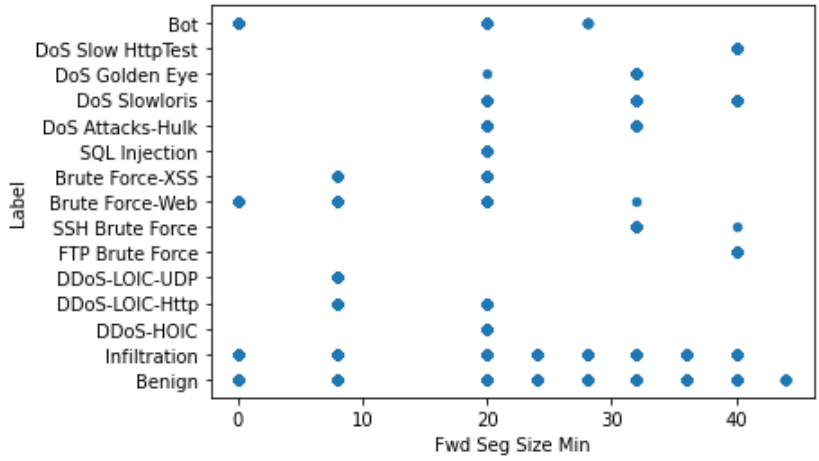
- Bot saldırıları 8080, 0 ve dinamik veya özel portlar denilen geçici bağlantı noktaları olan 49152 ile 65535 arası hedef portlar üzerinden gerçekleştirilmiştir. 8080 portu kullanıcı işlemleri, uygulamalar için ayrılmış kayıtlı port numarası (registered ports) adreslerindendir ve HTTP port numarası 80'e alternatif olarak kullanılmaktadır (Schneider, 1997).

- DoS saldırıları çeşitlerinden olan Golden Eye, Slowloris, Hulk saldırıları 80 hedef portu üzerinden, Slow HttpTest saldırısı ise 21 portu üzerinden gerçekleştirilmiştir. İyi bilinen port numaralarından (well known ports) olan 80 portu HTTP (Hyper Text Transfer Protocol) hizmetinde, 21 port numarası FTP (File Transfer Protocol) hizmetinde kullanılmaktadır (Schneider, 1997).
- DDoS LOIC UDP, LOIC HTTP, HOIC saldırıları 80 hedef portuna gerçekleştirilmiştir.
- SQL Injection saldırıları 80 hedef portuna gerçekleştirilmiştir.
- FTP Brute Force saldırıları 21 portundan; SSH Brute Force saldırıları 22 ve 21 portundan; Brute Force-XSS saldırıları 80, 500 ve 67 portlarından; Brute Force-Web saldırıları 80, 500, 67, 0, 22 hedef portundan gerçekleştirilmiştir. 67 portu BOOTP (BootStrap Protocol) ve DHCP (Dynamic Host Configuration Protocol) sunucu hizmeti için kullanılmaktadır. Bu protokol, diski olmayan bir istemci makinenin yada ilk kez başlatılan bir bilgisayarın ip adresi almasında, BOOTP sunucusundan önyükleme görüntüsü alarak işletim sisteminin yüklenmesinde görev almaktadır (Croft ve Gilmore, 1985). 500 portu IPSec için kullanılan portlardandır, internet üzerinde güvenli bir haberleşmenin sağlanabilmesi için Internet Güvenliği İlişkilendirmesi ve Anahtar Yönetimi ProtokolüI (ISAKMP), İnternet Anahtar Değişimi (IKE) protokolleri tarafından kullanılmaktadır (Maughan ve ark., 1998).



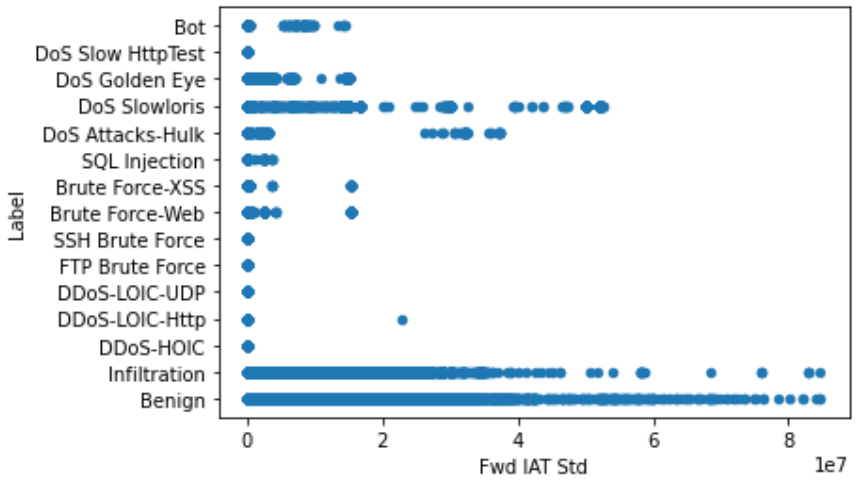
Şekil 4.27. "Dst Port" ve "Label" ilişkisi

Etirkete göre ileri yönde iletilen minimum segment boyutu dağılımı Şekil 4.28'de gösterilmiştir.



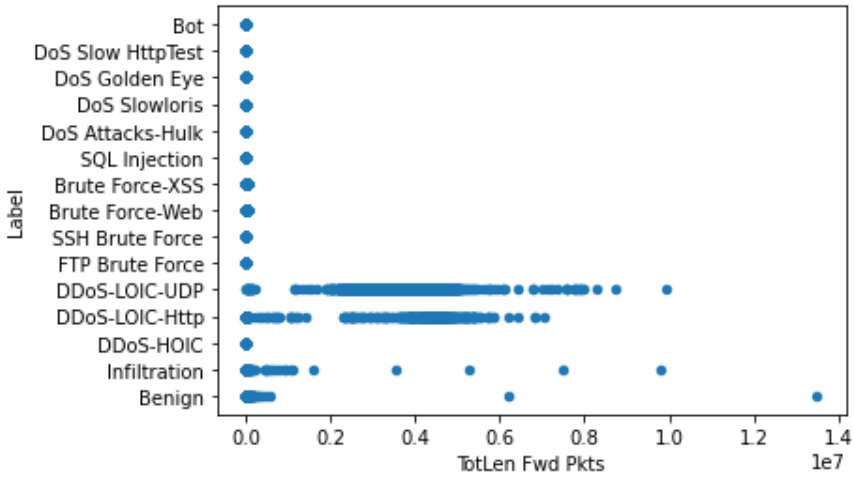
Şekil 4.28. "Fwd Seg Size Min" ve "Label" ilişkisi

Etikete göre ileri yönde gönderilen iki paket arasındaki standart sapma süresi dağılımı Şekil 4.29'da gösterilmiştir.



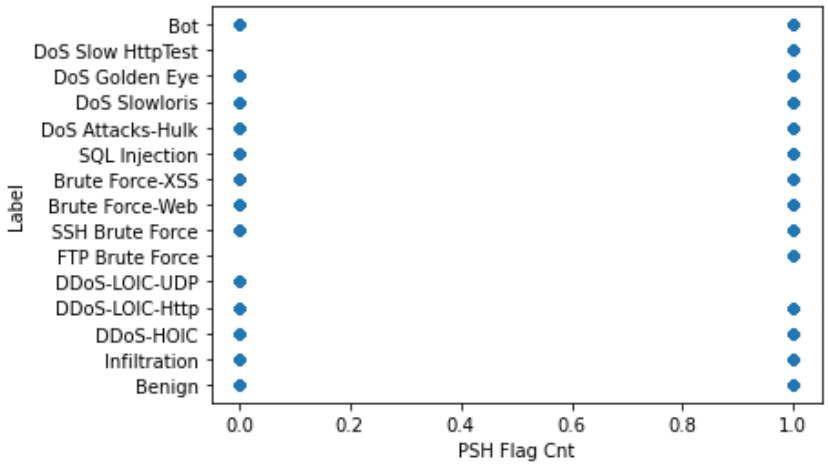
Şekil 4.29. "Fwd IAT Std" ve "Label" ilişkisi

Etikete göre ileri yönde iletilen paketlerin toplam uzunluğu dağılımı Şekil 4.30'da gösterilmiştir.



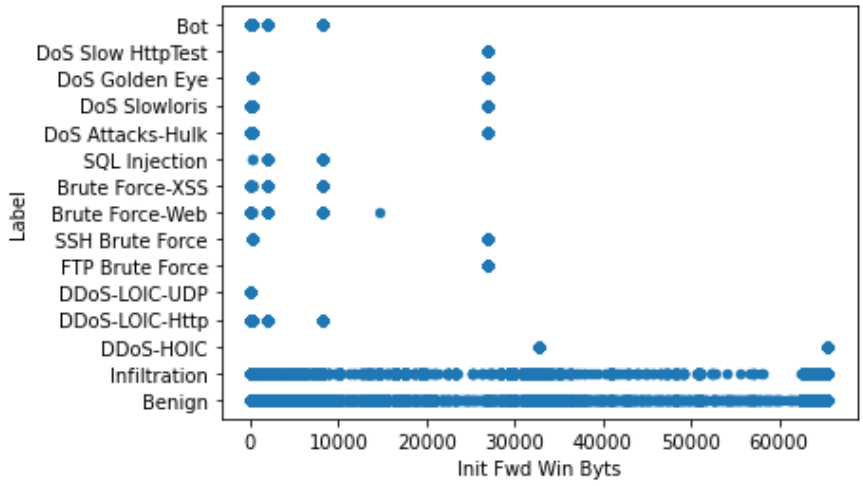
Şekil 4.30. "TotLen Fwd Pkts" ve "Label" ilişkisi

TCP iletişiminde iletilen paketlerin maksimum segment boyutu 1460 byte'dır. Normal şartlarda TCP iletişiminde veri iletiminin optimize edilebilmesi için arabelleğe alınarak iletim sağlanmaktadır. PSH (Push) bayrağı etkin olduğunda ise gönderilecek veri maksimum segment boyutundan küçük olduğunda segment boyutuna ulaşması beklenmeden, doğrudan paketler halinde gönderilmektedir. Bu bayrak kesintisiz iletişimi sağladığı için oyunlarda, gerçek zamanlı ses ve video uygulamalarda önemlidir (Wesley, 2022). PSH bayrak durumu ile etiket arasındaki ilişki Şekil 4.31'de gösterilmiştir.



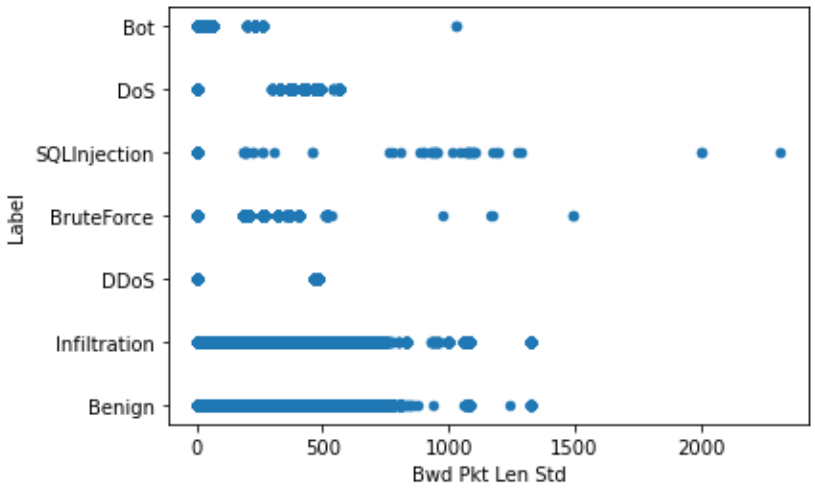
Şekil 4.31. "PSH Flag Cnt" ve "Label" ilişkisi

Etikete göre ileri yönde ilk çerçeve içinde gönderilen byte sayısı dağılımı Şekil 4.32'de gösterilmiştir.



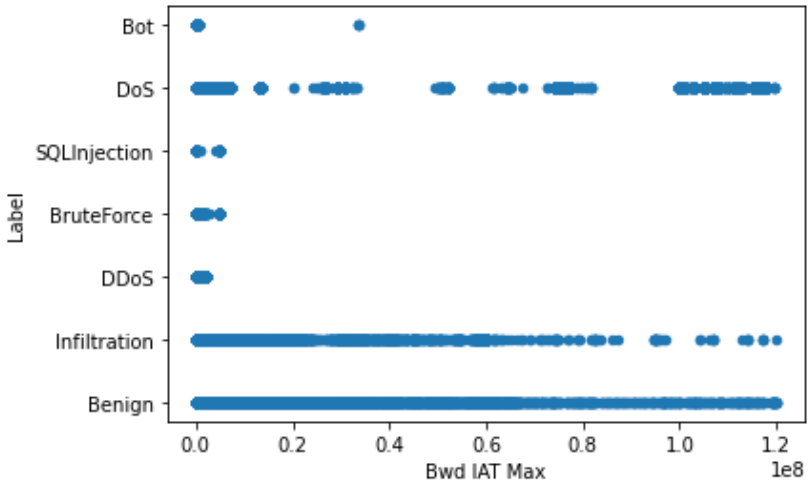
Şekil 4.32. "Init Fwd Win Byts" ve "Label" ilişkisi

Etikete göre geri yönde iletilen paketin standart sapma boyutu dağılımı Şekil 4.33'de yer almaktadır.



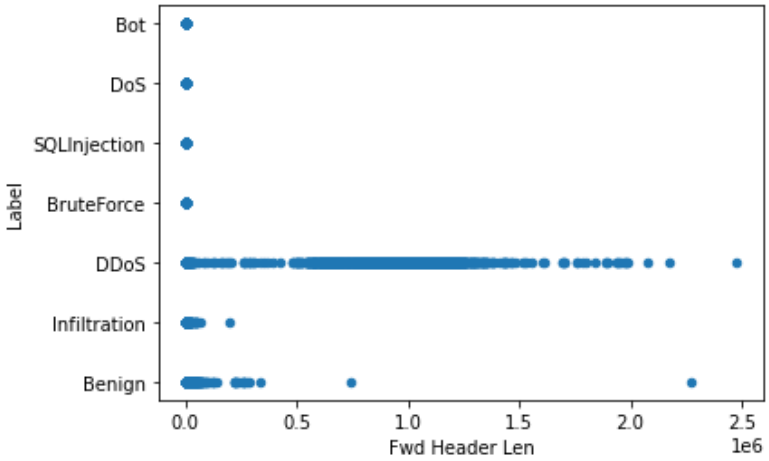
Şekil 4.33. "Bwd Pkt Len Std" ve "Label" ilişkisi

Etikete göre geri yönde gönderilen iki paket arasındaki maksimum süre dağılımı Şekil 4.34'de yer almaktadır.



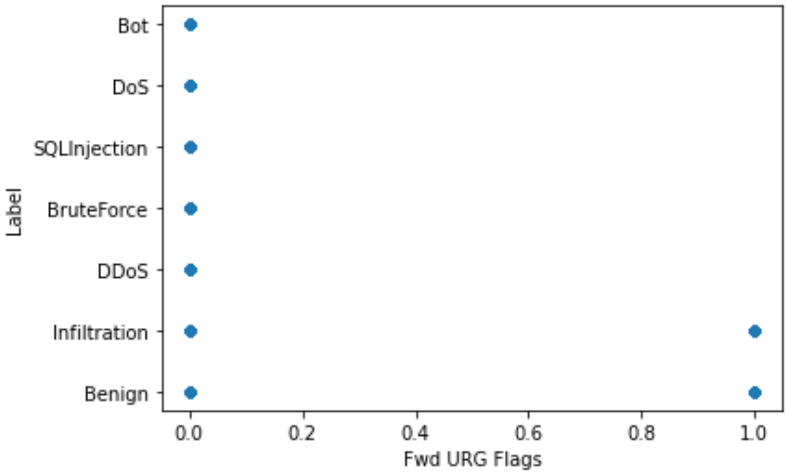
Şekil 4.34. "Bwd IAT Max" ve "Label" ilişkisi

Etikete göre ileri yönde iletilen başlıklar için kullanılan toplam başlık boyutu dağılımı Şekil 4.35'te yer almaktadır.



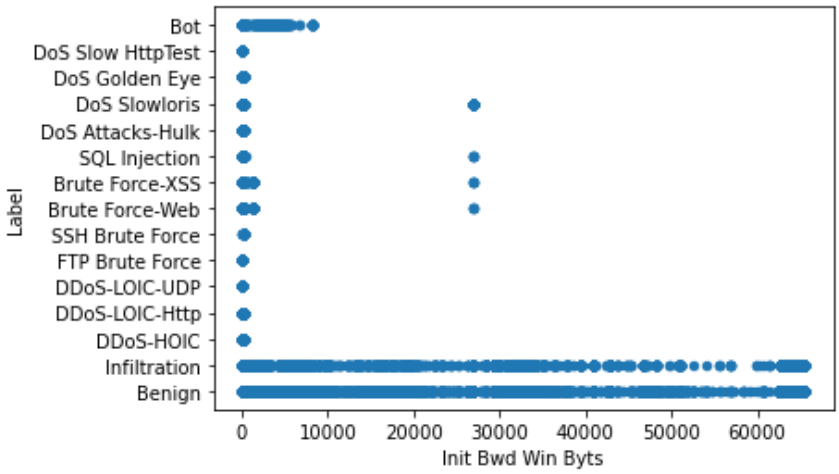
Şekil 4.35. "Fwd Header Len" ve "Label" ilişkisi

İleri yönde iletilen paketlerde URG (Urgent) bayrağının ayarlanma sayısı ile etiket arasındaki ilişki Şekil 4.36'da yer almaktadır. TCP haberleşmesinde URG bayrağı nadir olarak kullanılmaktadır. Bir segmentteki verinin acil olarak iletilmesi, öncelik verilmesi gerektiği durumlarda kullanılır (Wesley, 2022).



