

**T.C.
ONDOKUZ MAYIS ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ ANA BİLİM DALI**



**UYGULAMA PROGRAMLAMA ARAYÜZÜ ÇAĞRILARI
KULLANARAK SİSTEM GÜVENLİĞİNİN ANALİZ
EDİLMESİ**

Yüksek Lisans Tezi

Varol ŞAHİN

Danışman

Prof. Dr. Sedat AKLEYLEK

SAMSUN
2024

TEZ KABUL VE ONAYI

Varol ŞAHİN tarafından, Prof. Dr. Sedat AKLEYLEK danışmanlığında hazırlanan “Uygulama Programlama Arayüzü Çağrılarını Kullanarak Sistem Güvenliğinin Analiz Edilmesi” başlıklı bu çalışma, jürimiz tarafından 30.07.2024 tarihinde yapılan sınav sonucunda oy birliği ile başarılı bulunarak Yüksek Lisans Tezi olarak kabul edilmiştir.

	Unvanı Adı Soyadı Üniversitesi Ana Bilim/Ana Sanat Dalı	Sonuç
Başkan	Prof. Dr. Sedat AKLEYLEK İstinye Üniversitesi Yazılım Mühendisliği Anabilim Dalı	☒ Kabul ☐ Ret
Üye	Dr. Öğr. Üyesi Meryem SOYSALDI ŞAHİN Ondokuz Mayıs Üniversitesi Bilgisayar Yazılımı Ana Bilim Dalı	☒ Kabul ☐ Ret
Üye	Dr. Öğr. Üyesi Özgür TONKAL Samsun Üniversitesi Yazılım Mühendisliği Anabilim Dalı	☒ Kabul ☐ Ret

Bu tez, Enstitü Yönetim Kurulunca belirlenen ve yukarıda adları yazılı jüri üyeleri tarafından uygun görülmüştür.

Prof. Dr. Ahmet TABAK
Enstitü Müdürü

BİLİMSEL ETİĞE UYGUNLUK BEYANI

Hazırladığım Yüksek Lisans tezinin bütün aşamalarında bilimsel etiğe ve akademik kurallara riayet ettiğimi, çalışmada doğrudan veya dolaylı olarak kullandığım her alıntıya kaynak gösterdiğimi ve yararlandığım eserlerin Kaynaklar'da gösterilenlerden oluştuğunu, her unsurun enstitü yazım kılavuzuna uygun yazıldığını ve TÜBİTAK Araştırma ve Yayın Etiği Kurulu Yönetmeliği'nin 3. bölüm 9. maddesinde belirtilen durumlara aykırı davranılmadığını taahhüt ve beyan ederim.

Etik Kurul Gerekli mi ?

Evet ☐ (Gerekli ise ekler kısmına ekleyiniz)

Hayır ☒

20/06/2024
Varol ŞAHİN

TEZ ÇALIŞMASI ÖZGÜNLÜK RAPORU BEYANI

Tez Başlığı: UYGULAMA PROGRAMLAMA ARAYÜZÜ ÇAĞRILARI
KULLANARAK SİSTEM GÜVENLİĞİNİN ANALİZ EDİLMESİ

Yukarıda başlığı belirtilen tez çalışması için şahsım tarafından 20/06/2024 tarihinde intihal tespit programından alınmış olan özgünlük raporu sonucunda;

Benzerlik oranı : % 13.0

Tek kaynak oranı : % 1.0 çıkmıştır.

20/06/2024
Prof. Dr. Sedat AKLEYLEK

ÖZET

UYGULAMA PROGRAMLAMA ARAYÜZÜ ÇAĞRILARI KULLANARAK SİSTEM GÜVENLİĞİNİN ANALİZ EDİLMESİ

Varol ŞAHİN

Ondokuz Mayıs Üniversitesi

Lisansüstü Eğitim Enstitüsü

BİLGİSAYAR MÜHENDİSLİĞİ ANA BİLİM DALI

Yüksek Lisans, Temmuz/2024

Danışman: Prof. Dr. Sedat AKLEYLEK

Bu tez, uygulama programlama arayüzü (API) çağrılarını kullanarak makine öğrenimi (ML) yöntemleri ile anomali tespitine odaklanmaktadır. API/sistem çağrıları, sistemlerin ve uygulamaların işleyişini izlemek için kritik veri kaynaklarıdır ve bu çağrılardan elde edilen veriler, anomali tespiti için zengin bilgiler sunmaktadır. API çağrılarını, işletim sistemi ile kullanıcı uygulamaları arasındaki etkileşimi sağlayan temel yapı taşlarıdır ve bu çağrılar üzerinden yapılan analizler, sistemin güvenliğini sağlamak için önemli veriler sunmaktadır. Anomali tespiti, sistem güvenliği ve performansı açısından büyük önem taşımaktadır. ML modelleri, büyük miktarda veriyi işleyerek normal ve anormal davranışları öğrenmekte ve bu bilgileri kullanarak yeni verilerdeki anormallikleri tespit etmektedir. Sistem çağrılarını kullanarak yapılan anomali tespiti, ML algoritmaları ile birleştirildiğinde, daha hassas ve doğru tespitler yapabilmektedir. Tez kapsamında, AllMalwarePlusClean2 Subdataset, AllMalwarePlusClean Subdataset ve AllMalware Subdataset adlı üç farklı veri kümesi kullanılmıştır. Bu veri kümeleri, zararlı yazılım ve temiz veri örneklerini içermektedir. Veri kümesi üzerinde ön işleme aşamasında Bit Çifti Kodlaması (BPE) algoritması kullanılmıştır. BPE algoritması, veri boyutunu azaltarak işleme süresini ve hesaplama maliyetini düşürmektedir ve bu da büyük veri kümeleri ile çalışırken önemli avantajlar sağlamaktadır. Tezde, Bi-LSTM, Bi-GRU ve 1D-CNN modelleri kullanılarak anomali tespiti gerçekleştirilmiştir. Bu modeller, zaman serisi ve ardışık bağımlılıkları yakalamada etkin oldukları için tercih edilmiştir. Yapılan deneylerde, Bi-LSTM ve Bi-GRU modelleri, literatürde yaygın olarak kullanılan LSTM ve GRU modellerine kıyasla daha yüksek doğruluk oranları elde etmiştir. Bu modellerin çift yönlü öğrenme kapasiteleri sayesinde zaman serisi verilerindeki bağımlılıkları daha etkili bir şekilde yakaladıkları belirlenmiştir. Yapılan karşılaştırmalarda, Bi-LSTM ve Bi-GRU modellerinin, doğruluk, precision, recall ve F1 skoru gibi metrikler açısından üstün performans sergilediği görülmüştür. Bu durum, Bi-LSTM ve Bi-GRU modellerinin anomali tespiti uygulamalarında kullanılabilirliğini ve etkinliğini vurgulamaktadır.

Anahtar Sözcükler: API çağrılarını, Anomali tespiti, Makine öğrenimi, Bi-LSTM, Bi-GRU, 1D-CNN

ABSTRACT

ANALYZING SYSTEM SECURITY USING APPLICATION PROGRAMMING INTERFACE CALLS

Varol ŞAHİN

Ondokuz Mayıs University

Institute of Graduate Studies

Department of Computer Engineering

Master, July/2024

Supervisor: Prof. Dr. Sedat AKLEYLEK

This thesis focuses on anomaly detection with machine learning (ML) methods using API calls. API, or system calls, are critical data sources for monitoring the operation of systems and applications, and the data obtained from these calls provides a wealth of information for anomaly detection. API calls are the basic building blocks that enable the interaction between the operating system and user applications, and analyzing these calls provides important data to ensure the security of the system. ML models learn normal and abnormal behaviors by processing large amounts of data and use this information to detect anomalies in new data. When anomaly detection using system calls is combined with ML algorithms, it can make more sensitive and accurate detections. In this thesis, three different datasets named AllMalwarePlusClean2 Subdataset, AllMalwarePlusClean Subdataset and AllMalware Subdataset were used. These datasets contain malware and clean data samples. The Byte Pair Encoding (BPE) algorithm was used to preprocess the dataset. The BPE algorithm has reduced the processing time and computational cost by reducing the data size, which provides significant advantages when working with large datasets. In this thesis, anomaly detection is performed using Bi-LSTM, Bi-GRU and 1D-CNN models. These models are preferred because they are effective in capturing time series and sequential dependencies. In the experiments, Bi-LSTM and Bi-GRU models achieved higher accuracy rates compared to the LSTM and GRU models commonly used in the literature. Thanks to their bidirectional learning capacity, these models were found to capture dependencies in time series data more effectively. In the comparisons, Bi-LSTM and Bi-GRU models were found to have superior performance in terms of metrics such as accuracy, precision, recall and F1 score. This emphasizes the usability and effectiveness of Bi-LSTM and Bi-GRU models in anomaly detection applications.

Keywords: API calls, Anomaly detection, Machine learning, Bi-LSTM, Bi-GRU, 1D-CNN

ÖNSÖZ VE TEŞEKKÜR

Bu tez çalışmasında, makine öğrenimi modelleri ile uygulama programlama arayüzü çağrılarını kullanarak anomali tespiti hedeflenmiş ve elde edilen sonuçlar incelenmiştir.

Tez konumun belirlenmesinden tezimin bitimine kadar olan süreçte desteğini esirgemeyen, dahası lisans eğitimimden bu yana hep yanımda olan sayın danışmanım Prof. Dr. Sedat Akleylek'e çok teşekkür ederim.

