GRAF VE KAREKOD YÖNTEMLERİYLE DÖNÜŞTÜRÜLMÜŞ LOG KAYITLARI ÜZERİNDE DERİN ÖĞRENME TABANLI SİBER SALDIRI TESPİTİ

2023
DOKTORA TEZİ
BİLGİSAYAR MÜHENDİSLİĞİ

Yusuf ALACA

Tez Danışmanı
Doç. Dr. Yüksel ÇELİK

GRAF VE KAREKOD YÖNTEMLERİYLE DÖNÜŞTÜRÜLMÜŞ LOG KAYITLARI ÜZERİNDE DERİN ÖĞRENME TABANLI SİBER SALDIRI TESPİTİ

Yusuf ALACA

Tez Danışmanı
Doç. Dr. Yüksel ÇELİK

T.C.
Karabük Üniversitesi
Lisansüstü Eğitim Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalında
Doktora Tezi
Olarak Hazırlanmıştır

KARABÜK
Şubat 2023

Yusuf ALACA tarafından hazırlanan "GRAF VE KAREKOD YÖNTEMLERİYLE DÖNÜŞTÜRÜLMÜŞ LOG KAYITLARI ÜZERİNDE DERİN ÖĞRENME TABANLI SİBER SALDIRI TESPİTİ" başlıklı bu tezin Doktora Tezi olarak uygun olduğunu onaylarım.

Doç. Dr. Yüksel ÇELİK .....
Tez Danışmanı, Bilgisayar Mühendisliği Anabilim Dalı

Bu çalışma, jürimiz tarafından Oy Birliği ile Bilgisayar Mühendisliği Anabilim Dalında Doktora tezi olarak kabul edilmiştir. 24/02/2023

Ünvanı, Adı SOYADI (Kurumu) İmzası

Başkan : Dr. Öğr. Üyesi Eyüp Burak CEYHAN (BÜ) .....

Üye : Doç. Dr. Yüksel ÇELİK (KBÜ) .....

Üye : Doç. Dr. İlker TÜRKER (KBÜ) .....

Üye : Dr. Öğr. Üyesi Kürşat Mustafa KARAOĞLAN (KBÜ) .....

Üye : Dr. Öğr. Üyesi Erdal BAŞARAN (AİÇÜ) .....

KBÜ Lisansüstü Eğitim Enstitüsü Yönetim Kurulu, bu tez ile, Doktora derecesini onamıştır.

Prof. Dr. Müslüm KUZU .....
Lisansüstü Eğitim Enstitüsü Müdürü

"Bu tezdeki tüm bilgilerin akademik kurallara ve etik ilkelere uygun olarak elde edildiğini ve sunulduğunu; ayrıca bu kuralların ve ilkelerin gerektirdiği şekilde, bu çalışmadan kaynaklanmayan bütün atıfları yaptığımı beyan ederim."

Yusuf ALACA

ÖZET

Doktora Tezi

GRAF VE KAREKOD YÖNTEMLERİYLE DÖNÜŞTÜRÜLMÜŞ LOG KAYITLARI ÜZERİNDE DERİN ÖĞRENME TABANLI SİBER SALDIRI TESPİTİ

Yusuf ALACA

Karabük Üniversitesi
Lisansüstü Eğitim Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı

Tez Danışmanı:
Doç. Dr. Yüksel ÇELİK
Şubat 2023, 83 sayfa

Bilişim ve iletişim teknolojilerinin gelişip her alanda büyük faydalar sağlamalarının yanında, riskleri de barındırmaktadır. Bu risklerin başında bu teknolojileri kullanan devlet altyapıları, üretim tesisleri, ticaret, finans, haberleşme ve ulaşım sistemlerin siber saldırılara maruz kalmasıdır. Siber saldırılara maruz kalan sistemlere bağlı hizmetlerin sekteye uğraması, büyük mali kayıplara sebep olabilmektedir. Saldırıları tespit etmek ve engellemek için siber savunma teknolojileri geliştirilmesine rağmen, siber saldırı yöntemleri de aynı şekilde geliştirilmektedir. Siber savunmada, iletişim ağları üzerinde çok büyük verilerin olması, saldırı türlerinin fazlalığı ve bilgisayar korsanlığı becerilerinin giderek gelişmesinden dolayı geleneksel saldırı tespit yöntemlerinin yetersizliği önemli bir problemdir. Bu problemin üstesinden gelmek için araştırmacılar, son yıllarda makine öğrenmesi ve derin öğrenme yöntemleri başta olmak üzere birçok saldırı algılama mekanizmaları üzerinde yoğunlaşmışlardır. Biz

de bu çalışmamızda derin öğrenme temelli iki farklı özgün saldırı tespit metodu önererek bu problemlerin çözümüne katkı sunmaya çalıştık.

İlk olarak, graf tabanlı log ön işleme yöntemine, graf algoritması olan Node2Vec kullanarak karmaşık ve farklı yapılardaki log şablonları arasında ilişki kurularak tek tip hale dönüştürülerek vektörel hale getirilmesi sağlanmıştır. Daha sonra bu veriler LSTM derin öğrenme algoritmasına girdi olarak verilmiş ve model eğitilmiş, Graf tabanlı LSTM (Graph based LSTM, GLSTM) yöntemi önerilmiş ve siber saldırılar tespit edilmiştir. Önerilen GLSTM yaklaşımıyla, log kayıtlarının ön işlem adımlarından sonra graf tabanlı olarak vektörel hali oluşturularak derin öğrenme algoritması yardımıyla analiz edilmiştir. Önerilen modeli test etmek için Hadoop 'un farklı sistemlerden ve heterojen kaynaklardan topladığı, farklı yapılara sahip HDFS veri seti kullanılmıştır. GLSTM metodu ile yapılan deneysel çalışmalar sonucu elde edilen %97.01 doğruluk oranı, literatürdeki diğer benzer çalışmalarla kıyaslanmış ve başarısı gösterilmiştir.

İkinci olarak, siber saldırıların tespit edilmesinde her bir saldırının karekodunu oluşturarak, hafif derin öğrenme modellerini kullanarak en uygun özelliklerin seçimini optimizasyon algoritmasıyla otomatik olarak yapılmasını sağlayan çok amaçlı optimizasyon tabanlı hibrit bir yöntem olan CNN tabanlı karekodlarla log kayıtlarından siber saldırı tespiti yapan (CNN based QR Code Log, CNNQRLog) modeli önerilmiştir. İlk olarak çok sınıfa sahip hacimli verilerin QR kod resimleri oluşturulmuştur. Ardından MobileNetV2 ve ShuffleNet CNN modelleri kullanılarak QR kod görüntüleri eğitilmiştir. Eğitilen görüntülere ait derin CNN modelleri ile özellikleri çıkarılmış ve sınıflandırma amacına yönelik en etkili özelliklerin belirlenmesi için Harris Hawk Optimizasyon (HHO) algoritması özellik seçim amacıyla kullanılmıştır. Sonuç olarak önerilen hibrit model HHO ile seçilen özelliklerin sınıflandırılması sonucu saldırı türleri %95.89 doğruluk oranı elde edilmiş olup modelin diğer benzer CNN modeller ile kıyaslanarak daha iyi performans gösterdiği gösterilmiştir.

Anahtar Sözcükler : Anomali tespiti, Graf, Node2Vec, Derin öğrenme, HHO, ShuffleNet, MobileNetV2.

Bilim Kodu **:** 92432

ABSTRACT

Ph. D. Thesis

DEEP LEARNING-BASED CYBER ATTACK DEDECTION ON ENCODED LOG BY GRAPH AND QR CODE METHODS

Yusuf ALACA

Karabük University
Institute of Graduate Programs
Department of Computer Engineering

Thesis Advisor:
Assoc. Prof. Dr. Yüksel ÇELİK
February 2023, 83 pages

In addition to the fact that information and communication technologies develop and provide great benefits in every field, they also contain risks. At the beginning of these risks is the exposure of government infrastructures, production facilities, trade, finance, communication, and transportation systems using these technologies to cyber-attacks. Interruption of services connected to systems exposed to cyber-attacks can cause great financial losses. Although cyber defense technologies have been developed to detect and prevent attacks, cyber-attack methods are being developed similarly. In cyber defense, the inadequacy of traditional intrusion detection methods is an important problem due to the large amount of data on communication networks, the abundance of attack types, and the gradual development of hacking skills. To overcome this problem, researchers have focused on many attack detection mechanisms, especially machine learning and deep learning methods, in recent years

In this study, we tried to contribute to solving these problems by suggesting two unique attack detection methods based on deep learning.

First, the graph-based log preprocessing method was transformed into a vector by establishing a relationship between complex and different log templates using the graph algorithm Node2Vec. Then, these data were given as input to the LSTM deep learning algorithm, and the model was trained, Graph based LSTM (Graph-based LSTM, GLSTM) method was proposed, and cyber attacks were detected. With the proposed GLSTM approach, a graph-based vector version was created and analyzed with the help of a deep learning algorithm after the preprocessing steps of the log records. The HDFS dataset with different structures collected by Hadoop from different systems and heterogeneous sources was used to test the proposed model. The 97.01% accuracy rate obtained from experimental studies with the GLSTM method was compared with similar studies in the literature, and its success was demonstrated.

Secondly, the CNN-based QR Code detects cyber attacks from log records with a CNN-based data matrix. This multi-purpose optimization-based hybrid method creates the data matrix of each attack. It uses light deep learning models to automatically select the most convenient features with the optimization algorithm. Log, CNNQRLog) the model has been proposed. First, QR code images of bulky data with multiple classes were created. Then, QR code images were trained using MobileNetV2 and ShuffleNet CNN models. Deep CNN models and features of the trained images were extracted, and the Harris Hawk Optimization (HHO) algorithm was used to select the most effective features for classification purposes. As a result, a 95.89% accuracy rate of attack types was obtained due to the classification of selected features with the proposed hybrid model HHO. It was shown that the model performed better when compared with other similar CNN models.

Key Word **:** Anomaly detection, Graph, Node2Vec, Deep learning, HHO, ShuffleNet, MobileNetV2.

Science Code : 92432

TEŞEKKÜR

Bu tez çalışmasının planlanmasında, araştırılmasında, yürütülmesinde ve oluşumunda ilgi ve desteğini esirgemeyen, engin bilgi ve tecrübelerinden yararlandığım, yönlendirmeleri ve bilgilendirmeleriyle çalışmamı bilimsel temeller ışığında şekillendiren sayın hocam Doç. Dr. Yüksel ÇELİK'e sonsuz teşekkürlerimi sunarım.

Ayrıca doktora sürecinde önemli katkıları ve desteklerinden dolayı tez izleme kurulu ve tez savunma kurulu üyeleri Sayın Doç. Dr. İlker TÜRKER, Sayın Dr. Öğr. Üyesi Eyüp Burak CEYHAN, Sayın Dr. Öğr. Üyesi Erdal BAŞARAN ve Sayın Dr. Öğr. Üyesi Kürşat Mustafa KARAOĞLAN hocalarıma en kalbi durgularımla teşekkür ederim. Ayrıca tüm destekleri ve yönlendirmeleri için Sayın Doç. Dr. Akif AKGÜL hocama teşekkür ederim.

Her konuda sabırla yanımda olan haklarını hiçbir zaman ödeyemeyeceğim sevgili eşim Kübra ALACA ve biricik kızım Ayşe Hüma ALACA'ya tüm destekleri, büyük fedakarlıkları ve maddi, manevi hiçbir yardımı esirgemeden yanımda oldukları için tüm kalbimle teşekkür ederim.

Dua ve destekleriyle hep yanımda hissettiğim annem, abim Fuat ALACA, kardeşlerim Kevser ALACA, Hanımşah ALACA ve ablalarıma teşekkür ederim.

İÇİNDEKİLER

Sayfa

Sayfa

Sayfa

Sayfa

ŞEKİLLER DİZİNİ

Sayfa

ÇİZELGELER DİZİNİ

Sayfa

SİMGELER VE KISALTMALAR DİZİNİ

SİMGELER

x : giriş değeri

y : çıkış değeri

W, R : ağırlık

b : bias

c : hücre değeri

i : giriş kapısı

z : güncelleme değeri

$f^{(t)}$: unutma kapısı

$o^{(t)}$: çıkış kapısı

σ : sigmoid aktivasyon fonksiyonu

g, h : aktivasyon fonksiyonu

erf(z) : hata işlevi

t : zaman

r : sıfırlama kapısı

f : giriş katmanından gelen değer

j : satır

k : sütun

o : çıkış görüntü boyutu

s : filtrenin kayma miktarı

$\forall$: her

β : hiper parameter

m, θ : önceki gradyanların üssel ortalaması

v : gradyanların kareleri

KISALTMALAR

BGL : BlueGene/L

BLSTM : Bi Long Short-Term Memory (Bi Uzun Kısa Süreli Bellek)

CAPTCHA : Computers and Humans Apart'ı (Bilgisayar ve İnsan Apart'ı)

CNN : Convolutional Neural Network (Evrişimli Sinir Ağı)

CNNQRLOG : CNN tabanlı QR Log

DDoS : Distributed Denial of Service Attack (Dağıtılmış Hizmet Reddi Saldırıları)

DN : Doğru Negatif

DNN : Deep Neural Network (Derin Sinir Ağları)

DoS : Denial of Service (Hizmet Reddi Saldırıları)

DP : Doğru Pozitif

ELM : Extreme Learning Machines (Ekstrem Öğrenme Makineleri)

FS : Features Selection (Özellik Seçimi)

FTP : File Transfer Protocol (Dosya Transfer Protokol)

GD : Gradient Descent (Gradyan İniş)

GLSTM : Graf bazlı LSTM

GRU : Gated Recurrent Unit (Kapılı Tekraryan Birim)

HDFS : Hadoop Distributed File System (Hadoop Dağıtılmış Dosya Sistemi)

HHO : Harris Hawks Optimization (Harris Şahinleri Optimizasyonu)

IDS : Intrusion Detection System (Saldırı Tespit Sistemi)

IMDB : Internet Movie Database (İnterner Flim Veritabanı)

IoT : Internet of Things (Nesnelerin İnterneti)

IPLoM : Iterative Partitioning Log Mining (Yinelemeli Bölümleme Log Madenciliği)

KNN : K-Nearest Neighbors (K-En Yakın Komşu)

LSTM : Long Short-Term Memory (Uzun Kısa Süreli Bellek)

NB : Naive Bayes

ReLU :Rectified Linear Unit (Doğrultulmuş Doğrusal Birim)

RNN	: Recurrent Neural Network (Tekrarlamalı Sinir Ağı)
ROC	: Receiver Operating Characteristic Curve (Alıcı İşlem Karakteristiği)
SQL	: Structured Query Language (Yapılandırılmış Sorgu Dili)
SVM	: Support Vector Machine (Destek Vektör Makinesi)
TF-IDF	: Term Frequency - Inverse Document Frequency (Terim Frekansı - Ters Belge Frekansı)
XSS	: Cross Site Scripting (Siteler Arası Komut Dosyası Çalıştırma)
YN	: Yanlış Negatif
YP	: Yanlış Pozitif

BÖLÜM 1

GİRİŞ

1.1. PROBLEM TANIMI

Teknoloji hayatımızı kolaylaştırırken riskleri ve tehditleri de beraberinde getirmiştir. Bilgisayar, sunucu, bulut cihazları, cep telefonu, tablet gibi ürünlerin ve giyilebilir teknolojilerin hayatımıza girmesiyle birlikte kişisel veri depolama, siber güvenlik, siber saldırılar ve makine öğrenmesi gibi teknolojik kavramlar hayatımıza girmiştir. Bu teknolojik gelişmeler kişisel veri kavramını ve mahremiyetini ön plana çıkarırken, devletleri ve şirketleri bu verileri korumak ve kötü niyetli siber saldırganların eline geçmesini engellemek için siber güvenliğe büyük önem vermeye zorlamıştır. Kişisel veri güvenliğini sağlamak için gerekli tüm araçlar kullanılarak siber güvenlik önlemleri doğru bir şekilde alınsa dahi ihlaller meydana gelebilmektedir. Teknolojinin hızlı bir şekilde gelişmesiyle birlikte sunucular, ağlar, web siteleri, mobil ve masaüstü uygulamaları gibi pek çok bileşenin yüksek değerleri nedeniyle siber saldırıya uğrayan hedefler haline gelmiştir[1].

Siber saldırılar çok farklı amaç ve motivasyonlarla gerçekleştirilmektedir. Bu bileşenler birbirinden farklı olduğundan dolayı sunucu saldırıları[2], ağ saldırıları[3],web güvenliği için dosya aktarım protokolüne kaba kuvvet saldırıları (File Transfer Protocol, FTP-BruteForce), SQL enjeksiyon ve siteler arası komut çalıştırma (Cross Site Scripting, XSS) saldırıları[4], mobil telefon saldırıları[5] ve masaüstü uygulama saldırıları[6] gibi yapılan saldırılarda birbirinden farklı hizmet durdurma veya kaba kuvvet saldırıları olmaktadır.

Teknolojik gelişmelerle birlikte makine öğrenimi, derin öğrenme ve yapay zeka teknikleri görüntü, ses, sağlık, tarım, siber güvenlik gibi birçok alanda kullanılmaya başlanmıştır. Siber güvenliğin sağlanması için önceki siber saldırılardan elde edilen veriler kullanılarak bir sonraki siber saldırının nerede ve nasıl gerçekleşeceğinin

tahmin edilmesi ve bilgisayarların insan faktörünün yerini alması sağlanarak daha hızlı ve performanslı çözümler sunulmaktadır. Siber tehditleri tespit etmek, suç ve

suçlularla mücadele etmek için yapay zeka yöntemleri teknolojik gelişimde bizi bir adım daha ileriye taşıyarak insanlığın gücüne katkı sağlayacaktır. Elde edilen siber verilerin yapay zeka algoritmalarına öğretilmesi ile geleneksel insanlığın gücünden faydalanılan sistemlerden daha hızlı çalışan sistemler ile siber olayların tespit edilmesinde daha etkili olmaktadır [7–10].

Son zamanlarda siber saldırılar hem karmaşık hem de akıllıcı oldukları için saldırıların tespit edilmesi ve engellenmesi zorlaşmaktadır[11]. Bu zorlukları aşmak için birden çok saldırı tespit sistemleri geliştirilmiştir. Saldırı tespit sistemleri imza tabanlı ve anomali tespiti olmak üzere iki şekilde saldırıları tespit etmektedir[12]. Bu iki yöntemde saldırıları tespit etmek için makine öğrenmesi teknikleri kullanılarak daha önce yapılan saldırıların izleri takip edilmiş, gelecekte yapılan saldırıları tespit ve engellemek için bu izler kullanılmıştır[13]. Makine öğrenmesi yöntemleri ile saldırı tespit sistemlerinde daha önce yapılan saldırıları baz alıp, gelecekteki saldırıları tespit etmek için görüntü işleme teknikleri kullanarak yüksek doğruluk ve performans elde edildiği görülmüştür[14]. Bilgisayar ağlarında siber saldırılardan kullanıcıları korumak ve güvenliğini artırmak amacıyla her bir kullanıcı için QR kod oluşturulup, kullanıcıların bu QR kodlarla sisteme giriş yapması sağlanmıştır[15].

Bu tez çalışmasıyla bilgisayarlar, sunucular, IoT, bulut cihazları, bilgisayar ağları, akıllı telefonlar, tabletler ve masaüstü uygulamaları gibi yazılım, donanım ve uygulamaların güvenliğini sağlamak için yapay zekâ teknikleri ile oluşturulan iki farklı hibrit yöntem geliştirilmiştir. Bu yöntemler, siber saldırıları otomatik olarak tespit etmek ve saldırganların ilerde yapacak saldırıları tahmin edip bir daha saldırı başlatmamasını sağlamak için geliştirilmiştir.

1.2. TEZİN AMACI VE KAPSAMI

Bu tez, geleneksel saldırı tespit sistemlerinin sürekli değişen siber saldırılar ve gelişen bilgisayar korsanlığı teknikleri ile saldırıları önlemede yetersiz kaldığı

durumlarda otomatik siber saldırı tespit sistemi geliştirmek için yapay zekâ yöntemleri kullanılarak graf ve derin öğrenme ile hibrit olmak üzere iki farklı sistem geliştirmeyi amaçlamaktadır. Bu nedenle, akıllı ve karmaşık siber saldırıları tespit ederek gelecekteki saldırıları tahmin etmek için otomatik bir saldırı tespit sistemi önerilmiştir.

Bu tezde iki farklı hibrit yöntem önerilmiştir. GLSTM yönteminde, siber saldırıların otomatik tespiti nedeniyle siber saldırılardan kaynaklanan anormallikleri tespit etmek için bir graf ve derin öğrenme modeli kullanılarak bir sistem önerilmiştir. Bu model graf ve derin öğrenme algoritmasından Uzun Kısa Süreli Bellek (Long Short-Term Memory, LSTM) algoritmasını kullandığı için bu yöntem graf tabanlı LSTM (GLSTM) şeklinde adlandırılmıştır. Bu yöntem, farklı log verilerini tek bir formata dönüştürerek grafa dönüştüren ve derin öğrenme yöntemini kullanarak anormallikleri tespit eden bir model önermektedir. Önerilen bu modeli eğitmek için, yarı denetimli ve sezgisel bir yaklaşım olan Node2Vec algoritması ile farklı türdeki log verileri grafa dönüştürülmüştür. Bu vektör graf verileri LSTM algoritmasına girdi olarak verilmiş ve deneysel testler gerçekleştirilmiştir. Bu deneysel testler, anormallikleri yüksek bir doğrulukla saptadıklarını göstermiştir. Bu çalışmanın ikinci yöntemi, veri güvenliğini sağlamayı ve ağ hizmetlerinin aksamasına neden olan Dağıtılmış Hizmet Reddi Saldırıları (Distributed Denial of Service Attack, DDoS) saldırı sorununu tespit etmeyi amaçlamaktadır. DDoS saldırıları önlenebilirse ciddi maliyet problemlerini ortadan kaldıracaktır. Bu amaçla yeni, hızlı ve doğruluğu yüksek bir model önerisi sunmaktır. Log kayıtlarının her satırı QR kod görüntüsü ile temsil edildikten sonra MobileNetV2 ve ShuffleNet Evrişimli Sinir Ağı (Convolutional Neural Network, CNN) mimarilerinin üstün yönlerinden yararlanılarak Harris Şahinleri Optimizasyonu (Harris Hawks Optimization, HHO) optimizasyon algoritması kullanılarak hibrit bir model önerisi sunulmuştur. Bu yöntem log verilerinin QR kodları oluşturup saldırıları tespit ettiği için bu yöntem de CNN tabanlı QR kodlu Loglar (CNNQRLog) olarak adlandırılmıştır.

Bu tez, bilgisayar ağlarında veri güvenliğini sağlamak için siber saldırıların ve güvenlik tehditlerinin değerlendirilmesine ve yeni yöntemlerin geliştirilmesine olanak sağlayan hibrit yöntemler (GLSTM ve CNNQRLog) geliştirmeyi

amaçlamaktadır. Bu tezin ana hedefleri, siber saldırı ve günlük veri saldırı tespit sistemlerinin performansını ve doğruluğunu artırmak için hibrit bir modeller geliştirmektir. Bu hibrit modeller, graf, LSTM, MobileNetV2 ve ShuffleNet algoritmalarını kullanarak geliştirilmiştir. Bu çalışma, siber güvenliği sağlamanın en önemli adımı olarak, kötü niyetli siber saldırganlardan önce güvenlik açıklarını bulmayı ve önlem almayı amaçlamaktadır.

1.3. TEZİN BİLİME KATKISI

Bu tez çalışmasında ayrık olay tabanlı siber saldırıların hacimli log kayıtlarını kullanarak graf ve derin öğrenme modelleri ile birçok farklı saldırıları tespit eden sistemi geliştirmektir. Bu çalışmayla saldırı tespit sistemlerinde graf kullanılarak anomali tespiti yapılması ve ShuffleNet ile MobileNetV2 CNN mimarinin kullanılarak sistem tasarlanmasıyla yüksek performansta saldırıları tespit edilmesi yönüyle özgün ve yeni bir çalışmadır.

Geniş alanlarda kullanılmasına rağmen ağ sistemlerinde kullanılan teknolojilerin güvenlik altyapısı tam koruma sağlayacak şekilde oluşturulmamıştır[16]. Siber saldırılarda siber saldırıları tespit etmek üç nedenden dolayı zor bir iştir. İlk olarak, çok sayıda log kaydı nedeniyle düzenli manuel ifadeler oluşturmak kolay değildir. İkincisi, yazılımın boyutu ve karmaşıklığından kaynaklanan olay şablonlarının çeşitliliğidir. Üçüncüsü, sık yazılım güncellemeleri nedeniyle, log bildirimleri sık güncellemeler olarak listelenmesidir. Bu tezde, bu eksikliklerin üstesinden gelmek için grafikler ve derin öğrenme ile otomatik bir algılama sistemi geliştirilmiştir. Bu çalışmada yapay zekâ yaklaşımı kullanılarak esnek ve ölçeklenebilir mimaride çalışan yeni bir otomatik siber saldırı tespit sistemi geliştirilmiştir.

Ayrıca bu tez kapsamında optimizasyon algoritması kullanılarak sistemin performans ve doğruluğu artırılarak bilime katkıda bulunulmuştur. Önerilen modelin yenilikçi yönleri ve katkıları aşağıdaki gibidir:

- Her bir ağ hareketi QR kod görüntüsü ile temsil edilmiştir,

- Önerilen model MobileNetV2 modeline göre %5.49, ShuffleNet modeline göre %7.07 iyileştirme yapılmıştır,
- HHO algoritması özellik seçim amacı ile başarı bir şekilde uygulanmıştır,
- DDOS saldırılarını %95.71 doğruluk oranı ile sınıflandıran yeni bir hibrit model önerisi sunulmuştur.

1.4. TEZ PLANI

Bölüm 1, 'Giriş' bölümü olarak tanımlanmıştır. Bu bölümde problemin tanımı, yapılan tez çalışmasının amacı, geliştirilen otomatik siber saldırı algılama sistemi, kullanılan graf ve derin öğrenme modelleri, bilime katkısı ve tez planlaması hakkında bilgi verilmektedir.

Bölüm 2'de siber güvenlik, bilgi teknolojisi güvenliği ve siber saldırılar ile ilgili literatür araştırması yapılmış ve yapılan çalışmalardan kapsamlı bir şekilde bahsedilmiştir.

Bölüm 3'te siber güvenlik, graf ve derin öğrenme modelleme süreçleri ana ve alt başlıklar altında değerlendirilmektedir. Siber güvenlik ve siber saldırılar hakkında bilgiler verilerek otomatik saldırı tespit sistemi anlatılmaktadır. Önerilen hibrit modellerde kullanılan grafik algoritması Node2Vec, derin öğrenme algoritmaları LSTM, MobileNet ve ShuffleNet, optimizasyon algoritması HHO ve sınıflandırma algoritmalarının yapısından bahsedilmiştir. Ayrıca otomatik siber saldırı tespit sistemi geliştirmede bu algoritmaların avantajları hakkında bilgi verilmektedir.

Bölüm 4'te otomatik siber saldırı algılama sistemi oluşturmak için materyal ve metotlar ele alınmıştır. Bu bölümde deneysel testlerde kullanılan veri setlerinin yapısından, önerilen modellerde kullanım aşamasından önceki aşama olan ön işlemlerden bahsedilmiştir. Ayrıca önerilen hibrit yöntemler detaylı bir şekilde ele alınmıştır. Bu modellerin mimarileri, kullanılabilirlikleri ve ölçeklenebilirlikleri incelenmiştir. Bununla birlikte önerilen hibrit yöntemlerin performans metrikleri hakkında bilgi verilmiştir.