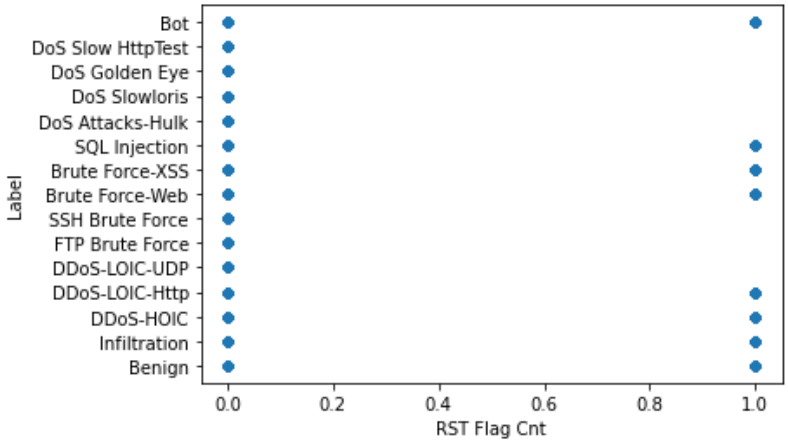
Şekil 4.36. "Fwd URG Flags" ve "Label" ilişkisi

Etikete göre geri yönde iletilen ilk pencere içinde gönderilen byte sayısı dağılımı Şekil 4.37'de gösterilmiştir.



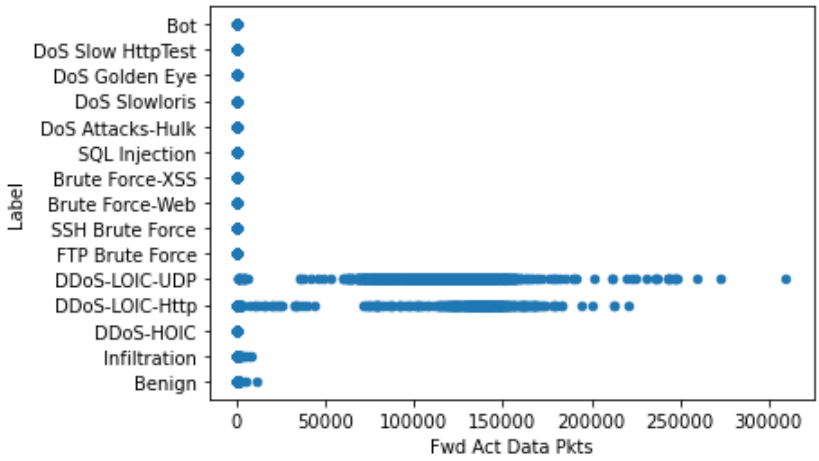
Şekil 4.37. "Init Bwd Win Byts" ve "Label" ilişkisi

RST bayrağı sıfırlama anlamına gelmektedir ve istemci ile sunucu arasındaki TCP bağlantısında bir sorun olduğunda bağlantının sonlandırılması için kullanılmaktadır (Wesley, 2022). Etikete göre RST bayrağı dağılımı Şekil 4.38'de gösterilmiştir.



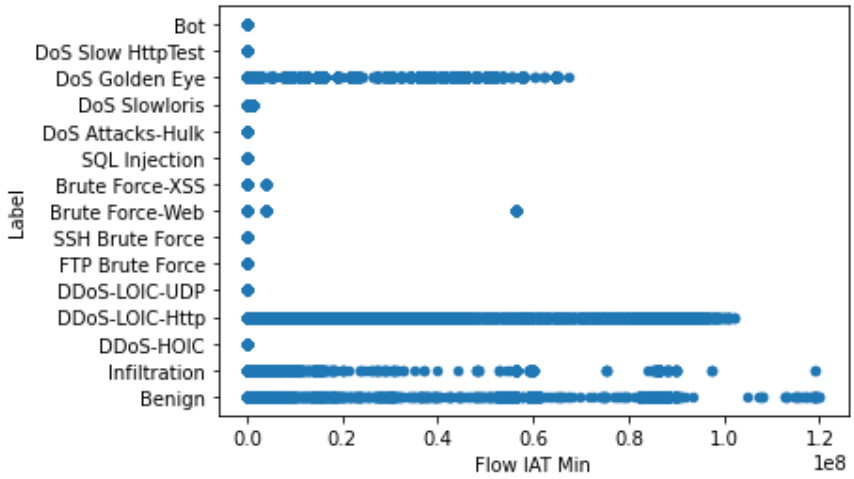
Şekil 4.38. "RST Flag Cnt" ve "Label" ilişkisi

Etikete göre ileri yönde iletilen en az 1 byte TCP veri yüküne sahip paket sayısı dağılımı Şekil 4.39'da gösterilmiştir.



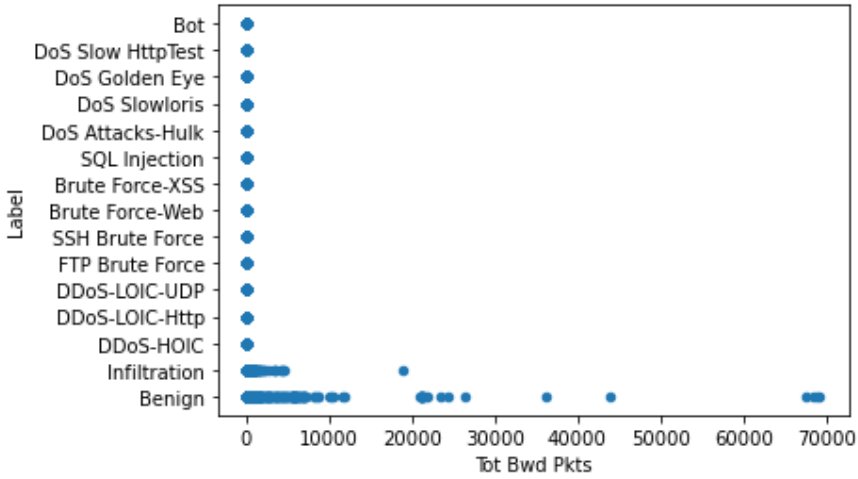
Şekil 4.39. "Fwd Act Data Pkts" ve "Label" ilişkisi

Etikete göre iki akış arasındaki minimum süre dağılımı Şekil 4.40'da gösterilmiştir.



Şekil 4.40. "Flow IAT Min" ve "Label" ilişkisi

Etikete göre geri yönde iletilen toplam paket sayısı dağılımı Şekil 4.41'de gösterilmiştir.



Şekil 4.41. "Tot Bwd Pkts" ve "Label" ilişkisi

4.3. Modelin Oluşturulması ve Performansının Değerlendirilmesi

4.3.1. K En Yakın Komşu (KNN-K Nearest Neighbor) Algoritması

Çalışmamızın bu kısmında "4.2. Veriler arası ilişkilerin tespit edilmesi" başlığı altında belirlenen saldırı çeşidine ve anomali durumuna etki eden değişkenler makine öğrenmesi tekniklerinden K En Yakın Komşu algoritmasına tabi tutularak model başarısı incelenmiştir.

İlk olarak eğitim ve test olamak üzere veri seti bölme işlemi gerçekleştirilir. Bölme işlemini gerçekleştirirken veri seti büyüklüğünden ve zaman kıstasından dolayı ayırarak çapraz doğrulama (holdout cross validation) yöntemi kullanılmıştır. "test_size=0,32" ve "random_state=42" olarak ayarlanarak 2.089.122 adet veriden rasgele olarak 1.420.602 adeti eğitim verisi, 668520 adeti test verisi olarak belirlenmiştir. Veri bölme işleminden sonra "StandardScaler" fonksiyonu ile eğitim verisinde normalizasyon işlemi gerçekleştirilmiştir. "StandardScaler" fonksiyonu veriler arasındaki ilişkiyi bozmadan özniteliklerin ölçeklendirilmesini sağlar bu sayede veri seti daha iyi anlamlandırılabilir.

KNN algoritması ile sınıflandırma işleminin gerçekleştirilebilmesi için "KNeighborsClassifier" kütüphanesi dahil edilmiştir ve en yakın komşulara göre yakınlığının belirlenebilmesi için "n_neighbors" parametresi varsayılan değer 5 ve 3 olarak girilmiştir. Veri normalizasyonu ile değişen değerler örneği Şekil 4.42'de gösterilmiştir.

80	0	0	-0.620222	-0.702041	-0.38368
48634	935	0	1.61411	1.65802	-0.38368
58990	935	0	2.09067	1.65802	-0.38368

Şekil 4.42. Veri normalizasyonu

DoS saldırılarının tespit edilmesi ve anomali durumunun tespit edilmesi için hazırlanan modellerin başarı durumlarının gözlemlenebilmesi precision (kesinlik: doğru olarak tahmin edilen değerlerin tüm doğru tahmin değerlerine oranı), recall (duyarlılık: doğru tahmin edilen değerlerin gerçekteki doğru değerlere oranı) ve f1 score (f skor: duyarlılık ve kesinlik değerlerinin harmonik ortalaması) değerleri hesaplanmıştır.

4.3.1.1. Senaryo 1 – KNN

"Senaryo 1" için KNN modeli ile saldırı tespitinin gerçekleştirilmesine yönelik testler aşağıda sunulmuştur:

1a) Isı haritası ile özellik seçimi yapılarak saldırı türünün tespitini sağlayan KNN modelinin oluşturulması

Veri seti girdisi olarak "Isı Haritası ile Korelasyon Matrisi" başlığı altındaki Şekil 4.13 ve Şekil 4.14'de belirlenen özellikler kullanılmıştır. En yakın kaç komşusuna göre

sınıflandırma işleminin sağlanabilmesi için "n_neighbors" parametresi 5 olarak tanımlanmıştır. Karışıklık matrisi ve sınıflandırma raporu Çizelge 4.12, Çizelge 4.11 yer almaktadır.

Çizelge 4.11. 1a maddesine ilişkin sınıflandırma raporu (classification report)

Saldırılar/Metrikler	Precision	Recall	F1	Support
SlowHTTPTest	1.000	1.000	1.000	44559
GoldenEye	0.999	0.999	0.999	13355
SlowLoris	0.999	0.999	0.999	3478
Hulk	1.000	1.000	1.000	147938
Benign	1.000	1.000	1.000	459190

Çizelge 4.12. 1a maddesine ilişkin karışıklık matrisi (confusion matrix)

	Benign	Hulk	SlowLoris	GoldenEye	SlowHTTPTest
Benign	459175	4	5	5	1
Hulk	0	147930	0	8	0
SlowLoris	3	0	3475	0	0
GoldenEye	1	8	0	13346	0
SlowHTTPTest	0	0	0	0	44559

1b) Isı haritası ile özellik seçimi yapılarak anomali durumunun tespitini sağlayan KNN modelinin oluşturulması

Veri seti girdisi olarak "Isı Haritası ile Korelasyon Matrisi" başlığı altındaki Şekil 4.15 ve Şekil 4.16'de belirlenen özellikler kullanılmıştır. Anomali durumunun tespit edilebilmesi için hazırlanan modelde ise "n_neighbors=3" olarak seçilmiştir ve algoritma çalışma süresi çok sınıflı modelden daha uzundur. Sınıflandırma raporu ve karışıklık matrisi Çizelge 4.13, Çizelge 4.14 yer almaktadır.

Çizelge 4.13. 1b maddesine ilişkin sınıflandırma raporu (classification report)

Saldırılar/Metrikler	Precision	Recall	F1	Support
Attack	1.000	1.000	1.000	209330
Benign	1.000	1.000	1.000	459190

Çizelge 4.14. 1b maddesine ilişkin karışıklık matrisi (confusion matrix)

	Benign	Attack
Benign	459175	4
Attack	0	147930

1c) Özellik seçimi yapılmadan saldırı türü tespitini sağlayan KNN modelinin oluşturulması

En yakın komşularına göre sınıflandırma işleminin sağlanabilmesi için "n_neighbors" parametresi 5 olarak tanımlanmıştır. Karışıklık matrisi ve sınıflandırma raporu Çizelge 4.16, Çizelge 4.15 yer almaktadır.

Çizelge 4.15. 1c maddesine ilişkin sınıflandırma raporu (classification report)

Saldırılar/Metrikler	Precision	Recall	F1	Support
Benign	1.000	1.000	1.000	459190
Hulk	1.000	1.000	1.000	147938
SlowLoris	1.000	1.000	1.000	3478
GoldenEye	1.000	1.000	1.000	13355
SlowHTTPTest	1.000	1.000	1.000	44559

Çizelge 4.16. 1c maddesine ilişkin karışıklık matrisi (confusion matrix)

	Benign	Hulk	SlowLoris	GoldenEye	SlowHTTPTest
Benign	459162	1	5	21	1
Hulk	1	147936	0	1	0
SlowLoris	1	0	3477	0	0
GoldenEye	6	5	0	13344	0
SlowHTTPTest	0	0	0	0	44559

1d) One Side Selection yöntemi ile dengelenen veri ile saldırı türü tespitini sağlayan KNN modelinin oluşturulması

Bu testte "4.2.3 Dengesiz Verinin Dengeli Hale Getirilmesi" başlığı altında belirtildiği üzere OSS yöntemi ile örneklem azaltılarak dengeli hale getirilen veri seti kullanılmıştır. (Şekil 4.6) Sınıflandırma raporu ve karışıklık matrisi Çizelge 4.17 ve Çizelge 4.18'de yer almaktadır.

Çizelge 4.17. 1d maddesine ilişkin sınıflandırma raporu (classification report)

Saldırılar/Metrikler	Precision	Recall	F1	Support
SlowHTTPTest	1.00	1.00	1.00	44559
GoldenEye	0.97	1.00	0.98	13355
SlowLoris	0.98	1.00	0.99	3478
Hulk	1.00	1.00	1.00	147938
Benign	1.00	1.00	1.00	459190

Çizelge 4.18. 1d maddesine ilişkin karışıklık matrisi (confusion matrix)

	Benign	Hulk	SlowLoris	GoldenEye	SlowHTTPTest
Benign	459175	4	5	5	1
Hulk	0	147930	0	8	0
SlowLoris	3	0	3475	0	0
GoldenEye	1	8	0	13346	0
SlowHTTPTest	0	0	0	0	44559

4.3.1.2. Senaryo 2 – KNN

"Senaryo 2" için KNN modeli ile saldırı tespitinin gerçekleştirilmesine yönelik testler aşağıda sunulmuştur:

2a) Özellik seçimi yapılmadan saldırı türünün tespitini sağlayan KNN modelinin oluşturulması

Isı haritasında [-0.3, 0.3] aralığı dışında özellikler tespit edilemediği için ısı haritası ile özellik seçimi kullanılmamıştır. Sınıflandırma raporu ve karışıklık matrisi Çizelge 4.19 ve Çizelge 4.20'de yer almaktadır.

Çizelge 4.19. 2a maddesine ilişkin sınıflandırma raporu (classification report)

Saldırılar/Metrikler	Precision	Recall	F1	Support
Benign	1.00	1.00	1.00	318438
Bot	1.00	1.00	1.00	91157
Infiltration	1.00	0.996	0.998	29671

Çizelge 4.20. 2a maddesine ilişkin karışıklık matrisi (confusion matrix)

	Benign	Bot	Infiltration
Benign	318427	2	9
Bot	5	91152	0
Infiltration	124	2	29545

2b) Anomali durumunun tespitini sağlayan KNN modelinin oluşturulması

Anomali durumunun tespit edilebilmesi için hazırlanan modelde ise "n_neighbors=3" olarak seçilmiştir. Teste ilişkin sınıflandırma raporu ve karışıklık matrisi Çizelge 4.21, Çizelge 4.22 sunulmuştur.

Çizelge 4.21. 2b maddesine ilişkin sınıflandırma raporu (classification report)

Saldırılar/Metrikler	Precision	Recall	F1	Support
Benign	0.94	0.99	0.96	318438
Attack	0.96	0.83	0.89	120828

Çizelge 4.22. 2b maddesine ilişkin karışıklık matrisi (confusion matrix)

	Benign	Attack
Benign	313878	4560
Attack	20236	100592

2c) One Side Selection yöntemi ile dengelenen veri kullanılarak saldırı türü tespitini sağlayan KNN modelinin oluşturulması

Bu testte "4.2.3 Dengesiz Verinin Dengeli Hale Getirilmesi" başlığı altında belirtildiği üzere OSS yöntemi ile örneklem azaltılarak dengeli hale getirilen veri seti kullanılmıştır. (Şekil 4.8) Teste ilişkin sınıflandırma raporu ve karışıklık matrisi Çizelge 4.23 ve Çizelge 4.24'de sunulmuştur.

Çizelge 4.23. 2c maddesine ilişkin sınıflandırma raporu (classification report)

Saldırılar/Metrikler	Precision	Recall	F1	Support
Benign	0.94	0.98	0.96	318438
Bot	1.00	1.00	1.00	91157
Infiltration	0.57	0.34	0.43	29671

Çizelge 4.24. 2c maddesine ilişkin karışıklık matrisi (confusion matrix)

	Benign	Bot	Infiltration
Benign	310705	64	7669
Bot	18	91139	0
Infiltration	19560	0	10111

2d) SMOTEENN yöntemi ile dengelenen veri kullanılarak saldırı türü tespitini sağlayan KNN modelinin oluşturulması

Bu testte "4.2.3 Dengesiz Verinin Dengeli Hale Getirilmesi" başlığı altında belirtildiği üzere sentetik veri sayısı arttırılarak veri seti dengeli hale getirilmiştir. (Şekil 4.9) Teste ilişkin sınıflandırma raporu ve karışıklık matrisi Çizelge 4.25 ve Çizelge 4.26 sunulmuştur.