Varol ŞAHİN



İÇİNDEKİLER

TEZ KABUL VE ONAYI.....	i
BİLİMSEL ETİĞE UYGUNLUK BEYANI.....	ii
TEZ ÇALIŞMASI ÖZGÜNLÜK RAPORU BEYANI.....	ii
ÖZET.....	iii
ABSTRACT	iv
ÖNSÖZ VE TEŞEKKÜR	v
İÇİNDEKİLER.....	vi
SİMGELER VE KISALTMALAR.....	viii
ŞEKİLLER DİZİNİ	xii
TABLOLAR DİZİNİ	xiii
1. GİRİŞ.....	1
1.1. Motivasyon ve Katkı	5
1.2. Organizasyon	8
2. KULLANILAN YÖNTEMLER VE ALTYAPI	10
2.1. Lojistik Regresyon (LR).....	10
2.2. Destek Vektör Makinesi (SVM)	10
2.3. Naive Bayes (NB).....	11
2.4. Rastgele Orman (RF).....	11
2.5. K-En Yakın Komşu (KNN)	11
2.6. Derin Sinir Ağı (DNN).....	12
2.7. LSTM	12
2.8. GRU (Gated Recurrent Unit)	14
2.9. CNN (Convolutional Neural Network)	16
2.10. Değerlendirme Metrikleri.....	17
3. LİTERATÜR TARAMASI	18
4. API ÇAĞRILARI KULLANILAN MAKİNE ÖĞRENİMİ MODELLERİ İLE ANOMALİ TESPİTİ	44
4.1. Veri Kümesi	44
4.2. Ön İşleme Aşaması	45
4.3. Eğitim Aşaması.....	45
4.4. Test Aşaması	46
4.5. Deney Sonuçları.....	47
4.6. Deney Ortamı.....	47
4.7. Sonuçlar.....	47
5. ÖNERİLEN MODEL.....	50
5.1. Veri Kümesi (Čeponis ve Goranin, 2018; Čeponis ve Goranin, 2019).....	50
5.2. Ön İzleme Aşaması.....	52
5.3. Bi-LSTM (Bidirectional Long Short-Term Memory).....	53
5.4. Bi-GRU (Bidirectional Gated Recurrent Unit)	55

5.5.	1D-CNN (1-Dimensional Convolutional Neural Network).....	56
5.6.	Benzetimler	58
5.6.1.	Önerilen Modellerin Literatür ile Karşılaştırılması	59
5.6.1.1.	Bi-LSTM	59
5.6.1.2.	Bi-GRU.....	61
5.6.1.3.	1D-CNN	63
5.6.2.	Oluşturulan Modellerin Elde Ettiği Sonuçlar	64
6.	SONUÇ VE DEĞERLENDİRME	77
	KAYNAKLAR.....	80
	ÖZ GEÇMİŞ	85



SİMGELER VE KISALTMALAR

ACC	: Doğruluk (Accuracy)
ADFA	: Avustralya Savunma Kuvvetleri Akademisi (Australian Defence Force Academy)
ADFA-LD	: Avustralya Savunma Kuvvetleri Akademisi Linux Veri Kümesi (Australian Defence Force Academy Linux Dataset)
ADFA-WD	: Avustralya Savunma Kuvvetleri Akademisi Windows Veri Kümesi (Australian Defence Force Academy Windows Dataset)
ADIM	: Anomali Tespit ve Tanımlama Modülü (Anomaly Detection and Identification Module)
ANN	: Yapay Sinir Ağları (Artificial Neural Networks)
AnRAD	: Nöromorfik Anomali Tespiti (A Neuromorphic Anomaly Detection)
API	: Uygulama Programlama Arayüzü (Application Programming Interface)
AUC	: Eğri Altındaki Alan (Area Under the Curve)
AUPR	: Doğruluk-Duyarlılık Eğrisi Altındaki Alan (Area Under the Precision-Recall Curve)
AUROC	: Alıcı İşletim Karakteristik Eğrisi Altındaki Alan (Area Under the Receiver Operating Characteristic Curve)
BLEU	: İki Dilli Değerlendirme Ölçütü (Bilingual Evaluation Understudy)
BoSC	: Sistem Çağrılarının Torbası (Bag of System Calls)
CNN	: Evrimsel Sinir Ağı (Convolutional Neural Network)
Co-sim	: Kosinüs Benzerliği (Cosine Similarity)
COV	: Kapsama (Coverage)
CPU	: Merkezi İşlem Birimi (The Central Processing Unit)
CSE-CIC-IDS2018	: İletişim Güvenliği Kurumu-Kanada Siber Güvenlik Enstitüsü-İzinsiz Giriş Tespit Sistemi 2018 (Communications Security Establishment-Canadian Institute for Cybersecurity-Intrusion Detection System 2018)
CSV	: Virgülle Ayrılmış Değerler (Comma Separated Values)
cuDNN	: Birleşik Hesaplama Mimarisi Derin Sinir Ağı (Compute Unified Device Architecture Deep Neural Network)
CV	: Karakteristik Değer (Characteristic Value)
D-MLP	: Dinamik Çok Katmanlı Algılayıcı (Dynamic Multilayer Perceptron)
DARPA	: Savunma İleri Araştırma Projeleri Ajansı (Defense Advanced Research Projects Agency)
DL	: Derin Öğrenme (Deep Learning)
DNN	: Derin SinirAğı (Deep Neural Network)

DReAM	: Derin Tekrarlamalı Dikkat Modeli (Deep Recursive Attentive Model)
DT	: Karar Ağacı (Decision Tree)
ECOD	: Ampirik Kümülatif Dağılım Fonksiyonları (Empirical Cumulative Distribution Functions)
EER	: Eşit Hata Oranı (Equal Error Rate)
EL	: Topluluk Öğrenimi (Ensemble Learning)
FAR	: Yanlış Alarm Oranı (False Alarm Rate)
FNR	: Yanlış Negatif Oranı (False Negative Rate)
FPR	: Yanlış Pozitif Oranı (False Positive Rate)
FVM	: Özellik Vektör Matrisi (Feature Vector Matrix)
GCN	: Grafik Evrişimsel Ağ (Graph Convolutional Network)
GNS3	: Grafik Ağ Simülatörü 3 (Graphical Network Simulator 3)
GPU	: Grafik İşleme Birimleri (Graphics Processing Unit)
GRU	: Kapılı Tekrarlayan Birim (Gated Recurrent Unit)
HIDS	: Ev Sahibi Tabanlı İzinsiz Giriş Tespit Sistemi (Host-Based Intrusion Detection System)
HMM	: Gizli Markov Model (Hidden Markov Model)
HTTP	: Köprü Metni Aktarım Protokolü (Hypertext Transfer Protocol)
IDS	: İzinsiz Giriş Tespit Sistemi (Intrusion Detection System)
IEEE	: Elektrik ve Elektronik Mühendisleri Enstitüsü (Institute of Electrical and Electronics Engineers)
IoT	: Nesnelerin İnterneti (Internet of Things)
KNN	: K-En Yakın Komşu (K-Nearest Neighbour)
LCS	: En Uzun Ortak Alt Dizi (Longest Common Subsequence)
LIME	: Yerel Yorumlanabilir Model Bağımsız Açıklamalar (Local Interpretable Model-agnostic Explanations)
LOF	: Yerel Aykırı Faktörü (Local Outlier Factor)
LR	: Lojistik Regresyon (Logistic Regression)
LSTM	: Uzun Kısa Süreli Bellek (Long Short Term Memory)
LTTng	: Linux İzleme Aracı Yeni Nesil (Linux Trace Toolkit Next Generation)
MA	: Hareketli Ortalama (Moving Average)
ML	: Makine Öğrenmesi (Machine Learning)
MRI	: Manyetik Rezonans (Magnetic Rezonans Imaging)
NAHSMM	: İç İç Geçmiş Gizli Yarı Markov Modeli (Nested-arc Hidden Semi-Markov Model)
NB	: Naive Bayes
NGIDS-DS	: Yeni Nesil İzinsiz Giriş Tespit Sistemi Veri Kümesi (Next Generation Intrusion Detection System Dataset)

NIDS	: Ağ Saldırı Tespit Sistemi (Network Intrusion Detection System)
NLP	: Doğal Dil İşleme (Natural Language Processing)
NPV	: Negatif Tahmin Değeri (Negative Predicted Value)
OC-SVM	: Tek Sınıflı Destek Vektör Makineleri (One-Class Support Vector Machines)
OCC	: Tek Sınıf Sınıflandırma (One Class Classification)
OGD	: Açık Kamu Verileri (Open Government Verileri)
OS	: İşletim Sistemi (Operating System)
PDA	: Nokta Tespit Algoritması (Point Detection Algorithm)
PI	: Tahmin Aralığı (Prediction Interval)
PPM-C	: Kısmi Eşleme İle Tahmin - Metod C (Prediction by Partial Matching - Method C)
PRC	: Kesinlik (Precision)
REC	: Duyarlılık (Recall)
Relu	: Doğrultulmuş Lineer Birim (Rectified Linear Unit)
RNN	: Tekrarlayan Sinir Ağları (Recurrent Neural Networks)
RNN-AE	: Tekrarlayan Sinir Ağı Tabanlı - Otomatik Kodlayıcı (Recurrent Neural Network Based - Autoencoder)
RNN-DAE	: Tekrarlayan Sinir Ağı Tabanlı - Gürültü Giderici Otomatik Kodlayıcı (Recurrent Neural Network Based - Denoising Autoencoder)
ROC	: Alıcı İşletim Karakteristiği (Receiver Operating Characteristic)
SHAP	: SHapley Toplam Açıklamaları (SHapley Additive exPlanations)
STIDE	: Sıralı Zaman Gecikmeli Gömme (Sequence Time-Delay Embedding)
STL	: Loess Kullanarak Mevsimsel Trend Ayrışımı (Seasonal-Trend Decomposition Using Loess)
STP	: İstatistiksel Örüntü Tabanlı Özellik Çıkarma Yöntemi (A Statistical Pattern Based Feature Extraction Method)
SVDD	: Destek Vektör Veri Tanımlama (Support Vector Data Description)
SVM	: Destek Vektör Makinesi (Support Vector Machine)
tCNN	: Zamansal Evrişimsel Sinir Ağı (Temporal Convolutional Neural Network)
TCP	: İletim Kontrol Protokolü (Transmission Control Protocol)
TF-IDF	: Terim Frekansı-Ters Doküman Frekansı (Term Frequency-Inverse Document Frequency)
tGCN	: Zamansal Grafik Evrişimsel Ağı (Temporal Graph Convolutional Network)
TNR	: Doğru Negatif Oranı (True Negative Rate)
TPR	: Doğru Pozitif Oranı (True Positive Rate)

UNM	: University of New Mexico
UNSW- NB15	: Yeni Güney Galler Üniversitesi- NB15 (University of New South Wales- NB15)
URI	: Tek Tip Kaynak Tanımlayıcı (Uniform Resource Identifier)
VM	: Sanal Makine (Virtual Machine)
VMI	: Sanal Makine İçgörüsü (Virtual Memory Introspection)
VMM	: Değişken Sıralı Markov Model (Variable-order Markov Model)