Bölüm 5’te siber saldırıların tespit edilmesi önerilen yöntemlerin deney sonuçlarına yer verilmektedir. Bu bölümde önerilen hibrit her iki yöntemin deneysel sonuçları detaylandırılmıştır. Karşılaştırma ve sonuç grafikleri gösterilip yorumlanmıştır.

Bölüm 6'da bulgular ve tartışmalara yer verilmektedir. Bu bölüm, önerilen her iki hibrit yöntemin deneysel sonuçlarını özetleyerek literatürde daha önce yapılmış benzer çalışmalarla karşılaştırılmıştır. Ayrıca her iki hibrit yöntem için daha önce yapılmış benzer çalışmaları tablolarda gösterilmiştir. Önerilen modellerin avantaj ve dezavantajlarından bahsedilmiştir.

Bölüm 7'de geliştirilen uygulamanın test sonuçları özetlenmiş ve bu yöntemlerin

BÖLÜM 2

LİTERATÜR ARAŞTIRMASI

Siber saldırılarla ilgili literatür incelendiğinde log anomali için yapılan çalışmalarda dört farklı yöntem kullanılmıştır. Bu yöntemler şu şekilde sıralanmaktadır;

- Şablon çıkarma,
- Dokümanları yönetmek,
- İzleme sistemi,
- Öğrenmeye dayalı anomali tespiti

Bu yöntemlerden şablon çıkarma ile ilgili birden çok çalışma yapılmıştır. Bu yöntemin temel amacı log dosyasında geçen kelimelerin frekansını çıkarıp, anormal olan kelimeleri tespit edip, anomali tespiti yapmaktır. Kelimelerin şablonda kalması için belli bir eşik değerini geçmesi gerekmektedir. Bu yöntemle yapılan çalışmaların başında Yinelemeli Bölümleme Log Madenciliği (Iterative Partitioning Log Mining, IPLoM) gelmektedir[17]. Log kayıtlarının tüm satırlarını eşit uzunlukta varsayarak yenilemeli olarak bölmektedir. Daha sonra bu bölümlerden boşluklar atılıp, bir eşik değer seçilmektedir. Böylece giriş verisi her adımdaki bölümlerden gelen aynı tür log verisinden seçilmektedir. Bölümleme dört farklı şekilde yapılmıştır. Birincisi belirtilen formata göre, ikincisi belirtilen konuma göre, üçüncüsü eşleştirme tipine göre ve son olarak da satır formatlarına göre bölümleme yapılmıştır.

Şablon çıkarma tekniğinin kullanıldığı başka bir çalışmada, doğal dil işleme kullanılarak belli bir sıra ve zaman damgasına göre tutulan log verilerinden derin öğrenme yaklaşımı kullanılıp, log anomali tespiti yapılmıştır. Düzenin bozulması veya belirli bir akışa göre düzenli tutulan log kayıtlarından farklı bir durum tespit edilmesi durumunda bir anormallik tespit edilmektedir. Log kayıtlarında anlamlı kelimeler alınır ve şablonlar düzenli tutulmaktadır. Bu şablon bir vektöre

aktarılmıştır ve bu yönteme tamplate2vector[18] adı verilmiştir. Derin öğrenme algoritmalarından biri olan LSTM kullanılmış ve HDFS ile BGL veri setleri kullanılarak bu yöntem test edilmiştir.

Word2Vec, dokümanları yönetmek ile yapılan çalışmaların ön saflarında yer almaktadır. Bu yöntemle kelime grupları oluşturulmaktadır. Veri boyutuna göre kelimeler, cümleler ve paragraflar gibi farklı gruplara ayrılmaktadır[19]. Bu yöntemle yapılan bir başka çalışmada, Thunderbird logları ve sistem log kayıtlarının doğal dil işleme tekniği kullanılarak anomali tespiti yapılmıştır. Deneylerde Word2vec ve TF-IDF öznitelik çıkarma algoritmaları kullanılmış ve sınıflandırma için derin öğrenme algoritması LSTM ile analiz tamamlanmıştır[20].

Anomali tespiti yapan bir diğer yöntem sistem izleme sistemidir. Bu yöntemle birbirinden farklı sistemlerden gelen log kayıtları izlenebilmektedir. Bu yöntemin geliştirilmesi konusunda Google tarafından geliştirilen Dapper aracı gelmektedir. Bu araç genellikle karmaşık ve büyük ölçekli dağıtılmış sistemlerde test edilmiş ve yüksek başarı elde etmiştir[21].

Literatürde öğrenmeyi temel alan birçok çalışma bulunmaktadır. Bu çalışmalar, çeşitli makine öğrenme tekniklerinin kullanımına dayanmaktadır. DeepLog, log kayıtlarından anormallikleri tespit eden önde gelen çalışmalardan biridir. Önerilen yaklaşım iki önemli bölümden oluşmaktadır. İlki, log anahtarını tanımlanmaktadır. Diğeri, anormallikler dahil tüm parametreleri tanımlayarak bir iş akışı sağlamaktadır. Log anahtarına bağlı olarak, anomali parametreleri bir vektöre aktarılmaktadır. Ardından, log anahtarına karşılık gelen anormallikleri tespit etmek için yapay sinir ağlarından LSTM algoritması kullanılmıştır. Böylece LSTM log anahtarına benzer anormallikleri tespit etmektedir. Algoritma ayrıca doğruluğu artırmak için yanlış pozitifler üzerinde manuel geri bildirim kullanmaktadır[22].

Başka bir çalışmada ise log kayıtlarından anomali tespiti için derin öğrenme tekniklerinden CNN algoritmasını kullanmışlardır. Bu çalışma, log kayıtlarındaki anahtar kelimeleri bularak, bu anahtar kelimelerdeki anormallikleri tespit etmektedir.

Belirlenen anahtar kelimeler sayısallaştırma, normalleştirme yapılmış ve 29x128 vektörüne aktarılmıştır. Bu yönteme logkey2vector adını verilmiştir[23].

Derin öğrenme üzerine yapılan başka bir çalışmada ise BGL(BlueGene/L), Thunderbird, Openstack ve IMDB (Internet Movie Database) veri setleri kullanılarak farklı modeller oluşturulmuş ve bu modeller arasında karşılaştırmalar yapılmıştır. Önerilen bu model IMDB veri seti kullanılarak diğer metin sınıflandırma problemleri genelleştirilmesini kanıtlamak için kullanılmıştır. Mimaride, orijinal verileri daha iyi anlamak için pozitif ve negatif etiketli veriler iki farklı otomatik kodlayıcıya aktarılmaktadır. Bu çıktı, derin öğrenme algoritmaları LSTM, BLSTM ve GRU için girdi olarak kullanılmaktadır[24,25].

Başka bir çalışmada sık desen madenciliği yöntemi ile tüm veri kümesi üzerinden birkaç geçiş yaparak orijinal veri uzayının alt kümeleri tespit etmeyi amaçlamaktadır. Üç adımdan oluşmaktadır. Birincisi, verilerin bir özetini alarak, tüm veri kümesinden geçiş yapmaktadır. İkincisi, küme adayları oluşturmak için başka bir geçiş yapmaktadır. Üçüncüsü, adaylardan uygun kümeleri seçmektedir[26].

Log kayıtlarından anomali tespiti için birden çok çalışmada graf yapısı kullanılmıştır. İşletim sistemine yetkisiz erişimi engellemek için graf yapısını kullanarak kimlik doğrulama loglarından anomali tespiti yapılmıştır. Adli bilişim için kullanılması amaçlanmış ve yeni bir yöntem olan graf kümeleme yöntemi geliştirilmiştir[27].

Günümüzde sürekli değişen siber saldırılar ve gelişen bilgisayar korsanlığı teknikleri ile yapılan saldırıları engellemek için birden çok çalışma yapılmıştır[7–10]. Hizmet durdurma saldırıları olan DoS/DDoS saldırıları özellikle sürekli çalışan sunucular, ağlar ve web siteleri gibi sistemlerde yapılan saldırıların en başında gelmektedir[28]. Gelişen teknolojilerle beraber birden fazla çeşitte DoS/DDoS saldırıları bulunmaktadır[29]. Birden çok sistemi ilgilendiren DoS/DDoS saldırısını gerçek zamanlı olarak tespit etmek için makine öğrenmesi tekniklerinden teknikleri kullanılmıştır[30]. Autoencoder kullanılarak hizmet engelleme saldırılarını engellemek için model önerilmiştir[31]. Nesnelerin internetinde hizmeti durduran saldırıları engellemek için derin öğrenme yöntemlerinden LSTM kullanılmıştır[32].

Kaba kuvvet saldırısı olarak bilinen FTP-BruteForce saldırı tespiti için yapılan çalışmalarda gerçek zamanlı saldırı tespiti[33], Naive Bayes özellik seçimi ile saldırı tespit sistemi geliştirilmesi[34] gibi birden çok sistem önerilmiştir.

Büyük log verilerinden anomali tespiti saldırı tespit sistemlerinde oldukça zor bir yöntemdir. Çünkü her yazılım veya donanım aygıtı, verileri gruplar halinde tutmak yerine düz metin olarak kendi içinde loglara kaydetmektedir. Büyük miktarda log verisi ve düzensizlik nedeniyle, anormallikleri hızlı ve etkili bir şekilde tespit etmek zordur. Yazılım hareketlerinin bir dosyayı açma girişimleri gibi minimum bellek koşulları kullanarak metin ve sayısal verilerin toplama işlemine log kaydı denilmektedir. Genellikle logları kaydeden uygulamalar logların neyin, nasıl ve nerede depolandığına odaklanmaktadır[35].

Geniş internet ağlarında, olay ve sistem tabanlı logların birden çok sistem, yazılım ve donanım kombinasyonu ile analiz edilmesi kritik öneme sahiptir. Siber saldırılar esnasında veya gerçekleştikten sonra saldırganların yaptığı saldırılara ilişkin log kayıtları sistem güvenliğinin yapıldığı cihaz ve yazılımlarda izler bırakmaktadır. Siber saldırıların tespitinde log verilerinin analizi yolu ile bu izlerin oluşturduğu anomalilerin tespit edilmesi önemli bir yaklaşımdır.

Graf yapısı kullanılarak yapılan bir başka çalışmada zaman serileri ve öldürme zinciri mekanizmaları ile log verilerinden anormallikler tespit edilmiştir. Geliştirilen bu yöntem ile saldırı profilleri oluşturulmuş, bilgisayar ağına yapılan saldırı logları simule edilmiştir[36].

Bulut bilişimde kullanılan yazılımların hatasını tespit etmek için graf yapısı kullanılmıştır. Yapılan yazılım hatalarını tespit etmek için log anomali algılama yöntemi geliştirilmiştir. Bu yöntemle, log kayıtları arasında karmaşık ilişki grafa dönüştürülerek, her bir log için önem puanı verilmiştir[37].

Modern Uç/Bulut teknolojilerine bağlanan fiziksel Nesnelerin İnterneti (Internet of Things, IoT) Uç cihazların doğası gereği kısıtlı olması, büyük hacimli saha eğitim verilerinin olması, yeterli depolanma alanlarına ve işlem yeteneklerine sahip

olmamalarından dolayı derin trafik denetimi ve sınıflandırma gibi ağır hesaplama işlemlerinin bulut tabanlı mimarilerden yararlanarak Ekstrem Öğrenme Makineleri (Extreme Learning Machines, ELM) modellerine dayalı trafik sınıflandırmasını modelini önermişlerdir[38]

Nesnelerin interneti (IoT) ağında dağıtılmış hizmet reddi (DDoS) saldırılarının tespiti için çok amaçlı optimizasyon tabanlı bir özellik seçimi Özellik Seçimi (Features Selection, FS) yöntemi kullanılmıştır. Siber saldırıların tespit edilmesinde performansı ve doğruluğu artırmak için uygun özniteliklerin seçilmesi gerekmektedir. Uygun özniteliklerin seçmek, verilerin boyutluluğunu azaltmak ve Saldırı Tespit Sistemi'nin (Intrusion Detection System, IDS) performansını iyileştirmek için uygun FS yöntemi kullanılmıştır [39]

Birden çok sistemi içeren DoS/DDoS saldırılarını gerçek zamanlı olarak tespit etmek için makine öğrenimi tekniklerinden teknikler kullanılmıştır[40]. Bir otomatik kodlayıcı kullanarak hizmet reddi saldırılarını önlemek için bir model önerilmiştir[41]. Derin öğrenme yöntemlerinden biri olan LSTM, Nesnelerin İnterneti üzerinde hizmet durdurma saldırılarını önlemek için kullanılmıştır[42]. Brute Force saldırısı olarak bilinen FTP-BruteForce saldırı tespiti için yapılan çalışmalarda, gerçek zamanlı saldırı tespiti[33] ve Naive Bayes özellik seçimi ile saldırı tespit sisteminin geliştirilmesi gibi birden fazla sistem önerilmiştir[34].

Son zamanlarda, siber saldırılar hem karmaşık hem de akıllı oldukları için tespit edilmesi ve önlenmesi daha zor hale gelmektedir[43]. Bu zorlukların üstesinden gelmek için çoklu saldırı tespit sistemleri geliştirilmiştir. İzinsiz giriş tespit sistemleri imza tabanlı ve anomali tespit olmak üzere saldırıları iki şekilde tespit etmektedir[44]. Bu iki yöntemde, saldırıları tespit etmek için makine öğrenme teknikleri kullanılarak daha önce yapılmış saldırıların izleri takip edilmesiyle bu izler gelecekteki saldırıları tespit etmek ve önlemek için kullanılmıştır[45]. Makine öğrenmesi yöntemleri ile saldırı tespit sistemlerinde önceki saldırılara dayalı gelecek saldırıları tespit etmek için görüntü işleme teknikleri kullanılarak yüksek doğruluk ve performans elde edildiği gözlemlenmiştir[46]. Kullanıcıları bilgisayar ağlarına yönelik siber saldırılardan korumak ve güvenliklerini artırmak amacıyla her kullanıcı

için QR kodlar oluşturulmuş ve kullanıcıların bu QR kodlar ile sisteme giriş yapması sağlanmıştır[47].

Son zamanlarda derin öğrenme yöntemlerinden biri olan CNN modelleri kullanılarak görüntülerdeki nesnelerin algılanmasında yüksek başarı elde edilmektedir[48]. CNN modelleri birçok farklı görüntü işleme tekniğinde başarılı olsa da kurulacak mimarilerde hiper parametrelerin doğru optimizasyonu hala zordur[49]. Modelin başarısını etkileyen bir diğer faktör de büyük miktarda veriye ihtiyaç duymasıdır[50]. Yakın zamanda geliştirilen CNN modelleri ile MobileNetV2[51]ve ShuffleNet[52] derin öğrenme mimarileri eğitilerek yüksek başarı oranları daha hızlı ve kesin olarak elde edilebilmektedir.

Modern ağlarda CNN modellerinin kullanımı, birçok mobil ve video uygulamasının yüksek hesaplamalı kaynak kullanımını ve yüksek işlem maliyetlerini gerektirmektedir [53]. Hiper parametrelerin kullanılması şunları mümkün kılmaktadır ve bu performans ölçümlerini azaltmak için MobileNetV2'yi kullanmaktadır[51]. MobileNetV2, görüntü işleme teknikleri ile birçok alanda kullanılmış ve yüksek başarı elde etmiştir[54–57]. Siber saldırıları tespit edebilmek için saldırılar gri ölçekli görüntülere dönüştürülerek MobilNetV2[51] kullanılarak sınıflandırılmıştır. UNSW-NB15 ve KDD-99/DARPA veri kümeleri görüntülere dönüştürülmüş, MobilNetV2 ile görüntü işleme teknikleri kullanılmış ve izinsiz giriş önleme sistemi önerilmiştir[58]. Kötü amaçlı yazılım tespiti, Android sistemlerde kötü amaçlı yazılım tespiti için XML ve DEX dosyalarındaki kötü amaçlı yazılımın görüntülere dönüştürülmesiyle MobilNetV2 CNN modeli kullanılarak gerçekleştirilmiştir[59].

ShuffleNet CNN modeli, görüntü işleme tekniklerinde sınırlı kaynakları en iyi ve verimli şekilde kullanmak için geliştirilmiştir[60]. ShuffleNetCNN modeli, nesnelerin internetinde otomatik modülasyon sınıflandırması yapmak için bilişsel radyo dalgalarından siber sistemlerdeki bilinmeyen sinyalleri tespit etmek için önerilmiştir [61]. Yüksek voltajlı elektrik sistemlerinde siber saldırıları önlemek için ShuffleNet kullanan bir CNN modeli önerilmiştir[62]. Siber saldırılar arasında yer alan kaba kuvvet saldırıları, genellikle son kullanıcıların bilgileri tahmin ederek

sisteme girmesini sağlamaktadır[63]. Web sitelerinde son kullanıcıları ve makine botlarını ayırt etmek için kullanılan güvenlik mekanizması olan Bilgisayar ve İnsan Apart'ı (Computers and Humans Apart'ı, CAPTCHA) derin öğrenme yöntemleriyle bypass etmek mümkündür. ShuffleNet CNN modeli oluşturularak bunu önlemek için makine botları ile son kullanıcıları birbirinden ayıran bir sistem önerilmiştir[64].

BÖLÜM 3

SİBER GÜVENLİK, GRAF VE DERİN ÖĞRENME MODELLERİ

3.1. SİBER GÜVENLİK UNSURLARI

Siber güvenlik, bilgisayar ağlarını, ekipmanı ve değerli verileri yetkisiz erişime veya yasa dışı kullanıma karşı korumak ve bilgi gizliliğini, bütünlüğünü ve kullanılabilirliğini sağlamaktır[65]. Siber alanda faaliyet gösteren bilgisayar ağları, sunucular, mobil cihazlar, elektronik sistemler veya veri ağları ile bu sistemlerde çalışan yazılımların siber tehditlere karşı korunması için alınan önlemleri kapsamaktadır. Bu alan, bilgisayar sistemlerine, internete ve çeşitli cihazlar gibi kablosuz ağ standartlarına giderek artan güven ve akıllı telefonlar, televizyonlar dahil nesnelerin internetini oluşturan akıllı cihazların büyümesi nedeniyle giderek daha önemli hale gelmektedir. Siber güvenlik hem politik kullanım hem de teknolojik karmaşıklık nedeniyle günümüzdeki en büyük zorluklardan biridir[66].

Bilgi sistemlerinin bütünlüğünün sağlanması, sistemlerden gerektiği gibi faydalanılması; sistemin istenilen koşullarda çalışmasına bağlıdır. Siber güvenlik, gizliliği sağlamak için sistemi oluşturan parçaların, yazılımların, donanımların ve sistemlerin savunulmasına katkıda bulunulmaktadır. Bilgisayar ağları dış erişime açık olduğundan bu ağlarda güvenlik riskleri ortaya çıkmaktadır[67].

3.1.1. Uygulama Güvenliği

Uygulama güvenliği, yazılımlardaki güvenlik açıklarının bulunup düzeltilmesi ve uygulamaların güvenliğinin artırılması sürecinden oluşmaktadır. Bu, uygulamaları çok daha güvenli hale getirmek ve onları saldırılardan korumak için yapılmaktadır. Uygulama güvenliği, yazılım geliştirme yaşam döngüsünü geliştirmek veya iyileştirmek için çok önemlidir. Özellikle, yeni güvenlik açıklarını tespit etmek ve

tehditleri belirlemek için dinamik bir yaklaşım gerektirmektedir. Kötü niyetli bilgisayar korsanları yeni saldırı yöntemleri geliştirirken, teknolojik ortam her zaman güvende kalabilmek için uygulama güvenliğinin önemini artırmaktadır. Uygulama güvenliği, güvenlik açıklarını belirlemek ve saldırıları önlemek için her oluşturma, test etme ve yayınlama aşamasında en iyi uygulamaları takip ederek ve farklı araçlar ve yöntemler kullanarak sağlanmaktadır[68].

3.1.2. Bilgi Güvenliği

Uluslararası standart, bilgi güvenliğini bilgi gizliliğini, bütünlüğünü ve kullanılabilirliğini korumak olarak tanımlanmaktadır. Bilgi, basılı veya kağıt, elektronik olarak saklanan, posta veya elektronik yollarla iletilen ve filmlerde, konuşmalarda ve benzerlerinde görüntülenen birçok biçimde olabilmektedir[69].

Bilgi güvenliği, bilgiyi ve onu kullanan, depolayan ve ileten sistemler ve donanım dahil olmak üzere kritik öğelerini korumaktadır[70]. Ayrıca, kuruluşlara değer katan bilgilerin birkaç kritik özelliğini de tanımlanmıştır. Bu özellikler, ISO/IEC 27002 (2005) tanımında belirtilen bilgilerin gizliliğini, bütünlüğünü ve kullanılabilirliğini içerir ancak bunlarla sınırlı değildir. Bilgi güvenliğinde CIA üçgeni olarak da bilinen bilginin gizliliğini, bütünlüğünü ve kullanılabilirliğini sağlamak, geleneksel olarak endüstri standardı olmuştur. Bilginin bu üç yönünün güvenliği her zaman olduğu gibi bugün de eşit derecede önemlidir, ancak CIA üçgen modeli artık bilgisayar endüstrisinin sürekli değişen manzarasını yeterince ele almamaktadır[70].

3.1.3. Ağ Güvenliği

Ağ güvenliği, ağ trafiği dahil olmak üzere teknolojik unsurları korurken izinsiz giriş yapıldığında ağ saldırılarını izlemek, durdurmak ve bunlara yanıt vermek için tasarlanmış cihazları, yazılımları, taktikleri ve güvenlik politikalarını tanımlayan bir terimdir. Ağ güvenliği, bilgili güvenlik analistlerinden, avcılardan, olay müdahale ekiplerinden vb. yararlanabilmektedir. Donanım ve yazılım teknolojileri ve gelen kaynaklar dahil olmak üzere ağı hedefleyen tüm tehditlere yanıt vermek üzere tasarlanmıştır [71].

Bilgisayar ağları, bilgilerin hızlı bir şekilde değiş tokuş edildiği ve kolayca erişildiği bilgi havuzlarıdır. Bu bilgi havuzu ortamını oluşturan ve kritik verileri içeren ağ güvenliğinin önemi her geçen gün artmaktadır. Dev bir bilgisayar ağı ve bunun sonucunda ortaya çıkan internet herkes için vazgeçilmez bilgi kaynaklarıdır. Her meslekte bilgisayar kullanımı, kişisel bilgisayarın her evde kullanılmaya başlanması ve internete erişimin çok kolay ve ucuz hale gelmesi, istisnasız her dijital cihazın bir bilgisayar ağına bağlanmasını zorunlu hale getirmektir[72].

3.2. SİBER SALDIRI TÜRLERİ

3.2.1. Hizmet Durdurma (DoS/DDoS) Saldırıları

Genel olarak, bir DDoS saldırısı, tek bir sisteme odaklanan, botlar veya zombiler olarak adlandırılan, güvenliği ihlal edilmiş birden çok dijital cihaz tarafından hedeflenmektedir. Motivasyonu, hizmeti yanlışlıkla bloke etmek veya durdurmak ve hizmetin kullanılamamasına neden olmak için hedef sistemi veya ağ kaynaklarını tüketmektir. DDoS saldırısı flood saldırısı, amplifikasyon saldırısı, çekirdek eritme saldırısı, yer saldırısı, TCP SYN saldırısı, CGI istek saldırısı ve kimlik doğrulama sunucusu saldırısı olmak üzere yedi ana sınıfa ayrılmaktadır[73].

3.2.2. SQL Enjeksiyon (SQL Injection) Saldırıları

SQL enjeksiyonu, bir saldırganın web uygulamasına kötü niyetli bir SQL sorgusunu giriş parametrelerine ekleyerek eklediği bir saldırı çeşididir. Geliştirici, kullanıcı tarafı değişkenleriyle birleştirilmiş dinamik sorgular kullandığında bir SQL enjeksiyonu gerçekleştirmektedir. Güçlü giriş doğrulaması olmadan, kötü niyetli SQL sorgusu web uygulamasına eklenir ve değişkenler yerine yasal sorguyla birleştirilmektedir. Sorguyu çalıştırmak için veritabanı yönetim sistemine gönderilmektedir. Sonuç olarak, kötü amaçlı sorgu yürütülmüş ve veritabanından istenilen veriler çekilmiş olmaktadır[74].

3.2.3. Kaba Kuvvet (Brute Force) Saldırıları

Kaba kuvvet saldırıları, bilgisayar ağlarında en yaygın saldırı türlerinden biri olarak görülmektedir. SSH protokolüne kaba kuvvet saldırısında, saldırgan bir kullanıcının hesabına giriş yapmaya çalışır ve kurbanın makinesinde oturum açma şifresini ortaya çıkarmak için farklı şifreler deneyerek doğru şifreyi bulmaya çalışmaktadır. Saldırganlar genellikle kurbanın makinesine karşı farklı parola kombinasyonları oluşturan otomatik yazılımlar kullanmaktadır. Kullanıcılar tarafından seçilen parolalar, kullanıcının bilgisinin sınırlı bir alanından seçildikleri için doğası gereği zayıf olmaktadırlar. Ayrıca, şifreleri ezberleme/geri çağırma ihtiyacı, şifreleri zayıflatmaya yardımcı olmaktadır. Kaba kuvvet saldırılarının popüler olmasının bir diğer önemli nedeni de varsayılan olarak otomatik olarak üretilen şifrelerin sürekli olarak kullanılması ve şifre olarak kullanıcı adının kullanılmasından kaynaklanmaktadır[75].

3.2.4. Siteler Arası Komut Çalıştırma (Cross-Site Scripting(XSS)) Saldırıları

Siteler Arası Komut Dosyası Çalıştırma (XSS), bir saldırganın çerezler, şifreler, kredi kartı numaraları vb. istemci tarafı web tarayıcısında bulunur, ancak yetenekleri web sunucusu tarafında kullanılmaktadır. Bir saldırgan, web uygulamalarındaki XSS güvenlik açıklarından yararlanmak için web uygulamasında kötü amaçlı bir JavaScript kodu oluşturarak enjekte etmektedir. Bu komut dosyası, web sitesinin zararsız bir bileşeni gibi görünecek şekilde enjekte edilir ve son olarak bu komut dosyası, web sitesinin güven alanı içinde yürütülerek çalıştırılmaktadır[76].

3.3. SALDIRI ALGILAMA YÖNTEMLERİ

3.3.1. İmza Tabanlı Yöntemler

İmza tabanlı algılama sisteminin ağ güvenliğinde kullanımı kolay olan sistemlerdendir. Siber saldırıların benzer imzalarını, davranışlarını veya aktivitelerini bulmaya çalışan açık bir yazılım platformudur. Bu algılama, bilinmeyen saldırıları bulamaz, yalnızca bilinen saldırıları bulabilmektedir. Bunun için her zaman yeni

imza anormallikleri hakkında bir güncellemesine gerek duymaktadır. Sırasıyla tüm saldırıları tespit etmek için mükemmel değildir. Bu sistemler sıfırıncı gün saldırılarını tespit etmekte yetersiz kalmıştır. Yeni kötü amaçlı yazılım algılandığında, her saldırı için özelleştirilmiş bir imza tasarlanmalıdır veya sisteme bağlı olarak diğerleriyle birleştirilmelidir[77].