Çizelge 4.25. 2d maddesine ilişkin sınıflandırma raporu (classification report)

Saldırılar/Metrikler	Precision	Recall	F1	Support
Benign	0.96	0.80	0.87	318438
Bot	1.00	1.00	1.00	91157
Infiltration	0.23	0.66	0.35	29671

Çizelge 4.26. 2d maddesine ilişkin karışıklık matrisi (confusion matrix)

	Benign	Bot	Infiltration
Benign	253641	13	64784
Bot	5	91151	1
Infiltration	9946	0	19725

2e) SMOTETomek yöntemi ile dengelenen veri kullanılarak saldırı türü tespitini sağlayan KNN modelinin oluşturulması

Bu testte "4.2.3 Dengesiz Verinin Dengeli Hale Getirilmesi" başlığı altında belirtildiği üzere sentetik veri sayısı arttırılarak veri seti dengeli hale getirilmiştir. (Şekil 4.10) Teste ilişkin sınıflandırma raporu Çizelge 4.27'de ve karışıklık matrisi Çizelge 4.28'de sunulmuştur.

Çizelge 4.27. 2e maddesine ilişkin sınıflandırma raporu (classification report)

Saldırılar/Metrikler	Precision	Recall	F1	Support
Benign	0.95	0.84	0.89	318438
Bot	1.00	1.00	1.00	91157
Infiltration	0.25	0.57	0.35	29671

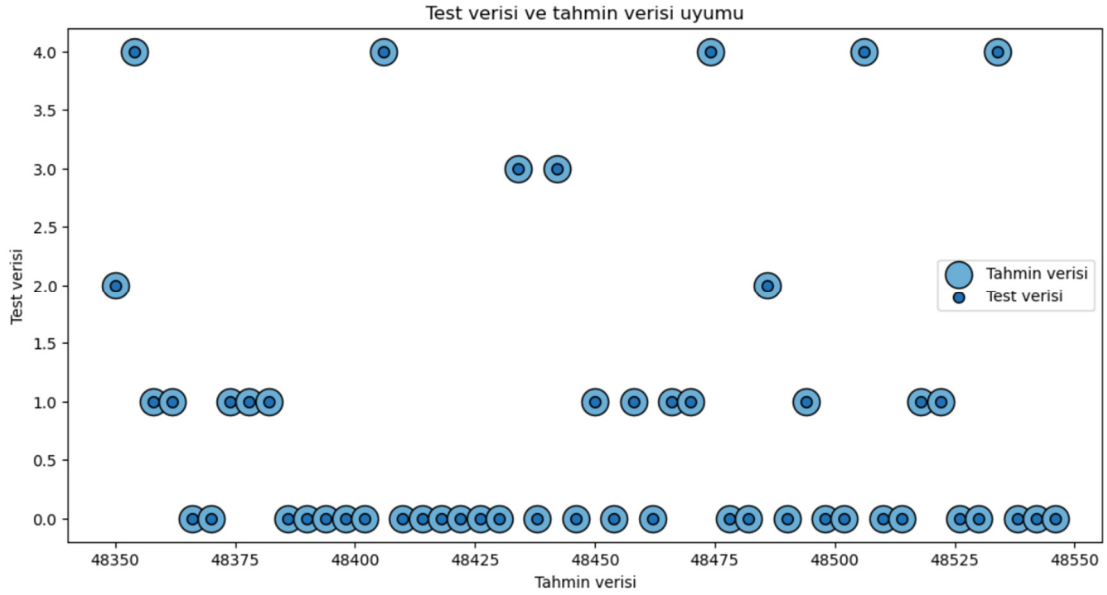
Çizelge 4.28. 2e maddesine ilişkin karışıklık matrisi (confusion matrix)

	Benign	Bot	Infiltration
Benign	267286	11	51141
Bot	6	91151	0
Infiltration	12815	0	16856

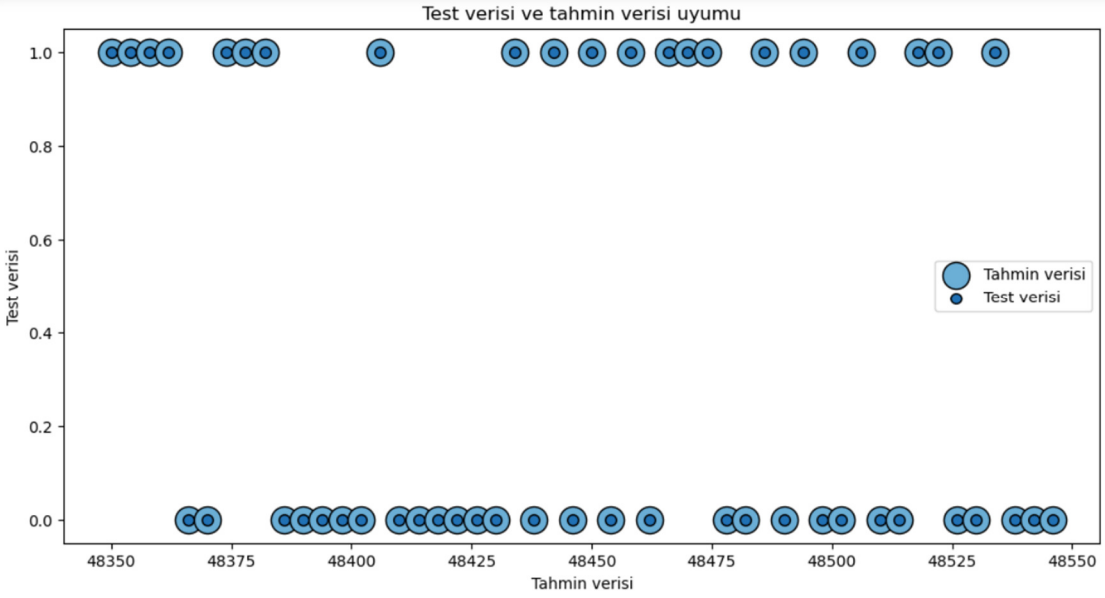
4.3.1.3. Sonuçlar - KNN

Hazırlanan modellerin doğru şekilde tahmin edilip edilmediğinin gözlemlenebilmesi için veri seti test kümesinden [48350,48550] değer aralığındaki 50 değer seçilerek tahmin edilmesi sağlanmıştır. Sonuçlar "scatter" grafiğinde gösterilerek

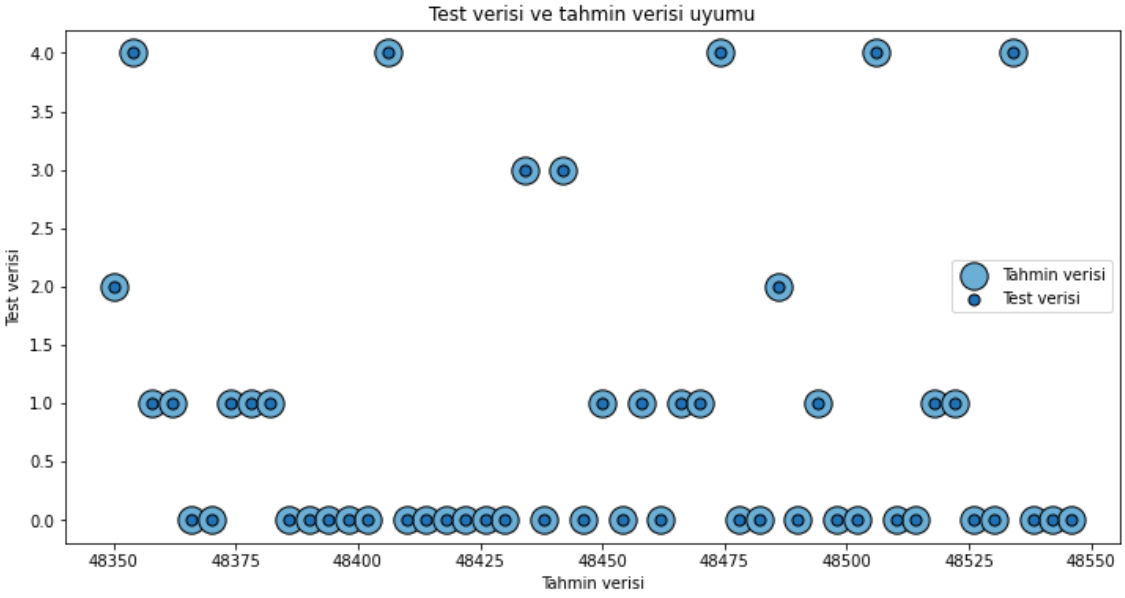
test verisi ve tahmin verisi uyumu çizdirilmiştir. (Şekil 4.43, Şekil 4.44, Şekil 4.45, Şekil 4.46, Şekil 4.47, Şekil 4.48, Şekil 4.49, Şekil 4.50, Şekil 4.51)



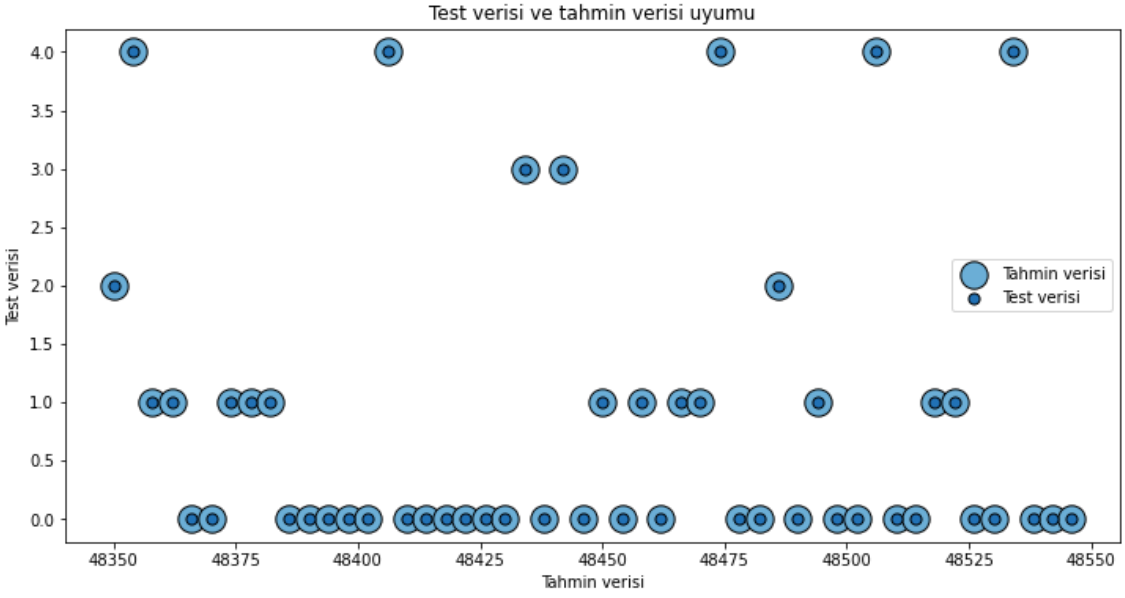
Şekil 4.43. 1a testi [48350,48550] değer aralığındaki 50 değer için model çıktısı



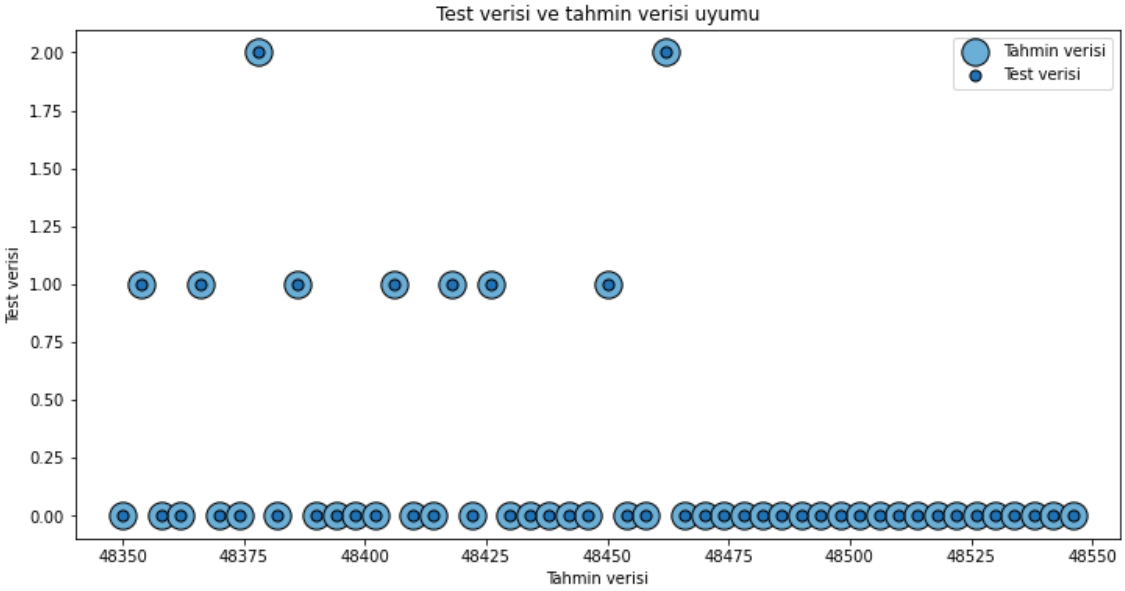
Şekil 4.44. 1b testi [48350,48550] değer aralığındaki 50 değer için model çıktısı



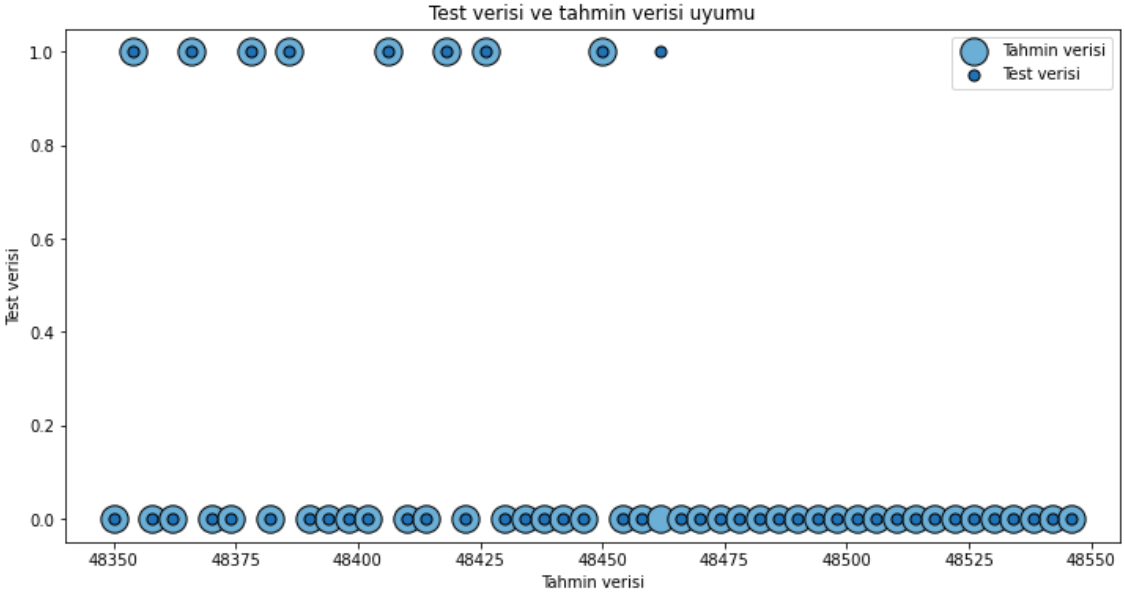
Şekil 4.45. 1c testi [48350,48550] değer aralığındaki 50 değer için model çıktısı



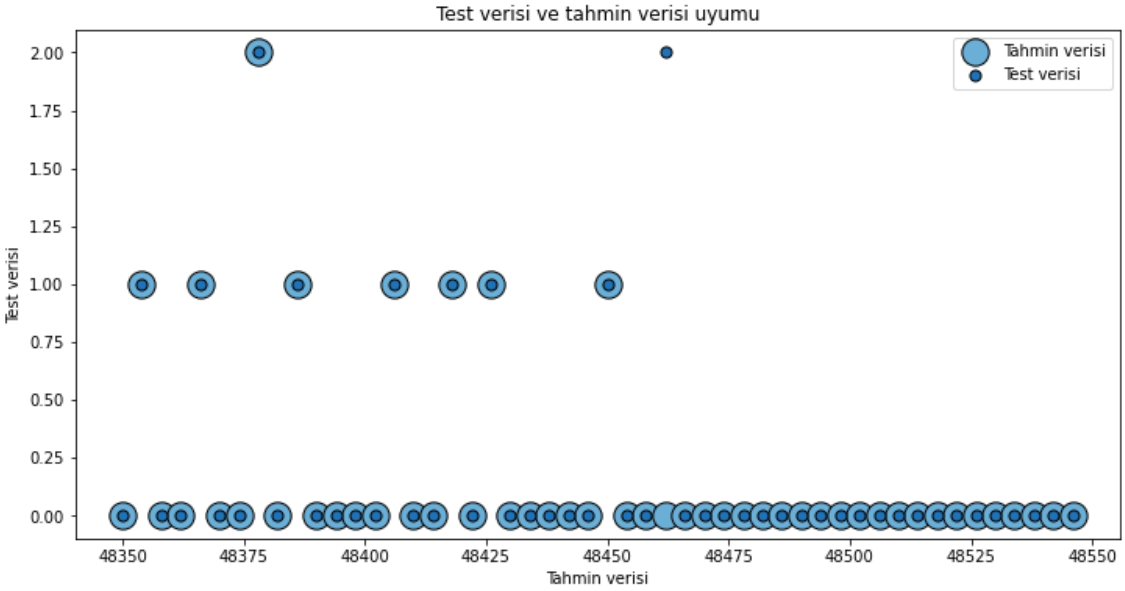
Şekil 4.46. 1d testi [48350,48550] değer aralığındaki 50 değer için model çıktısı