ŞEKİLLER DİZİNİ

Şekil 1.1. Makine öğrenmesi tekniklerinin sınıflandırılması.....	4
Şekil 2.1. RNN ve LSTM modellerinin iç mimarisi (Yang vd., 2022)	13
Şekil 2.2. GRU modelinin iç yapısı (Mahjoub vd., 2022).....	15
Şekil 4.1 Ön işleme, eğitim ve test aşamaları arasındaki ilişki	44
Şekil 4.2. Modellerin karışıklık matrisleri.....	48
Şekil 5.1. LSTM ve Bi-LSTM mimarileri (Hochreiter ve Schmidhuber, 1997; Graves ve Schmidhuber, 2005)	55
Şekil 5.2. Bi-GRU modelinin mimarisi (Maji ve Swain, 2022)	56
Şekil 5.3. 1D-CNN modelinin yapısı	58
Şekil 5.4. LSTM mimarisi (Čeponis ve Goranin, 2019).....	59
Şekil 5.5. Tez kapsamında önerilen Bi-LSTM mimarisi	60
Şekil 5.6. GRU (Čeponis ve Goranin, 2019).....	60
Şekil 5.7. Tez kapsamında önerilen Bi-GRU mimarisi	62
Şekil 5.8. 1D-CNN (Čeponis ve Goranin, 2019).....	62
Şekil 5.9. Tez kapsamında önerilen 1D-CNN mimarisi	63
Şekil 5.10. Bi-LSTM modelinin MalwarePlusClean veri kümesi üzerinde 1000 özellikli verilerle 10 dönem eğitilmesi sonucu elde edilen kayıp grafiği	72
Şekil 5.11. Bi-GRU modelinin MalwarePlusClean2 veri kümesi üzerinde 100 özellikli verilerle 10 dönem eğitilmesi sonucu elde edilen kayıp grafiği	73
Şekil 5.12. 1D-CNN modelinin AllMalwarePlusClean veri kümesi üzerinde 400 özellikli verilerle 10 dönem eğitilmesi sonucu elde edilen kayıp grafiği	74
Şekil 5.13. Bi-LSTM modelinin MalwarePlusClean veri kümesi üzerinde 1000 özellikli verilerle 20 dönem eğitilmesi sonucu elde edilen kayıp grafiği	75
Şekil 5.14. Bi-GRU modelinin MalwarePlusClean2 veri kümesi üzerinde 100 özellikli verilerle 20 dönem eğitilmesi sonucu elde edilen kayıp grafiği	75
Şekil 5.15. 1D-CNN modelinin AllMalwarePlusClean veri kümesi üzerinde 400 özellikli verilerle 20 dönem eğitilmesi sonucu elde edilen kayıp grafiği	76

TABLolar DİZİNİ

Tablo 3.1. Literatür çalışmalarının genel özeti	38
Tablo 3.2. Literatür çalışmalarının yöntem analizi	41
Tablo 3.3. Literatür çalışmalarının simülasyon bakımından karşılaştırılması	42
Tablo 4.1. Özelliklerin tanımı	45
Tablo 4.2. Deneysel sonuçlar	48
Tablo 5.1. Veri kümesi üretilirken kullanılan en yaygın kötücül yazılım kategorileri ve aileleri	51
Tablo 5.2. Kullanılan veri kümeleri ve özellikleri	52
Tablo 5.3. Tez kapsamında önerilen modeller ve Çeponis ve Goranin, (2019) çalışmasındaki modellerin AllMalware Subdataset için karşılaştırma sonuçları	66
Tablo 5.4. Tez kapsamında önerilen modeller ve Çeponis ve Goranin, (2019) çalışmasındaki modellerin AllMalwarePlusClean Subdataset için karşılaştırma sonuçları	67
Tablo 5.5. Tez kapsamında önerilen modeller ve Çeponis ve Goranin, (2019) çalışmasındaki modellerin AllMalwarePlusClean2 Subdataset için karşılaştırma sonuçları	68
Tablo 5.6. Modellerin AllMalware Subdataset veri kümesi ile eğitilmesi sonucunda elde edilen PRC, REC, F1-score ve ACC değerleri	69
Tablo 5.7. Modellerin AllMalwarePlusClean Subdataset veri kümesi ile eğitilmesi sonucunda elde edilen PRC, REC, F1-score ve ACC değerleri	70
Tablo 5.8. Modellerin AllMalwarePlusClean2 Subdataset veri kümesi ile eğitilmesi sonucunda elde edilen PRC, REC, F1-score ve ACC değerleri	71

1. GİRİŞ

Akıllı telefonlar, bilgisayarlar ve diğer elektronik cihazlar, günlük hayatın her noktasında biz kullanıcılar ile yer almaktadır. İnternetin de artık vazgeçilmez bir unsur haline gelmesi ile hem kişisel hem de kurumsal organizasyonlarda sistem güvenliği, olmazsa olmaz bir gereklilik haline gelmiştir. Sistem, birbiri ile etkileşim içerisinde bulunan ve genellikle karmaşık bir bütün oluşturan bileşenler bütünüdür. Alternatif olarak ise bir sistem, içerdiği fonksiyonlar ve davranışlarla da tanımlanabilmektedir. Sistemler arasındaki etkileşimler, bağlantıları, çevresel koşulları ve insan yönetici faktörlerine bağlı olarak sistem güvenliği hakkındaki soruları karmaşık hale getirmektedir. Örneğin, gözlemlene, ölçme ve kontrol ihtiyacı, sistem bağlantılarında dikkate alınması gereken kritik unsurlardır. Sistem güvenliği, yalnızca bireysel ve kurumsal düzeyde değil, toplum genelinde de önemli bir konudur. Hızla büyüyen ve evrilen teknolojik gelişmelere ayak uydurmak bu bağlamda kaçınılmazdır.

Günümüzde bilgi güvenliği, siber tehditlerin artışıyla daha da önemli hale gelmiştir. Siber saldırılar, kişisel verilerin çalınmasından kurumsal sistemlerin çökertilmesine kadar geniş bir yelpazede kullanıcı ve sistem düzeyinde zararlara yol açabilmektedir. Bu sebeple, güvenlik önlemlerinin sadece problemler ortaya çıktıktan sonra değil, proaktif bir yaklaşımla ele alınması gerekmektedir. Gelişmiş izleme ve saldırı tespit sistemleri, yapay zeka ve makine öğrenimi teknikleri kullanılarak, potansiyel tehditlerin önceden belirlenmesi ve engellenmesi sağlanmaktadır. Ayrıca, birçok güvenlik ihlalinin temelinde insan hatası bulunması sebebiyle kullanıcı eğitimi ve farkındalığı da sistem güvenliğinin önemli bir bileşeni olarak değerlendirilmektedir. Bu nedenle, kullanıcıların güvenlik protokolleri hakkında bilinçlendirilmesi ve gerekli uygulama ve sistemlerin öğretilmesi, genel güvenlik duruşunu güçlendirecektir (Ranshous vd., 2015).

Anomali tespiti, sistemlerin olağan dışı davranışlarını belirleyerek güvenlik tehditlerini önceden tespit etmeyi amaçlayan kritik bir alandır. Anomali tespiti ayrıca veri modeline uymayan veri örneklerini algılamakta kullanılan bir teknik olmakla birlikte birçok uygulama için üzerinde çalışılan önemli bir araştırma alanıdır (Garg ve Batra, 2017; Ranshous vd., 2015). Anomali tespiti yönteminin asıl amacı normal ve normal olmayan verileri ayırt etmektir. Bu yöntem ortaya çıkan örüntülerin (pattern), trendlerin ve anormalliklerin veri üzerinden tanımlanmasını sağladığından veri analizi

için oldukça önemlidir (Ahmed vd., 2016). Anomali tespiti, güvenlik, sağlık, ağ izleme gibi çeşitli uygulamalar için kritik öneme sahiptir (Alsalman, 2024). Bu bağlamda siber güvenlik, sağlık, bina yönetimi, hava tahmini vb. gibi çeşitli alanlar arasındaki uygulamalar ile gittikçe yükselen ve önemli araştırma alanı haline gelen bir alandır. Anormallikler, önemli ve nadir görülen olaylara işaret edebilmeleri açısından oldukça dikkate değerdir. Bu sayede çeşitli uygulama alanlarında kritik önlemlerin alınmasını sağlayabilmektedir. Örneğin bir ağdaki olağan akışın dışında oluşan bir trafik, bilgisayarın ele geçirildiği ve verilerin yetkisiz kişiler tarafından ele geçirildiği anlamına gelebilir. Benzer şekilde, kredi kartı işlemlerindeki alışılmışın dışında anormal davranışlar dolandırıcılık faaliyetlerine, MRI görüntüsündeki bir anormallik ise kötü huylu bir tümörün varlığına işaret edebilmektedir (Ahmed vd., 2016).

Sistem çağrıları (system calls) bir başka deyişle uygulama programlama arayüzü (API) çağrıları (API calls) kullanılarak gerçekleştirilen anomali tespiti, sistemlerin iç dinamiklerini ve davranışlarını detaylı bir şekilde analiz etmeyi sağlamaktadır. Sistem çağrıları, işletim sistemi ile kullanıcı uygulamaları arasındaki etkileşimi sağlayan temel yapı taşlarıdır ve bu çağrılar üzerinden yapılan analizler, sistemin güvenliğini sağlamak için önemli veriler sunmaktadır. API, bir yazılım programının başka programlara, kullanıcılara ve web API'lerin durumunda ise internet aracılığı ile dünyaya sunduğu bir arayüzdür (Jin vd., 2018). API, işletim sisteminin kritik bir ögesidir ve belirli kütüphanelerde bulundurulmuş bir dizi işlevi kapsamaktadır. Kullanıcılar bu işlevleri çeşitli dosyaların davranışlarını yansıtan işletim sistemi ile iletişimde kullanılmaktadırlar (Almaleh vd., 2023). Geleneksel olarak API bağlamında uygulama kelimesi işlevi bulunan her türlü yazılımı ifade etmek için kullanılmaktadır. Arabirim ise, iki uygulamanın kendi aralarında istekler ve yanıtlar aracılığıyla iletişim kurabilmelerine olanak sağlayan, uygulamalar arası bir hizmet sözleşmesi olarak düşünülebilir. Bir başka ifadeyle API'ler, iki yazılım bileşeninin belirli protokoller aracılığı ile birbirleriyle iletişim kurmalarına olanak sağlayan mekanizmalardır (AWS, 2024). API çağrıları ise, uygulamalar tarafından kullanılan programlama arayüzleridir (Kang vd., 2023). API çağrısında, sunucu başka bir sunucunun web arayüzüne iletim kontrol protokolü (TCP) aracılığı ile çağrıda bulunmaktadır. Bir istekte bulunabilmek için üç ana bileşen gereklidir. Bu bileşenler; tek tip kaynak tanımlayıcı (URI), başlıklar ve istek gövdesidir. Her API bu bileşenlerin farklı bir kombinasyonunu kullanmakta, farklı bir formatta haberleşmekte veya farklı veri gerektirmektedir. Fakat istek her