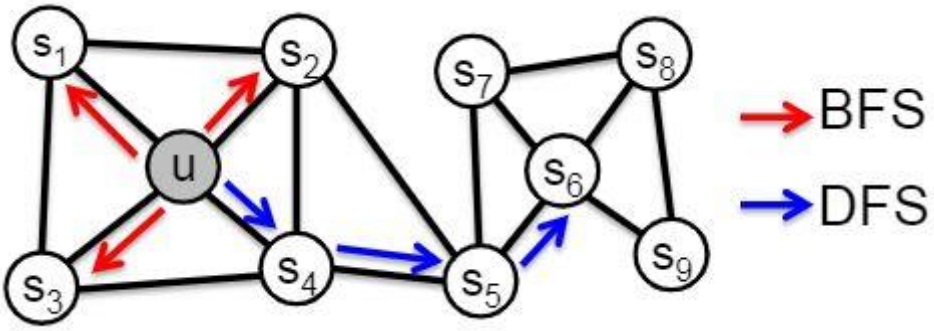
3.3.2. Anomali Tabanlı Yöntemler

Anormallik, normal davranıştan herhangi bir sapma olarak tanımlanmaktadır. Davranış tabanlı algılama olarak da adlandırılan anormallik tabanlı algılama, önemli sapmaları belirlemek için normal etkinlikleri gözlemlenen olaylarla karşılaştırmaktadır. Anomali tabanlı algılama üç genel modülden oluşmaktadır; (1) Parametreleştirme: ağ bağlantıları, ana bilgisayar ve uygulamalar gibi araştırılacak olanın farklı niteliklerinden ve özelliklerinden oluşan bir profilde gözlemlenen davranışı temsil etmektedir. (2) Eğitim: normal ve anormal davranışları birbirinden ayıran bir sınıflandırma modeli oluşturmak için parametreleştirilmiş profillerin işlenmesi aşamasıdır. (3) Tespit: Oluşturulan sınıflandırma modelini yeni trafik anormalliklerini tespit etmek için kullanmaktadır[78].

Anormallik algılama adımlarını gerçekleştirmek için farklı teknikler kullanılabilmektedir. İstatistik tabanlı teknikler: Anormallik, standart sapmalar, ortalamalar, eşikler ve olasılıklar kullanılarak belirli bir davranıştan sapma derecesinin puanlanmasıyla tanımlamaktadır. İlk yaklaşımlar tek değişkenli modeller kullanırken, sonraki yaklaşımlar çok değişkenli ve zaman serileri kullanmaktadır. Bilgiye dayalı teknikler: Bunlar, normal ve anormal işlemler altında gözlemlenen parametreler hakkındaki geçmiş bilgilerin mevcudiyetine dayanmaktadır. Bilgi tabanlı teknikler, uzman sistemler, sonlu durum makineleri, tanımlama dilleri ve veri kümeleme kullanmaktadır. Makine öğrenimi ve derin öğrenme teknikleri: Öğrenme algoritmaları, insan müdahalesi olmadan deneyimlerden öğrenerek bir IDS'nin performansını iyileştirmek için çeşitli makine öğrenimi algoritmaları kullanmaktadır[79].

3.4. GRAF: NODE2VEC ALGORİTMASI

Graf algoritmasından olan Node2Vec[80] algoritması doğal dil işleme algoritması olan word2vec algoritmasına alternatif olarak geliştirilmiştir. Doğal dil işleme düşünülerek geliştirilmesine rağmen birden çok alanda bu algoritma kullanılmıştır. Bu algoritmanın yaklaşımı d-boyutlu bir özellik uzayında her bir düğümün ağda komşuluklarını en üst düzeyde tutulmasını sağlamak için olasılık kullanmaktadır. Düğümlerin ağ komşuluklarını elde etmek için rasgele yürüyüş yaklaşımı kullanılmıştır.

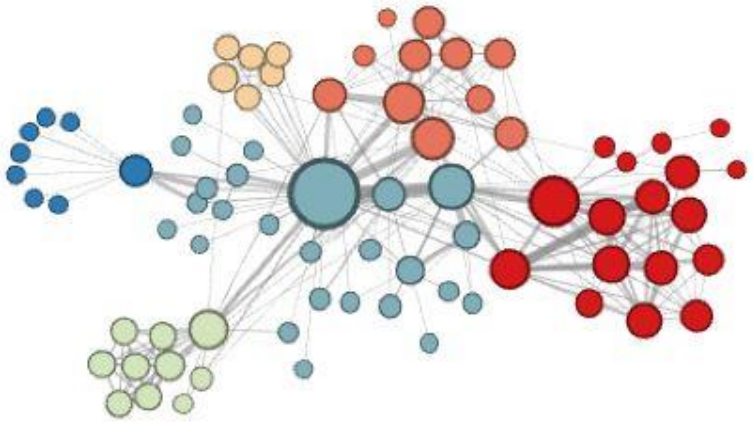


Şekil 3.1.Klasik arama Node2Vec graf algoritması[25].

Şekil 3.1'de graf üzerinde klasik arama algoritmaları gösterilmiştir. Bu algoritmaların biri Breadth-first Sampling (BFS) diğeri Depth-first Sampling (DFS) algoritmalarıdır. BFS yakın komşulukları DFS ise uzak komşulukları tespit edebildiği görülmektedir. Node2Vec bu iki yaklaşımı geliştirdiği esnek yapısı ile kullanmaktadır. Komşulukları bulmak için rasgele yürüyüş gerçekleştirerek olasılık kullanılmıştır. Ağırlıksız ve yönsüz ağlarda yarı denetimli çalışarak klasik arama yaklaşımları olan BFS ve DFS'den daha iyi sonuç elde edilmiştir[80].

Node2Vec algoritmasının yapısı diğer algoritmalara nazaran birçok farklılık göstermektedir. Bu algoritma dört parametre almaktadır. Bunlar p, q, rasgele yürüyüş ve yürüyüş uzunluğu parametreleridir. Bu parametreler değiştirilerek optimum sonuçlar elde edildiği için yarı denetim olarak çalışan bir algoritmadır. Bu parametrelerden p dönüş parametresidir. Daha önce ziyaret edilmiş düğümü

örnekleme olasılığını azaltır. Diğer parametre q ise giriş-çıkış parametresidir. Bu parametre ile daha önce ziyaret edilmemiş düğümler ziyaret edilmesini sağlar. Eğer q>1 ise rasgele yürüyüş daha ziyaret edilen düğüm etrafında gerçekleştirilmektedir. Bu yönüyle BFS algoritmasına benzemektedir. Eğer q<1 ise rasgele yürüyüş daha önce ziyaret edilmemiş düğümleri ziyaret etmektedir. Bu yönüyle de DFS algoritmasına benzemektedir. Şekil 3.2'te verilerin grafa aktarılmış şekli gösterilmiştir.



Şekil 3.2. Graf sonucunda verilerin gösterimi[25].

Verileri Node2Vec algoritmasından vektörel alabilmek için bir grafa aktarılması gerekmektedir. Bu çalışmada makine öğrenimi ve derin öğrenme yapılarını kolay kullanım sunduğundan StellarGraph kullanılmıştır[81]. Bu graf yapısının kullanılmasının başlıca sebepleri;

- Görselleştirme ve çeşitli makine öğrenmesi için kullanılabilmesi,
- Düğüm ve kenarlardan öznitelik çıkarabilmesi,
- Büyük veri üzerinde uygulanabilmesi,
- Düğümlerin sınıflandırılmasının yapılabilmesi,

gibi birçok işlemi kolay ve uygulanabilir bir şekilde yapabilmektedir. Bu graf yapısı kullanılarak derin öğrenme ve makine öğrenmesi ile ilgili birden çok çalışma yapılmıştır[82–84].

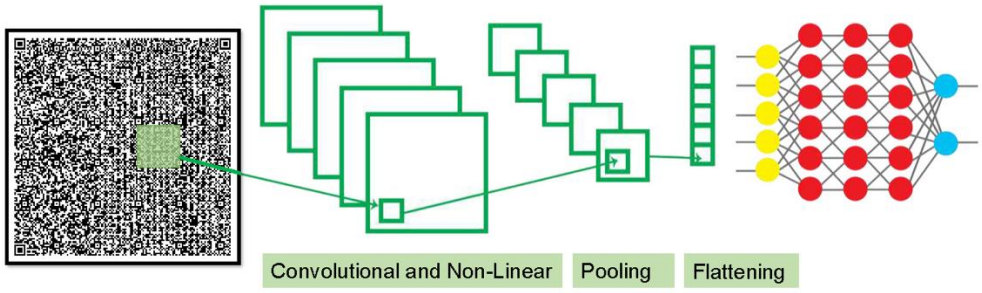
3.5. UZUN KISA-VADELİ BELLEK AĞLARIN YAPISI

LSTM'ler kendini tekrarlayan RNN'lerin bir üyesidir. RNN'ler ardışık verileri her seferinde bir öğeyi alarak kendini tekrarlayan modelleridir[85]. Markov modelleri ile karşılaştırıldığında durum uzay kümeleri artmasına rağmen uzun vadede bağımlılıktan dolayı daha iyi sonuç vermektedirler[86,87]. LSTM'ler ilk olarak [88] tarafından ortaya atıldı. LSTM'ler RNN'lerin dezavantajlarını ortadan kaldırmak için geliştirilmiştir. LSTM'ler RNN'ler gibi yenilemeli olarak çalışır bunlardan farkı kendi içinde gizli gösterim ile farklı hücreler üzerinde çalıştırmaktadır.

3.6. EVRİŞİMLİ SİNİR AĞLARIN YAPISI

Son zamanlarda görüntülerde nesne tespiti yapmak için derin öğrenme yöntemlerinde CNN modelleri kullanarak yüksek başarılar elde edilmiştir[89]. CNN modelleri birden çok farklı görüntü işleme tekniklerinde başarılı olsa da kurulacak olan mimarilerde hiper-parametrelerin doğru optimizasyonu hala zor bir süreçtir[90]. Modelin başarısını etkileyen bir diğer faktör de oldukça fazla veri miktarına ihtiyaç duymasıdır[91]. Son zamanlarda geliştirilen CNN modelleriyle hiper-parametre optimizasyonuna ihtiyaç duymayan MobileNetV2[51] ve ShuffleNet[52] derin öğrenme mimarilerinin eğitilmesiyle yüksek başarı oranlarına daha hızlı ve kesin bir şekilde ulaşılabilmektedir.

CNN modelleri, görüntülerdeki nesneleri algılamak için derin öğrenme yöntemlerinde kullanılmaktadır[92]. Şekil 3.3'te, CNN modelleri, birçok farklı görüntü işleme tekniğinde başarılı kullanımlarıyla ilgili olarak CNN modellerinin ana mimarisinin bir temsili olarak gösterilmektedir. Modelin başarısını etkileyen ana faktörler, özellikleri belirlemek için evrişimli katman, sistemin doğrusal olmaması için doğrusal olmayan katman, ağırlık sayısını azaltmak ve uyumu kontrol etmek için havuzlama katmanı, hazırlığı hazırlamak için düzleştirme katmanıdır. Klasik sinir ağı için veriler ve sınıflandırma kullanan standart sinir ağı için Tam Bağlantılı katman- bu beş katmandan oluşmaktadır.



Şekil 3.3. Geleneksel evrişimli sinir ağları mimarisi.

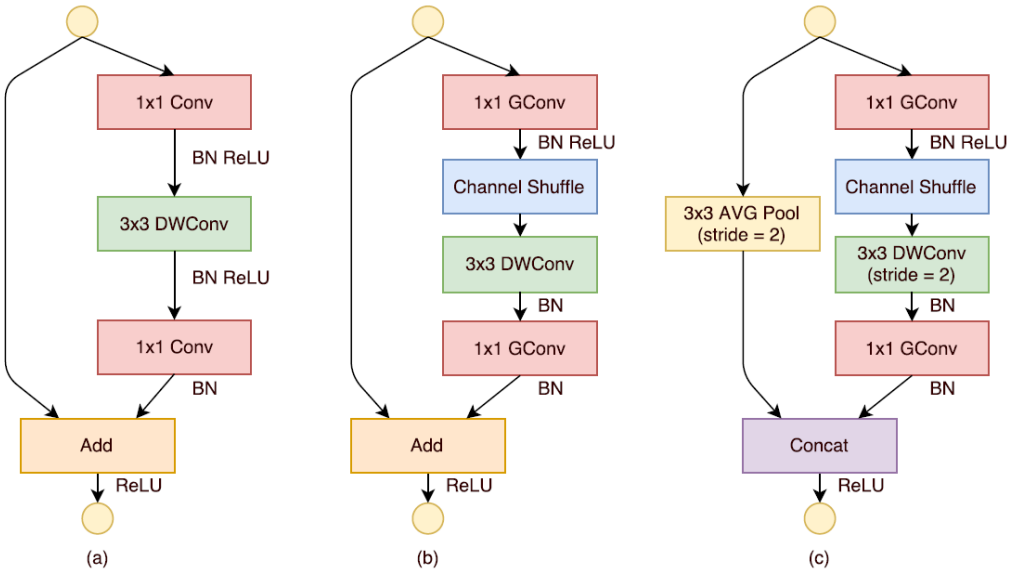
3.6.1. MobilnetV2 Algoritması

Modern ağlarda CNN modellerinin kullanılmasıyla, yüksek işlem maliyeti ile birlikte birçok mobil ve görüntü uygulamalarının yüksek hesaplama kaynak kullanımını gerektirmektedir[93]. Hiper-parametreler kullanarak, bu performans metriklerini azaltmak için MobileNetV2 kullanmak mümkündür[51]. MobileNetV2 birçok farklı alanda görüntü işleme teknikleriyle kullanılmış ve yüksek başarı elde edilmiştir[54–57]. Siber saldırı tespit etmek için saldırıları gri tonlu resimlere çevrilmiş, MobilNetV2 kullanılanarak sınıflandırma işlemi yapılmıştır[58]. UNSW-NB15 ve KDD-99/DARPA veri setleri görüntülere dönüştürülerek MobilNetV2 ile görüntü işleme teknikleri kullanılmış ve saldırı önleme sistemi önerilmiştir[94]. Android sistemlerinde malware tespit için XML ve DEX dosyalarındaki malwareleri görüntülere dönüştürerek, MobilNetV2 CNN modeli kullanılarak malware tespiti yapılmıştır[59].

3.6.2. ShuffleNet Algoritması

Görüntü işleme tekniklerinde kısıtlı kaynakları en iyi ve verimli bir şekilde kullanmak için ShuffleNet CNN modeli geliştirilmiştir[52]. Bu model Şekil 3.4'de gösterilen mimariyle siber güvenlikte görüntü işleme üzerine birden çok çalışmada kullanılmıştır. Nesnelerin internetinde otomatik modülasyon sınıflandırma yapmak için bilişsel radyo dalgalarından siber sistemlerde bilinmeyen sinyalleri tespit etmek için ShuffleNetCNN modeli önerilmiştir[60]. Yüksek voltajlı elektrik sistemlerinde

siber atakları önlemek için ShuffleNet kullanılarak bir CNN model önerilmiştir[62]. Siber saldırılardan olan kaba kuvvet saldırıları genellikle son kullanıcıların bilgileri tahmin ederek sisteme giriş yapmayı sağlamaktadır[63]. Web sitelerinde son kullanıcıları ve makine botlarını ayırt etmek için kullanılan güvenlik mekanizması Computers and Humans Apart'ı (CAPTCHA'lar) atlamak derin öğrenme yöntemleri ile mümkündür, bunu önlemek için ShuffleNet CNN model oluşturarak, makine botları ve son kullanıcıları ayırt eden sistem önerilmiştir[64].



Şekil 3.4. (a) Derinlemesine evrişimli darboğaz (DWConv), (b) Noktasal grup evrişimli GConv ile ShuffleNet birimi ve kanal karşılaştırma, (c) Adım = 2 olan ShuffleNet birimi. ShuffleNet mimari yapısı[55].

3.7. HARRİS HAWK OPTİMİZASYON ALGORİTMASININ YAPISI

Günümüzde birçok optimizasyon algoritması gibi Harris Hawk Optimization (HHO) algoritması da doğayı taklit ederek geliştirilmiştir[95]. Bu algoritma zeki kuşlardan olan şahinlerin avlanma stratejisini taklit ederek geliştirilmiştir. Bu algoritma popülasyon temelli, gradyansız bir optimizasyon olup, birçok mühendislik çalışmalarına uygun bir formilasyonla uygulanabilmektedir[96].

$$x(t+1) = \begin{cases} (x_{rabbit}(\mathrm{t}) - x_m(\mathrm{t})) - r_3(\mathrm{LB} + r_4(\mathrm{UB} - \mathrm{LB})), & q < 0.5 \\ x_{rand}(\mathrm{t}) - r_1|x_{rand}(\mathrm{t}) - 2r_2\mathrm{x}(\mathrm{t})|, & q \geq 0.5 \end{cases}$$

(3.1)

$$x_m(t) = \frac{1}{N}\sum_{i=1}^{N} x_i(t)$$

(3.2)

Harris şahinleri rastgele dolaşırken Eşitlik 3.1'de verildiği üzere iki keşif stratejisine sahiptirler. Burada x (t + 1), her yenilemede Harris Şahin'inin konum vektörüdür. $x_{rabbit}(t)$ Avın konum vektörüdür, x(t) şahinin mevcut konumudur ve r_1 , r_2, r_3, r_4 ve q ise rastgele sayılardır (0,1). LB, UB sırasıyla alt değer ve üst değerlerdir. $x_{rand}(t)$, mevcut popülasyondan rastgele seçilen bir şahini temsil ederken, $x_m(t)$ mevcut şahin popülasyonunun ortalama konumudur. Ortalama pozisyon için keşiften saldırıya geçiş Eşitlik 3.2'deki formülasyon kullanılarak bulunmaktadır.

$$E = 2E_0\left(1 - \frac{t}{T}\right)$$

(3.3)

Harris şahinleri keşif işlemini yaptıktan sonra avın enerjisine göre farklı saldırı stilleri geliştirebilmektedir. Kaçış sırasında avın enerjisi önemli ölçüde azalmaktadır. Bu durumun matematiksel modeli Eşitlik 3.3'te görüldüğü gibidir. Burada $E0$ avın ilk enerjisi, E kaçan avın enerjisi, T maksimum yineleme sayısı olarak kullanılmıştır.

Bu çalışmada, HHO algoritması MobileNetV2 CNN ve ShuffleNet CNN modeller tarafından oluşturulan ayrı 1000 derin özellikler birleştirilerek toplamda 2000 özellik oluşturulmuş, bu özellikleri optimize etmek için kullanılmıştır.

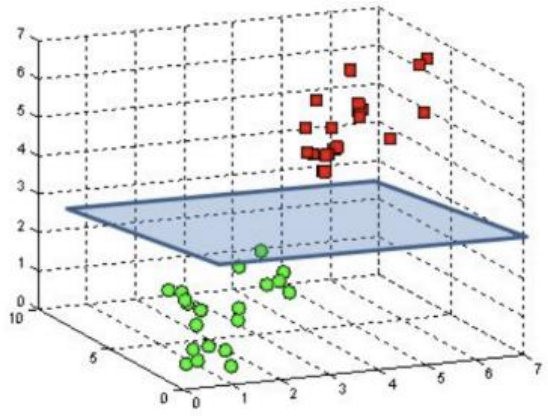
3.8. SINIFLANDIRMA ALGORİTMALARI

3.8.1. Destek Vektör Makinaları

Destek Vektör Makinaları, gözetimli öğrenme kullanarak sınıflandırma problemleri için kullanılmaktadır. Genellikle bir düzlem üzerinde bir doğru çizerek, bu doğrunun iki sınıf noktaları arasında maksimum uzaklığı bularak sınıflandırma işlemi yapar

[97]. Sınıflandırma yapılmasındaki asıl amaç gelecek olan verinin hangi sınıfa ait olduğunu bulmaktır.

Düşük boyuttaki verileri daha verimli sınıflandırma yapabilmek için çekirdek yöntemi kullanılmaktadır. Veri boyutunu büyütmeden elimizdeki verileri çekirdek fonksiyonları ile çarpma işlemi yaparak genişletip, anlamlı hale getirilmektedir [98]. Kullanılan çekirdeklerden biri Polinomal diğeri Gaussian RBF kübik çekirdeğidir. Polinomal çekirdeği ile verilerin boyutunu 2 boyuttan 3 ve daha fazla boyutta işlem yapılmasına olanak sağlamaktadır [99]. Her bir noktanın belirli bir noktaya benzerliğini normal dağılım ile hesaplayarak sınıflandırma yapmaktadır. Dağılım genişliği gamma hiperparametresi ile kontrol edilmektedir. Gamma parametresi ne kadar küçükse dağılım o kadar geniş olmaktadır. Overfit için gamma değerini düşürmek, underfit içinse gamma değerini artırmak gerekmektedir. Şekil 3.5'de üç boyutlu uzayda temsili verilerle ayrılma işlemi gösterilmiştir.



Şekil 3.5. Veri setinin üç boyutlu düzlemsel uzayda DVM kernel ile ayırılması.

3.8.2. K En Yakın Komşu

kNN, gözetimli öğrenme algoritması olup hem sınıflandırma hem de regresyon problemleri için kullanılmaktadır. kNN ile yeni noktaya en yakın olan aranıp, k yani bilinmeyen noktanın en yakın komşusu ile tahmin etme işlemi yapılmaktadır [100][101]. Uzaklık hesaplanması yapılması için bu çalışma kapsamında üç çeşit uzaklık hesaplama yöntemi kullanılmıştır.

Öklid uzaklığı kNN algoritmasında yakınlığı hesaplanması için kullanılmaktadır. Öklid doğrusal olarak iki nokta arasında uzaklıkları ölçmektedir. *P* ve *Q* arasındaki Öklid uzaklığı $P= (x_1, x_2, \ldots, x_n)$ ve $Q=(y_1, y_2, \ldots, y_n)$ olmak üzere Eşitlik 3.4'te hesaplaması yapılmıştır.

$$D(P,Q) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \tag{3.4}$$

Minkowski uzaklığı genel bir formülle ifade edilir ve ayrıca p'nin farklı değerleri için çeşitli mesafe ölçülerini tanımlamak için kullanılmaktadır. Minkowski mesafesi, Öklid uzayında tanımlanmış bir dizi olarak tanımlanmaktadır. Sınıflandırma ve kümeleme gibi makine öğrenimi ve veri madenciliği uygulamalarında sıklıkla kullanılan Öklid uzaklığı gibi uzaklık ölçülerinin genelleştirilmesidir. Herhangi iki P ve Q noktası arasındaki Minkowski mesafesi Eşitlik 3.5'e göre $P= (x_1, x_2, \ldots, x_n)$ ve $Q=(y_1, y_2, \ldots, y_n)$ ile hesaplanır.

$$D(P,Q) = (\sum_{i=1}^{n}|x_i - y_i|^p)^{1/p} \tag{3.5}$$

Mahalanobis mesafe, bilgisayar bilimi ve diğer birçok alanda kullanılan bir ölçüm sistemidir. Diğer ölçüm sistemlerinden en belirgin farkı mesafe ayrımını eliptik bir düzlem üzerinde yapmasıdır. Mahalanobis mesafesi, değer vektörü ile ortalama arasındaki farkın çarpımının karekökü, kovaryans matrisinin tersi ve yine değer vektörü ile ortalamalar arasındaki farkın devrik çarpımı olarak hesaplanır. Mahalanobis mesafesinin hesaplanması Eşitlik 3.6'da gösterilmiştir.

$$D_M(x) = \sqrt{(x-\mu)^T S^{-1}(x-\mu)} \tag{3.6}$$

BÖLÜM 4

MATERYAL VE METOT

Bu çalışmada bilgisayar ağlarında yeni saldırıların tespiti ve gelecekteki saldırıları tahmin etmek için iki farklı hibrit yöntem önerilmiştir. Yöntemlerden biri graf tabanlı LSTM kullanıldığı için GLSTM diğeri log kayıtlarının QR kod görüntüleri oluşturularak anomali tespiti yaptığı için QRLog olarak adlandırılmıştır. Sistem karmaşıklığından dolayı saldırı tespiti gerçek dünya uygulamalarına uyarlanmadan önce önemli miktarda test, değerlendirme ve ayarlama yapmak gerekmektedir. Bu yöntemlerden ilk olan GLSTM kapsamlı bir dizi izinsiz ve anormal davranışa sahip HDFS veri seti ile ikinci yöntem ise University of New Brunswick tarafından oluşturulan CSE-CIC-IDS2018[102] veri seti ile test ve değerlendirme yapmak için kullanılmıştır.

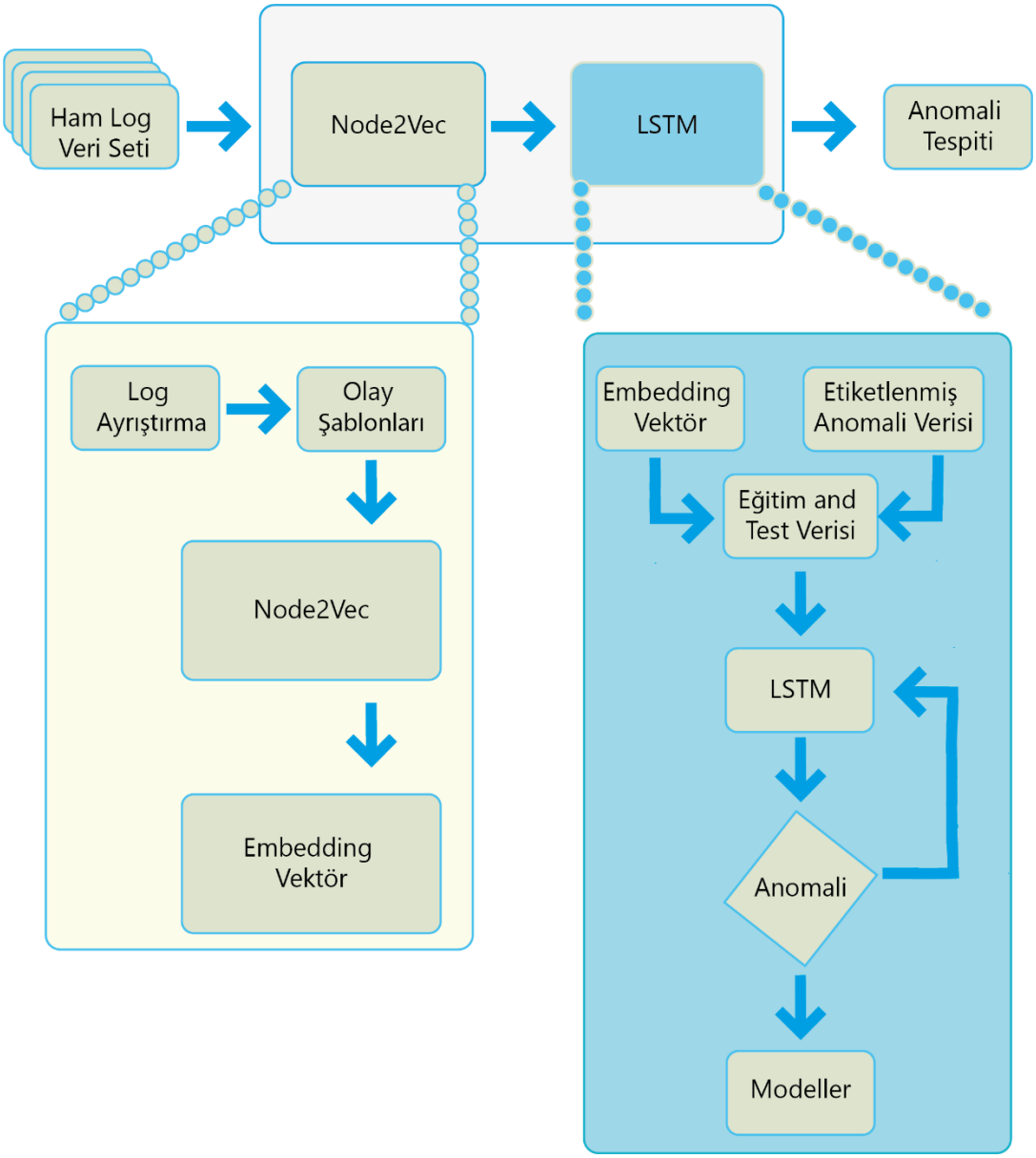
Bu çalışmanın temel amacı, ağda görülen olayların ve davranışların soyut temsillerini içeren kullanıcı profillerinin oluşturulmasına dayalı olarak izinsiz giriş tespiti için makine öğrenmesi ile graf ve görüntü işleme tekniklerini kullanarak bu saldırıları tespit etmek için sistematik bir yaklaşım geliştirmektir.