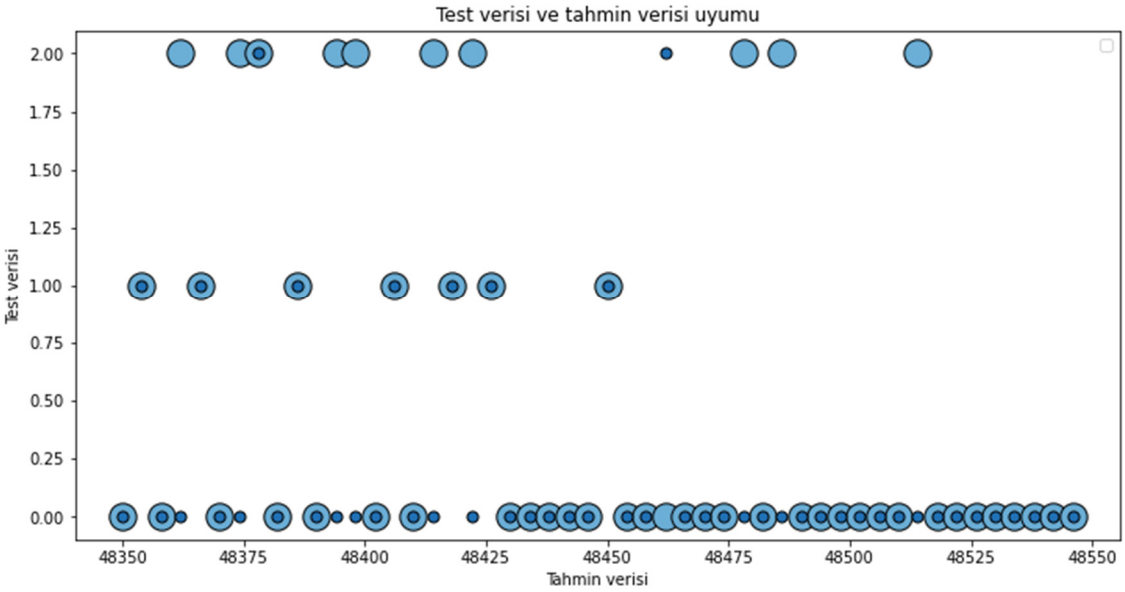
Şekil 4.47. 2a testi [48350,48550] değer aralığındaki 50 değer için model çıktısı



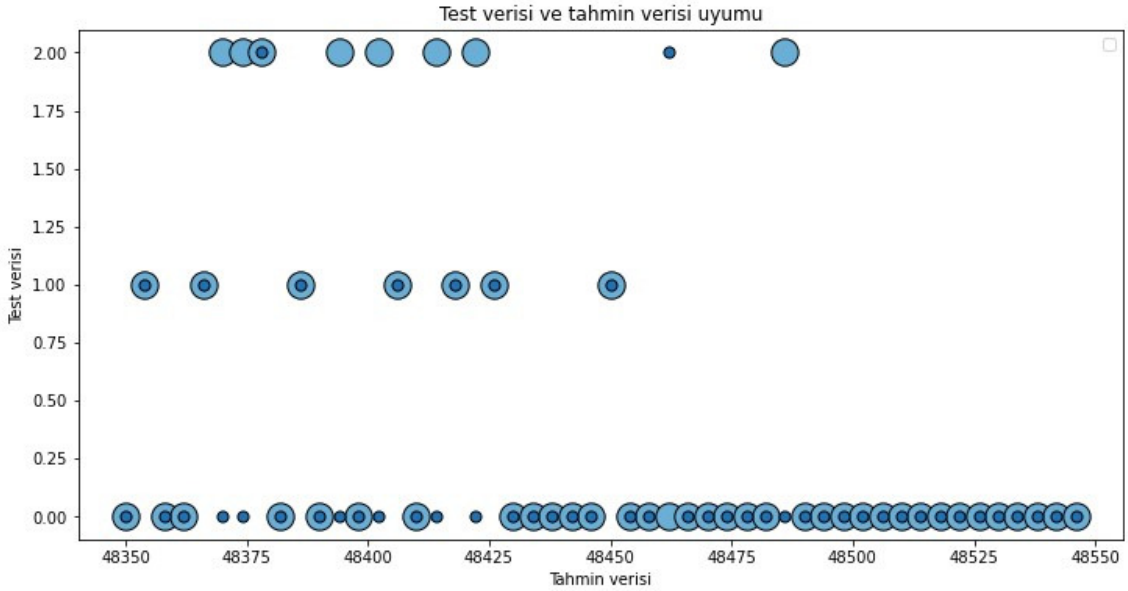
Şekil 4.48. 2b testi [48350,48550] değer aralığındaki 50 değer için model çıktısı



Şekil 4.49. 2c testi [48350,48550] değer aralığındaki 50 değer için model çıktısı



Şekil 4.50. 2d testi [48350,48550] değer aralığındaki 50 değer için model çıktısı



Şekil 4.51. 2e testi [48350,48550] değer aralığındaki 50 değer için model çıktısı

"Senaryo 1" ve "Senaryo 2" verisetlerinde KNN uygulaması için elde ettiğimiz test sonuçları Çizelge 4.29'da yer almaktadır.

Çizelge 4.29. "Senaryo 1" ve "Senaryo 2" KNN uygulaması performans değerlendirme sonuçları

Test No	Train Acc	Test Acc	Precision	Recall	F1 Score
1a	0.999964	0.999947	1	1	1
1b	0.999914	0.999913	1	1	1
1c	0.999956	0.999937	1	1	1
1d	0.998597	0.999014	0.99	1	0.99
2a	0.999803	0.999676	1	1	1
2b	0.952302	0.943553	0.95	0.91	0.93
2c	0.881936	0.937825	0.84	0.77	0.79
2d	0.994349	0.829832	0.73	0.82	0.74
2e	0.936767	0.854363	0.73	0.80	0.75

4.3.2. Evrişimli Sinir Ağları (ESA-CNN–Convolutional Neural Network)

Çalışmamızın bu kısmında derin öğrenme yöntemlerinden evrişimli sinir ağları (ESA) modeli kullanılarak ağ saldırılarının çoklu sınıflandırması ve ikili sınıflandırması konusu ele alınmıştır. ESA modeli, farklı nesneleri birbirinden ayırt edebilmek için görüntünün analiz edilmesini sağlamaktadır dolayısıyla görüntü verilerinde sıklıkla kullanılmaktadır. Bizim problemimizde ise metin verilerinin sınıflandırılması sağlanmıştır. ESA modeli oluşturulurken aşağıdaki sıra ile işlemler uygulanmıştır. (Şekil 4.52)

1- Derin öğrenme kodlarının çalıştırılabilmesi için “import tensorflow as tf”, model katmanlarının eklenebilmesi için “import tensorflow.keras.layers as ly” kütüphaneleri dahil edilmiştir.
2- “from sklearn.model_selection import train_test_split” kütüphanesi ile veri setinin x ve y değişkenleri olmak üzere %68’i eğitim, eğitim verisinin de %20’si geçerleme verisi olarak ayrılmıştır. Verisetinin dengeli hale getirildiği durumlarda ise arttırılan verisetinin tamamı eğitim, eğitim verisinin 0.25’i geçerleme, toplam verisetinin 0.32’si test veriseti olarak bölünmüştür.
3- Veri bölme işleminden sonra eğitim, test, geçerleme olmak üzere x girdileri “StandardScaler” ile ölçeklendirilerek normalizasyonu sağlanmıştır.
4- “y” çıktısı eğitim, test, geçerleme olmak üzere dataframe yapısından 5 sütunlu dizi haline getirilmiştir.
5- ESA modelinin uygulanabilmesi için eğitim, test, geçerleme olmak üzere x girdileri tekrar boyutlandırılmıştır.

x (dataframe) (2089122, 78)	Standard Scaler (3)	ESA modeli için dönüşüm (reshape) (5)	y (dataframe) (2089122, 1)	Kategorilerine göre dönüşüm (4)
x_train (dataframe) (1136481, 78)	X_train (array) (1136481, 78)	X_train1 (array) (1136481, 78, 1)	y_train (dataframe) (1136481, 1)	y_train1 (array) (1136481, 5)
x_val (dataframe) (284121, 78)	X_val (array) (284121, 78)	X_val1 (array) (284121, 78, 1)	y_val (dataframe) (284121, 1)	y_val1 (array) (284121, 5)
x_test (dataframe) (668520, 78)	X_test (array) (668520, 78)	X_test1 (array) (668520, 78, 1)	y_test (dataframe) (668520, 1)	y_test1 (array) (668520, 5)

Şekil 4.52. Veri seti dönüşümü

6- Bir boyutlu evrişimli sinir ağı oluşturulurken katmanların sıralı olarak eklenebilmesi için Keras’ın “Sequential” modeli eklenmiştir. Özelliklerin çıkarılması aşamasında metin verisi iki boyutlu olduğu için “Conv1D” katmanı eklenmiştir. Evrişim katmanı hiperparametreleri tanımlanırken kaç adet filtre kullanacağımızı belirten “filters” 64, filtrelerin boyutunu ifade eden “kernel-size” 6 ve girdi ile çıktı arasındaki boyutun nasıl doldurulacağını belirten “padding=same” olarak tanımlanarak girdi ve çıktının aynı boyutlu olması sağlanmıştır. Conv1D katmanında çıktıların normalleştirilmesi ve ağın daha

hızlı eğitilebilmesi için "BatchNormalization", boyut azaltarak hesaplama kolaylığı sağlaması için "pool_size=3" tanımlanarak "MaxPooling1D" katmanı eklenmiştir ve modele "Flatten" katmanı uygulanmıştır. Ardından Fully-Connected katmanı için Keras'ın "Dense" yoğunluk katmanları eklenerek model oluşturulmuştur. Her bir katmana Keras'ın "activition" özelliği eklenerek kullanacağımız aktivasyon fonksiyonları belirtilmiştir (Güler ve ark., 2023). Eklenen model katmanları Şekil 4.53'te gösterilmiştir.

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv1d (Conv1D)              (None, 78, 64)            448

batch_normalization (BatchN  (None, 78, 64)            256
ormalization)

max_pooling1d (MaxPooling1D  (None, 39, 64)            0
)

conv1d_1 (Conv1D)            (None, 39, 64)            24640

batch_normalization_1 (Batc  (None, 39, 64)            256
hNormalization)

max_pooling1d_1 (MaxPooling  (None, 20, 64)            0
1D)