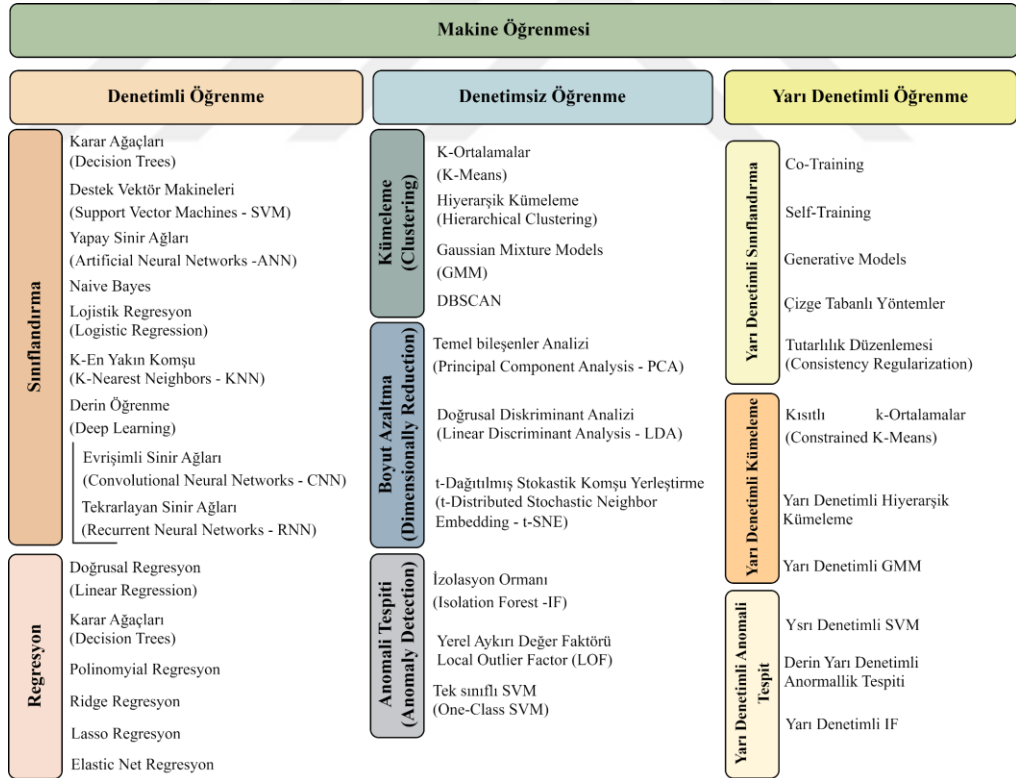
durumda HTTP olarak tanımlı bir mesaj yapısıdır. Bir analistin herhangi bir programın direkt olarak tersine analizini yapmasının zor olduğu durumlarda API, programın davranış ve işlevselliğini analiz etmesi için analiste önemli bir temel sağlamaktadır. Bu temel sayesinde sistemin davranışlarındaki anormallikler analist tarafından tespit edilebilmektedir. API/Sistem çağrısı, program tarafından herhangi bir servis için çekirdeğe (*kernel*) yapılan çağrıdır. Bu çağrının izleri (*trace*) işlemin (*process*) davranışını analiz edebilmektedir. Bu izler, işlemin normal ya da kötü niyetli olarak sınıflandırılmasında kullanılmaktadır (Osamor ve Wellman, 2022).

Anomali tespitinde kullanılan yöntemlerin doğruluğu ve güvenilirliği, kullanılan veri kümeleri ve modelin eğitimi ile doğrudan ilişkilidir. Eğitim sürecinde kullanılan veri kümelerinin normal ve anormal davranışları yeterince temsil edebiliyor olması gerekmektedir. Bu bağlamda, veri kümeleri, çeşitli saldırı türlerini ve sistem davranışlarını içermekte olup, anomali tespiti modellerinin performansını değerlendirmek için ideal test ortamları sunmaktadır (Chandola vd., 2009).

API çağrıları veya sistem çağrıları, kayıt değişiklikleri (*registry modification*) ve ağ aktiviteleri gibi çeşitli özellikler, kötü amaçlı yazılımların davranışlarını oluşturmaktadır. Dinamik analiz ile çıkarılan API çağrıları ve bu çağrılara ilişkin çeşitli bilgiler, davranış tabanlı tespit sistemlerinin en önemli özelliklerinden biri olarak kabul edilmektedir. Dizideki her API çağrısı, kendisinden bir önceki veya bir sonraki API çağrısı ile ilişkilendirilmektedir. Bu ve benzeri ilişkiler, kötü amaçlı yazılımların yıkıcı fonksiyonlarına ait örüntüleri içerebilmektedir. API ve sistem çağrıları ile ilgili çeşitli bilgilerin özellik olarak kullanıldığı, makine öğrenimi ve derin öğrenme modellerini de içeren birçok anormallik tespit sistemi önerilmiştir. Özellikle, makine öğrenimi yöntemleri ve derin öğrenme algoritmaları, bu tür anormallik tespit sistemlerinin performansını artırmak için kullanılmaktadır (Buskirk vd., 2018). Örneğin, derin öğrenme modelleri, karmaşık veri kümelerinde yüksek doğruluk oranlarına ulaşabilmekte ve kötü amaçlı yazılımların tespitinde etkin bir şekilde kullanılabilmektedir. Bu bağlamda, sistem güvenliğini artırmak ve kötü amaçlı yazılımları etkin bir şekilde tespit edebilmek için makine öğrenmesi (ML) ve derin öğrenme (DL) yöntemlerinin entegrasyonu büyük bir öneme sahiptir.

Denetimli, denetimsiz ve yarı denetimli olmak üzere üç farklı anomali tespit tekniği bulunmaktadır. Denetimli tekniklerde eğitim için etiketlenmiş veriye ihtiyaç duyulmaktadır ve bununla birlikte anormal verilerin açıkça etiketlenmiş olması

gerekmektedir (Chandola vd., 2009). Bu teknik, içinde hem normal hem de anormal örnekler bulunduran, etiketlenmiş bir veri kümesi üzerinde eğitilmiş bir algoritma kullanmaktadır. Fakat etiketli eğitim verilerinin genellikle mevcut olmaması ve sınıfların dengesiz olması nedeniyle bu teknik çok sık tercih edilmemektedir. Denetimli tekniğin aksine, denetimsiz teknikler etiketlenmiş verilere bağımlı değildir. Bu veriler yerine normal örüntü sapmalarına dayalı olarak anormallikleri tespit etmektedir (Chandola, vd. 2009). Denetimsiz teknikler ile veri mühendisleri, modelleri ve anormallikleri kendi kendine tespit edebilen etiketlenmemiş veri kümeleri aracılığı ile bir modeli eğitmektedir. Fakat bu teknikler daha geniş ve özelleştirilmiş uygulamalara sahip olması sebebiyle açık ara en çok kullanılan teknikler olmalarına rağmen, yüksek hacimli veri kümeleri ve işlem gücü gerektirmektedir. Yarı denetimli teknikler ise etiketlenmiş ve etiketlenmemiş verilerin her ikisini de kullanabiliyor olması sebebiyle denetimli ve denetimsiz tekniklerin arasında yer almaktadır (Chandola, vd. 2009). Şekil 1.1, makine öğrenmesi tekniklerinin sınıflandırmasını göstermektedir.



Şekil 1.1. Makine öğrenmesi tekniklerinin sınıflandırılması

Makine öğreniminin alt kümesi ve aynı zamanda bir evrimi olan derin öğrenme, insan bilişsel süreçlerini yansıtan karmaşık algoritmalar kullanarak daha sıkı olan sinir

ağlarını geride bırakmaktadır. Derin sinir ağları oluşturan bu algoritmalar, insan beyninin mantıksal yapısını taklit etmekte ve verileri analiz ederek sonuçlara ulaşmasını sağlamaktadır. Manuel olarak çıkarılan özelliklere dayanan geleneksel makine öğreniminden farklı olarak derin öğrenme, uçtan uca bir öğrenme çerçevesi üzerinde çalışarak insan müdahalesini minimum seviyeye indirmektedir. Derin sinir ağlarının mimarisi, karmaşık örüntüleri öğrenme yeteneğini artıran ve birbirine bağlı çok sayıda doğrusal olmayan katmanlardan oluşmaktadır (Manakitsa vd., 2024).

ML yöntemlerinin anomali tespiti alanında kullanımı yaygınlaşmaktadır. ML modelleri, büyük miktarda veriyi işleyerek normal ve anormal davranışları öğrenmekte ve bu bilgileri kullanarak yeni verilerdeki anormallikleri tespit etmektedir. Sistem çağrılarını kullanılarak yapılan anomali tespiti, ML algoritmaları ile birleştirildiğinde, daha hassas ve doğru tespitler yapabilmektedir. ML algoritmalarının anomali tespitinde kullanılması, birkaç adımdan oluşmaktadır. İlk adım, sistem çağrılarının toplanması ve bu verilerin ön işleme tabi tutulmasıdır. Bu süreçte, sistem çağrılarını belirli bir formatta düzenlenmekte ve gereksiz bilgilerden arındırılmaktadır. Daha sonra bu veriler özellik çıkarma (feature extraction) teknikleri kullanılarak anlamlı özellik vektörlerine dönüştürülmektedir. Özellik çıkarma adımında, n-gram analizi, frekans tabanlı yöntemler ve istatistiksel yöntemler gibi çeşitli teknikler kullanılmaktadır. Özellikle, n-gram analizi, sistem çağrılarının belirli bir ardışıklık düzeninde gruplandırılması ve bu grupların frekanslarının hesaplanması ile gerçekleştirilmektedir. Özellik vektörlerinin oluşturulmasının ardından ML modelleri bu veriler üzerinde eğitilmektedir. Sıkça kullanılan algoritmalar arasında, destek vektör makineleri (SVM), k-en yakın komşu (KNN), karar ağaçları (Decision Trees) ve sinir ağları (Neural Networks) yer almaktadır. Bu modeller, eğitim verisi üzerindeki normal ve anormal davranışları öğrenerek, yeni veriler üzerinde anomali tespiti yapmaktadır. Özellikle derin öğrenme modelleri büyük ve karmaşık veri kümeleri üzerinde yüksek performans göstermektedir. Tekrarlayan sinir ağları (RNN) ve uzun-kısa süreli bellek (LSTM) ağları, zaman serisi verilerinde etkili anomali tespiti yapabilen güçlü modellerdir (Čeponis ve Goranin, 2019).

1.1. Motivasyon ve Katkı

Günümüz teknolojik gelişmelerinin hızla ilerlemesi ile, bireyler ve kurumsal organizasyonların, operasyonları kolaylaştırmak, üretkenliği artırmak ve kullanıcılara sorunsuz bir deneyim sunmak amacıyla yürüttüğü, birbirine bağlı sistemler ve

uygulamaların kullanımı da paralel olarak artmaktadır. Bu sistemler ve uygulamalar, farklı yazılım bileşenleri arasında iletişim ve veri alışverişini sağlayarak köprü görevi gören API'ler sayesinde mümkün olmaktadır. API'lerin kullanımının artması ile, potansiyel tehditlerden korunmak için güvenlik önlemlerine duyulan ihtiyaç da artmaktadır.