4.1. MODELLERİN BELİRLENMESİ

4.1.1. Graf Tabanlı Anomali Tespiti (GLSTM)

Log analizinde anomali tespiti yapmak oldukça zordur. Çünkü log verileri hem sayısal hem de kategorik verilerden oluşmaktadır. Bu verilerinin analiz edilebilmesi belli başlı ön işlemlerden geçmesi gerekmektedir. Bölüm 2'de bahsedilen her bir çalışma veri seti üzerine farklı ön işleme teknikleri uygulanmaktadır. Böylece veri setinden öz nitelik çıkartıp, vektörel bir hale getirilmektedir. Daha sonra vektörel

olarak elde edilen bu veri seti derin öğrenme algoritmaları ile analiz edilerek, anomali tespiti yapılmıştır.



Şekil 4.1. Önerilen GLSTM yöntemin ana mimarisi.

Şekil 4.1'de önerilen metodun mimari yapısı gösterilmiştir. Yapı incelendiğinde öncelikle birden çok heterojen kaynaktan log verileri alınıp, şablonlar oluşturulmaktadır. Graf algoritması veri setini sayısal olarak kabul ettiğinden bu şablonların veri setinin kategorik olan kısmının sayısallaştırılma işlemi yapılmıştır. Sayısallaştırma işlemi literatürde kullanılan iki yöntemle yapılmıştır. Bunlar biri

Label Encoding diğeri One Hot Encoding işlemidir. Sayısallaştırılmış veri setini Node2Vec algoritmasına kenar ve düğüm olarak verilerek graf yapısı oluşturulmuştur. Bu graf yapısından vektörel bir sonuç elde edilmiştir. Bu sonucu da LSTM algoritmasına girdi olarak verilerek log anomali tespiti yapılmıştır.

Bu çalışmada önerilen GLSTM metodu iki aşamadan oluşmaktadır. Birinci aşamada veri setinden öznitelik yapılmadan veriler şablonlara dönüştürüldükten sonra bir grafa aktarılmıştır. Bu çalışma için graf algoritmalarından Node2Vec algoritması kullanılmıştır. Çünkü verilerin analiz edilmesi için vektörel olarak veri seti elde edilmesinde en etkili algoritmadır. Yapılan deneysel testler bu algoritmanın bu çalışma için uygun ve etkili olduğu kanıtlanmıştır. İkinci aşama analiz ve sınıflandırma işlemidir. Bu aşamada derin öğrenme algoritmalarından LSTM kullanılmıştır. Bu algoritma yenilemeli derin öğrenme algoritmasıdır. Log analizinde anomalinin tespit edilmesinde en çok tercih edilen algoritmaların başında gelmektedir. Yapılan deneysel testler sonucunda anomali tespitinde Node2Vec ve LSTM beraber kullanılarak, yüksek bir başarı oranı elde edildiğini kanıtlamıştır.

4.1.2. Hafif Derin Öğrenme Tabanlı Saldırı Tespiti (CNNQRLog)

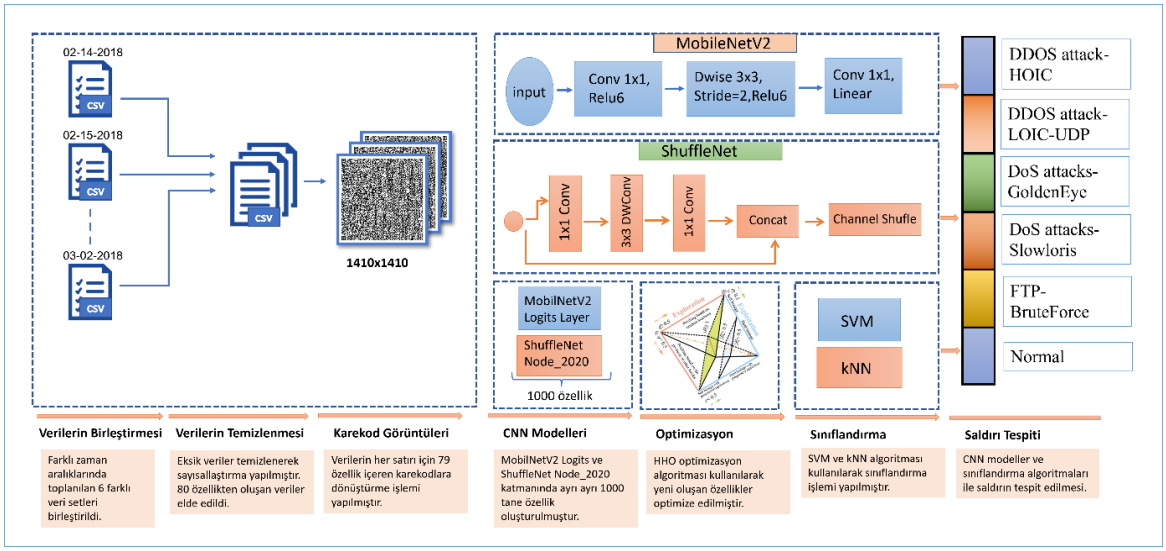
Bu çalışmanın amacı sunucu, web, ağ, mobil ve masaüstü uygulamalarına yapılan karmaşık, önceden tahmin edilemeyen siber saldırıları hafif derin öğrenme algoritmalarını kullanarak etkin, hızlı ve doğru bir şekilde tespit etmektir. Böylece bu çalışmanın veri ön işlemeden sonra ki kısım olan metotlar kısmında, üç aşamalı olarak gerçekleştirilmiş ve bu yöntemler sayesinde en etkin bir şekilde siber saldırılar tespit edilmiştir.

Bu çalışma; veri ön işleme, metotlar ve sonuç olmak üzere üç kısımdan oluşmaktadır. Şekil 4.2'de önerilen metotun ana hatları gösterilmiştir. Veri ön işleme kısmında, öncelikle farklı zaman aralıklarında toplanılan, içinde her bir saldırıya ait farklı özellikleri bulunan ve altı farklı saldırı sınıfına ait .csv uzantılı veri dosyaları bir araya getirilmiştir. Daha sonra normalizasyon işlemiyle eksik ve gereksiz veriler temizlenmiştir. Veri işlemenin son aşamasında her bir saldırı sınıfına ait özelliklerin

QR kod resimleri oluşturulmuştur. Daha sonra oluşturulan bu QR kod resimler metotlar kısmında hafif derin öğrenme modellerine girdi olarak verilmiştir.

Bu çalışmanın metotlar kısmı da üç aşamadan gerçekleşmektedir. Birinci aşamada, hafif derin öğreme algoritmalarında iki farklı CNN modeli MobilNetV2 ve ShuffleNet modellerine QR kod resimleri direkt olarak verilerek modeller eğitilmiştir. İkinci aşamasında, bu CNN modellerin MobileNetV2 CNN modeli için Logits ve ShuffleNet CNN modellinden ise node_202 katmanında elde edilen derin öznitelikle SVM, kNN sınıflandırma algoritmaları kullanılarak sınıflandırma yapılmıştır. Üçüncü aşamada, elde edilen bu derin öznitelikler HHO optimizasyon algoritması kullanılarak en iyi öznitelikler elde edilmiş ve SVM,kNN algoritmaları ile sınıflandırma yapılmıştır. Üçüncü aşamada, diğer aşamalarda elde edilen sonuçlardan daha iyi sonuçlar elde edilmiştir.

Bu çalışmanın en önemli katkılarından bazıları aşağıdaki gibidir. Öncelikle her türlü veri seti için uygundur ve siber saldırı veri seti QR kod görsellerine dönüştürülerek kullanılır. İkincisi, hafif derin öğrenme modellerinin analizi ve karşılaştırılması. Üçüncüsü, CNN modellerinden elde edilen derinlemesine özniteliklerle sınıflandırmadır. Son olarak elde edilen derinlemesine öznitelikler daha iyi sonuçlar elde etmek için optimizasyon algoritması ile optimize edilmiştir.



Şekil 4.2. Önerilen CNNQRLog yönteminin ana mimarisi.

4.2. DENEYSEL TESTLERDE KULLANILAN VERİ SETLERİ

4.2.1. GLSTM Yönteminde Kullanılan Veri Seti ve Veri Ön İşlemleri

Log verilerinin analizi, sayısal ve kategorik verileri girdi olarak almaktadır. Bu da ham log verilerin temizlenmesini, sıralanmasını ve normalleştirilmesini gerektirmektedir. Log kayıtları iki ana kısımdan oluşmaktadır. Baş kısım ve metin kısmı. Baş kısım genellikle zaman damgaları, ana bilgisayar adları ve olayların ciddiyeti gibi birkaç segmentten oluşmaktadır. Metin mesajı girişi geliştiriciler tarafından manuel olarak önceden tanımlanmaktadır. Bu da bir sistem içinde bile sistemler arasında önemli ölçüde farklılık gösterebilmektedir. Bu mesajlar da iki kısımdan oluşur sabit mesajlar ve değişken mesajlar.

Bu çalışmanın GLSTM yönteminde, siber saldırıları tespit etmek için log kayıtlarını analiz etmek ve saldırganların log kayıtlarında bıraktıkları izlerin oluşturdukları anormalikleri tespit etmek için bir model önermiştir. Bu modelin test edilmesinde HDFS veri seti kullanılmıştır. HDFS log veri kümesi 200'den fazla Amazon'un heterojen kaynağından toplanmış ve 11.175.629 log verisinden oluşmaktadır. HDFS log verileri, block_id kullanarak ayırma, çoğaltma ve silme gibi işlemleri belirli bir blokta kaydetmektedir. Bu veri kümesi 575.061 log bloğundan oluşup, Hadoop'un uzmanları tarafından 16.838 anormal olarak etiketlenmiştir. Çizelge 4.1'de HDFS veri seti ile ilgili bilgiler verilmiştir.

Çizelge 4.1. HDFS veri seti detayları.

Veri seti	Zaman	Log Satırı	Blok Sayısı
HDFS	38,7 saat	11,175,629	16,838(blok)

4.2.1.1. HDFS Veri Seti

Her ham log verisi iki kısımdan oluşmaktadır. Bunlardan biri zaman damgası diğeri log tamamlayıcı kısımdır. Zaman damgası, her log oluşumunun zamanını kaydetmektedir. Farklı biçimlerdeki zaman damgası düzenli ifadeler oldukları için

log ayrıştırma aşamasında ham log verilerinden kolayca çıkarılabilmektedir. Log tanımlayıcı, sistemin birden çok işlemi veya mesaj alışverişini tanımlayan bir belirteçtir. Örneğin, HDFS log veri kümesi 200'den fazla Amazon'un heterojen kaynağından toplanmış ve 11.175.629 satır log verisinden oluşmaktadır. HDFS log verileri, block_id kullanarak ayırma, çoğaltma ve silme gibi işlemleri belirli bir blokta kaydetmektedir. Bu veri kümesi 575.061 log bloğundan oluşup, Hadoop 'un uzmanları tarafından 16.838 anormal olarak etiketlenmiştir[92].

Log verisi X_1, X_2,X_3, X_4.... X_n oluşsun. Bu log verisi T_{k1}, T_{k2},T_{k3}, T_{k4}.... T_{kn} log şablonlarına karşılık gelmektedir. T_k log ayrıştırma metodu tarih(t), zaman(z), pid(p), tür(r), bileşen(b), içerik(i), şablonid(j), şablon(l) ve anomali(k) olmak üzere ayrıştırma işlemi yapmaktadır.

$$t, z, p, r, b, i, j, l, k = Tk(X) \quad (4.7)$$

Log ayrıştırma metodu sonucunda;

$$k = \begin{cases} 0, & Normal \\ 1, & Anormal \end{cases} \quad (4.8)$$

Eşitlik 4.7'de log şablonları oluşturulduktan sonra Node2Vec algoritmasına aktarılır. Bu algoritma sonucunda çıkan embedding vektör ile Eşitlik 4.8'deki etiketlenmiş anormal verilerden eğitim ve test verileri oluşturulmaktadır.

Çizelge 4.2'de görüldüğü gibi ilk kısmı zaman damgasından diğer kısmı log tamamlayıcısı olarak görülmektedir. Böylece log verilerinin bir kısmı sayısal veri diğer kısmı sözel veri içermektedir. Log verisindeki her bir sözcük log anahtar sözcüğü veya parametre olarak kullanılabilmektedir. Log parametreleri genellikle IP adresleri, MAC bilgileri veya kullanıcı bilgilerin olduğu kısımdan oluşmaktadır. Log anomali tespiti genel olarak log verilerinin anormal olmadığının tespit edilmesidir. Log verilerinde "INFO" olması o log verisini normal olduğu anlamına gelmemektedir. Bu için log verileri ayrıştırılırken anormal olup, olmadığı

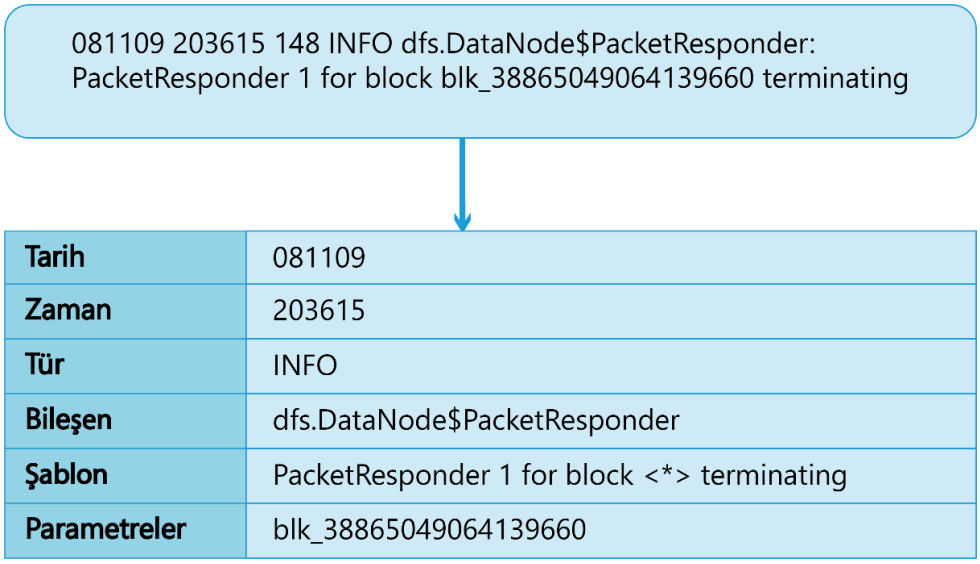
bilinmemektedir. Log ayrıştırmanın amacı ham log verilerinden anlamlı veriler elde etmektir. Böylece bu veriler kullanılarak analizler yapılıp, modeller oluşturulmaktadır.

Çizelge 4.2. Ham log veri yapısı.

No	**Ham Log Verisi**
1	081109 210248 1138 INFO dfs.DataNode$PacketResponder: Received block blk_6921674711959888070 of size 67108864 from /10.251.65.203
2	081109 211029 31 INFO dfs.FSNamesystem: BLOCK* NameSystem.addStoredBlock: blockMap updated: 10.251.106.50:50010 is added to blk_-29548654251973735 size 67108864
3	081109 212245 27 INFO dfs.FSNamesystem: BLOCK* NameSystem.allocateBlock: /user/root/rand/_temporary/_task_200811092030_0001_m_001648_0/part-01648. blk_2513940824125131775
4	081109 214910 2848 WARN dfs.DataNode$DataXceiver: 10.250.13.188:50010:Got exception while serving blk_6241141267506413726 to /10.251.194.245:
5	081110 030331 6561 WARN dfs.DataNode$DataXceiver: 10.251.42.191:50010:Got exception while serving blk_-8023826090828946372 to /10.251.214.130:

4.2.1.2. Logların Ayrıştırılması

Logları otomatik olarak analiz etmek için, logları metin ve makine öğrenimi algoritmalarına uyan uygun biçimlere dönüştürmek gerekmektedir. Log verilerinin analiz edilebilmesi için benzersiz kısımlarının belirlenmesi gerekmektedir. Şekil 4.3'de gösterildiği gibi log kayıtlarında benzerlik oranları farklı bulunan kısımlar etiketlenerek benzersiz şablonlar üretilmiştir.



Şekil 4.3. Log ayrıştırma adımları.

Log ayrıştırma işlemi yapılırken loglar ön işleme tabi tutulmaktadırlar. Çizelge 4.3'de zaman damgası içerisinde bulunan değerler de tarih, zaman ve PID olmak üzere ayrıştırıldı. Her bir log şablonu birbirinden farklı olduğu için her şablona ŞablonID şeklinde etiketlendirilmiştir. Bileşen ve içerik kısımları da farklı bir kolon altında ayrıştırmaya tabi tutulmuştur.

Çizelge 4.3. Ham log verinin şablonlara dönüştürülmesi.

NoID	1,2,3,
Tarih	081109 , 081110, …
Zaman	203615, 203807, 204005
PID	148, 222, 35, 308, 329, ….
Tür	INFO, WARN
Bileşen	dfs.DataNode$PacketResponder,dfs.FSNamesystem,dfs.DataNode$DataXceiver
İçerik	BLOCK* NameSystem.delete: blk_-1233005817943453613 is added to invalidSet of 10.251.75.49:50010
ŞablonID	E1,E2,E3,E4, …
Şablon	BLOCK* NameSystem.delete: <*> is added to invalidSet of <*>:<*>

4.2.2. CNNQRLog Yönteminde Kullanılan Veri Seti ve Veri Ön İşlemleri

Ağ davranışları değiştikçe, izinsiz girişler geliştikçe, statik ve tek seferlik veri kümelerinden yalnızca o zamanın trafik kompozisyonlarını ve izinsiz girişlerini yansıtmakla kalmayıp aynı zamanda değiştirilebilir olan daha dinamik genişletilebilir ve tekrarlanabilir sistemlere doğru hareket gerekli hale gelmiştir. Bu eksikliklerin üstesinden gelmek için, bu çalışmada ağ tabanlı saldırı detektörlerine odaklanarak izinsiz giriş tespit sistemlerini analiz etmek, test etmek ve değerlendirmek için farklı zaman aralıklarındaki veri setlerini birleştiren, temizleyen açık erişimli veri kümeleri[108] kullanılarak sistematik bir yaklaşım geliştirilmiştir.

4.2.2.1. CSE-CIC-IDS2018 Veri Seti

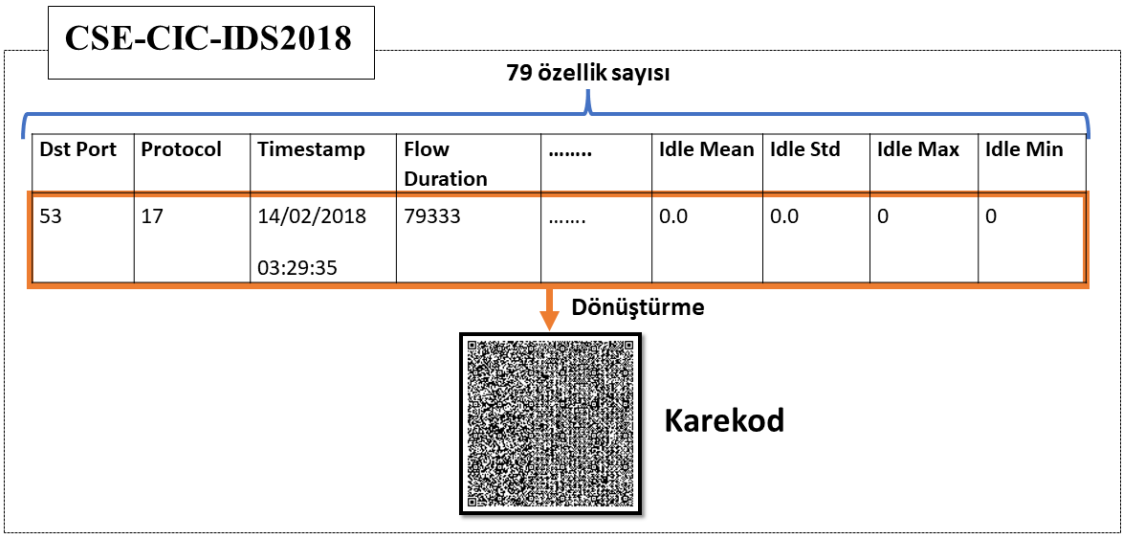
Bu çalışmada kullanılan veri seti : DDOS attack-HOIC, DDOS attack-LOIC-UDP, DoS attacks-GoldenEye, DoS attacks-Slowloris, FTP-BruteForce ve Normal olmak üzere altı farklı saldırı çeşidi içermektedir. Bu veri kümelerini oluşturmak için oluşturulan altyapıda 5 farklı bölümde kurban makinalar oluşturulmuş, 50 makine ile saldırılar gerçekleştirilmiştir. Bu altyapı için toplamda 420 makine ve 30 sunucu kullanılmıştır[102]. Yapılan her saldırı için 80 özellik içeren ham veri kümelerine ait bilgiler Çizelge 4.4'te verilmiştir.

Çizelge 4.4. Ham log veri setinin ilk dört satırı.

Dst Port	**Protocol**	**Timestamp**	**Flow Duration**	**……..**	**Idle Mean**	**Idle Std**	**Idle Max**	**Idle Min**	**Label**
53	17	14/02/2018 03:29:35	79333	…….	0.0	0.0	0	0	Benign
0	0	14/02/2018 08:36:39	112638623	…….	0.0	0.0	0	0	Benign
22	6	14/02/2018 08:40:13	6453966	…….	0.0	0.0	0	0	Benign
0	0	14/02/2018 08:33:50	112641466	…….	0.0	0.0	0	0	Benign

4.2.2.2. Veri Setinin QR Kodlara Dönüştürülmesi

CSE-CIC-IDS2018 veri kümesinde, izinsiz girişlerin ayrıntılı açıklamalarını ve uygulamalar, protokoller veya daha düşük seviyeli ağ varlıkları için soyut dağıtım modellerini içeren veri kümelerinden oluşmaktadır. Veri kümeleri ilk paketin ileri (kaynaktan hedefe) ve geri (hedeften kaynağa) yönleri belirlediği Çift Yönlü Akışlar (Biflow) üretmektedir. Bu nedenle Süre, Paket sayısı, Bayt sayısı, Paketlerin Uzunluğu gibi 80 istatistiksel özellik, vb. ayrıca ileri ve geri yönde ayrı ayrı hesaplanmaktadır. Şekil 4.4'te veri setinin her bir satırı için kare kodlar oluşturulmuştur. Bu veri setinin son satırı saldırı türlerini gösterdiğinden saldırı türleri her bir satır için ayrı ayrı oluşturulmuştur. Böylece altı farklı saldırı türünün herbiri için 1000 tane kare kod oluşturulmuştur. Kare kodlar .png uzantısında olup, bir bit derinliğinde 1410 x 1410 genişlik ve yükseklik boyutlarına sahiptir.



Şekil 4.4. Ham log veri setinin QR kodlara dönüştürülmesi.

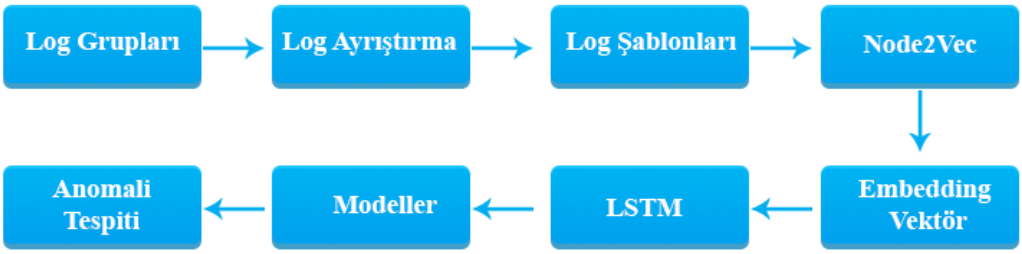
4.3. ÇALIŞMADA KAMSAMINDA ÖNERİLEN GLSTM YÖNTEMİ

Bu çalışmanın GLSTM yöntem mimarisi iki algoritmanın beraber kullanılmasına dayanmaktadır. İlki, anlamsız verinin temizlenip, anlamlı ve analiz edilebilecek bir duruma getirebilmek için Node2Vec algoritmasının kullanılmasıdır. İkincisi, analiz

edilebilecek verileri girdi olarak alıp, log kayıtlarında anomali tespiti için derin öğrenme algoritmalarından LSTM algoritmasının kullanılmasıdır.

Log verilerinden anormallik tespitinde üç tür anormallik vardır. Bu anomalilerden ilki nokta anomalisidir. Nokta anomalisi, kalan verilerin ortalama veya normal dağılımından önemli ölçüde sapan bir veridir[103]. İkincisi, bağlamsal anomalidir. Bağlamsal anormallik, belirli bir bağlam ve diğer bağlamlardaki standartla sınırlı anormal bir davranıştır[104]. Üçüncüsü, toplu anomalidir. Bağlamsal ve noktasal anormalliklerin aksine, toplu anormallikler verilerde bir dizi anormal değer olarak görünür. Toplu anomaliler, tüm veri kümesine göre bir veri örnekleri koleksiyonunun anormal davranışıdır[105].

Log anormalliği algılama, günlük verilerinde beklenen davranışa uymayan anormal sistem kalıplarını tanımlar. Bu bölümde, burada benimsenen algoritmalara dayalı olarak bu alandaki çalışmalarımız tartışılmaktadır. Çalışmamızın ana hatları Şekil 4.5'de gösterilmiştir. İlk olarak farklı log gruplarından ham log verileri alınmış ve gereksiz, gürültülü veriler çıkarılarak anlamlı hale getirilmiştir. Bu log verilerinden şablonlar oluşturulmuş ve özellik vektörünü oluşturmak için Node2Vec algoritmasına girdi verilmiştir. LSTM algoritması ile model eğitimi yapılmış ve bu eğitilen modeller ile anomali tespiti yapılmıştır.



Şekil 4.5. Önerilen GLSTM yöntemin ana hatları.