flatten (Flatten)            (None, 1280)              0

dense (Dense)                (None, 128)               163968

dense_1 (Dense)              (None, 64)                8256

dense_2 (Dense)              (None, 5)                 325

=================================================================
Total params: 198,149
Trainable params: 197,893
Non-trainable params: 256
_________________________________________________________________
```

Şekil 4.53. 1DCNN ile oluşturulan eğitim modeli

7- Modelin derlenmesi aşamasında tahmin işleminin doğru bir şekilde sağlanabilmesi için parametreler tanımlanmıştır. Sınıflandırma problemimizin hata değerlerinin hesaplanabilmesi için kayıp fonksiyonu olarak "categorical cross entropy", optimizasyon algoritması olarak ise "Adam" kullanılmıştır.

4.3.2.1. Senaryo 1 – 1DCNN

"Senaryo 1" için 1DCNN modeli ile saldırı tespitinin gerçekleştirilmesine yönelik testler aşağıda sunulmuştur:

1e) Saldırı türünün tespitini sağlayan 1DCNN modelinin oluşturulması

Model uygulanırken iterasyon sayısı 2 (epoch), aynı anda işleyebileceği veri sayısı 16 (batch_size), 70000 steps_per_epoch ayarları ile eğitim sağlanmıştır. Girdi olarak x_train=(1136481,78), x_test=(668520,78), x_val=(284121,78) boyutlu veriler kullanılmıştır. Teste ilişkin sınıflandırma raporu Çizelge 4.30'da ve karışıklık matrisi Çizelge 4.31'de sunulmuştur.

Çizelge 4.30. 1e maddesine ilişkin sınıflandırma raporu (classification report)

Saldırılar/Metrikler	Precision	Recall	F1	Support
Benign	1.000	1.000	1.000	459190
Hulk	1.000	1.000	1.000	147938
SlowLoris	0.99	0.77	0.86	3478
GoldenEye	1.000	0.98	0.99	13355
SlowHTTPTest	1.000	1.000	1.000	44559

Çizelge 4.31. 1e maddesine ilişkin karışıklık matrisi (confusion matrix)

	Benign	Hulk	SlowLoris	GoldenEye	SlowHTTPTest
Benign	459161	6	18	4	1
Hulk	5	147902	0	31	0
SlowLoris	801	14	2663	0	0
GoldenEye	113	120	2	13120	0
SlowHTTPTest	0	0	0	0	44559

1f) Anomali tespitini sağlayan 1DCNN modelinin oluşturulması

Anomali durumu tespitinin sağlanabilmesi için "1e" maddesinin eğitildiği ayarlar ile eğitim sağlanmıştır. Girdi olarak x_train=(1136481,78), x_test=(668520,78), x_val=(284121,78) boyutlu veriler kullanılmıştır. Teste ilişkin sınıflandırma raporu Çizelge 4.32 ve karışıklık matrisi Çizelge 4.33 sunulmuştur.

Çizelge 4.32. 1f maddesine ilişkin sınıflandırma raporu (classification report)

Saldırılar/Metrikler	Precision	Recall	F1	Support
Benign	1.000	1.000	1.000	459190
Attack	1.000	1.000	1.000	209330

Çizelge 4.33. 1f maddesine ilişkin karışıklık matrisi (confusion matrix)

	Benign	Attack
Benign	458976	214
Attack	36	209294

1g) One Side Selection yöntemi ile dengelenen veri kullanılarak saldırı türü tespitini sağlayan 1DCNN modelinin oluşturulması

Model uygulanırken iterasyon sayısı 2 (epoch), aynı anda işleyebileceği veri sayısı 16 (batch_size), 2500 steps_per_epoch ayarları ile eğitim sağlanmıştır. Bu testte "4.2.3 Dengesiz Verinin Dengeli Hale Getirilmesi" başlığı altında belirtildiği üzere OSS yöntemi ile örneklem azaltılarak dengeli hale getirilen veri seti kullanılmıştır. (Şekil 4.6) x_train=(40630,78), x_test=(655149,78), x_val=(13371,78) boyutlu veriler ile girdi sağlanmıştır. Teste ilişkin sınıflandırma raporu Çizelge 4.34 ve karışıklık matrisi Çizelge 4.35'te sunulmuştur.

Çizelge 4.34. 1g maddesine ilişkin sınıflandırma raporu (classification report)

Saldırılar/Metrikler	Precision	Recall	F1	Support
Benign	1.000	1.000	1.000	459190
Hulk	1.000	1.000	1.000	147938
SlowLoris	0.98	1.000	0.99	3478
GoldenEye	0.96	1.000	0.98	13355
SlowHTTPTest	1.000	1.000	1.000	44559

Çizelge 4.35. 1g maddesine ilişkin karışıklık matrisi (confusion matrix)

	Benign	Hulk	SlowLoris	GoldenEye	SlowHTTPTest
Benign	458428	6	18	4	1
Hulk	5	147902	0	31	0
SlowLoris	801	14	2663	0	0
GoldenEye	113	120	2	13120	0
SlowHTTPTest	0	0	0	0	44559

4.3.2.2. Senaryo 2 – 1DCNN

"Senaryo 2" için 1DCNN modeli ile saldırı tespitinin gerçekleştirilmesine yönelik testler aşağıda sunulmuştur:

2f) Saldırı türünün tespitini sağlayan 1DCNN modelinin oluşturulması

Model uygulanırken iterasyon sayısı 2 (epoch), aynı anda işleyebileceği veri sayısı 16 (batch_size), 20000 steps_per_epoch ayarları ile eğitim sağlanmıştır. x_train=(746752,78), x_test=(439266,78), x_val=(186688,78) boyutlu veriler ile girdi sağlanmıştır. Teste ilişkin sınıflandırma raporu Çizelge 4.36 ve karışıklık matrisi Çizelge 4.37 sunulmuştur.

Çizelge 4.36. 2f maddesine ilişkin sınıflandırma raporu (classification report)

Saldırılar/Metrikler	Precision	Recall	F1	Support
Benign	0.93	1.000	0.96	318438
Bot	1.000	1.000	1.000	91157
Infiltration	0.90	0.24	0.37	29671

Çizelge 4.37. 2f maddesine ilişkin karışıklık matrisi (confusion matrix)

	Benign	Bot	Infiltration
Benign	317625	66	747
Bot	26	91130	1
Infiltration	22664	0	7007

2g) One Side Selection yöntemi ile dengelenen veri kullanılarak saldırı türü tespitini sağlayan 1DCNN modelinin oluşturulması

Testte "4.2.3 Dengesiz Verinin Dengeli Hale Getirilmesi" başlığı altında belirtildiği üzere OSS yöntemi ile örneklem azaltılarak dengeli hale getirilen veri seti kullanılmıştır. (Şekil 4.8) Model uygulanırken iterasyon sayısı 2 (epoch), aynı anda işleyebileceği veri sayısı 16 (batch_size), 20000 steps_per_epoch ayarları ile eğitim sağlanmıştır. Test verisinin 0.02'si geçerleme olarak tanımlanarak x_train=(357096,78), x_test=(430480,78), x_val=(8786,78) boyutlu veriler ile girdi sağlanmıştır. Teste ilişkin sınıflandırma raporu Çizelge 4.38 ve karışıklık matrisi Çizelge 4.39 sunulmuştur.

Çizelge 4.38. 2g maddesine ilişkin sınıflandırma raporu (classification report)

Saldırılar/Metrikler	Precision	Recall	F1	Support
Benign	0.93	1.000	0.96	312020
Bot	1.000	1.000	1.000	89352
Infiltration	0.91	0.24	0.38	29108

Çizelge 4.39. 2g maddesine ilişkin karışıklık matrisi (confusion matrix)

	Benign	Bot	Infiltration
Benign	311169	123	728
Bot	105	89247	0
Infiltration	22050	1	6957

2h) SMOTEENN yöntemi ile dengelenen veri kullanılarak saldırı türü tespitini sağlayan 1DCNN modelinin oluşturulması

Bu testte "4.2.3 Dengesiz Verinin Dengeli Hale Getirilmesi" başlığı altında belirtildiği üzere sentetik veri sayısı arttırılarak veri seti dengeli hale getirilmiştir. (Şekil 4.9) Model uygulanırken iterasyon sayısı 2 (epoch), aynı anda işleyebileceği veri sayısı 16 (batch_size), 70000 steps_per_epoch ayarları ile eğitim sağlanmıştır. x_train=(1243834,78), x_test=(439266,78), x_val=(414612,78) boyutu veriler ile girdi sağlanmıştır. Teste ilişkin sınıflandırma raporu Çizelge 4.40 ve karışıklık matrisi Çizelge 4.41 sunulmuştur.

Çizelge 4.40. 2h maddesine ilişkin sınıflandırma raporu (classification report)

Saldırılar/Metrikler	Precision	Recall	F1	Support
Benign	0.96	0.78	0.86	318438
Bot	1.000	1.000	1.000	91157
Infiltration	0.21	0.65	0.32	29671

Çizelge 4.41. 2h maddesine ilişkin karışıklık matrisi (confusion matrix)

	Benign	Bot	Infiltration
Benign	247136	15	71287
Bot	49	91098	10
Infiltration	10462	0	19209

2i) SMOTETomek yöntemi ile dengelenen veri kullanılarak saldırı türü tespitini sağlayan 1DCNN modelinin oluşturulması

Bu testte "4.2.3 Dengesiz Verinin Dengeli Hale Getirilmesi" başlığı altında belirtildiği üzere sentetik veri sayısı arttırılarak veri seti dengeli hale getirilmiştir. (Şekil 4.10) Model uygulanırken iterasyon sayısı 2 (epoch), aynı anda işleyebileceği veri sayısı 16 (batch_size), 90000 steps_per_epoch ayarları ile eğitim sağlanmıştır. x_train=(1476121,78), x_test=(439266,78), x_val=(492041,78) boyutu veriler ile girdi

sağlanmıştır. Teste ilişkin sınıflandırma raporu Çizelge 4.42 ve karışıklık matrisi Çizelge 4.43'te sunulmuştur.

Çizelge 4.42. 2i maddesine ilişkin sınıflandırma raporu (classification report)

Saldırılar/Metrikler	Precision	Recall	F1	Support
Benign	0.97	0.59	0.74	318438
Bot	1.000	1.000	1.000	91157
Infiltration	0.16	0.82	0.27	29671

Çizelge 4.43. 2i maddesine ilişkin karışıklık matrisi (confusion matrix)

	Benign	Bot	Infiltration
Benign	189018	13	129427
Bot	25	91108	24
Infiltration	5253	0	24418

2j) SMOTETomek ile dengelenen veri kullanılarak saldırı türü tespitini sağlayan 1DCNN modelinde aktivasyon fonksiyonunun değiştirilmesi

Bu testte "4.2.3 Dengesiz Verinin Dengeli Hale Getirilmesi" başlığı altında belirtildiği üzere sentetik veri sayısı arttırılarak veri seti dengeli hale getirilmiştir. (Şekil 4.10) Model uygulanırken iterasyon sayısı 2 (epoch), aynı anda işleyebileceği veri sayısı 16 (batch_size), 90000 steps_per_epoch ayarları ile eğitim sağlanmıştır. model katmanlarında ReLU aktivasyon fonksiyonu değiştirilerek LeakyReLU kullanılmıştır. x_train=(1476121,78), x_test=(439266,78), x_val=(492041,78) boyutlu veriler ile girdi sağlanmıştır. Teste ilişkin sınıflandırma raporu Çizelge 4.44 ve karışıklık matrisi Çizelge 4.45 sunulmuştur.

Çizelge 4.44. 2j maddesine ilişkin sınıflandırma raporu (classification report)

Saldırılar/Metrikler	Precision	Recall	F1	Support
Benign	0.97	0.60	0.75	318438
Bot	1.000	1.000	1.000	91157
Infiltration	0.16	0.81	0.27	29671

Çizelge 4.45. 2j maddesine ilişkin karışıklık matrisi (confusion matrix)

	Benign	Bot	Infiltration
Benign	192438	148	125852
Bot	18	91135	4
Infiltration	5590	1	24080