İnternet teknolojilerinin benimsenerek günlük hayatın ayrılmaz bir parçası haline gelmiş olması, kötüye kullanımı ve suistimale açık olması gibi dezavantajları da beraberinde getirmiştir. İnternet ortamında dolaşan yüksek miktarda verilerin artan karmaşıklığı, anormalliklerin, beklenmedik kalıpların veya davranışların ortaya çıkma riskini de artırmaktadır. Anomaliler, performans sorunlarından güvenlik ihlallerine kadar çeşitli yelpazede birçok soruna işaret etmektedir. Bu nedenle, anomalilerin gerçek zamanlı olarak tespiti, sistemlerin bütünlüğünü, güvenliğini ve performansını korumak için hayati bir önem taşımaktadır. Özellikle API çağrıları aracılığıyla yapılan anomali tespiti, bu bağlamda çok kritik bir rol oynamaktadır. Uygulamalar arasındaki veri aktarımı ve iletişimi sağlayan API çağrıları, işlevsellik açısından oldukça önemlidir. Bir uygulamanın API çağrılarını izlemek, uygulamanın davranışlarını ve çalışma mantığını anlayabilmek, güvenlik zafiyetleri, enerji tüketimi ve sistem kaynaklarının işgali gibi sorunları tespit etme açısından kritik bir öneme sahiptir. Anomali tespiti, sistemde normalin dışında meydana gelen herhangi bir etkinliği tespit ederek, uygulamada veya sistemde hasar meydana gelmeden önce müdahale edilmesini mümkün kılmaktadır. Ayrıca, olağan dışı istekleri ve sızma girişimlerini tespit edip sistemi koruyarak, ortaya çıkan uygulama ve bakım maliyeti gibi sonuçları önleyebilmektedir. Bireysel kullanıcılar ve kurumsal organizasyonlar, API çağrılarını izleyip analiz ederek sorunların büyüyerek daha ciddi hale gelmesinin önüne geçebilirler. Bu anlamda, gerçek zamanlı anomali tespiti ile güvenlik açısından önleyici ve koruyucu önlemler alınması sağlanmaktadır.

Anomali tespiti, güvenliğin yanı sıra operasyonlar hakkında da değerli bilgiler sağlamaktadır. Normal API davranışından sapmalar, uygulamalardaki anormalliklere işaret edebilmektedir. Örneğin, bir API çağrısının alışılmadık derecede yavaş olması durumunda anomali tespiti bu sorunun tespit edilmesini ve kullanıcı deneyimini olumsuz etkilemeden önce çözülmesini sağlamaktadır. Böylece, anomalilerin erken tespit edilmesiyle uygulamaların sorunsuz ve verimli çalışması sağlanarak kullanıcı memnuniyetinin artırılmasına yardımcı olunmaktadır. Anomali tespitinin faydaları

arasında mali kayıplara karşı koruma da yer almaktadır. Dolandırıcılık faaliyetleri genellikle olağandışı API çağrı modelleri oluşturmaktadır. Anomali tespiti, kullanıcı hesaplarına yetkisiz erişim veya finansal verileri manipüle etme girişimleri gibi şüpheli faaliyetlerin tespit edilmesini mümkün kılmaktadır. Bu anomalilerin erkenden yakalanması ile dolandırıcılığın önüne geçilebilmekte ve işletmelerin finansal zararlardan korunması sağlanabilmektedir. Ayrıca, sistem ve kullanıcı sayısı arttıkça büyük hacimli API çağrılarının kontrolü de sağlanmakta ve anormallikler tespit edilebilmektedir. Bu durum, manuel izleme ihtiyacını azaltmakta ve insan gücü gereksinimini düşürmektedir. Böylece API'lerin güvenliği ve performansı sağlanırken çalışanların stratejik projelere odaklanması mümkün kılınmaktadır.

API çağrıları kullanılarak anomali tespiti çok önemli ve işlevsel olmasına rağmen bazı açık problemler henüz araştırmacılar ve uygulama geliştiriciler tarafından çözüme kavuşturulamamıştır. Bu açık problemlerden bazıları aşağıdaki gibidir:

- Anomali tespiti için kullanılacak veri kümelerinin dengesiz olması, makine öğrenimi yöntemlerinin başarımını etkilemektedir. Genellikle bir kaynaktan elde edilen anomali içeren veri sayısı, normal sistem davranışları içeren veri sayısına kıyasla daha düşük sayıdadır. Bu bağlamda, kullanılacak ML yöntemleri, veri dengesizliği karşısında eğitim ve test aşamalarında başarı metrikleri bakımından yetersiz sonuçlar üretebilmektedir. Ayrıca, mevcut veri kümesinin etiketlenmesi, veri hacmine ve çeşitliliğine bağlı olarak zaman ve kaynak kullanımı bakımından maliyetli olabilmektedir (Ahmed vd., 2016).
- Yüksek veri hacmi içeren veri kümeleri kullanılarak uygulanan ML modellerinin bu yük altında performansı düşük kalabilmektedir. API çağrı sayısının artışı ölçeklenebilirliği doğrudan etkilemektedir. Literatürde yer alan dağıtık sistemlerde ve bulut sistemlerinde çalışan modellerin performansı göz önüne alındığında, yüksek veri hacminin neden olduğu performans sorunları bir başka geliştirilmeye açık bir alandır (Muhammad ve Yan, 2015).
- Açık kaynaklarda yer alan veri kümeleri düşünüldüğünde, API çağrılarının içerdiği hassas bilgiler gizlilik ihlallerine sebep olabilmektedir. Bu bağlamda, API çağrıları içerisindeki uygulama ve sistem davranışlarına

ilişkin veri kümelerinin gizlilik ve veri güvenliği dikkate alınarak oluşturulması gerekmektedir (Almaleh vd., 2023).

Açık problemler dikkate alındığında, tez çalışmasının ana katkıları aşağıdaki şekilde özetlenmektedir:

- Kullanılan veri kümesi üzerinde yapılan deneyler Bi-LSTM ve Bi-GRU modelleri kullanılarak gerçekleştirilmiştir. Bu modeller, veri kümesindeki zaman serisi ve ardışık bağımlılıkları yakalamada etkin oldukları için tercih edilmiştir.
- Veri kümeleri üzerinde ön işleme aşamasında Bit Çifti Kodlaması (BPE) algoritması kullanılmıştır. BPE algoritması, veri kümesindeki kelimeleri veya karakter dizilerini sıkıştırarak daha kompakt ve yönetilebilir bir temsil oluşturmaktadır. Bu, özellikle büyük veri kümelerinde veri boyutunu azaltarak işleme süresini ve hesaplama maliyetini düşürmektedir.
- Literatürde önceki çalışmalarda elde edilen sonuçlar ile karşılaştırma yapılmıştır. Bu karşılaştırmalarda, literatürde yaygın olarak kullanılan LSTM ve GRU modelleri temel alınmıştır. Yapılan deneyler sonucunda, Bi-LSTM ve Bi-GRU modellerinin literatürdeki LSTM ve GRU modellerine kıyasla daha iyi sonuçlar verdiği görülmüştür. Bi-LSTM ve Bi-GRU modelleri, çift yönlü öğrenme kapasiteleri sayesinde zaman serisi verilerindeki bağımlılıkları daha etkili bir şekilde yakalamış ve anomali tespitinde daha yüksek doğruluk oranları elde etmiştir. Bi-LSTM ile Bi-GRU modellerinin anomali tespiti gibi uygulama alanlarında kullanılabilirliğini gösterilmiştir.

1.2. Organizasyon

Tez çalışmasının içeriği şu şekilde organize edilmiştir: Giriş bölümünde araştırmanın motivasyonu ve katkıları ile organizasyonu açıklanmaktadır. İkinci bölüm, Kullanılan Yöntemler ve Altyapı kısmında, çeşitli makine öğrenimi modelleri ve bu modellerin kullanımında kullanılan metrikler detaylandırılmaktadır. Üçüncü bölüm, Literatür kısmında, ilgili literatür çalışmaları incelenmekte ve mevcut çalışmalar özetlenmektedir. Dördüncü bölümde, API Çağrılarını Kullanılan Makine Öğrenimi Modelleri ile Anomali Tespiti kısmında, veri kümesi, ön işleme, eğitim, test

ařamaları ve deney sonuçları ele alınmaktadır. Beřinci bölüm, Önerilen Model kısmında, veri kümesi, Bi-LSTM, Bi-GRU ve 1D-CNN gibi modeller ile yapılan benzetimlerin ve bu modellerin literatürle karşılaştırılmasının sonuçları sunulmaktadır. Son olarak, altıncı bölümde sonuç ve deęerlendirme kısmında yapılan çalışmaların genel bir deęerlendirmesi ve elde edilen sonuçların analizi yer almaktadır.



2. KULLANILAN YÖNTEMLER VE ALTYAPI

ML yöntemleri, verilere dayalı tahminler (regresyon) veya sınıflandırmalar yapmak için kullanılan tekniklerdir (Buskirk vd., 2018). Bu yöntemler, bağımsız ve bağımlı değişkenler arasındaki doğrusal olmayan ilişkileri belirleme yeteneğine sahiptir. ML teknikleri iki kategoriye ayrılır: denetimli ve denetimsiz (Buskirk vd., 2018; Muhammad ve Yan, 2015). Denetimli öğrenme, eğitim için önceden etiketlenmiş verilerin kullanılmasına dayanmaktadır. Sınıflandırma, denetimli öğrenmede en yaygın süreçtir, ancak verilerin manuel olarak etiketlenmesi gerekmektedir. Yeterli sayıda etiketli veri olmaması, denetimli öğrenme yöntemlerinin başarılı sonuçlar elde etmesini engelleyebilmektedir. Denetimsiz öğrenme, etiketsiz verilerden özelliklerin çıkarılmasını gerektirmekte ve tespit performansı genellikle denetimli öğrenme yöntemlerine kıyasla yetersiz kalmaktadır. Yapılan ön çalışma öncelikle denetimli öğrenme modellerine odaklanmıştır. Daha sonra, bu bölümde çalışmada kullanılan denetimli öğrenme yöntemleri ve bu yöntemlerin performanslarını değerlendirmek için kullanılan metrikler hakkında kısa bilgi verilecektir.

2.1. Lojistik Regresyon (LR)

LR modeli, iki sınıflı sınıflandırma problemlerini ele almak için kullanılan bir makine öğrenimi yöntemidir. Girdi değişkenleri bir veya daha fazla özelliği kapsayabilir. İkili lojistik regresyonda, sonuçlar tipik olarak 0 veya 1 olarak temsil edilmektedir. LR, çıktı olasılık değerlerini hesaplamak için sigmoid fonksiyonunu kullanır ve sınıflandırma görevlerini yerine getirmektedir. Sigmoid fonksiyonu 0 ile 1 aralığında çıktıları vermektedir. 0,5'in altında değerlere sahip örnekler negatif sınıfa ait olarak belirlenirken, 0,5'e eşit veya 0,5'i aşanlar pozitif sınıfa atanmaktadır (Shanthamallu vd., 2017).