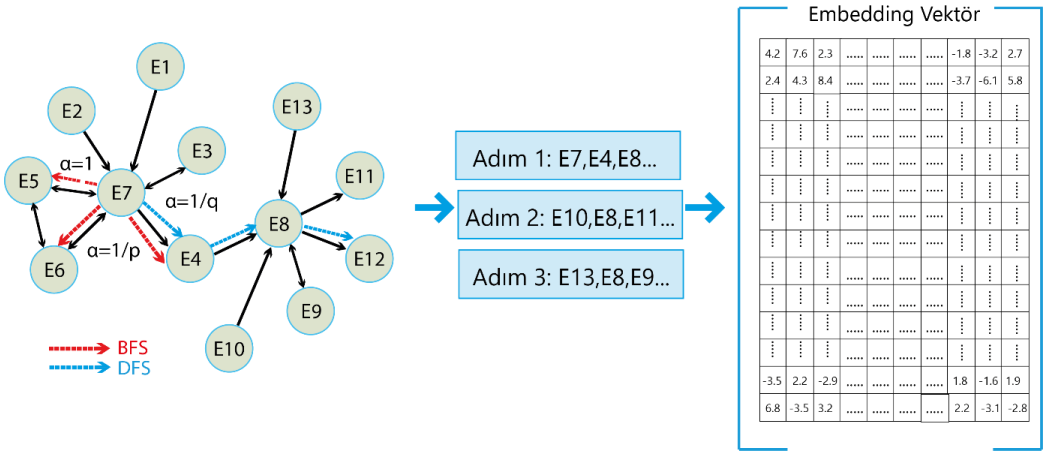
4.3.1. Verilerin Graf'a Aktarılması

Verileri Node2Vec algoritmasından vektörel alabilmek için bir grafa aktarılması gerekmektedir. Bu çalışmada makine öğrenimi ve derin öğrenme yapılarını kolay kullanım sunduğundan StellarGraph kullanılmıştır[81]. Bu graf yapısının kullanılmasının başlıca sebepleri;

- Görselleştirme ve çeşitli makine öğrenmesi için kullanılabilmesi,
- Düğüm ve kenarlardan öznitelik çıkarabilmesi,
- Büyük veri üzerinde uygulanabilmesi,
- Düğümlerin sınıflandırılmasının yapılabilmesi,

gibi birçok işlemi kolay ve uygulanabilir bir şekilde yapabilmektedir. Bu graf yapısı kullanılarak derin öğrenme ve makine öğrenmesi ile ilgili birden çok çalışma yapılmıştır[82–84].

Graflar kenar ve düğümlerden oluştukları için verilerin grafa aktarılmasın kenar ile düğümlerden yararlanılmıştır. Düğümler ise belli bir kaynaktan hedefe doğru gitmesi gerekmektedir. Bunun için kaynak olarak her bir şablon için farklı olan ŞablonID seçilmiş ve hedef için de anomali veya normal kolon seçilmiştir. Diğer kalan kolonlar ise düğümler için kullanılmıştır. Şekil 4.6'da grafa aktarılmış verilerin kenar ve düğümleri gösterilmiştir.



Şekil 4.6. Graf ile vektörel verilerinin oluşturulması.

4.3.2. Tekrarlamalı Sinir Ağları

LSTM'ler kendini tekrarlayan RNN'lerin bir üyesidir. RNN'ler ardışık verileri her seferinde bir öğeyi alarak kendini tekrarlayan modelleridir[85]. Markov modelleri ile karşılaştırıldığında durum uzay kümeleri artmasına rağmen uzun vadede bağımlılıktan dolayı daha iyi sonuç vermektedirler[107]. LSTM'ler ilk olarak [88] tarafından ortaya atıldı. LSTM'ler RNN'lerin dezavantajlarını ortadan kaldırmak için geliştirilmiştir. LSTM'ler RNN'ler gibi yenilemeli olarak çalışır bunlardan farkı kendi içinde gizli gösterim ile farklı hücreler üzerinde çalıştırmaktadır.

4.3.2.1. Uzun Kısa-Vadeli Bellek Ağları

LSTM'ler özet olarak kapı adı verilen bir dizi işlemi tüm ağ üzerinde bulunan hücrelerden geçerek hesaplama yapılmaktadır. LSTM'lerde üç önemli kapı vardır. Bunlar unutma, giriş ve çıkış kapılarıdır. Unutma kapısı, Şekil 4.7'de gösterildiği gibi önceki hücre C_{t-1}'nin durumuna bağlı olarak hangi bilgilerin tutulması veya atılması gerektiğini belirlemektedir. Unutma kapısı, önceki durumun gizli hücresi h_t ve girdi hücresi X_t 'yi sigmoid fonksiyonu ile 0 veya 1 olacak düzeye getirmektedir. Giriş kapısı, C_t'deki yeni bilginin saklanması kontrol etmektir. Bunu da gizli durum h_t ve girdi hücre X_t bilgilerini aynı anda sigmoid ile tanh aktivasyon fonksiyonundan geçirilerek hesaplanmaktadır. Çıkış kapısı, h_{t+1} gizli durum bilgisini belirlemek için C_t'deki bilgiyi filtreleyerek hesaplamaktadır. Eş. 4.9,4.10,4.11,4.12 ve 4.13'de bu kapıların formülü verilmiştir.

Unutma kapısı:

$$ft = \sigma(W_{fx}\, x_t + W_{fh}\, h_t + b_f), \tag{4.9}$$

Giriş kapısı:

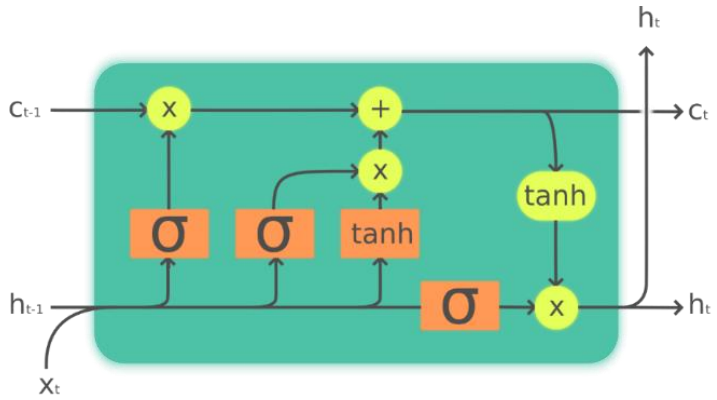
$$i_{t+1} = \sigma\,(W_{ix}\, x_t + W_{ih}\, h_{t-1} + b_i), \tag{4.10}$$

$$C_t = f_{t+1}\, c_{t-1} + i_{t+1} \tanh(W_{cx}\, x_t + b_c), \tag{4.11}$$

Çıkış kapısı:

$$O_{t+1} = \sigma\ (W_{ox}\ x_t + W_{oh}\ h_{t-1} + b_o\ ]), \tag{4.12}$$

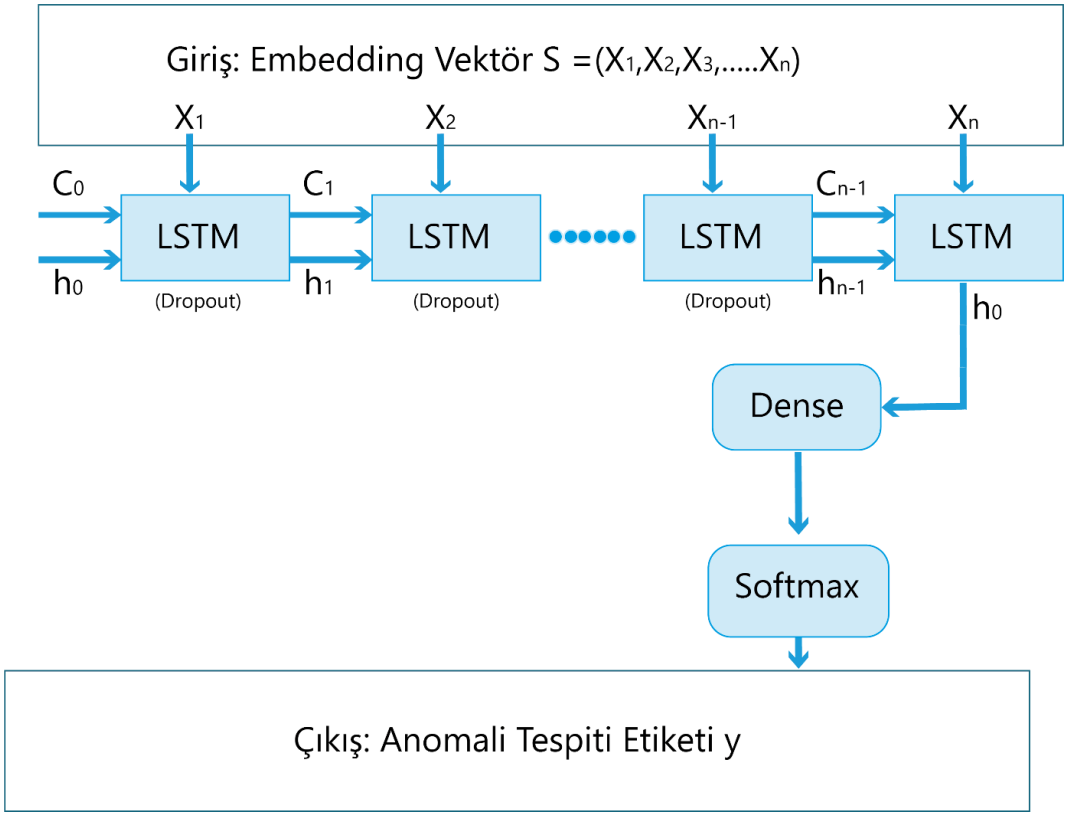
$$h_t = O_{t+1}\ \tanh\ (C_t) \tag{4.13}$$



Şekil 4.7. LSTM ana mimarisi.

4.3.3. Anomali Tespiti

Şekil 4.8'de LSTM algoritmasının bu çalışmada kullanım şekli gösterilmiştir. Giriş verileri Hadoop 'un birden çok kaynaktan topladığı HDFS ver seti kullanılmıştır. Bu veriler 1-hot encoding ve label encoding metotlarıyla sayısallaştırma işlemi yapılmıştır. Daha sonra bu veriler Node2Vec algoritmasına girdi parametresi olarak verilerek, çıkış olarak embbeding vektör oluşturulmaktadır. Bu embbeding vektör ile etiketlenmiş anormal log verileri LSTM algoritmasına giriş parametresi olarak verilmektedir. Böylelikle anormal değerleri tespit eden modeller oluşturulmaktadır.



Şekil 4.8. LSTM katmanlarının modellenmesi.

4.4. ÇALIŞMADA KAMSAMINDA ÖNERİLEN CNNQRLOG YÖNTEMİ

Bilgi teknolojileri hızla geliştikçe sunucular, bulut, IoT, mobil ve masaüstü uygulamaları gibi yüksek değerleri nedeniyle siber saldırıya uğramaktadır. Bu nedenle, siber saldırılar birçok alanda büyük endişe uyandırmıştır. Saldırı tespit sistemleri, siber güvenlik alanında önemli rol oynamakla birlikte ayrıntılı sistem çalışma verilerinden oluşmasından dolayı önemli veri analiz nesnesi haline gelmiştir. Geleneksel saldırı tespit sistemleri daha önce tespit edilen saldırıların kaydedilip yeni gelen saldırılarla karşılaştırmasıyla veya sistem anormalliklerini bakarak siber saldırıları tespit etmektedir. Saldırı tespit verileri çok büyük, saldırı türleri çeşitli ve bilgisayar korsanlığı becerilerinin gelişmesinden dolayı geleneksel tespit yöntemlerinin verimli olmamasına neden olmaktadır. Geleneksel saldırı tespit teknolojisini geliştirmek için son yıllarda makine öğrenmesi ve derin öğrenme yöntemi başta olmak üzere birçok saldırı algılama mekanizması önerilmiştir.

Bu çalışmada önerilen CNNQRLog yönteminde, siber saldırıların tespit edilmesinde hafif derin öğrenme modellerinin en uygun özniteliklerin kullanılmasını sağlayan çok amaçlı optimizasyon tabanlı hibrit bir yöntem önermektedir. İlk olarak çok sınıfa sahip hacimli verilerin QR kod resimleri oluşturulmuştur. Ardından MobileNetV2 ve ShuffleNet CNN modelleri kullanılarak QR kod görüntüleri eğitilmiştir. Eğitilen görüntülere ait derin CNN modelleri ile özellikleri çıkarılmış ve sınıflandırma amacına yönelik en etkili özelliklerin belirlenmesi için Harris Hawk Optimizasyon (HHO) algoritması özellik seçim amacıyla kullanılmıştır.

4.4.1. CNN Hafif Derin Öğrenme Modelleri

Nesnelerin interneti (IoT) ağında dağıtılmış hizmet reddi (Distributed Denial-Of-Service, DDoS) saldırılarının tespiti için çok amaçlı optimizasyon tabanlı bir özellik seçimi (Feature Selection, FS) yöntemi kullanılmıştır. Siber saldırıların tespit edilmesinde perfornması ve doğruluğu artırmak için uygun özniteliklerin seçilmesi gerekmektedir. Uygun özniteliklerin seçmek, verilerin boyutluluğunu azaltmak ve IDS'nin performansını iyileştirmek için uygun FS yöntemi kullanılmıştır [39].

Modern Uç/Bulut teknolojilerine bağlanan fiziksel IoT Uç cihazların doğası gereği kısıtlı olması, büyük hacimli saha eğitim verilerinin olması, yeterli depolanma alanlarına ve işlem yeteneklerine sahip olmamalarından dolayı derin trafik denetimi ve sınıflandırma gibi ağır hesaplama işlemlerinin bulut tabanlı mimarilerden yararlanarak Extreme Learning Machines modellerine dayalı trafik sınıflandırmasını modelini önermişlerdir[38].

Bu çalışmada önerilen CNNQRLog yönteminde, veri güvenliğinin sağlanması ve ağ hizmetlerinin aksamasına neden olan DDOS saldırı probleminin tespit edilmesi amaçlanmıştır. DDOS saldırıları önelenebildiği takdirde ciddi maliyet problemlerini ortadan kaldıracaktır [107]. Bu amaç doğrultusunda yeni, hızlı ve yüksek doğruluk oranı ile tespit eden bir model önerisi sunmaktır. Ağ kayıtlarının QRCode resmi ile temsil edildikten sonra CNN mimarilerinin üstün yönlerinden faydalanılarak hibrit bir model önerisi sunulmuştur.

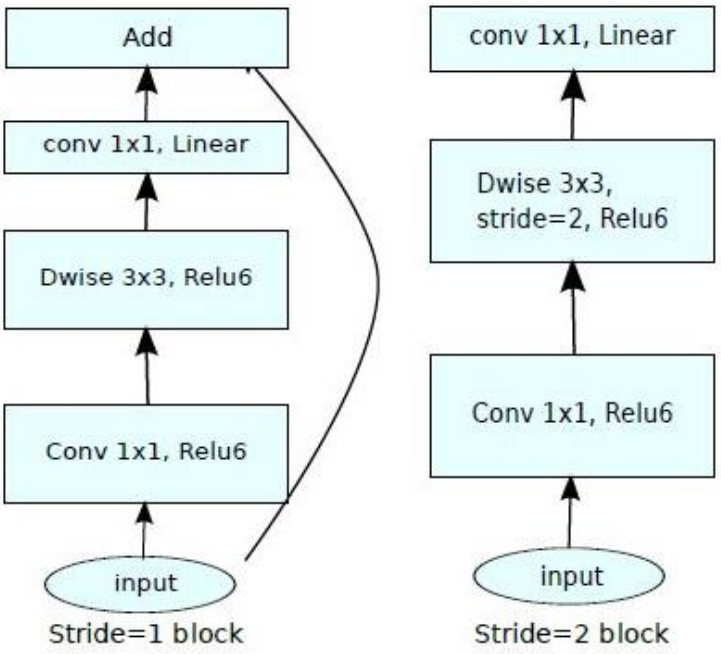
4.4.2. MobileNetV2 Algoritması Uygulanışı

MobileNetV2 algoritması Google çalışanları tarafından geliştirilmiştir[51]. Önceki sürüm olan MobileNetV1'de mobil cihazlar veya düşük hesaplama gücüne sahip tüm cihazlar için uygun olan ağın karmaşıklık maliyetini ve model boyutunu önemli ölçüde azaltan Derinlemesine Ayrılabilir Evrişim tanıtılmıştır[109]. MobileNetV2'de ters artık yapı ile daha iyi bir modül sunulmuştur. Dar katmanlardaki doğrusal olmayan durumları ortadan kaldırmıştır. Özellik çıkarma için omurga olarak MobileNetV2 ile nesne algılama ve semantik bölümleme için son teknoloji performanslar elde edilmiştir. Bu algoritma mobil cihazlar, düşük hesaplama maliyetine sahip olması ve iyi performansları elde edebildiği için bu çalışma kapsamında seçilmiştir.

MobileNetV2 ağı kendi içinde 16 blok katmanına sahiptir. Her katmandaki her blok aynı değildir ancak bloklarda Batch Normalization, Conv2D, Depthwise Conv2D, Zero Padding2D, Expansion Layer, Projection Layer gibi katmanlar bulunur. Katmanların ayrıca bir artık bağlantısı vardır ve her bloğa Darboğaz Artık Bloğu adı verilir[51].

MobileNetV2 ana blok yapısı aşağıdaki Şekil 4.9'da gösterilmektedir. Çoklu sınıflandırma için bu çalışmanda önerilen CNNQRLog yönteminin ağ mimarisi, bir daralma yolundan (sol taraf) ve bir sınıflandırıcı kafasından (sağ taraf) oluşur. Büzülme yolu, iki 3x3 evirişimin (padded konvolüsyonlar) tekrarlanan uygulamasını kullanarak tipik evrişimli bir ağın mimarisini takip eder; bunların her birini bir düzeltilmiş doğrusal birim (ReLU) ve aşağı örnekleme için adım 2 ile 2x2 maksimum havuzlama işlemi takip eder. "Blok" olarak adlandırılan bu üç işleme aşaması, birden çok kez tekrarlanır (ağı derinleştirir), sonuçta bir dizi tamamen bağlantılı katman (sınıflandırıcı aşaması) elde edilir. Evrişim katmanları, eğitim verimliliğini artırmak için tüm veri setine tekrar tekrar uygulanan bilgi işlem filtreleri ile her katmanda yerel ağırlıklı toplamlar ("özellik haritaları" olarak adlandırılır) elde eder. Doğrusal olmayan katmanlar daha sonra özellik haritalarının doğrusal olmayan özelliklerini artırır. Son olarak, havuzlama katmanı, ağın karmaşık özellikleri tanımlamak üzere yerel özellikleri toplamasını sağlamak için özellik haritalarında

örtüşmeyen bölgelerin alt örneklemesini gerçekleştirir. Her aşağı örnekleme adımında, özellik kanallarının sayısını iki katına çıkarırız. Maksimum havuzlama, düzeltilmiş özellik haritasından en büyük öğeyi alır. En büyük öğeyi almak, ortalama birleştirmeyi de alabilir.



Şekil 4.9. MobileNetV2 mimari yapısı[54].

4.4.2.1. Derinlemesine Ayrılabilir Konvolüsyonlar

Derinlemesine Ayrılabilir Evrişimler, birçok verimli sinir ağı mimarisi için önemli bir yapı taşıdır [109][110]. Temel fikir, tam evrişim operatörünü evrişimi iki ayrı katmana bölen çarpanlara ayrılmış bir sürümle değiştirmektir. İlk katman, derinlemesine evrişim olarak adlandırılır; giriş kanalı başına tek bir evrişimli filtre uygulayarak ışık filtrelemesi gerçekleştirir. İkinci katman, giriş kanallarının doğrusal bilgi işlem kombinasyonları aracılığıyla yeni özellikler oluşturmaktan sorumlu olan noktasal evrişim adı verilen 1 × 1 evrişimdir.

Standart evrişim, giriş tensörü olarak L_i'yi $h_i \; x \; w_i \; x \; d_i$ olarak alır ve çıkış tensörü L_j'yi üretmek için evrişim çekirdeğini $R^{k \; x \; k \; x \; d_i \; x \; d_j}$ olarak uygulamaktadır. Standart

evrişimli katmanların hesaplama maliyeti $h_i\ x\ w_i\ x\ d_i\ x\ k\ x\ k$ olarak hesaplanmaktadır.

Derinlemesine ayrılabilir evrişimler, standart evrişimli katmanların yerine geçmeli olarak kullanılır. Ampirik olarak, neredeyse düzenli kıvrımlar kadar iyi çalışırlar, ancak yalnızca derinlemesine ve 1 × 1 noktasal kıvrımların toplamı Eşitlik 4.14'teki kadar maliyeti vardır.

$$h_i\ x\ w_i\ x\ d_i(k^2 +\ d_j)$$

(4.14)

Etkili derinlemesine ayrılabilir evrişim, geleneksel katmanlara kıyasla hesaplamayı neredeyse k faktörü kadar azaltır. MobileNetV2, k = 3 (3 × 3 derinlemesine ayrılabilir evrişim) kullanır, bu nedenle hesaplama maliyeti, doğrulukta yalnızca küçük bir azalmayla standart evrişimlerden 8 ila 9 kat daha küçüktür[51].

4.4.2.2. Doğrusal Darboğazlar (Bottlenecks) Katmanı

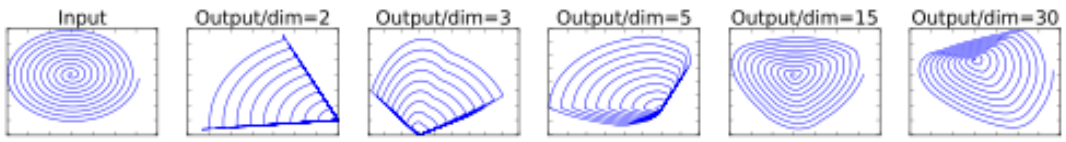
Derin Sinir Ağı her biri $h_i\ x\ w_i\ x\ d_i$ boyutlarında bir aktivasyon tensörüne sahip olan n adet Li katmanından oluşmaktadır. Bu katmanlar, $d_i$boyutlu $h_i\ x\ w_i$ "piksel" kapıları olarak bu aktivasyon tensörlerinin temel özelliklerini belirlemektedir. Gerçek görüntülerden oluşan bir girdi seti için, katman aktivasyonları setinin (herhangi bir Li katmanı için) bir "ilgili anakatmandan" oluşturulmaktadır. Sinir ağlarındaki ilgi anakatmanların düşük boyutlu alt uzaylara gömülebileceği uzun süredir varsayılmıştır. Başka bir deyişle, derin bir evrişimli katmanın tüm bireysel d-kanal piksellerine baktığımızda, bu değerlerde kodlanan bilgi, düşük boyutlu bir alt uzaya gömülebilen bir ana katman bulunmaktadır.

Bir katmanın boyutsallığını basit bir şekilde azaltarak ve böylece çalışma alanının boyutsallığını azaltarak kullanılmaktadır. Bu, MobileNetV1[109] tarafından bir genişlik çarpanı parametresi aracılığıyla hesaplama ve doğruluk arasında etkili bir şekilde değiş tokuş yapmak için başarılı bir şekilde diğer ağların verimli model tasarımlarına da dahil edilmiştir. Bu sezgiyi takiben, genişlik çarpanı yaklaşımı,

ilgilenilen ana katman tüm bu alanı kapsayana kadar, aktivasyon alanının boyutsallığının azaltılmasına izin vermektedir. Bununla birlikte, derin evrişimli sinir ağlarının aslında ReLU gibi koordinat başına doğrusal olmayan dönüşümlere sahip olduğunu hatırladığımızda bu sezgi bozulmaktadır. Örneğin, 1B uzayda bir çizgiye uygulanan ReLU, bir 'ışın' üretirken, R_n uzayında olduğu gibi, genellikle n eklemli parçalı doğrusal bir eğri ile sonuçlanmaktadır.

Genel olarak, bir ReLU(B_x) katman dönüşümünün sonucu sıfır olmayan bir S hacmine sahipse, iç S'ye eşlenen noktaların girdinin doğrusal bir B dönüşümü aracılığıyla elde edildiğini görmek kolaydır; tam boyutlu çıktıya karşılık gelen girdi alanı, doğrusal bir dönüşümle sınırlıdır. Başka bir deyişle, derin ağlar yalnızca çıktı alanının sıfır olmayan hacim kısmında doğrusal bir sınıflandırıcı gücüne sahiptir.

Öte yandan, ReLU kanalı çökerttiğinde, kaçınılmaz olarak o kanaldaki bilgileri kaybeder. Ancak, çok sayıda kanalımız varsa ve aktivasyon manifoldunda bir yapı varsa, bu bilgi diğer kanallarda hala korunabilir. Tamamlayıcı malzemelerde, giriş manifoldu aktivasyon alanının önemli ölçüde daha düşük boyutlu bir alt uzayına gömülebilirse, ReLU dönüşümünün gerekli karmaşıklığı ifade edilebilir fonksiyonlar kümesine dahil ederken bilgiyi koruduğunu gösteriyoruz. Şekil 4.10'da daha yüksek boyutlu uzaylara gömülü düşük boyutlu manifoldların ReLU dönüşümlerine örnekler gösterilmektedir. Bu örneklerde, ilk sarmal, rasgele matris T ve ardından ReLU kullanılarak n boyutlu bir uzaya gömülür ve ardından T −1 kullanılarak 2B uzaya geri yansıtılır. Yukarıdaki örneklerde n = 2; 3, manifoldun belirli noktalarının birbirine çöktüğü bilgi kaybına neden olurken, n = 15 ila 30 için dönüşüm oldukça dışbükey değildir.



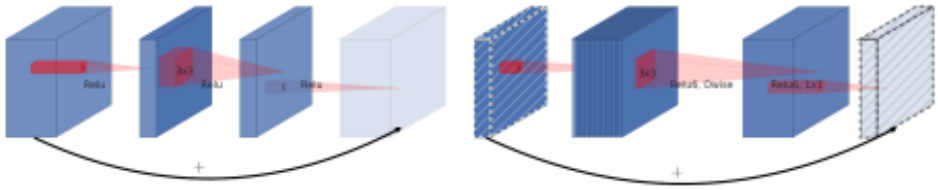
Şekil 4.10. Yüksek boyutlu uzaylara gömülü düşük boyutlu ana katmanların ReLU dönüşümüne örnekler.

Özetlemek gerekirse, ilgilenilen manifold yüksek boyutlu aktivasyon uzayının düşük boyutlu bir alt uzayında bulunması gerekliliğinin göstergesi olan iki özelliği vurgulamıştır.

- İlgili manifold, ReLU dönüşümünden sonra sıfır olmayan bir hacim olarak kalırsa, doğrusal bir dönüşüm karşılık gelmektedir.
- ReLU, yalnızca giriş manifoldu giriş uzayının düşük boyutlu bir alt uzayında bulunuyorsa, giriş manifoldu hakkındaki tüm bilgileri koruyabilmektedir.

4.4.2.3. Ters Artıklar (Inverted residuals) Katmanı

Darboğaz blokları, her bloğun bir giriş ve ardından birkaç darboğaz ve ardından genişleme içerdiği artık bloğa benzer görünmektedir. Bununla birlikte, darboğazların aslında tüm gerekli bilgileri içerdiği sezgisinden esinlenerek, bir genişletme katmanı yalnızca tensörün doğrusal olmayan dönüşümüne eşlik eden bir uygulama detayı görevi görürken, darboğazlar arasında doğrudan kısa yollar kullanılmaktadır.