2k) SMOTETomek ile dengelenen veri kullanılarak saldırı türü tespitini sağlayan 1DCNN modelinde optimizasyon fonksiyonunun değiştirilmesi

Model uygulanırken iterasyon sayısı 2 (epoch), aynı anda işleyebileceği veri sayısı 16 (batch_size), 90000 steps_per_epoch ayarları ile eğitim sağlanmıştır. Model derlenirken Adam optimizasyon fonksiyonu değiştirilerek RMSProp kullanılmıştır. Bu testte "4.2.3 Dengesiz Verinin Dengeli Hale Getirilmesi" başlığı altında belirtildiği üzere sentetik veri sayısı arttırılarak veri seti dengeli hale getirilmiştir. (Şekil 4.10) Girdi olarak x_train=(1476121,78), x_test=(439266,78), x_val=(492041,78) boyutlu veriler kullanılmıştır. Teste ilişkin sınıflandırma raporu Çizelge 4.46 ve karışıklık matrisi Çizelge 4.47 sunulmuştur.

Çizelge 4.46. 2k maddesine ilişkin sınıflandırma raporu (classification report)

Saldırılar/Metrikler	Precision	Recall	F1	Support
Benign	0.95	0.88	0.91	318438
Bot	1.000	1.000	1.000	91157
Infiltration	0.28	0.49	0.36	29671

Çizelge 4.47. 2k maddesine ilişkin karışıklık matrisi (confusion matrix)

	Benign	Bot	Infiltration
Benign	280288	18	38032
Bot	93	91007	57
Infiltration	14996	0	14675

2l) One Side Selection yöntemi ile dengelenen verisetinde XGBoost algoritması ile seçilen özellikler kullanılarak saldırı türü tespitini sağlayan 1DCNN modelinin oluşturulması

Bu testte "4.2.3 Dengesiz Verinin Dengeli Hale Getirilmesi" başlığı altında belirtildiği üzere OSS yöntemi ile örneklem azaltılarak dengeli hale getirilen veri seti kullanılmıştır. (Şekil 4.8) "4.2.4.2 XGBoost Algoritması ile Özellik Seçimi" başlığı

altında tespit edilen öznitelikler model eğitimi için kullanılmıştır. (Şekil 4.22) Teste ilişkin sınıflandırma raporu Çizelge 4.48 ve karışıklık matrisi Çizelge 4.49 sunulmuştur.

Çizelge 4.48. 2l maddesine ilişkin sınıflandırma raporu (classification report)

Saldırılar/Metrikler	Precision	Recall	F1	Support
Benign	0.93	0.99	0.96	312020
Bot	1.000	1.000	1.000	89352
Infiltration	0.76	0.24	0.36	29108

Çizelge 4.49. 2l maddesine ilişkin karışıklık matrisi (confusion matrix)

	Benign	Bot	Infiltration
Benign	309597	194	2229
Bot	96	89256	0
Infiltration	22229	1	6878

2m) Anomali tespitini sağlayan 1DCNN modelinin oluşturulması

Teste ilişkin sınıflandırma raporu Çizelge 4.50 ve karışıklık matrisi Çizelge 4.51 aşağıda sunulmuştur.

Çizelge 4.50. 2m maddesine ilişkin sınıflandırma raporu (classification report)

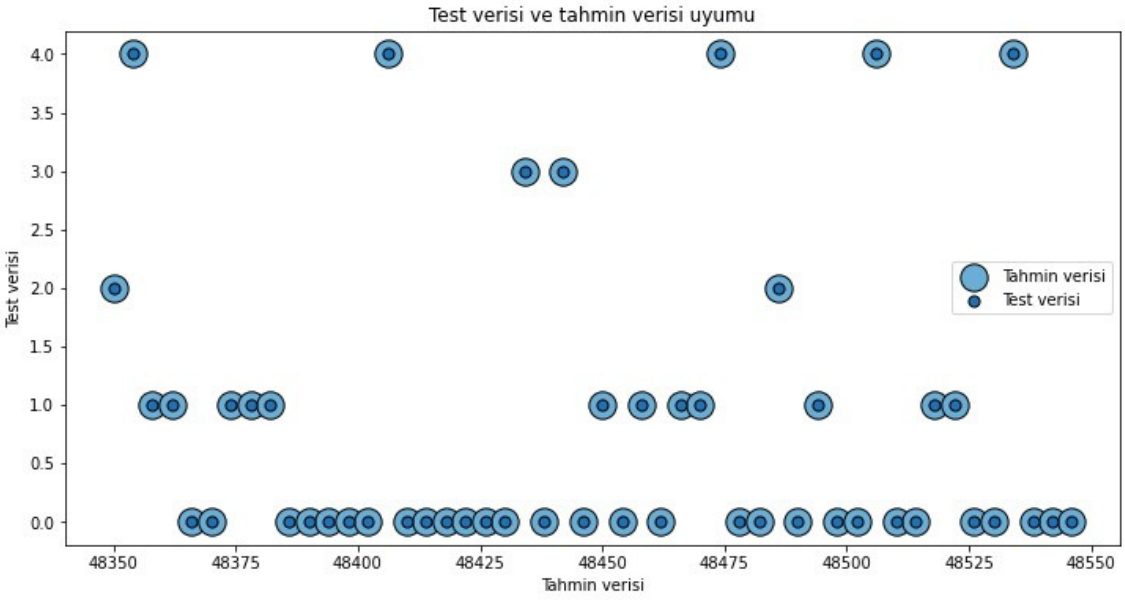
Saldırılar/Metrikler	Precision	Recall	F1	Support
Benign	0.93	1.000	0.96	318438
Attack	0.99	0.81	0.89	120828

Çizelge 4.51. 2m maddesine ilişkin karışıklık matrisi (confusion matrix)

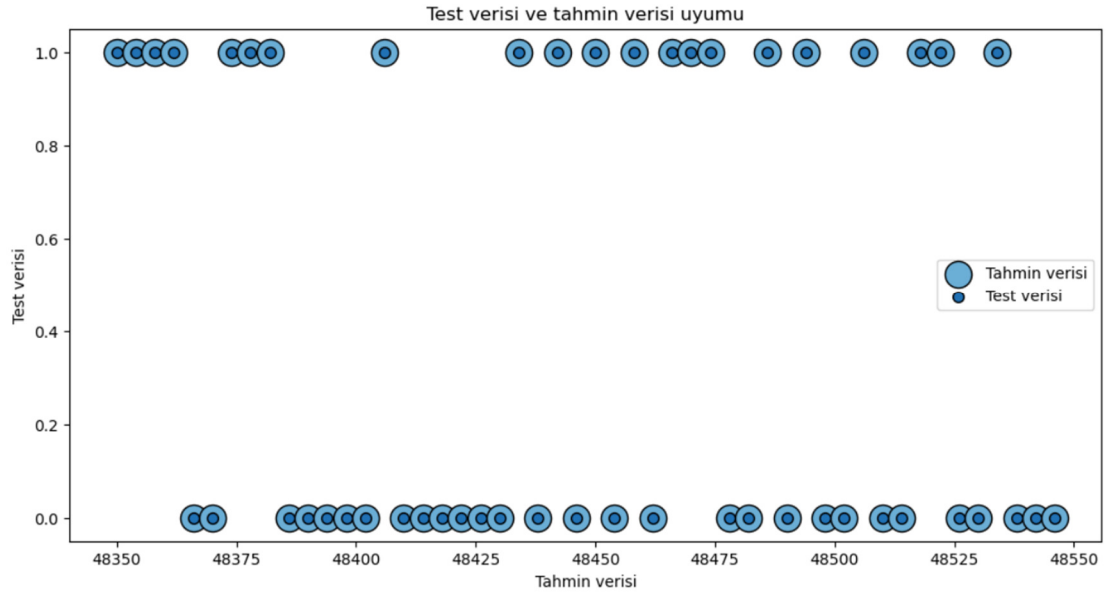
	Benign	Attack
Benign	317707	731
Attack	22623	98205

4.3.2.3. Sonuçlar – 1DCNN

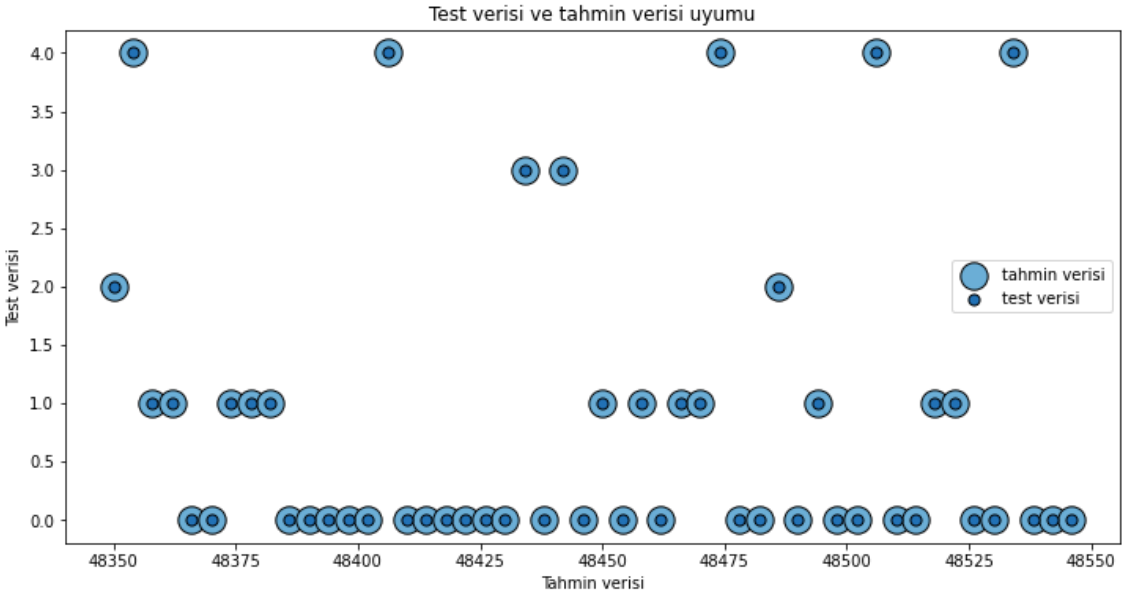
Hazırlanan modellerin doğru şekilde tahmin edilip edilmediğinin gözlemlenebilmesi için veri seti test kümesinden [48350,48550] değer aralığındaki 50 değer seçilerek tahmin etmesi sağlanmıştır. Sonuçlar "scatter" grafiğinde gösterilerek test verisi ve tahmin verisi uyumu çizdirilmiştir. (Şekil 4.54, Şekil 4.55, Şekil 4.56, **Şekil 4.55**Şekil 4.57, Şekil 4.58, Şekil 4.59, Şekil 4.60, Şekil 4.61, Şekil 4.62, Şekil 4.63, Şekil 4.64)



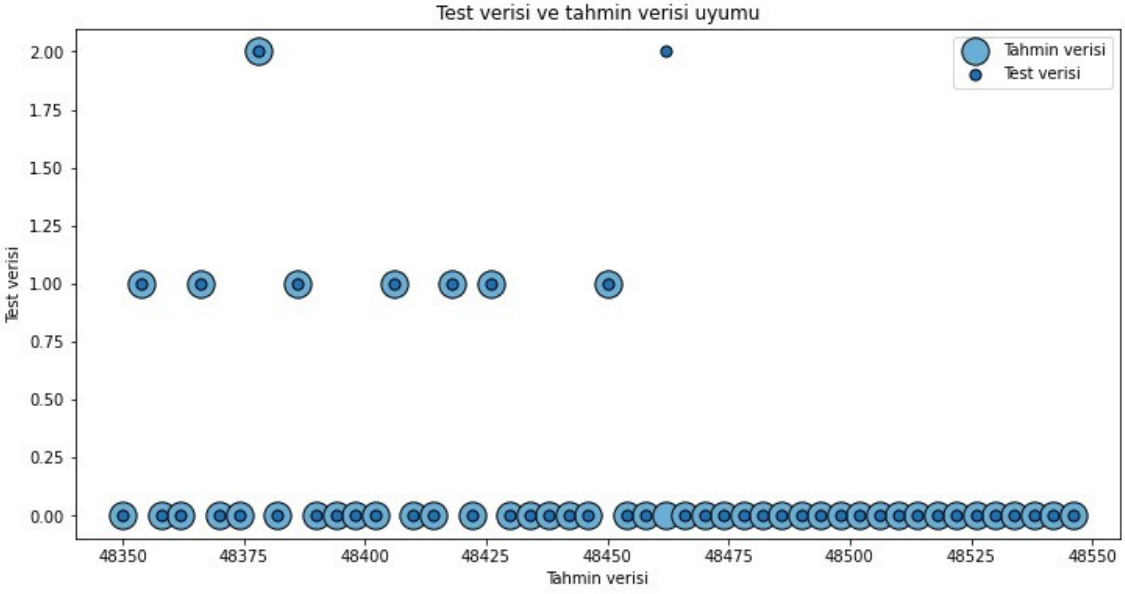
Şekil 4.54. 1e testi [48350,48550] değer aralığındaki 50 değer için model çıktısı



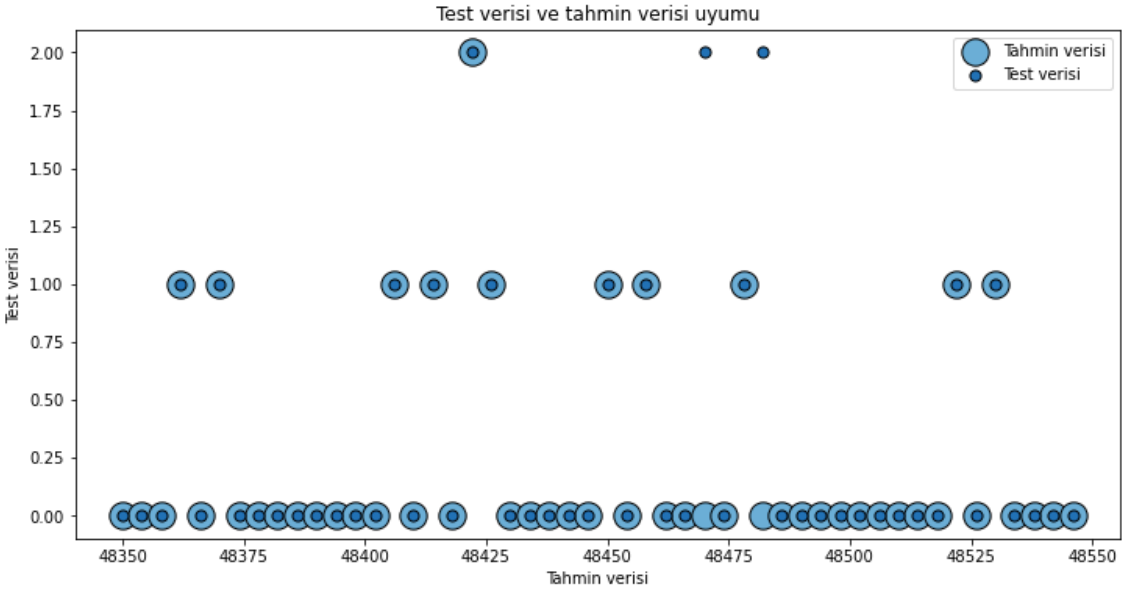
Şekil 4.55. 1f testi [48350,48550] değer aralığındaki 50 değer için model çıktısı



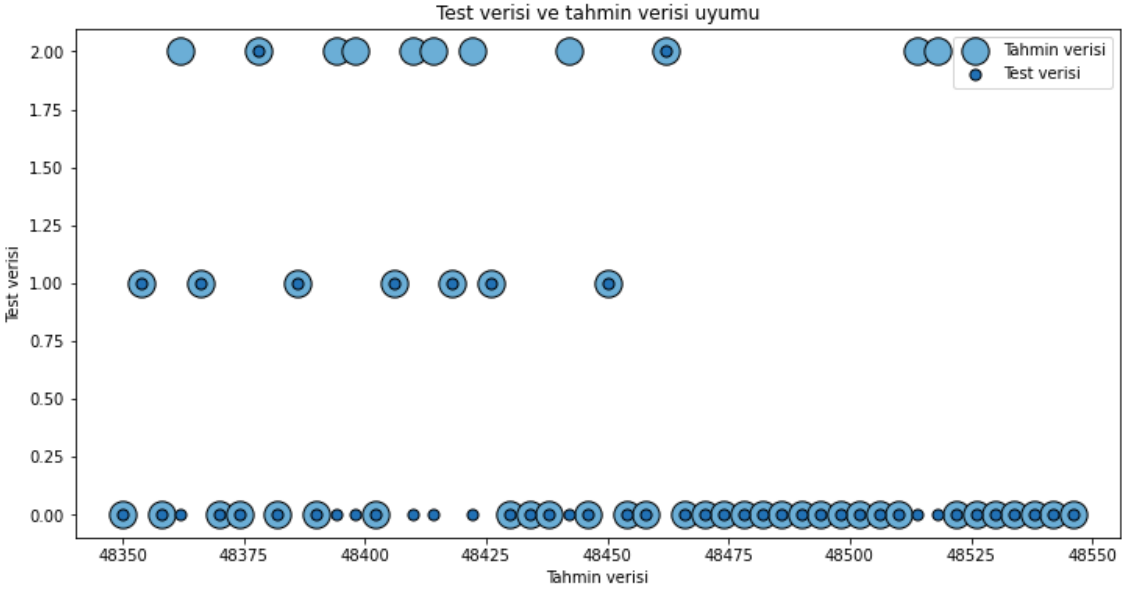
Şekil 4.56. 1g testi [48350,48550] değer aralığındaki 50 değer için model çıktısı



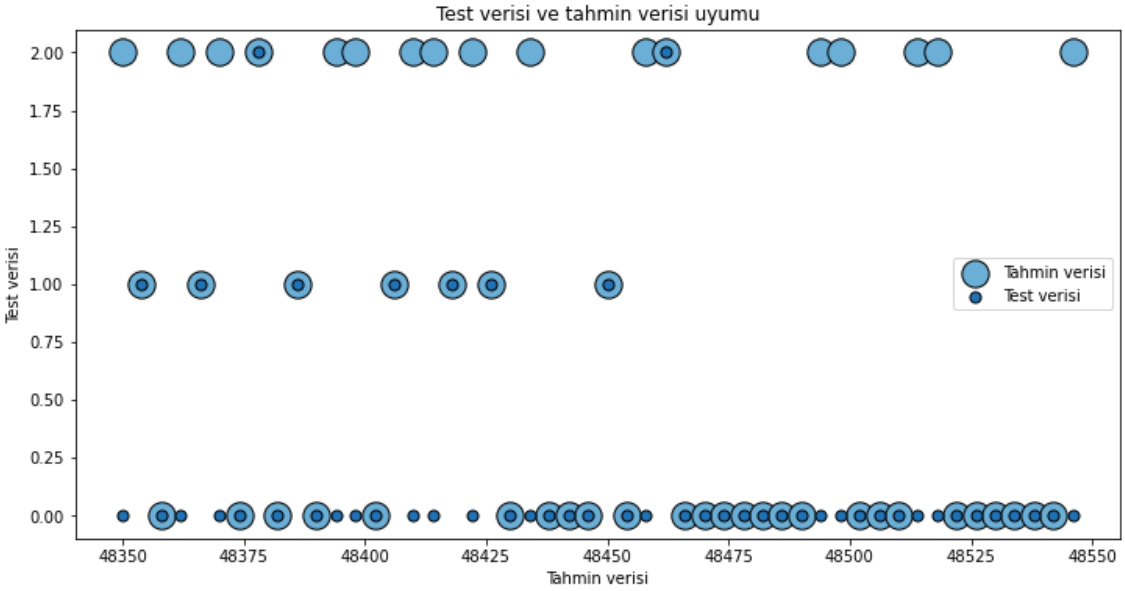
Şekil 4.57. 2f testi [48350,48550] değer aralığındaki 50 değer için model çıktısı



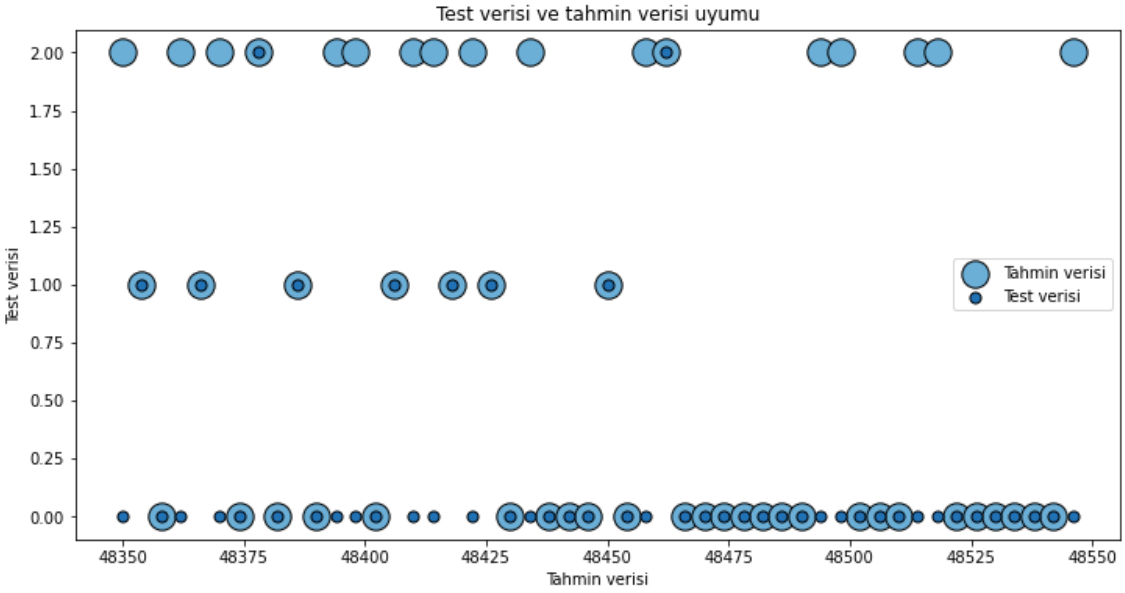
Şekil 4.58. 2g testi [48350,48550] değer aralığındaki 50 değer için model çıktısı



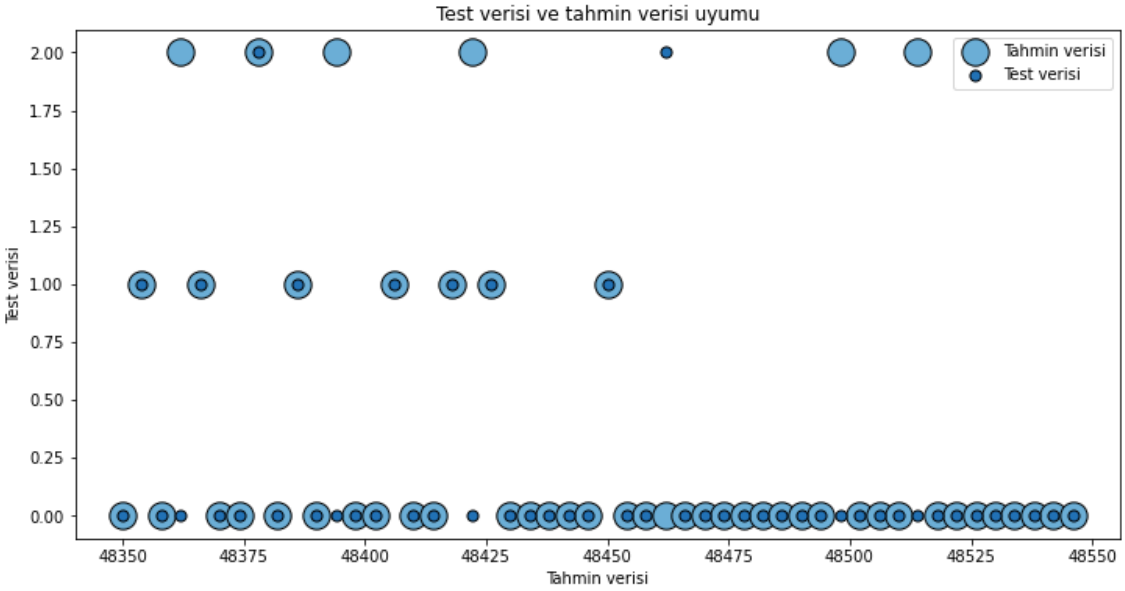
Şekil 4.59. 2h testi [48350,48550] değer aralığındaki 50 değer için model çıktısı



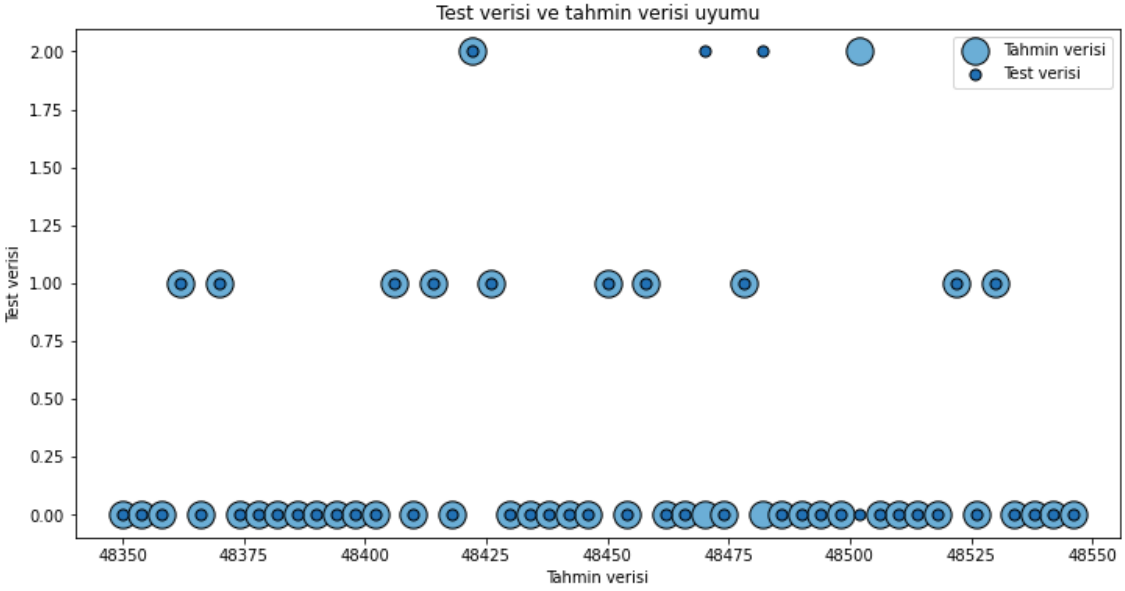
Şekil 4.60. 2i testi [48350,48550] değer aralığındaki 50 değer için model çıktısı



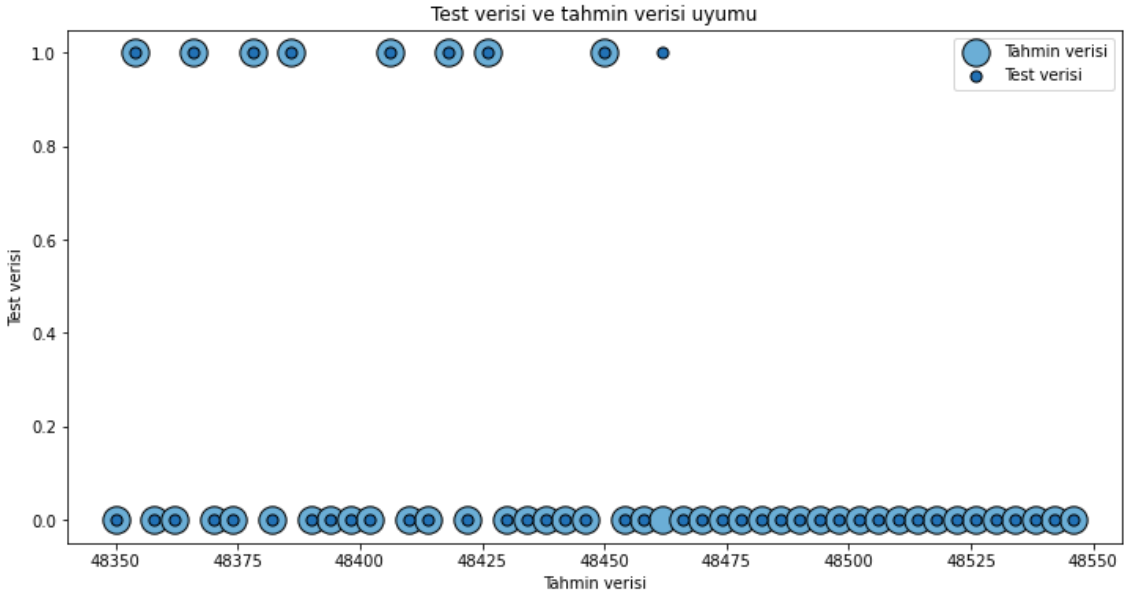
Şekil 4.61. 2j testi [48350,48550] değer aralığındaki 50 değer için model çıktısı



Şekil 4.62. 2k testi [48350,48550] değer aralığındaki 50 değer için model çıktısı



Şekil 4.63. 2l testi [48350,48550] değer aralığındaki 50 değer için model çıktısı



Şekil 4.64. 2m testi [48350,48550] değer aralığındaki 50 değer için model çıktısı

"Senaryo 1" ve "Senaryo 2" verisetlerinde 1DCNN uygulaması sonucu elde ettiğimiz test sonuçları Çizelge 4.52'de yer almaktadır.

Çizelge 4.52. "Senaryo 1" ve "Senaryo 2" 1DCNN uygulaması performans değerlendirme sonuçları

Test No	Train Acc	Test Acc	Precision	Recall	F1 Score
1e	0.999736	0.999711	1.0	1.0	1.0
1f	1.0	1.0	1.0	1.0	1.0
1g	0.998937	0.998824	0.99	1.0	0.99
2f	0.946993	0.946492	0.95	0.74	0.78
2g	0.863692	0.946322	0.95	0.75	0.78
2h	0.875692	0.813727	0.72	0.81	0.73
2i	0.816292	0.693302	0.71	0.81	0.67
2j	0.815740	0.700379	0.71	0.81	0.67
2k	0.800376	0.878670	0.74	0.79	0.76
2l	0.859	0.942508	0.90	0.74	0.77
2m	0.947277	0.946834	0.96	0.91	0.93

Çalışmanın devamında modelin veri setindeki diğer saldırıların tespitinde başarısının nasıl olduğunun tespit edilebilmesi için OSS yöntemi ile boyutu azaltılarak model başarısı değerlendirilmiştir. İkinci olarak ise OSS ile boyutu azaltılan veri setine SMOTEENN uygulanarak model başarısı incelenmiştir. Veri setindeki değişimler Çizelge 4.53'te gösterilmiştir.

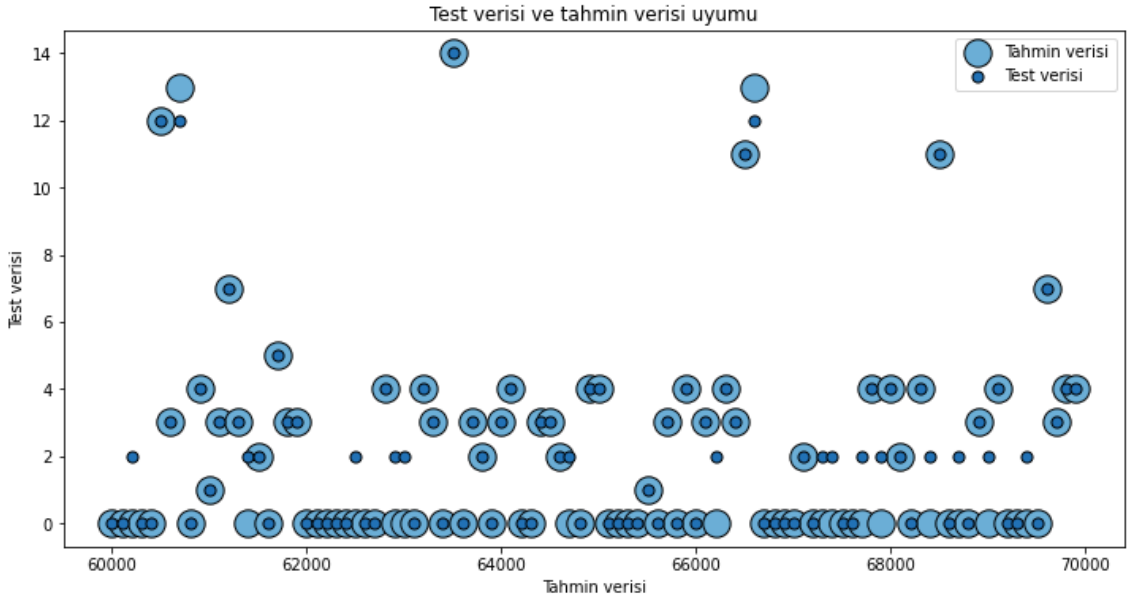
Çizelge 4.53. "Senaryo 3" veri dengeleme işlemi sonrası veri sayıları

Etiket	İlk Durum	OSS	OSS + SMOTEENN
0	6995422	583267	3403
1	286191	90032	5096
2	161204	238788	3657
3	686012	149999	5096
4	576191	150149	4777
5	1730	1730	5090
6	193360	998	5069
7	187589	1104	5088
8	611	998	5045
9	230	999	5091
10	87	87	4756
11	461912	20506	4906
12	10990	10990	5004
13	41508	20025	4858
14	139890	20000	1901

3a) OSS yöntemi ile boyutunu azalttığımız veri setini 1DCNN modeline verdiğimizde Çizelge x'deki başarı değerleri elde edilmiştir. Modelin fit işlemi 10 dk 44 sn sürmüştür. Test verisinin tahmini 16 sn'de gerçekleşmişitr. 3a testine ait sınıflandırma raporu Çizelge 4.54'de, model tahminin gerçekleştirildiği dağılım grafiği Şekil 4.65'te gösterilmiştir.

Çizelge 4.54. 3a maddesine ilişkin sınıflandırma raporu (classification report)

No	Saldırılar/Metrikler	Precision	Recall	F1	Support
0	Benign	0.761889	0.946226	0.844111	116654
1	Bot	0.992063	0.999667	0.995851	18006
2	Infiltration	0.695463	0.28724	0.406562	47758
3	DDoS-HOIC	0.999967	1	0.999983	30000
4	DDoS-LOIC-Http	0.99465	0.996836	0.995742	30030
5	DDoS-LOIC-UDP	0.799534	0.991329	0.885161	346
6	FTP Brute Force	0.731707	0.3	0.425532	200
7	SSH Brute Force	0.99095	0.99095	0.99095	221
8	Brute Force-Web	0.543103	0.633166	0.584687	199
9	Brute Force-XSS	1	0.5	0.666667	200
10	SQL Injection	0	0	0	17
11	DoS Attacks-Hulk	0.998293	0.998293	0.998293	4101
12	DoS Slowloris	0.960193	0.362147	0.525933	2198
13	DoS Golden Eye	0.805192	0.999001	0.891687	4005
14	DoS Slow HttpTest	0.963663	0.9945	0.978839	4000
	Accuracy			0.84	257935
	Macro Avg	0.82	0.73	0.75	257935
	Weighted Avg	0.83	0.84	0.81	257935

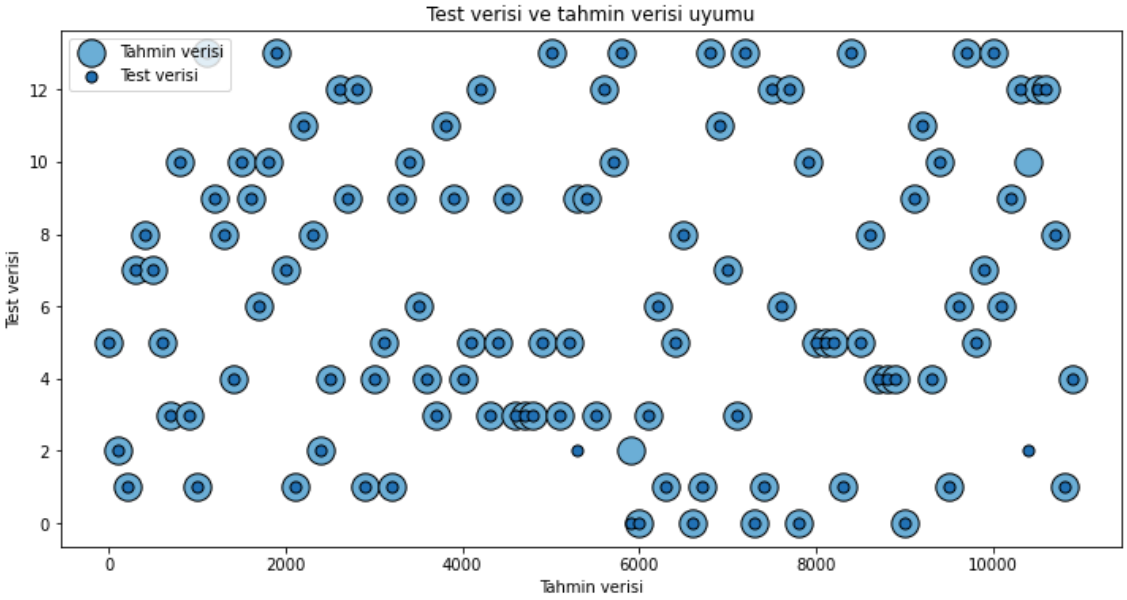


Şekil 4.65. 3a testi 1DCNN modelinde test verisi ve tahmin verisi uyumu

3b) Saldırıların doğru olarak sınıflandırılabilmesi için ikinci çalışmada OSS yöntemi ile boyut azaltıldıktan sonra SMOTEENN tekniği uygulanarak veri seti dengeli hale getirilmeye çalışılmıştır. Modelin fit işlemi 38 sn sürmüştür. Test verisi tahmini 3 sn'dir. 3b testine ait sınıflandırma raporu Çizelge 4.55'de, model tahminin gerçekleştirildiği dağılım grafiği Şekil 4.66'da gösterilmiştir.

Çizelge 4.55. 3b maddesine ilişkin sınıflandırma raporu (classification report)

No	Saldırılar/Metrikler	Precision	Recall	F1	Support
0	Benign	0.873964	0.768222	0.817688	686
1	Bot	1	0.997154	0.998575	1054
2	Infiltration	0.798956	0.858345	0.827586	713
3	DDoS-HOIC	0.999009	1	0.999504	1008
4	DDoS-LOIC-Http	1	0.980932	0.990374	944
5	DDoS-LOIC-UDP	0.991228	1	0.995595	1017
6	FTP Brute Force	0.96425	0.945472	0.954769	1027
7	SSH Brute Force	1	0.994214	0.997099	1037
8	Brute Force-Web	0.792536	0.732995	0.761603	985
9	Brute Force-XSS	0.753128	0.926154	0.830727	975
10	SQL Injection	0.877069	0.778594	0.824903	953
11	DoS Attacks-Hulk	0.994018	0.996004	0.99501	1001
12	DoS Slowloris	1	0.998088	0.999043	1046
13	DoS Golden Eye	0.995758	1	0.997875	939
14	DoS Slow HttpTest	0.861042	0.906005	0.882952	383
	Accuracy			0.93	13768
	Macro Avg	0.93	0.93	0.92	13768
	Weighted Avg	0.93	0.93	0.93	13768



Şekil 4.66. 3b testi 1DCNN modelinde test verisi ve tahmin verisi uyumu

5. SONUÇLAR VE ÖNERİLER

5.1 Sonuçlar

Teknolojinin gelişmesi ve tehdit alanının büyümesinden dolayı ağlar üzerinde siber güvenlik olaylarının tespiti ve iyileştirmesi çalışmaları eylem planlarına girmiş durumdadır. Saldırı tespitini imza tabanlı olarak gerçekleştiren sistemler mevcuttur ve son zamanlarda güvenlik ürünlerine yapay zeka özellikleri de kazandırılmaya başlanmıştır. Yapay zeka ile geliştirilen sistemlerde işlemlerin nasıl sağlandığı, tespit oranının başarılı olup olmadığı merakından haraketle hazırlanan bu tezde, ağ saldırı trafiği veri setinden öğrenme gerçekleştirilerek saldırı tespitinin otomatize olarak sağlanması amaçlanmıştır.