2.2. Destek Vektör Makinesi (SVM)

SVM, ikili ve çok sınıflı sınıflandırma problemleri için yaygın olarak kullanılan bir denetimli öğrenme modelidir. SVM, giriş veri noktalarını n boyutlu bir uzaya eşleyerek ve verileri farklı gruplara ayırmak için $n-1$ boyutlu bir hiper düzlem oluşturarak çalışmaktadır (Shanthamallu vd., 2017). SVM algoritmasının amacı, iki grup arasındaki marjı maksimize ederek bu noktaları bir hiper düzlem aracılığıyla iki ayrı kümeye ayırmaktır. Ayrıca doğrusal ayrımın elde edilemediği durumlarda,

çekirdek hilesi (Kernel Trick) olarak adlandırılan bir teknik kullanılmaktadır (Muhammad ve Yan, 2015). Daha sonra, bu dönüştürülmüş uzayda doğrusal bir hiper düzlem (Hyperplane) tanımlanarak orijinal özellik uzayına geri yansıtılmaktadır. Gauss, radyal tabanlı fonksiyon ve polinom çekirdeği gibi çeşitli çekirdek türleri uygulamada yaygın olarak kullanılmaktadır. SVM modelinin en önemli avantajları aşırı uyumdan kaçınması ve olasılıksal olmamasıdır (Muhammad ve Yan, 2015).

2.3. Naive Bayes (NB)

NB sınıflandırıcıları basit olasılıksal sınıflandırıcılardır (Muhammad ve Yan, 2015). "Naive" terimi, girdi özelliklerinin hiçbirisi arasında korelasyon olmadığı ve hepsinin birbirinden bağımsız olduğu varsayımını ifade etmektedir. Naive Bayes algoritması temelde Bayes teoremine dayanmaktadır. Bu yöntem, girdi özelliklerinin belirli bir sınıfa ait olma olasılığını vermektedir (Rish, 2001).

2.4. Rastgele Orman (RF)

RF, bir dizi ağaç yapılı sınıflandırıcıdan oluşan bir topluluk öğrenme algoritmasıdır. Model içinde her ağaç, bir özellik alt kümesi seçmek için bir karar ağacı algoritması kullanılmaktadır. Orman RF yöntemi kullanılarak oluşturulduktan sonra, yeni bir örneği sınıflandırmak için her bir ağaca veri sunulmakta ve sonrasında, her ağaç belirli bir sınıf için oy vererek kararını belirtir. Orman, verilen örnek için en çok oyu alan sınıfı seçer. RF yönteminin başlıca avantajları arasında gürültüye karşı dayanıklılığı ve diğer modellere kıyasla aşırı uyuma daha az duyarlı olması yer almaktadır (Min vd., 2018).

2.5. K-En Yakın Komşu (KNN)

En basit denetimli makine öğrenimi yöntemlerinden biri olan KNN algoritması, girdi verilerini ayrık sonuçlara sınıflandırmaktadır (Muhammad ve Yan, 2015). Olasılıksal ve parametrik olmayan doğası nedeniyle veri dağılımı hakkında ön bilgi olmadığında sınıflandırma problemlerinde kullanılmaktadır. Sınıflandırma işlemi, girdiler arasında bir benzerlik ölçüsünün (mesafe) hesaplanmasına dayanmaktadır. Daha sonra, herhangi bir bilinmeyen örnek, en yakın k tane komşusunun durumuna göre sınıflandırılmaktadır. Ancak veri boyutu büyüdükçe karmaşıklık da artmaktadır. KNN kullanılmadan önce boyutsallığın etkilerini azaltmak için genellikle boyut azaltma teknikleri uygulanmaktadır (Beyer vd., 1999).

2.6. Derin Sinir Ağı (DNN)

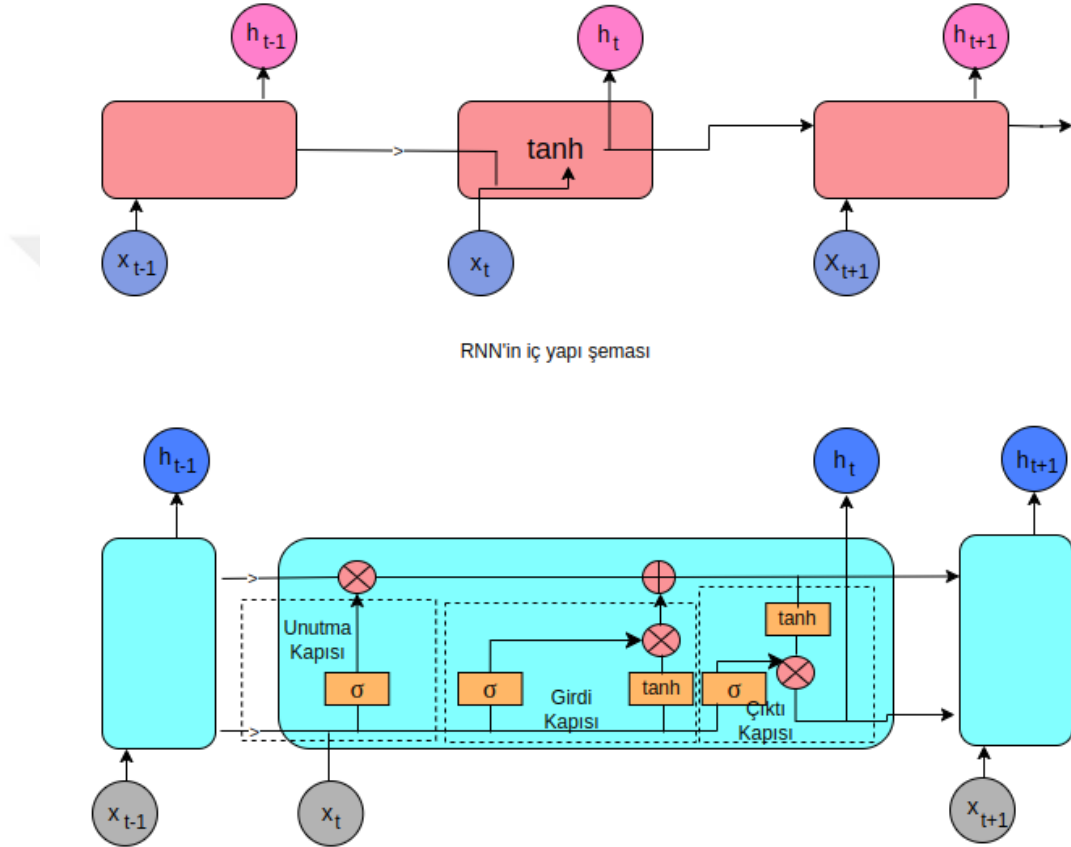
DNN, makine öğreniminde en sık kullanılan yöntemlerden biridir. Geleneksel makine öğrenimi tekniklerinin aksine, büyük veri kümelerini işlerken üstün performans göstermektedir. DNN modellerinin dikkate değer bir özelliği, birden fazla gizli katmandan oluşan derin mimarileridir (Liu ve Lang, 2019). Bir DNN'in tasarımı insan beyninin çalışma mantığını yansıtmakta ve tipik olarak girdi katmanı, birkaç gizli katman ve çıktı katmanlarından oluşmaktadır. Sonuç olarak DNN modelleri birçok birimi kapsamakta ve bu durum da onları doğrusal olmayan ve karmaşık verileri sınıflandırmak için uygun hale getirmektedir. Ancak DNN modellerinin eğitimi, karmaşık model yapıları ve boyutları nedeniyle diğer yöntemlere göre daha fazla zaman gerektirmektedir (Liu ve Lang, 2019).

2.7. LSTM

Geleneksel RNN yapılarında, girdiler bir dizi ilişkili sıralı verilerden oluşmakta ve ileri yönde özyinelemeli hesaplama gerçekleştirilmektedir. Tüm nöronlar zincirleme olarak birbirine bağlanmakta ve böylece önceki bilgiler mevcut anın bilgileriyle entegre edilerek bir sonraki anda birlikte çalışabilmektedir. Bu yapı RNN'lerin hafıza işlevini görmesini ve zaman dizilerini işlemesini sağlamaktadır. RNN sinir ağlarının bellek kapasitesinin sınırlı olması nedeniyle önceki zaman aşamalarında yer alan bilgiler daha az tutulurken, yakın anların bilgileri daha fazla tutulmaktadır. Eğitim sürecinde bilgiler çok sayıda tekrarlı döngü gövdesinden geçerken bilgilerin bozulmasına sebep olmaktadır. Bu durum gradyan kaybolması olarak adlandırılmaktadır.

LSTM modeli, RNN modelinin gelişmiş bir versiyonudur. Bir derin öğrenme modeli olarak RNN yapılarında, ağ katman sayısının artmasıyla birlikte kaybolan gradyan problemi oluşmaktadır. Bu da ağın önceki zaman adımlarındaki bilgileri öğrenmesini engellemektedir. LSTM modeli ise, bilginin akışını kontrol edebilmek için bir geçit mekanizması kullanan, bileşik bir tekrarlayan ünite ile RNN yapısındaki gradyan kaybolması sorununun çözümü için önerilmiştir. LSTM yapısında durum değerinin uzun süre tutulabilmesi için sabit ağırlıklı ve kendisine bağlı tekrarlayan düğüm içeren bir bellek hücresi bulunmaktadır. Böylece gradyan kaybolma problemi oluşmadan birçok zaman adımından geçebilmektedir (Liu vd., 2020b).

LSTM modelinde ise bellek hücreleri kullanılarak geleneksel nöronların sinir ağı yapısı geliştirilmiştir. Bellek hücrelerinin durumu, bilgi transferi ve işleme kapasitesi LSTM sinir ağı mimarisinin en belirgin özellikleridir. Bellek hücrelerinde geçmiş, bir diğer ifadeyle tarihsel bilgiler iyi bir şekilde muhafaza edilebilmektedir. Hücrelerin içindeki giriş kapısı, değerlerin korunma süresini kendi başına yönetebilir, böylece önemli bilgiler daha iyi tahmin edilebilmektedir (Yang vd., 2022).