Şekil 4.11. Artık blok ve ters artık arasındaki farkın gösterilmesi.

Şekil 4.11'de çapraz olarak taranmış katmanlar, doğrusal olmama durumlarını kullanmaz. Göreceli kanal sayısını belirtmek için her bloğun kalınlığını kullanılmıştır. Klasik artıkların çok sayıda kanala sahip katmanları birbirine bağladığına, tersine çevrilmiş artıkların ise darboğazları nasıl birleştirdiği gösterilmektedir.

Çizelge 4.5. s adımı ve t genişleme faktörü ile k'dan k¯kanallarına dönüşüm yapan darboğaz rezidüel blok.

Giriş	**Operatör**	**Çıkış**
h x w x k	1x1 conv2d, ReLU6	h x w x (tk)
h x w x tk	3x3 dwise s=s, ReLU6	h/s x w/s x (tk)
h/s x w/s x tk	Liner 1x1 conv2d	h/s x w/s x $\bar{k}$

Darboğaz evrişimi için çalışma süresi ve parametre sayısı Temel uygulama yapısı Çizelge 4.5'te gösterilmektedir. $\bar{d}$ giriş kanalları ve $\bar{\bar{d}}$ çıkış kanalları ile h × w, genişleme faktörü t ve çekirdek boyutu k olan bir blok için, gereken toplam çarpma sayısı h · w · $\bar{d}$ · t($\bar{d}$ + k + $\bar{\bar{d}}$) ile karşılaştırıldığında, bu ifadenin fazladan bir terimi vardır, çünkü aslında fazladan 1 × 1 evrişim vardır, ancak ağlarımızın doğası çok daha küçük girdi ve çıktı boyutları kullanmamıza izin verir.

4.4.2.4. Bilgi Akışı Yorumu (Information Flow Interpretation)

Mimariminin ilginç bir özelliği, yapı taşlarının girdi/çıktı alanları (darboğaz katmanları) ile girdiyi çıktıya dönüştüren doğrusal olmayan bir işlev olan katman dönüşümü arasında doğal bir ayrım sağlamasıdır. İlki, ağın her katmandaki kapasitesi olarak görülebilirken, ikincisi ifade gücü olarak görülebilmektedir. Bu, hem ifade gücü hem de kapasitenin birbirine karıştığı ve çıktı katmanı derinliğinin işlevleri olduğu, hem düzenli hem de ayrılabilir olan geleneksel evrişimli blokların tersidir.

Özellikle, iç katman derinliği 0 olduğunda, temel evrişim, kısayol bağlantısı sayesinde özdeşlik işlevidir. Genişleme oranı 1'den küçük olduğunda, bu klasik bir artık evrişim bloğudur. Bununla birlikte,1'den büyük genişleme oranının en yararlı olduğunu gösterülmektedir. Bu da, ağın ifade gücünü kapasitesinden ayrı olarak incelenemesine izin vermektedir. Ağ özelliklerinin daha iyi anlaşılmasını sağlamak için bu ayrımın daha fazla araştırılmasının garanti edilmektedir.

MobileNetV2'nin mimarisi, 32 filtreli ilk tam konvolüsyon katmanını, ardından Çizelge 4.6'da açıklanan 19 artık darboğaz katmanını içermektedir. Düşük hassasiyetli hesaplama ile kullanıldığında sağlamlığı nedeniyle doğrusal olmayan olarak ReLU6 kullanılmaktadır. Modern ağlar için standart olarak her zaman 3 × 3 çekirdek boyutunu kullanılmış, eğitim sırasında bırakma ve toplu normalleştirmeden yararlanılmıştır. Aynı sıradaki tüm katmanlar aynı sayıda c çıkış kanalına sahiptir. Her sekansın ilk katmanı bir adım s'ye sahiptir ve diğerleri adım 1'i kullanır.

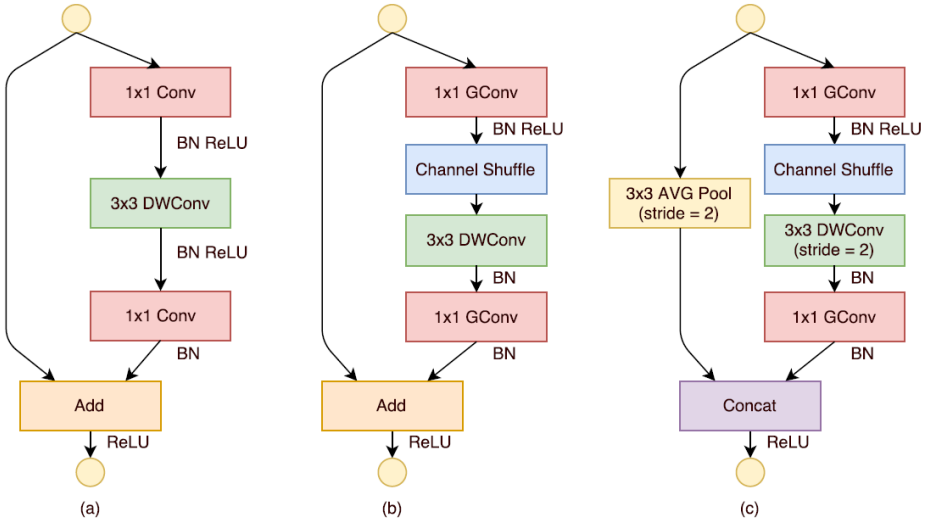
Çizelge 4.6. MobileNetV2 : Her satır, n kez tekrarlanan 1 veya daha fazla özdeş (modulo adım) katman dizisini tanımlar.

Giriş	**Operatör**	**t**	**c**	**n**	**s**
$224^2\ x\ 3$	Conv2d	-	32	1	2
$112^2\ x\ 32$	Bottleneck	1	16	1	1
$112^2\ x\ 16$	Bottleneck	6	24	2	2
$56^2\ x\ 24$	Bottleneck	6	32	3	2
$28^2\ x\ 32$	Bottleneck	6	64	4	2
$14^2\ x\ 64$	Bottleneck	6	96	3	1
$14^2\ x\ 96$	Bottleneck	6	160	3	2
$7^2\ x\ 160$	Bottleneck	6	320	1	1
$7^2\ x\ 320$	Conv2d 1x1	-	1280	1	1
$7^2\ x\ 1280$	Avgpool 7x7	-	-	1	-
$1\ x\ 1\ x\ 1280$	Conv2d 1x1	-	-	-	

4.4.3. SuffleNet Algoritması Uygulanışı

Bu çalışmada ShuffleNet CNN modeli düşük hesaplama maliyeti, ayarlanabilen parametreler sayesinde esnek olması ve karmaşıklığı azaltması nedeniyle seçilmiştir. ShuffleNet CNN modeli, üç katman halinde gruplandırılmış bir ShuffleNet birimleri yığınından oluşur[52]. Şekil 4.12'de (a) ile gösterilen kısım artık standart Darboğaz birimidir ancak derinlik kıvrımı olarak kullanılmaktadır. MobileNetV2'deki bir darboğaz türü, bu ünitede 1x1, 3x3 DW ve 1x1 evrişim olarak derinlemesine ayrılabilir evrişim olarak kullanılabilir. (b) ile gösterilen birim ShuffleNet birimidir.

Bu birimdeki birinci ve ikinci 1x1 kıvrımlar, grup kıvrımları ile dönüşümlü olabilir. İlk 1x1 evrişimden sonra, karşılaştırma olarak bir kanal uygulanır. (c) ile gösterilen kısım, Adım=2 ile ShuffleNet Hacmi olarak kullanılır. Bu bölüme 3x3 ortalama havuzlama eklenir. Bununla birlikte, eleman bazında ekleme, kanal birleştirme ile değiştirilir ve bu da kanal boyutunu çok az ekstra hesaplama maliyeti ile büyütmeyi kolaylaştırır. c x h x w girişi ve m darboğaz kanalları göz önüne alındığında, ShuffleNet yalnızca hw (2 cm) gerektirir. Başka bir deyişle, bir hesaplama bütçesi verildiğinde, ShuffleNet daha büyük özellik haritalarını kullanabilir. Küçük ağlar genellikle bilgileri işlemek için yetersiz sayıda kanala sahip olduğundan, bu küçük ağlar için çok önemlidir. Darboğaz kanalları, her ShuffleNet birimi için çıkış kanallarının 1/4'üne ayarlanmıştır. Kanal sayısına bir ölçek faktörü s uygulanır. Yukarıdaki Çizelgedaki ağlar "ShuffleNet 1^x" olarak gösterilir, ardından "ShuffleNet s^x", ShuffleNet 1^x'teki filtre sayısının s kez ölçeklendiği anlamına gelir, böylece toplam karmaşıklık kabaca s^2 katı olacaktır. ShuffleNet 1^x.



Şekil 4.12. ShuffleNet mimari yapısı[55].

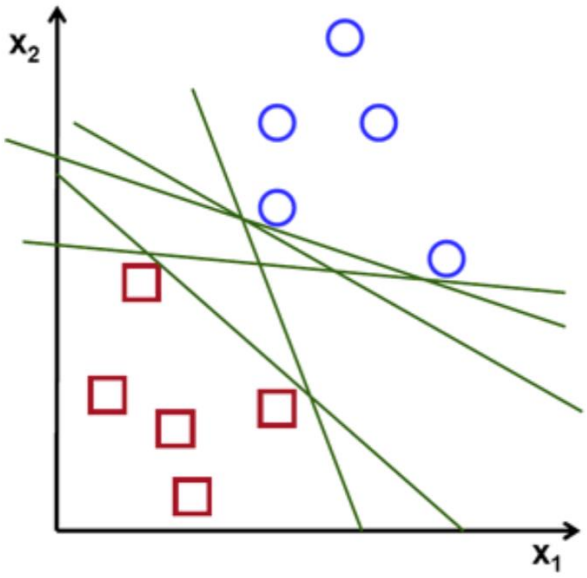
4.4.4. Sınıflandırma Algoritmaları

4.4.4.1. Destek Vektör Makinaları

Destek Vektör Makineleri, istatistiksel öğrenme teorisinin yapısal risk minimizasyon teorisinden türetilmiştir. DVM'nin temel fikri, giriş vektörlerini yüksek boyutlu bir özellik uzayına eşlemek ve en uygun ayırıcı hiperdüzlemi oluşturmaktır. DVM, ayırıcı hiper düzlem ile veri arasındaki marjı maksimize ederek genelleme hatasının üst sınırını en aza indirmeyi amaçlar.

Ek olarak, bir destek vektör makinesi, sınıflandırma, regresyon veya diğer görevler için kullanılabilen yüksek veya sonsuz boyutlu bir uzayda bir hiper düzlem veya hiper düzlemler kümesi oluşturmaktadır. Sezgisel olarak, herhangi bir sınıfın en yakın eğitim veri noktasına (işlevsel sınır adı verilir) en büyük mesafeye sahip olan hiperdüzlem tarafından iyi bir ayrım elde edilmektedir. Genel olarak, marj ne kadar büyük olursa, sınıflandırıcının genelleme hatası o kadar düşük olmaktadır.

Verileri sınıflandırmak, makine öğreniminde yaygın bir görevdir. Verilen bazı veri noktalarının her biri iki sınıftan birine aittir ve amaç, yeni bir veri noktasının hangi sınıfta olacağına karar vermektir. Destek vektör makinelerinde, bir veri noktası p-boyutlu olarak görülmektedir. Şekil 4.13'te vektör (p sayılarının bir listesi) ve bu tür noktaları bir (p -boyutlu hiperdüzlem) ile ayırıp ayıramayacağımızı göstermektedir. Buna lineer sınıflandırıcı denilmektedir ve birçok hiper düzlem, verileri sınıflandırabilmektedir. Hiperdüzlem, iki sınıf arasındaki en büyük ayrımı veya marjı temsil eden hiperdüzlem olduğundan, her iki taraftaki en yakın veri noktasına olan mesafe maksimum olacak şekilde hiperdüzlemi seçmek en iyisidir. Böyle bir hiper düzlem varsa, maksimum marj hiper düzlemi olarak bilinmektedir. Tanımladığı doğrusal sınıflandırıcı, maksimum marj sınıflandırıcısı veya eşdeğer olarak optimal kararlılık algısıdır.



Şekil 4.13. DVM makinaları hiperdüzlem seçimi.

Maliyet Fonksiyonu ve Gradyan Güncellemeleri

DVM algoritmasında, veri noktaları ile hiperdüzlem arasındaki marjı maksimize etmeye çalışılmaktadır. Kenar boşluğunu en üst düzeye çıkarmaya yardımcı olan kayıp işlevi hesaplamak için Eşitlik 4.15'te verilmiştir.

$$c(x, y, f(x)) = f(x) = \begin{cases} 0, & Eğer\ y \times f(x) \geq 1 \\ 1 - y \times f(x), & x < 1 \end{cases} \quad (4.15)$$

Öngörülen değer ile gerçekleşen değer aynı işaretli ise maliyet 0'dır. Değillerse, kayıp değerini hesaplamak gerekmektedir. Maliyet fonksiyonuna bir düzenlileştirme parametresi de eklenmektedir. Düzenleme parametresinin amacı, marj maksimizasyonunu ve kaybı dengelemektir. Normalleştirme parametresini ekledikten sonra maliyet fonksiyonları Eşitlik 4.16'da hesaplanmaktadır. Parametre nerede $\lambda > 0$ olduğunda marjin boyutu artırılması ile x_i kenar boşluğu doğru orantılı bir şekilde artmaktadır.

$$\lambda\|w\|^2 + \left[\frac{1}{n}\sum_{i=1}^{n} \max(0, 1 - y_i(w^t x_i - b))\right] \quad (4.16)$$

Yanlış sınıflandırma olmadığında model, veri noktasının sınıfını doğru bir şekilde tahmin eder; sadece normalizasyon parametresinden gradyanı güncellemek gerekmektedir. Bu güncelleme Eşitlik 4.17'de hesaplanmıştır.

$$w = w - \alpha \times (2\lambda w) \tag{4.17}$$

Bir yanlış sınıflandırma olduğunda, yani model, veri noktamızın sınıfının tahmininde hata yaptığında, gradyan güncellemesi gerçekleştirmek için düzenlileştirme parametresiyle birlikte kaybı da dahil etmek gerekmektedir. Bu güncelleme de Eşitlik 4.18'de hesaplanmıştır.

$$w = w + \alpha \times (y_i . x_i - 2\lambda w) \tag{4.18}$$

Doğrusal Olmayan Çekirdekler

1963'te Vapnik tarafından önerilen orijinal maksimum marjlı hiperdüzlem algoritmasında doğrusal bir sınıflandırıcı oluşturuldu. Ancak, 1992'de Bernhard Boser, Isabelle Guyon ve Vladimir Vapnik (başlangıçta Aizerman ve diğerleri [18] tarafından önerilen), çekirdek hilesini uygulayarak doğrusal olmayan maksimum marjlı hiperdüzlemlere sınıflandırıcılar üretmenin bir yolunu önerdiler[5]. Ortaya çıkan algoritma, doğrusal olmayan bir çekirdek işlevinin her bir iç çarpımı değiştirmesi dışında biçimsel olarak benzerdir. Bu, algoritmanın dönüştürülmüş bir versiyonunu vermektedir. Dönüşüm doğrusal olmayabilir ve dönüştürülmüş alan oldukça boyutlu olabilir; Sınıflandırıcı, dönüştürülmüş öznitelik uzayında bir hiper düzlem olmasına rağmen, orijinal girdi uzayında doğrusal olmayabilir.

Algoritma yeterince örnek verildiğinde hala iyi performans göstermesine rağmen, daha yüksek boyutlu bir öznitelik uzayında çalışmanın destek vektör makinelerinin genelleme hatasını arttırdığı dikkat çekicidir. Bazı yaygın çekirdekler polinom (homojen) Eşitlik 4.19'da, Polinom (homojen olmayan) Eşitlik 4.20'de, Gauss çekirdeği Eşitlik 4.21'de, sigmoid çekirdeği Eşitliği 4.22'de hesaplanmıştır.

$$k(x_i, x_j) = (x_i . x_j)^d \quad (4.19)$$

$$k(x_i, x_j) = (x_i . x_j + r)^d \quad (4.20)$$

$$k(x_i, x_j) = \exp(-\gamma \|x_i . x_j\|^2) \quad (4.21)$$

$$k(x_i, x_j) = \tanh(\zeta x_i . x_j + c) \quad (4.22)$$

4.4.4.2. K En Yakın Komşu

K-en yakın komşu (KNN) sınıflandırması temel ve basit bir sınıflandırma yöntemidir. K-en yakın komşu sınıflandırması, bilinmeyen veya belirlenmesi zor olasılık yoğunluklarını tahmin etmek için geliştirilmiştir.

Örüntü tanımada, k-En Yakın Komşular algoritması (veya kısaca KNN), sınıflandırma ve regresyon için kullanılan parametrik olmayan bir yöntemdir. KNN sınıflandırmasında çıktı bir sınıf üyeliğidir. Komşularının çoğunluk oyu bir nesneyi sınıflandırır ve nesneye en yakın k komşusu arasında en yaygın olan sınıf atanır (k pozitif bir tamsayıdır, k = 1 ise tipik olarak küçüktür), nesneye en yakın komşusunun sınıfı atanır. KNN, işlevin yalnızca yerel olarak tahmin edildiği ve tüm hesaplamaların sınıflandırmaya kadar ertelendiği, örneğe dayalı bir öğrenme veya tembel öğrenmedir. KNN algoritması, tüm makine öğrenimi algoritmalarının en basitlerinden biridir. Yakın komşuların ortalamaya uzaktakilerden daha fazla katkıda bulunması için komşuların katkılarını tartmak yararlı olabilmektedir. Örneğin, ortak bir ağırlık şeması, her bir komşuya 1/d'lik bir ağırlık verir; burada d, komşuya olan mesafedir. Komşular, sınıfı (KNN sınıflandırması için) veya nesne özellik değeri (KNN regresyonu için) bilinen nesnelerden alınır. Bu, açık bir eğitim adımı gerekmese de, algoritma için eğitim seti olarak düşünülebilir. KNN algoritmasının bir eksikliği, verilerin yerel doğasına duyarlı olmasıdır.

4.5. PERFORMANS DEĞERLENDİRME METRİKLERİ

Log kaydetme, düşük bellek koşulları veya bir dosyaya erişme girişimleri gibi yazılım davranışının sayısal ve metinsel verilerini toplamaktadır. Modern yazılım mühendisliğinde günlük anomalisi, üç nedenden dolayı hala zorlayıcıdır.

Bunun başlıca nedenleri;

- Logların büyük hacimli olması ve bundan dolayı manuel düzenli ifade oluşturma için büyük çaba sarf etmek gerekir,
- Yazılımın karmaşıklığından dolayı olay şablonlarının çeşitli olması,
- Yazılım güncellemelerinin sıklığı olmasından dolayı loglama ifadelerinin sık güncellenmesi şeklinde sıralanmaktadır.

Bu çalışmada saldırı tespit sistemleri için iki farklı yöntem önerilmiştir. Önerilen GLSTM yöntemde log kayıtlarının her bir satırı için şablonlar oluşturulmuş ve bu büyük hacimli loglar şablonlar vasıtasıyla boyutları küçültülerek düzenli bir hale getirilmiştir. Daha sonra Node2Vec algoritmasıyla bu şablonlar arasında ilişki kurularak embedding vektör oluşturulmuştur. LSTM ile model eğitilmiş ve yeni oluşturulan log şablonunda anomali tespiti yapılmıştır. Önerilen CNNQRLog yöntemde saldırı verileri QR kodlara dönüştürülmüş, hafif derin öğrenme algoritmaları kullanılarak sınıflandırma işlemi yapılmıştır. Daha sonra performans düzeyin en üst seviyeye çıkarmak için optimizasyon uygulanmış ve daha iyi sonuçlar elde edilmiştir.

4.5.1. Graf ve Derin Öğrenme Modellerinin Hiperparametre Değerleri

Bu çalışmada önerilen GLSTM yönteminde kullanılan LSTM ve Node2Vec graf algoritmasının hiperparametre değerleri Çizelge 4.7, ve Çizelge 4.8'de verilmiştir. CNNQRLog yöntemlerinin gerçekletirilmesinde kullanılan hiperparametreler bölüm 4.4.2'deki MobilNetV2 algoritmasının uygulanması kısmında ve bölüm 4.4.3'deki ShuffleNet algoritmasının ugulanışı kısmında bahsedilmiştir.

Çizelge 4.7 GLSTM modelinin hiperparatmetre değerleri.

GLSTM Modeli Eğitim Hiperparametreleri	**LSTM**
Eğitim verisi (Train Data) satır sayısı	7.822.940(%70)
Test verisi (Test Data) satır sayısı	3.352.688(%30)
Giriş katmanı (Input Layer) sayısı	128
Conv1D evrişim katmanı (Conv1D filters size) filtre sayısı	32
MaxPooling1D havuzlama katmanı sayısı	2
Her bir evrişim katmanı için aktivasyon fonksiyonu	ReLu,Sigmoid
Evrişim katmanı çekirdek sayısı	3
Her katmanın nöron sayısı	128,64,32, çıktı uzunluğu
Kayıp fonksiyonu	"binary crossentropy"
Optimize edici fonksiyon	"Adam"
Tur (Epochs) sayısı	30
batch_size değeri	64

Çizelge 4.8 Graf algoritmasının hiperparametreleri değerleri.

Graf Algoritmasının Hiperparametreleri	**Node2Vec**
Düğüm (Node) sayısı	Veri seti uzunluğu
Kenar (Edge) sayısı	2000
Rasgele yürüyüş (Randoms Walks) sayısı	20000
Maksimum yürüyüş uzunluğu	100
Kök düğüm(root node) başına rastgele yürüyüş sayısı (n)	10
Kaynak düğüme dönme olasılık değeri (p)	0.5
Kaynak düğümden uzaklaşma olasılık değeri (q)	2

4.5.2. Performans Metrik Değerlerinin Hesaplanması

Bu çalışmada önerilen yöntemlerin başarısını ölçmek için sırasıyla şu kriterler kullanılmıştır. Eşitlik 4.14 ve Eşitlik 4.15'te Doğruluk ve Kesinlik ölçümü yapılmıştır. Bu eşitliklerde DN doğru negatifler, DP doğru pozitifler, YN yanlış negatifler ve YP yanlış pozitifler olmak üzere parametreler kullanılmıştır. Eşitlik 4.16 ve Eşitlik 4.17'de Özgünlük ve Hassasiyet değerleri hesaplanmıştır. Doğruluk ve Kesinlik 'in kümülatif toplamından F-Skor Eşlik 4.18'da hesaplanmıştır.

$$\text{Doğruluk} = \frac{\text{DP+DN}}{\text{DP+DN+YP+YN}} \tag{4.14}$$

$$\text{Kesinlik} = \frac{\text{DP}}{\text{DP+YN}} \tag{4.15}$$

$$\text{Özgünlük} = \frac{\text{DN}}{\text{DN+YP}} \tag{4.16}$$

$$\text{Hassasiyet} = \frac{\text{DP}}{\text{DP+YP}} \tag{4.17}$$

$$\text{F} - \text{Skor} = \frac{2*\text{TP}}{2*\text{TP+FP+FN}} \tag{4.18}$$

BÖLÜM 5

DENEYSEL ÇALIŞMALAR

Bu çalışmada GLSTM ve CNNQRLog önerilen yöntemlerinin deneysel testleri yapılmıştır. Yapılan deneysel testler sonucunda siber saldırıların tespit edilmesinde yüksek doğrulukta sonuçlar elde edilmiştir. Deneysel testler yapılırken karmaşıklık matrisinden yararlanılmıştır. Karmaşıklık matrisi sonucunda her bir yönteme ait doğruluk, kesinlik, hassasiyet, özgünlük ve f-skor değerleri elde edilmiştir. Bu değerlerin hesaplanması için eşitlik denklemleri Bölüm 4'te hesaplanmıştır. Bu çalışmada iki farklı yöntem için iki farklı veri seti kullanılarak testler yapılmıştır. Bölüm 4'te veri setleri ile ilgili detaylı bilgi verilmiştir. Bu kısımda veri setlerinin deneylerde kullanıp şekillerine değinilecektir.

5.1. ÖNERİLEN GLSTM YÖNTEMİNİN PERFORMANS SONUÇLARI

5.1.1. HDFS Veri Setinin Uygulanışı

HDFS veri seti Bölüm 4'te bahsedilen ön işlem adımları yapıldıktan sonra graf verisine dönüştürülmüştür. Graf verisine dönüştürmek için Node2Vec algoritması kullanılmıştır. Graf verisi sonucunda her bir sınıfa ait vektör verileri elde edilmiştir. Bu verilerin sınıflandırma işlemi yapmak için LSTM algoritması kullanılmış ve bu veriler bu algoritmaya giriş verisi olarak verilmiştir.

HDFS verilerininın sınıflandırılması işlemlerinde deneysel çalışmalarının performans testi için karmaşıklık matrisinden yararlanılacaktır. Deneysel çalışmalarda veri setinin %70'I modelleri eğitmek için, %30'u test etmek için kullanılmıştır. Karmaşıklık matrisi ile modelin doğruluk, kesinlik, hassasiyet, özgünlük ve f-skor performans metrikleri hesaplanmıştır.

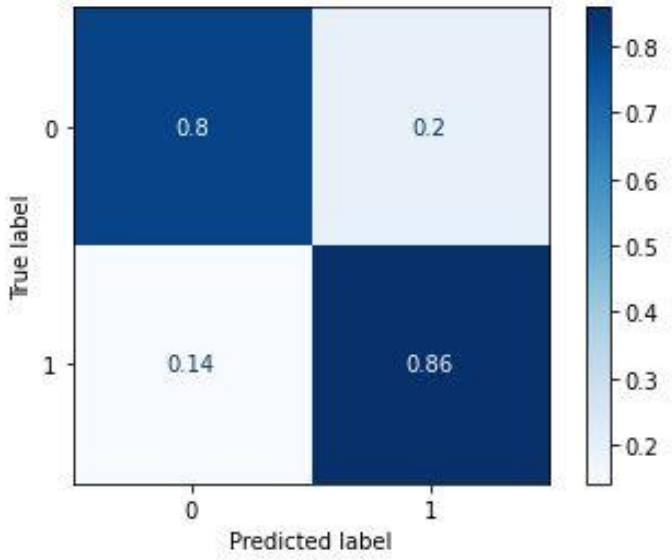
Deneysel çalışmalarda ilk aşamada büyük hacimli log verilerinin boyutlarının azaltmak ve düzenli hale getirmek için şablonlarda dönüştürülmüştür. Bu şablonların arasında ilişki kurmak ve derin öğrenme algoritmasıyla modelin eğitimini gerçekleştirmek için Node2Vec algoritması kullanılmıştır. Daha sonra anomali tespiti yapılması için LSTM algoritmasıyla model eğitimi yapılmıştır. Bu veri seti ile oluşturulan modelde LSTM giriş katmanı 128, gizli katmanı 64 ve çıkış katmanı normal ya da anormal sonucu elde etmek için 1 nörondan oluşmaktadır. Deneysel çalışma sonucunda önerilen model ile % 97.01 doğruluk oranı elde edilmiştir. Önerilen modelle ile elde edilen performans sonuçları Çizelge 5.1'de verilmiştir.