Çalışmalarımız kapsamında derin öğrenme algoritmalarından Evrişimli Sinir Ağları (ESA, CNN), makine öğrenmesi algoritmalarından K En Yakın Komşu (KNN) algoritması kullanılmıştır. Evrişimli Sinir Ağları algoritmalarının bilgisayarlı görü alanında kullanımı örneğin sürücüsüz araç teknolojileri; görüntülerin işlenip değerlendirilmesi ile ilgili olarak biyoloji alanında kullanımı örneğin MR görüntülerinden hastalık teşhisi, bitki hastalıklarının tespit edilerek tarımda iyileştirmelerin sağlanabilmesi; doğal dil işleme (NLP) gibi farklı alanlardaki karmaşık görevleri başarıyla yerine getirebilmesi sebebiyle saldırı tespitinde incelenmesi tercih sebebi olmuştur. Evrişimli sinir ağlarının bir çeşidi olan Bir Boyutlu Evrişimli Sinir Ağları (1B-ESA, 1D-CNN) zamansal serilerin sınıflandırılması, sinyal analizi gibi bir boyutlu verilerde başarılıdır ve model ile ilgili olarak literatürde çok az miktarda çalışma mevcuttur. Bu nedenle saldırı veriseti özniteliklerinden elde edilen özellik vektörünün bir boyutlu matris görüntüsünde olduğu ve zamansal olarak sıralı özellik vektörlerinden oluştuğu düşünülerek Bir Boyutlu Evrişimli Sinir Ağları (1B-ESA, 1D-CNN) kullanılmıştır. Büyük veri kümelerinde kullanım kolaylığı sebebiyle de K En Yakın Komşu algoritması kullanılmıştır.

Modellerin eğitilip test edilmesi aşamasında güncelliğinden dolayı CSE-CIC-IDS2018 veriseti tercih edilmiştir. CSE-CIC-IDS2018 veriseti büyüklüğünden dolayı testler alt gruplara ayrılarak gerçekleştirilmiştir. Oluşturulan test grupları “Senaryo 1” ve “Senaryo 2” olarak adlandırılmıştır. “Senaryo 1” testleri DoS saldırılarının yer aldığı Hulk, GoldenEye, Slowloris, SlowHTTPTest etiketli verilerden oluşmaktadır. “Senaryo 2” testleri sızma (infiltration) ve bot saldırılarından oluşmaktadır. “Senaryo 3” testleri ise

örneklem azaltma tekniklerinden olan OSS yöntemi ile boyutu azaltılmış DoS (Hulk, GoldenEye, Slowloris, SlowHTTPTest), DDoS (HOIC, LOIC Http, LOIC UDP), Bot, infiltration, Brute Force (XSS, Web, FTP, SSH), SQL Injection etiketli saldırılardan oluşmaktadır.

Çalışma kapsamında aşağıdaki sorulara yanıt bulmaya çalışılmıştır:

- "K En Yakın Komşu" ve "Bir Boyutlu Evrişimli Sinir Ağları" modellerinin saldırı tespiti konusundaki başarısı nasıldır?
- Modeller hangi saldırı türlerinin tespit edilmesinde daha başarılıdır?
- Dengesiz verisetinin dengeli hale getirilmesi için kullanılan örneklem azaltma ve örneklem artırma tekniklerinin model başarısına olan etkisi nasıldır?
- Özellik seçiminin model başarısına olan etkisi nasıldır?

Elde edilen sonuçlara göre: (Çizelge 4.56, Çizelge 4.57, Çizelge 4.58, Çizelge 4.59, Çizelge 4.60)

- DoS saldırılarının tahmininde ve anormal durum tespitinde KNN ve 1DCNN modellerine ait performans değerleri incelendiğinde her iki yöntem de başarılıdır ve benzer sonuçlar üretmiştir.
- DoS saldırılarının tahmininde KNN modeli sonuçları incelendiğinde ısı haritası ile özellik azaltmanın başarının arttırılmasında öneminin olmadığı görülmüştür.
- DoS saldırılarının tahmininde tek taraflı seçim (OSS) yöntemi ile örneklem azaltılarak dengelenmeye çalışıldığında KNN yöntemi için eğitim ve test doğruluğunun 0.001 oranında düştüğü gözlenmiştir.
- Bot ve sızma saldırılarından oluşan veriseti 1DCNN ve KNN modellerine uygulandığında en yüksek doğruluğun KNN modeline ait olduğu görülmüştür. Dengesiz verisetlerinin sınıflandırılmasında KNN yönteminin daha başarılı olduğu değerlendirilmiştir.
- Bot ve sızma saldırıları ile zararsız trafiğin tahmin edilmesinde her iki modelde de en iyi tahminin bot saldırılarında olduğu, en kötü tahminin sızma saldırılarında olduğu tespit edilmiştir.
- Senaryo 2 KNN modeli uygulamasında veriseti dengesizliğinin giderilmesi için örneklem arttırma (SMOTEENN, SMOTETomek) ve azaltma teknikleri (OSS) uygulandığında eğitim doğruluğu en yüksek SMOTEENN en düşük OSS'de tespit edilmiştir. Test doğruluğu ise en yüksek OSS, en düşük SMOTEENN'de tespit

edilmiştir. F1 puanlarına bakıldığında ise en iyi yöntem OSS, en düşük SMOTEENN'de tespit edilmiştir.

- Senaryo 2 1DCNN modeli uygulamasında veriseti dengesizliğini gidermek için OSS yöntemi kullanıldığında model performansının benzer olduğu tespit edilmiştir. En düşük performans ise SMOTETomek ile sentetik veri arttırıldığında tespit edilmiştir.
- SMOTETomek ile arttırılan verinin kullanıldığı 1DCNN modelinde aktivasyon fonksiyonu değiştirilerek LeakyReLU uygulandığında performans değişiklik göstermezken, optimize edici olarak RMSProp kullanıldığında en düşük olarak tespit edilen F1 skor değerinin %13 olarak arttığı görülmüştür.
- Senaryo 2 verisi 1DCNN uygulamasında orijinal veriseti ile OSS uygulanan veriseti benzer sonuçlara sahiptir. OSS uygulanan verisetinde XGBoost algoritması kullanılarak özellik seçimi gerçekleştirildiğinde benzer sonuçlar elde edilmiştir.
- Uygulanan tüm modellerde veriseti dengesizliğinin giderilmesi için OSS yöntemi kullanıldığında F1 puanının daha yüksek olduğu ve sentetik veri arttırımı ile kıyaslandığında daha kısa sürede verisetinin dengeye ulaştığı görülmüştür.
- Her iki çalışmada incelendiğinde KNN algoritmasının büyük veri setlerinde başarılı olduğu, veri seti dengesizliğinin çalışmasını olumsuz etkilemediği sadece tahmin işleminin daha uzun sürede gerçekleştiği tespit edilmiştir.
- OSS yönteminin orijinal veri seti özelliklerini koruyarak örneklem azaltımı gerçekleştirdiği sonucundan hareketle model, saldırı gruplarının tamamına uygulandığında F1 puanının Bot, DDoS, Brute Force ve DoS saldırılarında yüksek olduğu; Infiltration ve SQL Injection saldırılarının tahmininde ise düşük olduğu görülmüştür.
- OSS yöntemi ile örneklem azaltılıp SMOTEENN yöntemi ile model dengelenerek örneklem arttırıldığında 0.75 olan F1 puanının 0.92 puana ulaşarak arttığı, saldırı çeşidi tahminin doğru gerçekleştiği görülmüştür.
- 1DCNN modelinde yalnızca OSS uygulanmış olan veri seti ile gerçekleştirilen testte 0.84 olan doğruluk değerinin OSS ve SMOTEENN birlikte kullanıldığında 0.93'e ulaştığı görülmüştür. Örneklem artırma ve azaltma işleminde hibrit yöntemler kullanılmasının başarıyı olumlu olarak etkilediği gözlemlenmiştir.

Çizelge 4.56. Senaryo 1 testleri sınıflandırma raporu (classification report)

Yöntem	No	Değişiklik	Saldırılar/Metrikler	Precision	Recall	F1	Support
KNN	1a	Isı Haritası	Benign	1	1	1	459190
			Hulk	1	1	1	147938
			SlowLoris	0.999	0.999	0.999	3478
			GoldenEye	0.999	0.999	0.999	13355
			SlowHTTPTest	1	1	1	44559
	1c	X	Benign	1	1	1	459190
			Hulk	1	1	1	147938
			SlowLoris	1	1	1	3478
			GoldenEye	1	1	1	13355
			SlowHTTPTest	1	1	1	44559
	1d	OSS	Benign	1	1	1	459190
			Hulk	1	1	1	147938
			SlowLoris	0.98	1	0.99	3478
			GoldenEye	0.97	1	0.98	13355
			SlowHTTPTest	1	1	1	44559
1DCNN	1e	X	Benign	1	1	1	459190
			Hulk	1	1	1	147938
			SlowLoris	0.99	0.77	0.86	3478
			GoldenEye	1	0.98	0.99	13355
			SlowHTTPTest	1	1	1	44559
	1g	OSS	Benign	1	1	1	459190
			Hulk	1	1	1	147938
			SlowLoris	0.98	1	0.99	3478
			GoldenEye	0.96	1	0.98	13355
			SlowHTTPTest	1	1	1	44559
KNN	1b	Isı Haritası	Attack	1	1	1	209330
			Benign	1	1	1	459190
1DCNN	1f	X	Attack	1	1	1	209330
			Benign	1	1	1	459190

Çizelge 4.57. Senaryo 1 testleri performans değerlendirme sonuçları

Test No	Train Acc	Test Acc	Precision	Recall	F1 Score
1a	0.999964	0.999947	1	1	1
1b	0.999914	0.999913	1	1	1
1c	0.999956	0.999937	1	1	1
1d	0.998597	0.999014	0.99	1	0.99
1e	0.999736	0.999711	1.0	1.0	1.0
1f	1.0	1.0	1.0	1.0	1.0
1g	0.998937	0.998824	0.99	1.0	0.99

Çizelge 4.58. Senaryo 2 testleri sınıflandırma raporu (classification report)

Yöntem	No	Değişiklik	Saldırılar/Metrikler	Precision	Recall	F1	Support
KNN	2a	X	Benign	1	1	1	318438
			Bot	1	1	1	91157
			Infiltration	1	0.996	0.998	29671
	2c	OSS	Benign	0.94	0.98	0.96	318438
			Bot	1	1	1	91157
			Infiltration	0.57	0.34	0.43	29671
	2d	SMOTEENN	Benign	0.96	0.8	0.87	318438
			Bot	1	1	1	91157
			Infiltration	0.23	0.66	0.35	29671
	2e	SMOTETomek	Benign	0.95	0.84	0.89	318438
			Bot	1	1	1	91157
			Infiltration	0.25	0.57	0.35	29671
1DCNN	2f	X	Benign	0.93	1	0.96	318438
			Bot	1	1	1	91157
			Infiltration	0.9	0.24	0.37	29671
	2g	OSS	Benign	0.93	1	0.96	312020
			Bot	1	1	1	89352
			Infiltration	0.91	0.24	0.38	29108
	2h	SMOTEENN	Benign	0.96	0.78	0.86	318438
			Bot	1	1	1	91157
			Infiltration	0.21	0.65	0.32	29671
	2i	SMOTETomek	Benign	0.97	0.59	0.74	318438
			Bot	1	1	1	91157
			Infiltration	0.16	0.82	0.27	29671
	2j	SMOTETomek + LeakyReLU	Benign	0.97	0.6	0.75	318438
			Bot	1	1	1	91157
			Infiltration	0.16	0.81	0.27	29671
	2k	SMOTETomek + RMSProp	Benign	0.95	0.88	0.91	318438
			Bot	1	1	1	91157
			Infiltration	0.28	0.49	0.36	29671
	2l	OSS + XGBoost	Benign	0.93	0.99	0.96	312020
			Bot	1	1	1	89352
			Infiltration	0.76	0.24	0.36	29108
KNN	2b	X	Benign	0.94	0.99	0.96	318438
			Attack	0.96	0.83	0.89	120828
1DCNN	2m	X	Benign	0.93	1.00	0.96	318438
			Attack	0.99	0.81	0.89	120828

Çizelge 4.59. Senaryo 2 testleri performans değerlendirme sonuçları

Test No	Train Acc	Test Acc	Precision	Recall	F1 Score