Şekil 2.1. RNN ve LSTM modellerinin iç mimarisi (Yang vd., 2022)

LSTM, dizi (sequence) tahmini ve uzun süreli bağımlılıkları yakalamada oldukça başarılıdır. Bu modeli bu kadar başarılı kılan başlıca sebep düzen bağımlılığını anlayabilme yeteneğidir. Bu yetenek sayesinde makine çevirisi ve konuşma tanıma gibi karmaşık problemler çözülebilmektedir. Geleneksel RNN yapıları kısa süreli bir hafızaya ve tek bir gizli duruma sahip olmaları, ağı uzun süreli bağımlılıkları öğrenmesini zorlaştırabilir. LSTM bu durumu, bilgiyi daha uzun süreli tutabilecek bir konteyner olan bir bellek hücresi ile ele almaktadır. Bu bellek hücresi, giriş kapısı, unutma kapısı ve çıkış kapısı olmak üzere üç adet kapı ile kontrol edilmektedir. Giriş kapısı bellek hücresine hangi bilginin eklendiğini, unutma kapısı hangi bilginin

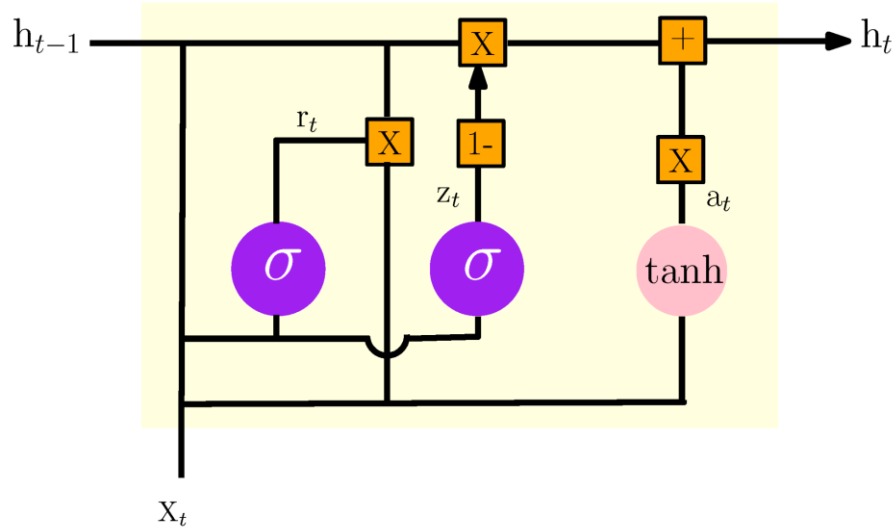
silineceğini, çıkış kapısı ise hangi bilginin çıktı olarak verileceğini kontrol etmektedir. Bu durum, LSTM ağlarının bilgileri ağ üzerinde akarken seçici bir şekilde tutması ya da silmesine olanak tanımakta ve uzun süreli bağımlılıkları öğrenmelerini sağlamaktadır. Bu kapılar aşağıdaki gibi tanımlanmaktadır (Yang vd., 2022):

- **Giriş Kapısı:** Eğitim sürecine yeni bilgilerin eklenip eklenmeyeceğinin kontrolü ve o andaki hücre durumunun tespiti için kullanılmaktadır. Giriş kapısında sigmoid ve tanh katmanları bulunmaktadır. Sigmoid katmanı bilgi güncellemesinin içeriğini belirlerken tanh katmanı mevcut duruma eklenecek yeni aday değerler vektörü oluşturmakla görevlidir.
- **Unutma Kapısı:** Mevcut giriş bilgisine ve bir önceki zaman dilimindeki bilgilere dayanarak bellek biriminin değerinin tutulup tutulmayacağını belirlenmesi için bir sigmoid fonksiyonu kullanılmaktadır.
- **Çıkış Kapısı:** Sigmoid fonksiyonu tarafından belirlenen çıkış değeri tanh fonksiyonları ile çarpılarak nihai çıkış değeri elde edilmektedir.

2.8. GRU (Gated Recurrent Unit)

GRU, RNN yapısının özel bir türüdür ve RNN yapılarındaki problemlerin çözümü amacıyla geliştirilmiştir. RNN yapılarında patlayan veya kaybolan gradyanlar problemleri bulunmaktadır. Bu durum da sensör verilerindeki uzun vadeli aktivite dizilerini ve zaman korelasyonlarını modellemedeki etkinliğini azaltmaktadır. Bu nedenle LSTM ve GRU gibi RNN varyantları önerilmiştir. Her iki model de zamansal faaliyet dizilerini yakalamak için bellek hücreleri ve kapı çeşitlerini entegre etmektedir. LSTM, ağa bilgi akışının kontrol edilmesi amacıyla bağlam bilgilerini derlemek için bir bellek birimi ile birleşmektedir. LSTM ve GRU, bir sonraki gizli durumun güncellenmesi ve içerikle etkileşim mekanizmaları açısından farklılıklar göstermektedirler. LSTM, toplama yoluyla güncellenirken, GRU, bir sonraki gizli durumu güncelleyebilmek için belleğe kaydedilme süresine göre hareket etmektedir. (Chen vd., 2022). GRU modelinin iç yapısı LSTM modelinin iç yapısı ile benzerlik göstermektedir. GRU'lar LSTM'lere oranla daha az kapı kullanırlar ve bu sayede akıcı bir yapıya sahiptir. Bundan dolayı LSTM modellerine göre hesaplama açısından daha az maliyetli ve uygulanması daha kolaydır. GRU, ayrı bir bellek birimi kullanmadan gizli durum içindeki bilgi akışını düzenleyen kontrol birimlerine sahiptir (Liu vd.,

2020c). GRU, LSTM ağlarındaki giriş kapısı ve unutma kapısını güncelleme kapısı adında tek bir kapıda birleştirmektedir. Bir GRU sıfırlama (reset) kapısı ve güncelleme (update) kapısı olmak üzere iki adet kapıya sahiptir. Sıfırlama kapısı, geçmişten gelen bilgiler ile mevcut durumun ilişkilendirilip ilişkilendirilmeyeceğine ve hangi bilginin tutulup hangisinin unutulacağına karar verir. Güncelleme kapısı ise kapsamı kontrol eden geçmişten gelen bilgileri mevcut durumda tutan ve ne kadarının geleceğe aktarılacağına karar veren kapıdır. Uzun süreli bağımlılıkları korumak için ayrı bir bellek hücresi bulunduran LSTM ağlarının aksine GRU'lar saklı durum ve hücre durumunu tek bir vektörde birleştirebilirler. Böylece modelin karmaşıklığının azalması sağlanmaktadır. GRU'lar daha basit yapılarına rağmen özellikle konuşma tanıma, dil modelleme, zaman serisi tahmini gibi modelleme gereken durumlarda LSTM'lere eşit veya onlardan daha iyi performans sergilemektedir. Şekil 2.2, temel bir GRU biriminin dizaynını göstermektedir (Mahjoub vd., 2022).



Şekil 2.2. GRU modelinin iç yapısı (Mahjoub vd., 2022)

GRU'nun kendine özgü yapısında görüldüğü gibi GRU yapısında saklı durum (hidden state - h_t) ve aday aktivasyonu (candidate activation - a_t) birleşmektedir. Burada h_{t-1} bir önceki zaman adımından gelen gizli durumu temsil etmektedir. X_t , mevcut zaman adımının girişini, h_t ise mevcut zaman adımının çıkışını temsil etmektedir. Güncelleme kapısı z_t ve sıfırlama kapısı r_t , GRU hücresinde bulunan iki adet kontrol kapılarıdır. Mor daire içindeki " σ " ifadesi sigmoid fonksiyonunu ifade etmektedir. Pembe daire içindeki " $\tanh$ " ise hiperbolik tanjant fonksiyonunu ifade etmektedir. Güncelleme kapısı z_t , bir önceki zaman adımı $t-1$ 'den mevcut zaman adımı

t'ye iletilen durum bilgisi $ht-1$ miktarını düzenlemek için kullanılmaktadır. Güncellenen kapının değeri büyüdükçe bir önceki zaman adımından o kadar fazla durum bilgisi getirilmektedir. Sıfırlama kapısı rt , bir önceki durumdan ne kadar bilginin mevcut aday küme at'ye aktarılacağını belirlemektedir. Bir önceki durumdan ne kadar az bilgi yazılıyorsa sıfırlama kapısı rt o kadar küçük olarak değerlendirilmektedir (Xie vd., 2022).

2.9. CNN (Convolutional Neural Network)

Evrişimsel sinir ağları (CNN), görsel amaçlı derin öğrenme uygulamalarının önde gelen dallarından biri olarak, nesne tanıma, algılama ve segmentasyon gibi çeşitli bilgisayarla görme uygulamalarında kaydettikleri çığır açan performanslar sayesinde son yıllarda büyük ilgi görmüştür. CNN, yalnızca probleme özgü senaryolarda yüksek performans göstermekle kalmamış, aynı zamanda görsel nesne tanıma görevlerinde ve özellikle hassas tıbbi uygulamalarda başarılı sonuçlar göstermiştir (Bernal vd., 2019). En popüler derin öğrenme tekniklerinden birisi olan CNN, doğrusal olmayan ilişkileri modelleyebilme yeteneğine sahiptir. Geleneksel DNN'lere göre çok daha optimize olmaları ve soyut görüntü özelliklerini öğrenmede daha başarılı olmaları CNN'leri görüntü işleme için avantajlı hale getirmektedir. Paylaşılan ağırlıklara sahip seyrek bağlantılı bir CNN benzer büyüklükteki tam bağlantılı bir derin ağdan önemli ölçüde daha az parametreye sahiptir. CNN, evrişim katmanları, havuzlama katmanları ve tam bağlı katmanlar gibi birden fazla yapı taşı kullanarak geriye yayılım yoluyla özelliklerin uzamsal hiyerarşilerini otomatik ve uyarlanabilir bir şekilde öğrenmek için tasarlanmıştır (Yamashita vd., 2018). Tipik bir yapay sinir ağında (YSA) her gizli katman ağırlıklar, girdiler ve çıktılar içermektedir. Bir CNN'deki her nöron giriş ve çıkışlar için iki boyutlu düzlemlere, ağırlıklar için ise çekirdek (kernel) adı verilen özellik haritalarına sahiptir. Evrişim katmanları giriş görüntülerinden özellikler çıkarmaktadır. Bu görüntüler ağa beslendiğinde evrişim ve havuzlama katmanlarının çıktıları özellik haritalama adı verilen iki boyutlu bir düzlem halinde düzenlenir. Bir katmanın sahip olduğu her düzlem önceki katmandan elde edilen çıktıların birleştirilmesiyle oluşmaktadır. Özelliklerin daha yüksek katmanlara yayılmasıyla birlikte boyutları da evrişim ve havuzlama katmanlarında kullanılan filtre boyutuna göre azalmakta, ancak giriş görüntülerinden en iyi özellikleri çıkarmak ve sınıflandırma doğruluğunu artırmak için özellik haritalarının sayısı artmaktadır. Havuzlama katmanları tipik olarak modellerde evrişim katmanlarından sonra

kullanılmaktadır. Konvolüsyon ve havuzlama işlemlerinden sonra çıkarılan özellikler bir vektöre dönüştürülerek ve devamında tam bağlı katmanlar kullanılarak sınıflandırma gerçekleştirilmektedir (LeCun vd., 1998; Yao vd., 2019).