Model	**Doğruluk**	**Hassasiyet**	**Özgünlük**	**Kesinlik**	**F1_Skor**
Önerilen GLSTM Model	97.01	97.23	96.06	83.40	84.25

Çizelge 5.1. Önerilen GLSTM modelin performans sonuçları.

5.1.2. GLSTM Yönteminin Performans Metrik Sonuçları

Bu çalışmamızda yapılan testler sonucunda başarı durumunu gösteren Şekil 5.1'de Hata Matrisi(Confusion Matrix) verilmiştir. Bu grafikte gerçek ve tahmin edilen değerlerin etkinliği hesaplanmıştır. Burada önemli olan modelimizi eğitikten sonra çıkan tahmin değerlerin gerçek değerler ile karşılaştırılarak doğruluğu tespit edilmiştir. Bu grafik gerçekte olan anomalinin, model eğitildikten sonra bu anomalilerin ne kadarını tespit ettiğini göstermektedir. Böylelikle modelimizin yüksek bir başarı elde ettiğini bu grafik gösterilmektedir.



Şekil 5.1. Karmaşıklık matrisi sonuçları.

5.1.3. Eğri Grafikleri Sonuçları

Yapılan deneylerin sonucunu doğru bir şekilde ölçmek için faydalı iki araç olan AUC eğrileri kullanılmaktadır. Bu eğriler iki farklı hatayı ortadan kaldırmak için kullanılmaktadır. Bundan biri YP'lerdir. Bu hata olay yokken olay varmış gibi sonuç vermektedir. Diğeri YN'dir. Bu hata da olay varken olayı tespit etmemesinden dolayı hatalı sonuçlar üretmektedir. Bu iki hatadan dolayı yapılan deneylerin sonucu net bir şekilde anlaşılmamaktadır. Bunun önüne geçmek için AUC eğrileri kullanılmaktadır.

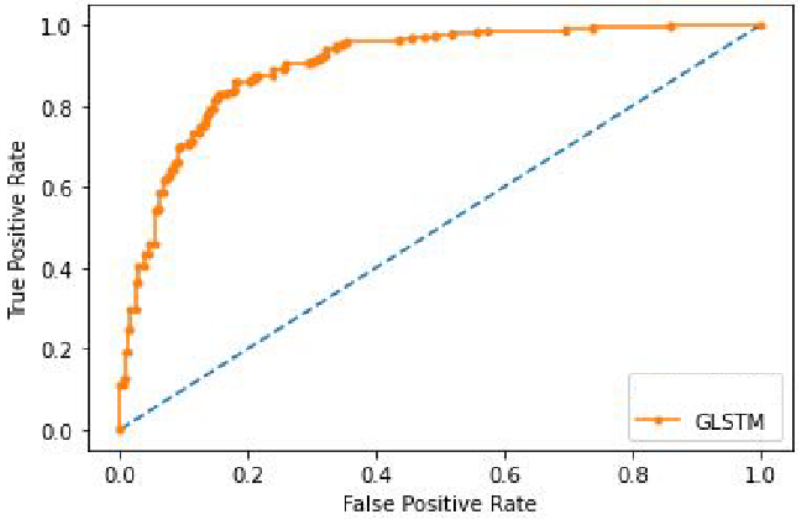
$$Gerçek\ Pozitif\ Oranı = \frac{|\mathbf{DP}|}{|\mathbf{YN}| + |\mathbf{DP}|} \tag{5.19}$$

$$Gerçek\ Negatif\ Oranı = \frac{|\mathbf{DN}|}{|\mathbf{YN}| + |\mathbf{DN}|}$$

(5.20)

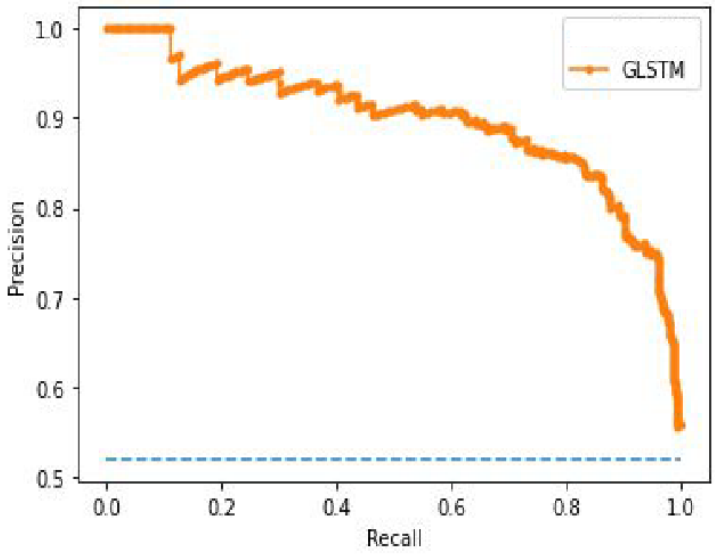
AUC eğrisinde iki önemli oran hesaplanmaktadır. Bunlardan biri Eşitlik 5.19'da gösterilen Gerçek Pozitif Oranı'dır. Diğeri de Eşitlik 5.20'de gösterilen Gerçek Negatif Oranı'dır. Şekil 5.2'de AUC Eğrisi'nin grafiği gösterilmiştir. Grafiğin x ekseninde daha küçük değerler yani daha düşük yanlış pozitifleri ve daha yüksek gerçek negatifleri göstermektedir. Grafiğin y ekseni de daha büyük değerler yani daha yüksek gerçek pozitifleri ve daha düşük yanlış negatifleri göstermektedir. Bu da

şunu göstermektedir iyi bir model grafikte kesikli çizgilerle gösterilen kısım yani eşik değeri 0,5'den daha yüksek bir değer göstermektedir. Bu da modelin iyi bir sonuç ortaya koyduğunu göstermektedir.



Şekil 5.2. AUC eğri grafiği.

Modelin doğru ölçen bir diğer grafik Kesinlik – Doğruluk grafiğidir. Bu eğrileri Hassas Geri Çağırma Eğrileri de denilmektedir. Kesinlik Eşitlik 5.19'da gösterildiği gibi modelin pozitif kısmın ne kadar iyi tahmin ettiğini göstermektedir. Doğruluk Eşitlik 5.20'de gösterilmektedir. Bu da gerçek pozitiflerin daha doğru tahmin edilmesini sağlamaktadır. Şekil 5.3'te Kesinlik – Doğruluk grafiği gösterilmektedir. Eğri altında kalan alanın integrali modelin ne kadar kesin ve doğru bir şekilde çalıştığını göstermektedir.



Şekil 5.3. Kesinlik - Doğruluk grafiği.

5.2. ÖNERİLEN CNNQRLOG YÖNTEMİNİN PERFORMANS SONUÇLARI

5.2.1. CSE-CIC-IDS2018 Veri Setinin Uygulanışı

CSE-CIC-IDS2018 veri seti Bölüm 4'te yapılan ön işlem adımları sonucunda deneysel testlerde kullanılmıştır. Öncelikle farklı tarihlere ait veri setleri bir araya getirilmiştir. Daha sonra veri setindeki eksik veriler sonucu etkilememesi için veri setinden çıkarılmıştır. Bu veri setinin sınıf alanını gösteren son sütunu haricindeki kısımların her bir satırı için QR kod görüntüleri oluşturulmuştur.

CSE-CIC-IDS2018 verilerinin sınıflandırılması işlemlerinde deneysel çalışmalarının performans testi için karmaşıklık matrisinden yararlanılacaktır. Deneysel çalışmalarda veri setinin %70'i (4200 QR kod görüntüleri) modelleri eğitmek için, %30'u (1800 QR kod görüntüleri) test etmek için kullanılmıştır. Confusion matrix ile modelin doğruluk, kesinlik, özgünlük, hassasiyet ve f-skor performans metrikleri hesaplanmıştır.

5.2.2. CNN Modelleri Deneysel Testleri

Deneysel çalışmanın ilk aşamasında iki farklı CNN modeli ile QR kod görüntüleri eğitilmiştir. CNN modellerinin hiper parametreleri seçilirken donanım özellikleri göz önünde bulundurulmuştur. Ağın her iterasyonda eğitmiş olduğu veri miktarı olan mini-batch size değeri 32 olarak belirlenmiştir. Tüm veri setinin bir sefer eğitilmiş olduğu epoch değeri ise 16 olarak belirlenmiştir. Veri setinde bulunan altı sınıflı QR koda görüntülerine dönüştürülmüş veri seti ilk olarak her iki CNN modeline doğrudan giriş olarak verilmiştir. Deneysel çalışma sonucunda MobileNetV2 CNN modeli ile % 90.28 doğruluk oranı elde edilirken ShuffleNet CNN modeli ile % 88.78 doğruluk oranı elde edilmiştir. MobileNetV2 modeli ile görüntüler 25.80 dakikada eğtirilirken ShuffleNet CNN modeli ile 126.16 dakikada eğitilmiştir. CNN modelleri ile elde edilen performans sonuçları Çizelge 5.2'te verilmiştir.

Çizelge 5.2. CNN modellerin performans sonuçları.

CNN Model	Doğruluk	Hassasiyet	Özgünlük	Kesinlik	F1_Skor	Zaman
MobileNetV2	90.28	90.28	98.06	90.40	90.25	25.80(dk)
ShuffleNet	88.78	88.78	97.76	89.05	88.82	126.16(dk)

5.2.3. CNN Modelleri ve Sınıflandırma Algoritmaları Deneysel Testleri

MobileNetV2 CNN modeli için Logits, ShuffleNet CNN modellinden ise node_202 katmanından her bir görüntüye ait 1000 derin özellik elde edilmiştir. Elde edilen bu özellikler daha sonra SVM ve kNN mamkine öğrenme yöntemlerine giriş olarak verilerek analiz sonuçları ayrı incelenemiştir. Deneysel çalışma sonucunda MobileNetV2 CNN modeli ve ShuffleNet CNN modellerinin 1000 özelliği ile en iyi sonuçlar SVM quadratic kernel çekirdeği ile sırasıyla %94.67 ve %89.56 doğruluk oranları elde edilmiştir. Elde edilen performans sonuçları Çizelge 5.3 ve Çizelge 5.4'te sırasıyla verilmiştir.

Çizelge 5.3. MobileNetV2 DVM ve KNN sınıflandırma algoritmaları deneysel test sonuçları.

Algoritma	Çekirdek & Uzaklık Foksiyonları	Doğruluk	Hassasiyet	Özgünlük	Kesinlik	F1_Skor
SVM	Linear	91.06	91.06	98.21	91.49	91.06
	Quadratic	**94.67**	**94.67**	**98.93**	**94.94**	**94.66**
kNN	Euclidean	92.11	92.11	98.42	92.52	92.08
	City Blok	92.00	92.00	98.40	92.37	91.96
	Minkowski	91.78	91.78	98.36	92.17	91.74

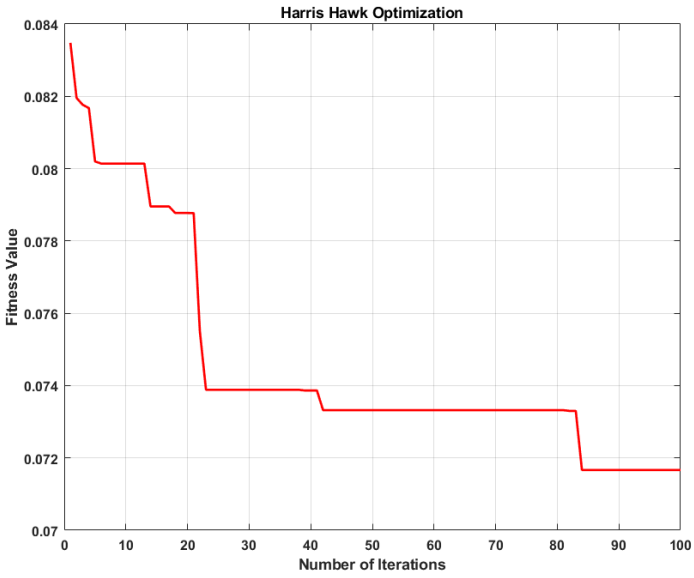
Çizelge 5.4. ShuffleNet ile DVM ve KNN sınıflandırma algoritmaları deneysel test sonuçları.

Algoritma	Çekirdek & Uzaklık Foksiyonları	Doğruluk	Hassasiyet	Özgünlük	Kesinlik	F1_Skor
SVM	Linear	83.80	83.78	96.76	83.99	83.69
	Quadratic	**89.56**	**89.56**	**97.91**	**89.73**	**89.52**
kNN	Euclidean	84.28	84.28	96.86	84.86	84.17
	City Blok	83.83	83.83	96.77	84.50	83.74
	Minkowski	85.06	84.96	97.01	85.75	84.95

5.2.4. Hibrit Hafif Derin Öğrenme Modelleri Deneysel Testleri

Deneysel çalışmanın üçüncü aşamasında MobileNetV2 CNN modelinin Logits katmanı ile ShuffleNet CNN modelinin node 202 katmanından elde edilen öznitelikler birleştirilmiştir. Böylece her bir QR kod görüntüsüne ait 2000 özellikten oluşan yeni bir özellik seti elde edilmiştir. Daha sonra HHO optimizasyon

algoritması ile öznitelik seçimi yapılmıştır. İterasyona göre değişen uygunluk değeri Şekil 5.4'te verilmiştir.



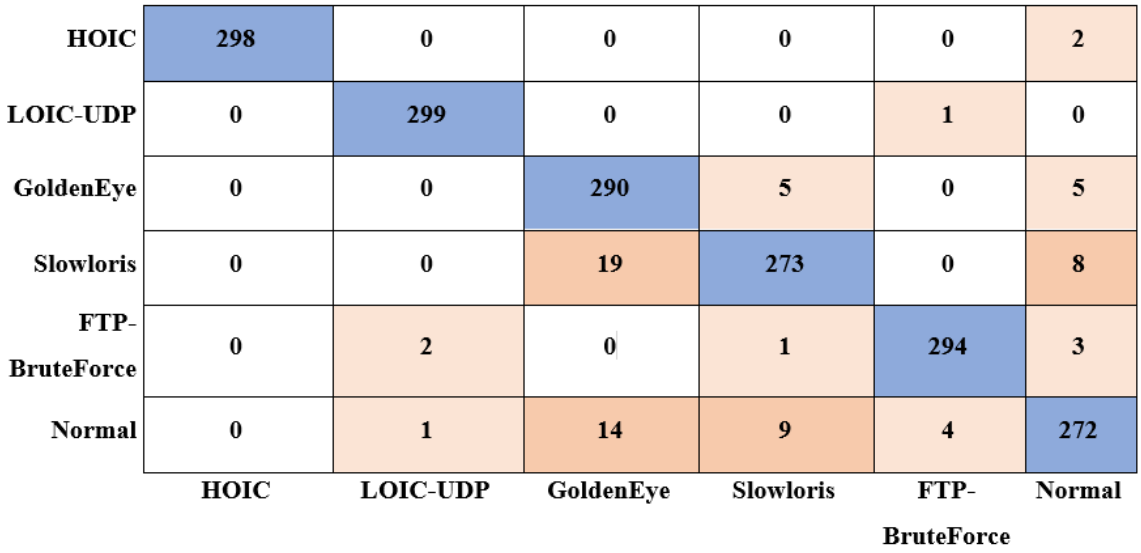
Şekil 5.4. İterasyona göre değişen fitness değeri.

En iyi fitness değeri belirlendikten sonra HHO optimizasyon algoritması ile 253 adet özellik seçilmiştir. Elde edilen yeni özellik seti ardından SVM ve kNN makine öğrenme algoritmaları ile test edilmiştir. Deneysel çalışma sonucunda en yüksek doğruluk oranı SVM algoritması ile % 95.89 doğruluk oranı elde edilmiştir. Önerilen modelin performans sonuçları Çizelge 5.5'te verilmiştir. En iyi sonuçların elde edildiği SVM qaudratic kernel'a ait confusion matrixi Şekil 5.5'te verilmiştir.

Çizelge 5.5. Hibrit olarak uygulanan hafif derin öğrenme algoritmalarının optimizasyon sonucu deneysel sonuçları.

Algoritma	**Çekirdek & Uzaklık Foksiyonları**	**Doğruluk**	**Hassasiyet**	**Özgünlük**	**Kesinlik**	**F1_Skor**
SVM	Linear	95.33	95.40	99.07	95.46	95.37
	Quadratic	**95.89**	**95.89**	**99.18**	**95.95**	**95.89**
kNN	Euclidean	91.44	91.44	98.29	91.72	91.42

	City Blok	9156	9156	9831	9188	9154
	Minkowski	91.39	91.39	98.28	91.67	91.36



Şekil 5.5. Hibrit modellerin HHO optimizasyon algoritması deneysel test sonuçlarında oluşan performans metriklerinin sonuçları.

5.3. MODELLERİN KARŞILAŞTIRILMASI

5.3.1. GLSTM Yönteminin Kıyaslanması

Bu çalışma, daha önce yapılmış çalışmalar ile karşılaştırmalar kullanılan metot, kullanılan veriseti ve doğruluk oranına göre yapılmıştır. Bu kriterler göz önünde bulundurularak bu çalışmanın diğer çalışmalar ile yapılan karşılaştırmaları Çizelge 5.6'da verilmiştir. LogAnomaly[18], önerdiğimiz çalışmayla aynı veriseti ve geliştirdiği metotta aynı derin öğrenme algoritması kullanılmaktadır. LogAnomaly ile log verisinde öncelikle eş anlamlı ve zıt anlımlı kelimeler Word2Vec ile tespit edilerek herbir log satırı için bir şablon oluşturulmuştur. Daha sonra bu şablonlar bir vektöre aktarılıp LSTM ile analiz edilmiştir. Log kayıtları hem sayısal hem de metinsel verilerden oluştuğu için LogAnomaly önemli zorluklarla karşılaşmaktadır. Önerdiğimiz metod hem sayısal hem de metinsel verileri grafa dönüştürdüğünden bu tür zorlukların üstesinden gelmiş ve nitekim LogAnomay'den daha iyi bir başarı

sonucu elde edilmiştir. DeepLog[22], doğal dil işleme yöntemiyle her bir log satırı için bir anahtar oluşturup ve bu anahtara karşılık gelen kelimelerle vektörel bir sonuç elde edilmiştir. Bu vektörelli LSTM kullanarak anomali tespiti yapılmıştır. Bu yöntem hacimli log verilerinde özellikle sayısal olan kısımların analiz etmesinde zorluklar yaşamaktadır. Önerdiğimiz yöntem tüm veri setinin herbir özelliğini kullanarak analiz yaptığından bu yöntem ile aynı veriseti ve algoritma kullanmasına rağmen daha başarılı sonuç elde etmiştir. Log anomali tespiti yapmak için önerdiğimiz çalışmayla benzer veriseti ile Bi-LSTM ve PCA algoritmaları[111] kullanımıştır. Bu çalışmayla veriseti öncelikle parçalara ayrılmış ve daha sonra şablonlar haline getirilmiştir. Daha sonra sayısallaştırma ve normalizasyon işlemleriyle vektörel hale dönüştürülmüştür. Önerdiğimiz yöntemle aynı veriseti kullanmalarına rağmen özellikle önerdiğimiz model, PCA ile elde edilen sonuçlardan daha başarılı sonuçlar elde edilmiştir. Sonuç olarak önerdiğimiz metod daha önce yapılmış birçok model daha başarılı sonuçlar ürettiği ve log anomalide daha etkin bir şekilde kullanılabileceğini gösterilmiştir.

Çizelge 5.6. Önerilen GLSTM yöntemin diğer modeller ile karşılaştırılması.

Yazarlar	**Metot**	**Veri seti**	**Doğruluk(%)**
2019, Weibin Meng et al. [112]	LSTM,Word2Vec	BGL,HDFS	96.00
2017,Min Du et al. [22]	LSTM,tamplate2Vec	BGL,HDFS	92.00
2022, Zhang Yue et al.[111]	Bi-LSTM,PCA	HDFS	95.60
2023, Önerilen GLSTM yöntemi	**LSTM,Node2Vec**	**HDFS**	**97.01**

5.3.2. CNNQRLog Yönteminin Kıyaslanması

Bu çalışmada önerilen CNNQRlog yöntemde, CSE-CIC-IDS2018 veri setinin her satırı için QR kod görüntüleri oluşturulmuş ve hafif derin öğrenme algoritmaları MobilNetV2 ve ShuffleNet CNN algoritmaları ile test edilmiştir. Test sonucunu optimize etmek için Bölüm 4'te bahsedilen HHO optimizasyon algoritması uygulanmış ve en iyi 253 özellik çıkarıltılmıştır. Sınıflandırma algoritmalarından SVM ve KNN kullanılmıştır.

Çizelge 5.7. Önerilen CNNQRlog yöntemin diğer modeller ile karşılaştırılması.

Çalışmalar	**Metot**	**Veriseti**	**Sınıf Sayısı**	**Doğruluk(%)**
2020,Farhan et al.[113]	Deep Neural Network (DNN)	CSE-CIC-IDS2018	7	90.25
2020,Farhan et al.[114]	Binary Particle Swarm Optimization (BPSO)	CSE-CIC-IDS2018	7	95.00
2021,Cil et al.[7]	Deep Neural Network (DNN)	CICDDoS2019	4	94.57
2021, Noever et al.[94]	MobileNetV2	UNSW-NB15,NSL-KDD datasets	2,9	56.00
2023, Önerilen CNNQRLog yöntemi	**MobilNetV2,ShuffleNet, HHO**	**CSE-CIC-IDS2018**	**6**	**95.89**

Önerilen CNNQRlog yöntemdeki çalışmamızla aynı veriseti ve benzer çalışmaların karşılaştırılması Çizelge 5.7'de verilmiştir. Öncelikle önerdiğimiz yöntem ile aynı dataset fakat farklı yöntemlerin kullanıldığı Deep Neural Network (DNN) [113] ve Binary Particle Swarm Optimization (BPSO) [114] incelendiğinde önişleme kısmında ham veride kullanılmayan bazı verilerin atılmış ve sayısallaştırma işlemi yapılmıştır. Bu da hacimli verilerde hız, performans ve doğruluğu etkilediği gibi aşırı öğrenmeye sebep olmaktadır. Önerdiğimiz çalışmada verileri QR kodlara dönüştürüdümüzden herhangi bir veri kayıbı olamamakla birlikte büyük hacimli verilerin hız,performans ve doğruluğu artırmak için optimizasyon ile en uygun öznitelik seçimi yapılmıştır. Önerdiğimiz model ile benzer veriseti kullanan ve belli bir saldırı çeşidi olan DoS/DDoS saldırılarını tespit eden Deep Neural Network (DNN) yöntemi önermişlerdir[7]. Öznitelik seçimi manuel olarak yapıldığından en uygun öznitelik seçimi beceri ve uzmanlık gerektirdiğinden önemli zorluklarla karşılaşılmaktadır. Önerdiğimiz çalışma en uygun öznitelik seçimini yapmak için optimizasyon algoritması kullanılarak yüksek doğrulukta ve performansta özniteliklerin seçilmesi sağlanmıştır. Önerdiğimiz çalışmaya benzer bir çalışmada verisetlerin gri tonlamalı resimleri elde edilmiş ve MobileNetV2 algoritmasıyla test

edilmiştir[94]. Gri tonlamalı verisetlerini elde etmek için veri önişleme kısmında one-hot-encoding işlemi yapılarak sayısallaştırma işlemi yapılmış ve resimleri oluşturulmuştur. İki sınıfı kullanılarak sınıflandırma yapıldığında %97 oranında başarı sağlanmış fakat dokuz sınıf kullanıldığında başarı oranı %56 düşmüştür. Bu da sınıf sayılarının artması veya farklı verisetlerinde doğruluk oranını yüksek oranda değişmesine sebeb olacaktır. Önerdiğimiz modelde sayısallaştırma işlemi yapılmadan ve yüksek doğruluk oranı elde etmek için en uygun öznitelikler optimizasyon algoritmasıyla seçilerek modelimiz eğitilmiştir. Sonuç olarak önerdiğimiz model literatürde benzer veriseti ve çalışmalarla karşılaştırıldığında daha performanslı ve yüksek doğrulukta çalıştığı görülmektedir.

BÖLÜM 6

BULGULAR VE TARTIŞMA

Bu çalışmamızın GLSTM yönteminde, özellikle bir çok çalışmada kullanılan doğal dil işleme tekniği yerine graf yapısı kullanılmıştır. Graf algoritmalarından Node2Vec kullanılmıştır. Bu algoritma Bölüm 3'te bahsedildiği gibi word2vec algoritmalarına alternatif olarak geliştirilmiştir. Yapılan bu çalışmamızda logların ayrıştırılması ve özellik çıkarma noktasında diğer algoritmalara göre daha iyi bir sonuç elde ettiğini yapılan testlerle gösterilmiştir.

Log anomali tespitinde derin öğrenme ağları modellerinin eğitilip yüksek başarı elde edilmesi için özellikle log ayrıştırma işlemi yapıldıktan sonra modele girdi olarak verilecek düzeye getirilmesi gerekmektedir. Bu çalışmada node2vec çıkış vektörünü LSTM'e giriş verisi olarak verilmiş ve %97,01 oranında başarı elde edilmiştir. DDİ tekniği kullanan metodlardan daha iyi bir sonuç elde edilmiştir.

Siber saldırılar, sunucu saldırılarını, ağ saldırılarını, web saldırılarını, mobil saldırıları ve masaüstü uygulamalarına yönelik saldırıları içerir. Bölüm 2'de bu tür saldırılarla ilgili çalışmalardan bahsedilmektedir. İlgili çalışmalar, Çizelge 6'da gösterilen çeşitli veri kümeleri, yöntemler ve doğruluk performans ölçütleri dikkate alınarak karşılaştırılmıştır. Önerilen çalışmanın CNNQRlog yöntemini ve önceki çalışmalar karşılaştırıldı, ancak siber saldırıların çeşitliliği ve yöntemlerin farklı olması nedeniyle kesin bir karşılaştırma yapılamamıştır. Çizelge 5.7, araştırmacıların siber saldırıları tespit ederken çoğunlukla herhangi bir optimizasyon işlemi yapmadan ve görüntü işleme tekniklerini kullanmadan veri setlerini doğrudan sınıflandırdıklarını açıkça göstermektedir.

Bu çalışmanın CNNQRlog yönteminde, siber saldırıları otomatik olarak tespit eden, her saldırı için QR kod görüntüleri üreten ve saldırı tespiti için en iyi özellikleri

seçmek üzere özellik seçimi için bir optimizasyon algoritması kullanan bir derin öğrenme modeli geliştirilmiştir. Model, girdi olarak aldığı QR kod görüntülerinden en iyi özellikleri seçerek maksimum doğruluk sağlamaktadır.

Geliştirilen yöntemleri test etmek için kullanılan saldırıların çeşitliliği ve veri setlerinin çeşitliliği nedeniyle, KDD Cup'99[115], Kyoto 2006+ [116], NSL-KDD[117], UNSW-NB15[118], CIC-IDS2017[119], CSE-CIC-IDS2018[113]. Böyle bir sistem geliştirmek için veri setlerinin hızlı bir şekilde analiz edilmesi ve her bir veri setine uygun olarak görüntü dosyalarına dönüştürülmesi gerekir[6]. NSL-KDD ve UNSW-NB15 veri setlerinin öznitelik sayılarına göre farklı matrislerden görüntüler oluşturarak çalışmalarında görüntü işleme teknikleri ve derin öğrenme yöntemlerini kullanmışlardır [14]. Her veri kümesinin özellikleri farklı olacağından bunları her veri kümesine otomatik olarak uygulamak kolay değildir. Önerdiğimiz yöntem herhangi bir veri kümesine uygulanabilir, çünkü her veri kümesinin öznitelik sayısına bakılmaksızın QR kodları otomatik olarak üretilebilir. Bu QR kodlu görüntüler ile görüntü işleme ve derin öğrenme teknikleri kullanılarak hızlı, etkili sonuçlar elde edilmektedir.