2a	0.999803	0.999676	1	1	1
2b	0.952302	0.943553	0.95	0.91	0.93
2c	0.881936	0.937825	0.84	0.77	0.79
2d	0.994349	0.829832	0.73	0.82	0.74
2e	0.936767	0.854363	0.73	0.80	0.75
2f	0.946993	0.946492	0.95	0.74	0.78
2g	0.863692	0.946322	0.95	0.75	0.78
2h	0.875692	0.813727	0.72	0.81	0.73
2i	0.816292	0.693302	0.71	0.81	0.67
2j	0.815740	0.700379	0.71	0.81	0.67
2k	0.800376	0.878670	0.74	0.79	0.76
2l	0.859	0.942508	0.90	0.74	0.77
2m	0.947277	0.946834	0.96	0.91	0.93

Çizelge 4.60. Senaryo 3 testleri performans değerlendirme sonuçları

Test No	Train Acc	Test Acc	Precision	Recall	F1 Score
3a	0.84	0.84	0.82	0.73	0.75
3b	0.93	0.93	0.93	0.93	0.92

5.2. Öneriler

Siber saldırıların tespiti konusunda incelemesini yaptığımız yapay zekâ algoritmalarının başarılı olduğu görülmüştür. Siber güvenliğin farklı alanlarında çalışan uzman kişilerin görüşünün alınması, sistemlerdeki eksikliklerin giderilmesi için ihtiyaçların doğru bir şekilde belirlenmesi, çeşitli saldırılara maruz kalan firma ve kurumların saldırıya ait trafik bilgilerini paylaşması ile veya saldırı senaryolarının geliştirilerek veri setlerinin geliştirilmesi, yeteneklerine göre kullanılacak algoritmaların belirlenmesi, algoritmalarda hiperparametre ince ayarlarının yapılması ile faydalı çalışmalar yapılabileceği değerlendirilmiştir.

6. KAYNAKLAR

, Sinir hücresi - Vikipedi, https://tr.wikipedia.org/wiki/Sinir_h%C3%BCcresi:

Neuron Hand-tuned - Sinir hücresi - Vikipedi.

CS 230 - Derin Öğrenme püf noktaları ve ipuçları el kitabı.

Activation Functions — ML Glossary documentation.

, Conv1D layer, https://keras.io/api/layers/convolution_layers/convolution1d/:

Akyel, N. ve Seçkin, K., 2012, k-En Yakın Komşuluk Algoritmasının Hile Denetiminde Kullanımı, *Muhasebe ve Vergi Uygulamaları Dergisi*.

Altun, S. ve Talu, M. F., 2020, Derin Sinir Ağları için Hiperparametre Metodlarının ve Kitlerinin İncelenmesi, *DÜMF Mühendislik Dergisi*, 12 (2), 187-199.

Altunay, H. C. ve Albayrak, Z., 2021, Network Intrusion Detection Approach Based on Convolutional Neural Network, *European Journal of Science and Technology Special Issue*, 26 (26), 22-29.

Amidi, S. ve Amidi, A., CS 230 - Evrişimli Sinir Ağları El Kitabı, https://stanford.edu/~shervine/l/tr/teaching/cs-230/cheatsheet-convolutional-neural-networks:

Ataş, M. ve Alhajahmad, B., 2023, Bilgisayarlı Görmede Topluluk Öğrenimi (Ensemble Learning) Yaklaşımları, *AS-Proceedings*, 1 (2), 451-455.

Atefi, K., Hashim, H. ve Kassim, M., 2019, Anomaly analysis for the classification purpose of intrusion detection system with K-nearest neighbors and deep neural network, *Proceeding - 2019 IEEE 7th Conference on Systems, Process and Control, ICSPC 2019*.

Baykara, M. ve Daş, R., 2019, Saldırı tespit ve engelleme araçlarının incelenmesi, *DÜMF Mühendislik Dergisi*, 10 (1), 57-75.

Brownlee, J., 2019a, A Gentle Introduction to Cross-Entropy for Machine Learning, https://machinelearningmastery.com/cross-entropy-for-machine-learning/:

Brownlee, J., 2019b, A Gentle Introduction to Information Entropy, https://machinelearningmastery.com/what-is-information-entropy/:

Çarkacı, N., 2018, Derin Öğrenme Uygulamalarında En Sık kullanılan Hiper-parametreler, https://medium.com/deep-learning-turkiye/derin-ogrenme-uygulamalarinda-en-sik-kullanilan-hiper-parametreler-ece8e9125c4:

Chen, T. ve Guestrin, C., 2016, XGBoost: A Scalable Tree Boosting System.

Croft, B. ve Gilmore, J., 1985, RFC 951: Bootstrap Protocol (BOOTP).

Ding, B., Qian, H. ve Zhou, J., 2018, Activation functions and their characteristics in deep neural networks, *Proceedings of the 30th Chinese Control and Decision Conference, CCDC 2018*, 1836-1841.

Güler, E., Kakiz, T., Günay, F. B., Şanal, B. ve Çavdar, T., 2023, Kapalı Mekân Ortamında 1D-CNN Kullanarak Yapılan Doluluk Tespiti Sınıflandırması, *Karadeniz Fen Bilimleri Dergisi*, 13 (1), 60-71.

Gulsen, F., 2021, Sinir Ağları Regresyonu - TensorFlow.

Hacıbeyoğlu, M., Çelik, M., Erdaş Çiçek, Ö., Yazar, □., Author, C., Erbakan Üniver□ite□i, N., Bilimleri Enstitüsü, F., Mühendisliği Anabilim Dalı, B., Fakültesi, M., Mühendisliği Bölümü, B., Bilgileri, M. ve Geçmişi, M., 2023, En Yakın Komşu Algoritması ile Binalarda Enerji Verimliliği Tahmini, *Fen ve Mühendislik Bilimleri Dergisi*.

İnik, Ö. ve Ülker, E., 2017, Derin Öğrenme ve Görüntü Analizinde Kullanılan Derin Öğrenme Modelleri. Gaziosmanpaşa Bilimsel Hazırlık Dergisi (GBAD).

Karakuş, C., Makine Öğrenmesi Temelleri Ders Notu, p.

Karaman, M. S., 2020, Anomali Tabanlı Saldırı Tespit Sistemlerinde Makine Öğrenimi Modellerinin Performans Değerlendirmesi. YL Tezi**:** 1-93.

Karaman, M. S., Turan, M. ve Aydin, M. A., 2020, Yapay Sinir Ağı Kullanılarak Anomali Tabanlı Saldırı Tespit Modeli Uygulaması, *European Journal of Science and Technology Special Issue*, 17-25.

Karataş, G., 2020, Derin Öğrenme Tabanlı Saldırı Tespit Sistemi. İstanbul.

Kılıçarslan, S. ve Adem, K., 2021, An overview of the activation functions used in deep learning algorithms, *Journal of New Results in Science*, 10 (3), 75-88.

Kilichev, D. ve Kim, W., 2023, Hyperparameter Optimization for 1D-CNN-Based Network Intrusion Detection Using GA and PSO, *Mathematics 2023, Vol. 11, Page 3724*, 11 (17), 3724-3724.

Kılınç, F. ve Eyüpoğlu, C., 2023, Ağ Ortamındaki Saldırı Türleri: Saldırı Senaryo Örnekleri, *İstanbul Ticaret Üniversitesi Teknoloji ve Uygulamalı Bilimler Dergisi*, 6 (1), 99-109.

Kiranyaz, S., Avci, O., Abdeljaber, O., Ince, T., Gabbouj, M. ve Inman, D. J., 2021, 1D convolutional neural networks and applications: A survey, *Mechanical Systems and Signal Processing*, 151, 107398-107398.

Kızrak, A., 2018, Şu Kara Kutuyu Açalım: Yapay Sinir Ağları, https://ayyucekizrak.medium.com/şu-kara-kutuyu-açalim-yapay-sinir-ağları-7b65c6a5264a:

Kızrak, A., 2019, Derin Öğrenme İçin Aktivasyon Fonksiyonlarının Karşılaştırılması, https://ayyucekizrak.medium.com/derin-öğrenme-için-aktivasyon-fonksiyonlarının-karşılaştırılması-cee17fd1d9cd:

Koşan, M. A., Coşkun, A. ve Karacan, H., 2019, Yapay Zeka Yöntemlerinde Entropi, *Journal of Information Systems and Management Research*.

Kurt, F. ve Efe, M. Ö., 2018, Evrişimli Sinir Ağlarında Hiper Parametrelerin Etkisinin İncelenmesi.

Lam, N. T., 2021, Detecting Unauthorized Network Intrusion based on Network Traffic using Behavior Analysis Techniques, *International Journal of Advanced Computer Science and Applications*, 12 (4).

Leevy, J. L. ve Khoshgoftaar, T. M., 2020, A survey and analysis of intrusion detection models based on CSE-CIC-IDS2018 Big Data, *Journal of Big Data*, 7 (1).

Luo, J., Jin, J. ve Shan, F., 2017, Standardization of Low-Latency TCP with Explicit Congestion Notification: A Survey. IEEE Internet Computing, Institute of Electrical and Electronics Engineers Inc. 21**:** 48-55.

Mahendru, K., 2019, Loss Function | Loss Function In Machine Learning, https://www.analyticsvidhya.com/blog/2019/08/detailed-guide-7-loss-functions-machine-learning-python-code/:

Maughan, D., Schertler, M., Schneider, M. ve Turner, J., 1998, RFC 2408: Internet Security Association and Key Management Protocol (ISAKMP).

Mohammadpour, L., Ling, T. C., Liew, C. S. ve Aryanfar, A., 2022, A Survey of CNN-Based Network Intrusion Detection. Applied Sciences (Switzerland). 12.

Osken, S., Yildirim, E. N., Karatas, G. ve Cuhaci, L., 2019, Intrusion detection systems with deep learning: A systematic mapping study, *2019 Scientific Meeting on Electrical-Electronics and Biomedical Engineering and Computer Science, EBBT 2019*.

Prabhu, 2018, Understanding of Convolutional Neural Network (CNN).

Qazi, E. U. H., Almorjan, A. ve Zia, T., 2022, A One-Dimensional Convolutional Neural Network (1D-CNN) Based Deep Learning System for Network Intrusion Detection, *Applied Sciences (Switzerland)*, 12 (16).

Schneider, P., 1997, TCP/IP Traffic Classification Based on Port Numbers, *Division of Applied Sciences Harvard University*.

Şeker, A., Diri, B. ve Balik, H. H., 2017, Derin Öğrenme Yöntemleri ve Uygulamaları Hakkında Bir İnceleme. 3**:** 47-64.

Ser, G. ve Bati, C. T., 2019, Determining the Best Model with Deep Neural Networks: Keras Application on Mushroom Data, *Yüzüncü Yıl Üniversitesi Tarım Bilimleri Dergisi Cilt*, 29.

Sert, Z., 2020, Evrişimsel Sinir Ağları (ESA).

Seyyarer, E., Ayata, F., Uçkan, T. ve Karci, A., 2020, Derin Öğrenmede Kullanılan Optimizasyon Algoritmalarının Uygulanması ve Kıyaslanması.

Tensorflow, Windows'ta kaynaktan derle, https://www.tensorflow.org/install/source_windows?hl=tr:

Tuğrul, B. ve Ahmed, A. S. A., 2022, Makine öğrenme yöntemleri ile ağ trafik analizi, *Ömer Halisdemir Üniversitesi Mühendislik Bilimleri Dergisi*.

UNB, IDS 2018 | Datasets | Research | Canadian Institute for Cybersecurity | UNB, https://www.unb.ca/cic/datasets/ids-2018.html:

Versloot, C., 2019, How to use binary categorical crossentropy with keras.

Wesley, E., 2022, RFC 9293: Transmission Control Protocol (TCP).

Yu, J., Shi, S., Zhang, F., Chen, G. ve Cao, M., 2019, PredGly: Predicting lysine glycation sites for Homo sapiens based on XGboost feature optimization, *Bioinformatics*, 35 (16).