BÖLÜM 7

SONUÇLAR VE ÖNERİLER

Bu çalışmanın GLSTM yönteminde, kompleks ağlardaki farklı kaynaklardan elde edilen ve farklı özellikleri barındıran büyük miktardaki logların, tek tip haline getirilmesi ve bunlardan anomali tespiti yapılması amaçlanmıştır. Logların çok büyük ve farklı verilerden oluşması, bunlardan hızlı ve etkili anomali tespitinin yapılmasını oldukça zorlaştırmaktadır. Bunun için bu farklı log verilerinin, etkili ve hızlı bir şekilde işlenebilmesi için bu çalışmamızda, farklı yapılardaki loglar bir şablon haline getirilmiş daha sonra bu şablonlar arasındaki ilişkileri elde etmek için Graf yapısına dönüştürülmüştür. Graf dönüşümü için bir Graf algoritması olan Node2Vec kullanılmıştır. Bu dönüşümden log şablonlarının embbeding vektörü elde edilmiştir. Elde edilen bu vektör anomali etiketleri içeren veriler derin öğrenme algoritması için %70 eğitim ve %30 test verisi olarak ayrılmıştır. Bu veriler derin öğrenme metotlarından LSTM algoritması kullanılarak eğitilmiş ve test edilmiştir. Yapılan testler sonucunda önerdiğimiz Graf tabanlı LSTM modelimiz %97,01 doğruluk oranında başarılı sonuçlar elde edilmiştir.

Bu çalışmanın CNNQRLog yönteminde, siber saldırıları log kayıtlarının QR kodları oluşturulmuş ve bu QR kod görüntüleriyle makine öğrenmesi yöntemlerinde görüntü işleme teknikleri kullanılarak yüksek doğruluk ve performans elde edilmiştir. QR kod görüntüleri ilk aşamada MobileNetV2 ve ShuffleNet CNN modellerine doğrudan verilmiş, sırasıyla % 90.28 ve %88.78 doğruluk oranında başarı elde edilmiştir. İkinci aşamada QR kod görüntülerinin ayrı ayrı 1000 derin özellikler çıkartılmış, SVM ve kNN ile sınıflandırma işlemi ile sınıflandırma yapılandırma yapılarak, en iyi SVM quadratic kernel çekirdeği ile sırasıyla %94.67 ve %89.56 doğruluk oranları elde edilmiştir. Daha iyi sonuçlar elde etmek amacıyla üçücü aşamada 253 adet derin özellik seçilmiş, optimize etmek için HHO optimizasyon algoritması uygulanarak, SVM ve kNN sınıflandırma yapılmış %95,89 ile en yüksek başarı elde edilmiştir.

Önerilen model QR kod görüntülerinin MobileNetV2 ve ShuffleNet mimarilerine göre sırasıyla %5.49 ve %7.07 iyileştirme yapılmıştır. Literatüre DDOS saldırılarını tespit eden özgün bir hibrit model sunulmuştur.

Gelecekteki çalışmalar için, farklı veri setleri gri renkli görüntülere dönüştürülerek farklı CNN modelleri ve optimizasyon algoritmaları kullanılrak daha yüksek doğruluk oranına sahip bir model geliştirilmesi planlanmaktadır.

KAYNAKLAR

1. Oliveira, N., Praça, I., Maia, E., and Sousa, O., "Intelligent cyber attack detection and classification for network-based intrusion detection systems", ***Applied Sciences***, 11 (4): 1674 (2021).
2. Al-talak, K. and Abbass, O., "Detecting Server-Side Request Forgery (SSRF) Attack by using Deep Learning Techniques", ***International Journal Of Advanced Computer Science And Applications***, 12 (12): (2021).
3. Liu, L., Zhang, L., Liao, S., Liu, J., and Wang, Z., "A generalized approach to solve perfect Bayesian Nash equilibrium for practical network attack and defense", ***Information Sciences***, 577: 245–264 (2021).
4. Ravishankar, N., Raju, M. B., and Vyuha, N. C. S., "Secure Software Immune Receptors from SQL Injection and Cross Site Scripting Attacks in Content Delivery Network Web Applications", (2021).
5. Almaiah, M. A., Al-Zahrani, A., Almomani, O., and Alhwaitat, A. K., "Classification of cyber security threats on mobile devices and applications", Artificial Intelligence and Blockchain for Future Cybersecurity Applications, ***Springer***, 107–123 (2021).
6. Namukuwa, R., Chitauro, M., and Fungai, B. S., "A Desktop Review of Security Techniques Applicable for Optimised Water Network", (2021).
7. Cil, A. E., Yildiz, K., and Buldu, A., "Detection of DDoS attacks with feed forward based deep neural network model", ***Expert Systems With Applications***, 169: 114520 (2021).
8. Lipp, M., Kogler, A., Oswald, D., Schwarz, M., Easdon, C., Canella, C., and Gruss, D., "PLATYPUS: Software-based power side-channel attacks on x86", (2021).
9. Rodríguez, E., Otero, B., Gutiérrez, N., and Canal, R., "A survey of deep learning techniques for cybersecurity in mobile networks", ***IEEE Communications Surveys & Tutorials***, 23 (3): 1920–1955 (2021).
10. Park, J., Kim, J., Gupta, B. B., and Park, N., "Network log-based SSH brute-force attack detection model", ***CMC-Computers Materials & Continua***, 68 (1): 887–901 (2021).
11. Thakkar, A. and Lohiya, R., "A survey on intrusion detection system: feature selection, model, performance measures, application perspective, challenges, and future research directions", ***Artificial Intelligence Review***, 1–111 (2021).

12. Ahmad, Z., Shahid Khan, A., Wai Shiang, C., Abdullah, J., and Ahmad, F., "Network intrusion detection system: A systematic study of machine learning and deep learning approaches", ***Transactions On Emerging Telecommunications Technologies***, 32 (1): e4150 (2021).

13. Mendonça, R. v, Teodoro, A. A. M., Rosa, R. L., Saadi, M., Melgarejo, D. C., Nardelli, P. H. J., and Rodríguez, D. Z., "Intrusion detection system based on fast hierarchical deep convolutional neural network", ***IEEE Access***, 9: 61024–61034 (2021).

14. Toldinas, J., Venčkauskas, A., Damaševičius, R., Grigaliūnas, Š., Morkevičius, N., and Baranauskas, E., "A novel approach for network intrusion detection using multistage deep learning image recognition", ***Electronics***, 10 (15): 1854 (2021).

15. Pallavi, C., Girija, R., and Jayalakshmi, S. L., "An Analysis on Network Security Tools and Systems", (2021).

16. He, S., He, P., Chen, Z., Yang, T., Su, Y., and Lyu, M. R., "A Survey on Automated Log Analysis for Reliability Engineering", ***ArXiv Preprint ArXiv:2009.07237***, (2020).

17. Makanju, A. A. O., Zincir-Heywood, A. N., and Milios, E. E., "Clustering event logs using iterative partitioning", (2009).

18. Meng, W., Liu, Y., Zhu, Y., Zhang, S., Pei, D., Liu, Y., Chen, Y., Zhang, R., Tao, S., Sun, P., and Zhou, R., "Loganomaly: Unsupervised detection of sequential and quantitative anomalies in unstructured logs", ***IJCAI International Joint Conference On Artificial Intelligence***, 2019-Augus: 4739–4745 (2019).

19. Church, K. W., "Word2Vec", ***Natural Language Engineering***, 23 (1): 155–162 (2017).

20. Ahmed, M., Mahmood, A. N., and Hu, J., "A survey of network anomaly detection techniques", ***Journal Of Network And Computer Applications***, 60: 19–31 (2016).

21. Sigelman, B. H., Andr, L., Burrows, M., Stephenson, P., Plakal, M., Beaver, D., Jaspan, S., and Shanbhag, C., "Dapper , a Large-Scale Distributed Systems Tracing Infrastructure", ***Google Research***, (April): 14 (2010).

22. Du, M., Li, F., Zheng, G., and Srikumar, V., "DeepLog: Anomaly detection and diagnosis from system logs through deep learning", ***Proceedings Of The ACM Conference On Computer And Communications Security***, 1285–1298 (2017).

23. Li, H. and Li, Y., "LogSpy: System Log Anomaly Detection for Distributed Systems", ***Proceedings - 2020 International Conference On Artificial Intelligence And Computer Engineering, ICAICE 2020***, 347–352 (2020).

24. Xu, W., Huang, L., Fox, A., Patterson, D., and Jordan, M. I., "Detecting large-scale system problems by mining console logs", (2009).

25. Wang, M., Xu, L., and Guo, L., "Anomaly detection of system logs based on natural language processing and deep learning", ***2018 4th International Conference On Frontiers Of Signal Processing, ICFSP 2018***, 140–144 (2018).

26. Vaarandi, R., "A data clustering algorithm for mining patterns from event logs", (2003).

27. Studiawan, H., Payne, C., and Sohel, F., "Graph clustering and anomaly detection of access control log for forensic purposes", ***Digital Investigation***, 21: 76–87 (2017).

28. Khader, R. and Eleyan, D., "Survey of DoS/DDoS attacks in IoT", ***Sustainable Engineering And Innovation***, 3 (1): 23–28 (2021).

29. Swe, Y. M., Aung, P. P., and Hlaing, A. S., "A slow ddos attack detection mechanism using feature weighing and ranking", (2021).

30. Awan, M. J., Farooq, U., Babar, H. M. A., Yasin, A., Nobanee, H., Hussain, M., Hakeem, O., and Zain, A. M., "Real-time DDoS attack detection system using big data approach", ***Sustainability***, 13 (19): 10743 (2021).

31. Long, C., Xiao, J., Wei, J., Zhao, J., Wan, W., and Du, G., "Autoencoder ensembles for network intrusion detection", (2022).

32. Thavamani, S. and Sinthuja, U., "LSTM based Deep Learning Technique to Forecast Internet of Things Attacks in MQTT Protocol", (2022).

33. Fahrnberger, G., "Realtime Risk Monitoring of SSH Brute Force Attacks", (2022).

34. Panigrahi, R., Borah, S., Pramanik, M., Bhoi, A. K., Barsocchi, P., Nayak, S. R., and Alnumay, W., "Intrusion detection in cyber–physical environment using hybrid Naïve Bayes—Decision table and multi-objective evolutionary feature selection", ***Computer Communications***, 188: 133–144 (2022).

35. Wang, X., Wang, D., Zhang, Y., Jin, L., and Song, M., "Unsupervised learning for log data analysis based on behavior and attribute features", (2019).

36. Schindler, T., "Anomaly Detection in Log Data using Graph Databases and Machine Learning to Defend Advanced Persistent Threats", (2017).

37. Yan, X., Zhou, W., Gao, Y., Zhang, Z., Han, J., and Fu, G., "PADM: Page rank-based anomaly detection method of log sequences by graph computing", ***Proceedings Of The International Conference On Cloud Computing Technology And Science, CloudCom***, 2015-Febru (February): 700–703 (2015).

38. Kozik, R., Choraś, M., Ficco, M., and Palmieri, F., "A scalable distributed machine learning approach for attack detection in edge computing environments", ***Journal Of Parallel And Distributed Computing***, 119: 18–26 (2018).

39. Roopak, M., Tian, G. Y., and Chambers, J., "Multi- objective- based feature selection for DDoS attack detection in IoT networks", ***IET Networks***, 9 (3): 120–127 (2020).

40. Vidal, J. M., Orozco, A. L. S., and Villalba, L. J. G., "Adaptive artificial immune networks for mitigating DoS flooding attacks", ***Swarm And Evolutionary Computation***, 38: 94–108 (2018).

41. Zhao, H., Li, J., Nie, J., Ge, J., Yang, S., Yu, L., Pu, Y., and Wang, K., "Identification Method for Cone Yarn Based on the Improved Faster R-CNN Model", ***Processes***, 10 (4): 634 (2022).

42. Elbasani, E. and Kim, J.-D., "LLAD: Life-Log Anomaly Detection Based on Recurrent Neural Network LSTM", ***Journal Of Healthcare Engineering***, 2021: (2021).

43. Hamooni, H., Debnath, B., Xu, J., Zhang, H., Jiang, G., and Mueen, A., "Logmine: Fast pattern recognition for log analytics", (2016).

44. Omid, M., Firouz, M. S., Nouri-Ahmadabadi, H., and Mohtasebi, S. S., "Classification of peeled pistachio kernels using computer vision and color features", ***Engineering In Agriculture, Environment And Food***, 10 (4): 259–265 (2017).

45. Dai, H., Li, H., Chen, C. S., Shang, W., and Chen, T.-H., "Logram: Efficient log parsing using n-gram dictionaries", ***IEEE Transactions On Software Engineering***, (2020).

46. Singh, D., Taspinar, Y. S., Kursun, R., Cinar, I., Koklu, M., Ozkan, I. A., and Lee, H.-N., "Classification and Analysis of Pistachio Species with Pre-Trained Deep Learning Models", ***Electronics***, 11 (7): 981 (2022).

47. Sapegin, A., Jaeger, D., Cheng, F., and Meinel, C., "Towards a system for complex analysis of security events in large-scale networks", ***Computers \& Security***, 67: 16–34 (2017).

48. Ren, S., He, K., Girshick, R., and Sun, J., "Faster r-cnn: Towards real-time object detection with region proposal networks", ***Advances In Neural Information Processing Systems***, 28: (2015).

49. Dubey, S. R. and Jalal, A. S., "Species and variety detection of fruits and vegetables from images", ***International Journal Of Applied Pattern Recognition***, 1 (1): 108–126 (2013).

50. Shen, D., Wu, G., and Suk, H.-I., "Deep learning in medical image analysis", ***Annual Review Of Biomedical Engineering***, 19: 221 (2017).

51. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., and Chen, L.-C., "Mobilenetv2: Inverted residuals and linear bottlenecks", (2018).

52. Zhang, X., Zhou, X., Lin, M., and Sun, J., "Shufflenet: An extremely efficient convolutional neural network for mobile devices", (2018).

53. Karadal, C. H., Kaya, M. C., Tuncer, T., Dogan, S., and Acharya, U. R., "Automated classification of remote sensing images using multileveled MobileNetV2 and DWT techniques", ***Expert Systems With Applications***, 185: 115659 (2021).

54. AlOwais, A., Naseem, S., Dawdi, T., Abdisalam, M., Elkalyoubi, Y., Adwan, A., Hassan, K., and Fernini, I., "Meteorite hunting using deep learning and UAVs", (2019).

55. Alsayed, A., Alsabei, A., and Arif, M., "Classification of apple tree leaves diseases using deep learning methods", ***International Journal Of Computer Science & Network Security***, 21 (7): 324–330 (2021).

56. Patel, R. and Chaware, A., "Transfer learning with fine-tuned MobileNetV2 for diabetic retinopathy", (2020).

57. Topalli, M. T., Yilmaz, M., and Çorapsiz, M. F., "Real Time Implementation of Drone Detection using TensorFlow and MobileNetV2-SSD", (2021).

58. Noever, D. A. and Noever, S. E. M., "Deep Learning Classification Methods Applied to Tabular Cybersecurity Benchmarks", ***International Journal Of Network Security & Its Applications (IJNSA) Vol***, 13: (2021).

59. Zhang, W., Luktarhan, N., Ding, C., and Lu, B., .

60. Yin, J., Guo, L., Jiang, W., Hong, S., and Yang, J., "ShuffleNet-inspired lightweight neural network design for automatic modulation classification methods in ubiquitous IoT cyber–physical systems", ***Computer Communications***, 176: 249–257 (2021).

61. Duan, X., Ying, S., Cheng, H., Yuan, W., and Yin, X., "OILog: An online incremental log keyword extraction approach based on MDP-LSTM neural network", ***Information Systems***, 95: 101618 (2021).

62. Sun, K., Qiu, W., Dong, Y., Zhang, C., Yin, H., Yao, W., and Liu, Y., "WAMS-based HVDC Damping Control for Cyber Attack Defense", ***IEEE Transactions On Power Systems***, (2022).

63. Stiawan, D., Idris, M., Malik, R. F., Nurmaini, S., Alsharif, N., and Budiarto, R., "Investigating brute force attack patterns in IoT network", ***Journal Of Electrical And Computer Engineering***, 2019: (2019).

64. Dankwa, S. and Yang, L., "An efficient and accurate depth-wise separable convolutional neural network for cybersecurity vulnerability assessment based on CAPTCHA breaking", ***Electronics***, 10 (4): 480 (2021).

65. Schatz, D., Bashroush, R., and Wall, J., "Towards a more representative definition of cyber security", ***Journal Of Digital Forensics, Security And Law***, 12 (2): 8 (2017).

66. Kianpour, M., Kowalski, S. J., and Øverby, H., "Systematically Understanding Cybersecurity Economics: A Survey", ***Sustainability***, 13 (24): 13677 (2021).

67. Duncan, A. J., Creese, S., and Goldsmith, M., "Insider attacks in cloud computing", (2012).

68. Alazab, M. and Tang, M., "Deep Learning Applications for Cyber Security", ***Springer***, (2019).

69. Disterer, G., "ISO/IEC 27000, 27001 and 27002 for information security management", ***Journal Of Information Security***, 4 (2): (2013).

70. Whitman, M. E. and Mattord, H. J., "Principles of Information Security", ***Cengage Learning***, (2021).

71. Kuhl, M. E., Sudit, M., Kistner, J., and Costantini, K., "Cyber attack modeling and simulation for network security analysis", (2007).

72. Soceanu, A., Vasylenko, M., and Gradinaru, A., "Improving cybersecurity skills using network security virtual labs", (2017).

73. Dong, S., Abbas, K., and Jain, R., "A survey on distributed denial of service (DDoS) attacks in SDN and cloud computing environments", ***IEEE Access***, 7: 80813–80828 (2019).

74. Sadeghian, A., Zamani, M., and Abdullah, S. M., "A taxonomy of SQL injection attacks", (2013).

75. Najafabadi, M. M., Khoshgoftaar, T. M., Kemp, C., Seliya, N., and Zuech, R., "Machine learning for detecting brute force attacks at the network level", (2014).

76. Gupta, S. and Gupta, B. B., "Cross-Site Scripting (XSS) attacks and defense mechanisms: classification and state-of-the-art", ***International Journal Of System Assurance Engineering And Management***, 8 (1): 512–530 (2017).

77. Ganesan, A., Parameshwarappa, P., Peshave, A., Chen, Z., and Oates, T., "Extending Signature-based Intrusion Detection Systems WithBayesian Abductive Reasoning", ***ArXiv Preprint ArXiv:1903.12101***, (2019).

78. Aldweesh, A., Derhab, A., and Emam, A. Z., "Deep learning approaches for anomaly-based intrusion detection systems: A survey, taxonomy, and open issues", ***Knowledge-Based Systems***, 189: 105124 (2020).

79. Hamada, Y., Inoue, M., Ueda, H., Miyashita, Y., and Hata, Y., "Anomaly-based intrusion detection using the density estimation of reception cycle periods for in-vehicle networks", ***SAE International Journal Of Transportation Cybersecurity And Privacy***, 1 (11-01-01–0003): 39–56 (2018).

80. Grover, A. and Leskovec, J., "Node2vec: Scalable Feature Learning for Networks", .

81. Internet: CSIRO's Data61, "StellarGraph Machine Learning Library", **https://github.com/stellargraph/stellargraph** .

82. Rong, X., "Word2vec Parameter Learning Explained", 1–21 (2014).

83. Demeester, T., Rocktäschel, T., and Riedel, S., "Lifted rule injection for relation embeddings", ***EMNLP 2016 - Conference On Empirical Methods In Natural Language Processing, Proceedings***, 1389–1399 (2016).

84. Kipf, T. N. and Welling, M., "SEMI-SUPERVISED CLASSIFICATION WITH GRAPH CONVOLUTIONAL NETWORKS", .

85. Specht, D. F., "Probabilistic neural networks", ***Neural Networks***, 3 (1): 109–118 (1990).

86. Werbos, P. J., "Generalization of backpropagation with application to a recurrent gas market model", ***Neural Networks***, 1 (4): 339–356 (1988).

87. Rodriguez, P., Wiles, J., and Elman, J. L., "A recurrent neural network that learns to count", ***Connection Science***, 11 (1): 5–40 (1999).

88. Hochreiter, S. and Schmidhuber, J., "Long Short-Term Memory", ***Neural Computation***, 9 (8): 1735–1780 (1997).

89. Liu, Q., Zhang, N., Yang, W., Wang, S., Cui, Z., Chen, X., and Chen, L., "A review of image recognition with deep convolutional neural network", (2017).

90. Feurer, M. and Hutter, F., "Hyperparameter optimization", Automated Machine Learning, ***Springer, Cham***, 3–33 (2019).

91. Hluchyj, M. G. and Karol, M. J., "Shuffle net: An application of generalized perfect shuffles to multihop lightwave networks", ***Journal Of Lightwave Technology***, 9 (10): 1386–1397 (1991).

92. Ma, H., Liu, Y., Ren, Y., and Yu, J., "Detection of collapsed buildings in post-earthquake remote sensing images based on the improved YOLOv3", ***Remote Sensing***, 12 (1): 44 (2019).

93. Chen, S., Wang, W., and van Zuylen, H., "Construct support vector machine ensemble to detect traffic incident", ***Expert Systems With Applications***, 36 (8): 10976–10986 (2009).

94. Noever, D. A. and Noever, S. E. M., "Image Classifiers for Network Intrusions", ***ArXiv Preprint ArXiv:2103.07765***, (2021).

95. Heidari, A. A., Mirjalili, S., Faris, H., Aljarah, I., Mafarja, M., and Chen, H., "Harris hawks optimization: Algorithm and applications", ***Future Generation Computer Systems***, 97: 849–872 (2019).

96. Gupta, S., Deep, K., Heidari, A. A., Moayedi, H., and Wang, M., "Opposition-based learning Harris hawks optimization with advanced transition rules: Principles and analysis", ***Expert Systems With Applications***, 158: 113510 (2020).

97. Hearst, M. A., Dumais, S. T., Osuna, E., Platt, J., and Scholkopf, B., "Support vector machines", ***IEEE Intelligent Systems And Their Applications***, 13 (4): 18–28 (1998).

98. Steinwart, I. and Christmann, A., "Support Vector Machines", ***Springer Science & Business Media***, (2008).

99. Moghaddam, V. H. and Hamidzadeh, J., "New Hermite orthogonal polynomial kernel and combined kernels in support vector machine classifier", ***Pattern Recognition***, 60: 921–935 (2016).

100. Guo, G., Wang, H., Bell, D., Bi, Y., and Greer, K., "KNN model-based approach in classification", (2003).

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

102. Internet: University of New Brunswick, "A Realistic Cyber Defense Dataset (CSE-CIC-IDS2018)", **https://www.unb.ca/cic/datasets/ids-2018.html** .

103. Ahmed, M., Mahmood, A. N., and Islam, M. R., "A survey of anomaly detection techniques in financial domain", ***Future Generation Computer Systems***, 55: 278–288 (2016).

104. Gogoi, P., Bhattacharyya, D. K., Borah, B., and Kalita, J. K., "A survey of outlier detection methods in network anomaly identification", ***The Computer Journal***, 54 (4): 570–588 (2011).

105. Saurabh, P. and Verma, B., "An efficient proactive artificial immune system based anomaly detection and prevention system", ***Expert Systems With Applications***, 60: 311–320 (2016).

106. Jia, T., Yang, L., Chen, P., Li, Y., Meng, F., and Xu, J., "Logsed: Anomaly diagnosis through mining time-weighted control flow graph in logs", (2017).

107. Zhou, Y., Cheng, G., Jiang, S., Zhao, Y., and Chen, Z., "Cost-effective moving target defense against DDoS attacks using trilateral game and multi-

objective Markov decision processes", ***Computers & Security***, 97: 101976 (2020).

108. Internet: KHARISMADHANY, E., "IDS 2018 Intrusion CSVs (CSE-CIC-IDS2018)", **https://www.kaggle.com/code/ekkykharismadhany/dataset-checking/data** .

109. Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., and Adam, H., "Mobilenets: Efficient convolutional neural networks for mobile vision applications", ***ArXiv Preprint ArXiv:1704.04861***, (2017).

110. Chollet, F., "Xception: Deep learning with depthwise separable convolutions", (2017).

111. Zhang, Y., Zhang, D., Guo, F., Wang, X., Duan, Y., and Zhang, X., "Log Anomaly Detection Based on Bi-LSTM Feature Extraction", (2022).

112. Meng, W., Liu, Y., Zhu, Y., Zhang, S., Pei, D., Liu, Y., Chen, Y., Zhang, R., Tao, S., Sun, P., and Zhou, R., "Loganomaly: Unsupervised detection of sequential and quantitative anomalies in unstructured logs", ***IJCAI International Joint Conference On Artificial Intelligence***, 2019-Augus (August): 4739–4745 (2019).

113. Farhan, R. I., Maolood, A. T., and Hassan, N., "Performance analysis of flow-based attacks detection on CSE-CIC-IDS2018 dataset using deep learning", ***Indones. J. Electr. Eng. Comput. Sci***, 20 (3): 1413–1418 (2020).

114. Farhan, R. I., Maolood, A. T., and Hassan, N. F., "Optimized Deep Learning with Binary PSO for Intrusion Detection on CSE-CIC-IDS2018 Dataset", ***Journal Of Al-Qadisiyah For Computer Science And Mathematics***, 12 (3): Page-16 (2020).

115. Bay, S. D., "The uci kdd archive", ***Http://Kdd. Ics. Uci. Edu***, (1999).

116. Song, J., Takakura, H., Okabe, Y., Eto, M., Inoue, D., and Nakao, K., "Statistical analysis of honeypot data and building of Kyoto 2006+ dataset for NIDS evaluation", (2011).

117. Tavallaee, M., Bagheri, E., Lu, W., and Ghorbani, A. A., "A detailed analysis of the KDD CUP 99 data set", (2009).

118. Moustafa, N. and Slay, J., "UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)", (2015).

119. Sharafaldin, I., Lashkari, A. H., and Ghorbani, A. A., "Toward generating a new intrusion detection dataset and intrusion traffic characterization.", ***ICISSp***, 1: 108–116 (2018).

ÖZGEÇMİŞ

Yusuf ALACA 2009 yılında Erciyes Üniversitesi Mühendislik Fakültesi Bilgisayar Mühendisliği Bölümü'nde öğrenime başlayıp 2013 yılında mezun oldu. 2014 yılında Karabük Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Anabilim Dalında yüksek lisans eğitimine başladı ve 2017 yılının ortalarında tamamladı. 2014 yılının başında Zonguldak Belediyesi'nde Bilgisayar Mühendisi olarak göreve başladı ve 2017 yılında aynı kurumda Bilgi İşlem Müdürü olarak görevine devam etti. 2018 yılında Karabük Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Anabilim Dalında doktora eğitimine başladı. 2020 yılının başında Hitit Üniversitesi Osmancık Ömer Derindere MYO öğretim görevlisi olarak göreve başladı. Halen bu kurumda öğretim görevlisi müdür yardımcısı olarak çalışmaya devam etmektedir.