2.10. Değerlendirme Metrikleri

ML algoritmalarının sınıflandırma performansları çeşitli ölçütler kullanılarak değerlendirilmektedir. Karmaşıklık matrisi (Confusion Matrix), bir sınıflandırma algoritmasının etkinliğini incelemek için önemli bir yöntemdir. Karmaşıklık matrislerinden elde edilen doğru pozitif (True Positive - TP), doğru negatif (True Negative - TN), yanlış pozitif (False Positive - FP) ve yanlış negatif (False Negative - FN) değerleri, incelenen metriklerin hesaplanmasında önemli yer tutmaktadır (Liu ve Lang, 2019). Bu çalışmada kullanılan metrikler aşağıdaki gibi özetlenmektedir:

- Doğruluk (ACC): Doğruluk, doğru sınıflandırılmış örneklerin toplam örnek sayısına oranını temsil eder. Bu metrik dengeli bir veri kümesi ile çalışırken uygundur. Ancak, gerçek dünya veri kümeleri sıklıkla dengesizlikler sergiler. Sonuç olarak, doğruluğun yanı sıra alternatif metriklerin de keşfedilmesi elzem hale gelmektedir. Doğruluk hesaplaması aşağıdaki gibidir:

$$ACC = (TP + TN) / (TP + FP + FN + TN)$$

- Kesinlik (PRC): Kesinlik, gerçek pozitif örneklerin tahmin edilen pozitif örneklere oranıdır ve aşağıdaki şekilde hesaplanır:

$$PRC = TP / (TP + FP)$$

- Duyarlılık (REC): Duyarlılık, gerçek pozitif örneklerin toplam pozitif örneklere oranıdır ve aşağıdaki şekilde hesaplanır:

$$REC = TP / (TP + FN)$$

- F1-skoru (F1): F1-skoru (F1-score), kesinlik ve geri çağırmanın harmonik ortalaması olarak tanımlanır. Veri kümeleri dengesiz olduğunda daha güvenilir sonuçlar sağladığı için genellikle diğer metriklere göre tercih edilir. F1-skoru aşağıdaki gibi hesaplanır:

$$F1 = (2 * PRC * REC) / (PRC + REC)$$

3. LİTERATÜR TARAMASI

Bu bölümde sistem ve API çağrılarını kullanarak yapılan makine öğrenimi (ML) tabanlı anomali tespitine odaklanan mevcut çalışmalar özetlenecektir.

Sistem çağrılarının veri kümeleri, örneğin Avustralya Savunma Kuvvetleri Akademisi Linux veri kümesi (ADFA-LD), çeşitli uygulamalar için normal ve kötü niyetli süreçlerin sistem çağrısı izlerini içeren etiketli verilerden oluşan mevcut koleksiyonun bir parçasıdır. Osamor ve Wellman (2022) tarafından yapılan çalışmada hem doğruluğu hem de verimliliği artırmayı amaçlayan DL tekniklerine dayalı bir hibrit anomali tespit sistemi önerilmiştir. Önerilen sistem, evrimsel sinir ağı (CNN) ve LSTM ağlarını birleştirerek oluşturulmuş ve gelişmiş tespit yeteneklerine ulaşmayı amaçlamaktadır. İlk adımda, sistem çağrısı izlerinin ham dizisi, izlerin boyutunu azaltmak için CNN ağına verilmektedir. Ardından, azaltılmış iz vektörü, sistem çağrılarının dizisini anlamak ve nihai tespit sonucunu oluşturmak için LSTM ağına iletilmektedir. Hibrit model, TensorFlow-GPU kullanılarak uygulanmış ve eğitilmiştir; performansı ise ADFA-LD veri kümesi üzerinde değerlendirilmiştir. ADFA-LD sunucu tabanlı izinsiz giriş tespit veri kümesi, Avustralya Savunma Kuvvetleri Akademisi (ADFA) tarafından oluşturulmuştur. Bu veri kümesi, Linux çekirdeği tarafından sağlanan standart arayüzler aracılığıyla kullanıcı ve çekirdek modları arasındaki iletişimi kolaylaştıran Linux sistem çağrılarını kaydetmektedir. İz dizisindeki (trace sequence) her sistem çağrısının benzersiz bir kimlik numarası bulunmaktadır. Sunucu, belirli bir zaman diliminde sistem çağrısı izlerini kaydeden güncel bir Linux sunucusunu profilemek için tasarlanmıştır. Önerilen çerçeve 3 aşamadan oluşmaktadır: veri toplama, veri ön işleme ve algoritma eğitimi. İlk aşamada ADFA-LD veri kümesi kullanılmıştır. İkinci aşamada ise, tabakalı örnekleme uygulanmış ve veri kümesi eğitim, doğrulama ve test olarak bölünmüştür. Algoritma eğitimi aşamasında, CNN ve LSTM algoritmasını birleştiren hibrit bir DL modeli eğitilmiştir. CNN, anlamlı özellikler çıkarmak için giriş görüntüsü üzerinde evrimsel işlemler gerçekleştiren evrim katmanı ve görüntülerin boyutunu azaltan havuzlama katmanından oluşmaktadır. Ayrıca, verinin doğrusal olmamasıyla başa çıkmak için doğrultulmuş lineer birim (ReLU) aktivasyon fonksiyonu kullanılmıştır. Son olarak, sistem çağrısı dizilerindeki anormallikleri tespit etmek için hibrit DL tabanlı CNN ve LSTM önerilmiştir. CNN, anlamlı özellikler çıkarmak amacıyla, LSTM ise azaltılmış özelliklerden dizinin örüntülerini öğrenmek amacıyla kullanılmıştır. Deneysel

sonular, nerilen yntemin daha kısa eēitim sresi ve geliřtirilmiř anormali tespit oranları sergilediēini, dolayısıyla yanlış alarm oranlarını azalttıēını gstermektedir. Yapılan deneyler sonucunda modelin doēruluk oranı %96 olarak tespit edilmiřtir. Bununla birlikte %0.04 kayıp oranı ve modelin alıřma zamanında da azalma gzlemlenmiřtir.

Yeni teknolojiler, zellikle konteyner platformları gibi, ortaya ıktıka, bu ortamlarda izinsiz giriř faaliyetlerini verimli bir řekilde tespit edebilen izinsiz giriř tespit sistemlerine (IDS) duyulan ihtiya artmaktadır. Bu alanda hl daha fazla geliřmeye ihtiya vardır. Bu nedenle Rocha vd. (2023) tarafından yapılan alıřmada, zellikle konteyner platformları iin izinsiz giriř tespit sistemini uygulamak amacıyla bir ereve nerilmiřtir. Bu erevede, sistem aērılarındaki anomalileri tespit etmek iin ML teknikleri kullanılmıřtır. Beř katmanlı bir mimari, serbeste eriřilebilen aralar kullanılarak oluřturulmuř ve grafik aē simlatr 3 (GNS3) kullanılarak simle edilen bir kurumsal ortamda deēerlendirilmiřtir. Bu katmanlar řunlardır: veri yakalama, zellik mhendisliēi, indeksleme ve arama, anormali tespiti, uyarı oluřturma ve veri analizi. Bu nedenle IDS alıřma srecinde eēitim ařaması, ardından test ařaması ve son olarak retim ařaması ile alıřmaktadır. Genel eriřime aık bir sistem aērısı veri kmesi kullanılarak yapılan bir deneyle, ereve katmanlarının iřlevselliēi ve entegrasyonu bařarılı bir řekilde doērulanmıř ve elde edilen tespit sonuları, orijinal veri kmesi kaynaēının sonularını ařmıřtır. Veri kmesini oluřturmak iin Sysdig aracı kullanılmıřtır. Bu ereve, konteyner platformlarına odaklanmış, sunucu tabanlı izinsiz giriř tespit sistemi (HIDS) uygulamak iin bir referans mimari iermektedir. Bu ereve, konteyner dzenleme kmesinde bir HIDS uygulamak iin entegre edilmiř ara ve prosedrlerden oluřmaktadır. Konteyner platformunda, konteynerlerin ana iřletim sisteminin ekirdeēine yaptıēı sistem aērılarındaki anomalilerin tespit edilmesi, sistem zerindeki izinsiz giriřleri veya saldırıları potansiyel olarak belirleyebilmektedir. Bu nedenle, ereve, konteyner platformlarında IDS'yi geliřtirmeyi amalamaktadır. Veri kmesi, virglle ayrılmıř deēerler (CSV) formatında birden fazla segment halinde GitHub platformu zerinde eriřilebilir olduēundan, bu segmentlerin her bir davranıř kategorisi (normal veya saldırı) iin tek bir dosyada birleřtirilmesi gerekmektedir. Verilerin Elasticsearch isimli arama ve analitik motorunda indekslenmesi iin, veri toplama aracı, her oluřturulan CSV dosyasını bireysel olarak okurken, her veri kmesi iin bir indeks oluřturulmaktadır.

Yedi saldırı türü seçilmiş ve uygulamanın anormal davranışını tespit etmek amacıyla gerçekleştirilmiştir. Her saldırı türü için yapılan anomali tespiti sonucunda, tespit bilgilerini ve elde edilen sonuçları saklamak üzere bir indeks oluşturulmuştur. Yedi farklı saldırı türü için elde edilen sonuçlar, kapsama oranı ve yanlış pozitif oranı (FPR) gibi değerler ile performans değerlendirmesi yapılmıştır. Yapılan deneyler sonucunda modelin anomali tespit doğruluk oranı %87-%97 arasında değişkenlik göstermiştir. Kesinlik oranı en az %98, kapsama (COV) ölçütü ise %99 olarak elde edilmiştir.

Peddoju vd. (2020) tarafından yapılan çalışmada, Xen hipervizörü üzerinde çalışan Windows sanal makinelerinde uygulamaların davranışlarını öğrenmek için sistem çağrılarının torbası (BoSC) temelli doğal dil işleme (NLP) kullanılarak bir anormallik tespit yöntemi uygulanmıştır. Bu süreç boyunca, sanal bellek denetimi yardımıyla hem normal uygulamalardan hem de kötü amaçlı yazılımlardan etkilenen uygulamalardan sistem çağrısı izleri çıkarılmıştır. Çıkarılan sistem çağrısı dizilerinin ön işleme, gereksiz sistem çağrılarını filtreleyerek ve düzenleyerek geçerli diziler elde etmek için yapılmıştır. Daha sonra bu dizilerin davranışları, NLP tabanlı anormallik tespit teknikleri kullanılarak analiz edilmiştir. Bu süreçte, sanal makine (VM) üzerinde çalışan kötü niyetli işlemleri tanımlamak için kosinüs benzerliği (Co-Sim) algoritması uygulanmıştır. Ayrıca, sistem çağrısı dizilerinde yer alan tehlike noktasını belirlemek için nokta tespit algoritması (PDA) uygulanmıştır. sanal makine içgörüsü (VMI), sanal makineler üzerindeki siber tehditlerin izlenmesi ve analiz edilmesi için en esnek kötü amaçlı yazılım tespit tekniği olarak öne çıkmaktadır. Bu teknik, hipervizör seviyesinde sanal makine çalışma zamanı durumunu kontrol eder. Özellik çıkarma aşamasında, anormallik tespiti için metin sınıflandırılması ile benzerlik gösteren bir teknik olan açı benzerliği tekniği kullanılmıştır. Bu teknikte, bir sistem çağrısı dizisi bir belge olarak kabul edilirken, bireysel sistem çağrıları ise bir kelime olarak görünmektedir. Simülasyon sonuçlarına göre, önerilen algoritmalar anormallikleri tespit etme konusunda %99 gibi yüksek bir doğruluk oranı sergilemektedir.

Zhan vd. (2023) tarafından yapılan çalışmada, mevcut sistemlerin pertürbasyon saldırılarına karşı olan hassasiyetini ele almak amacıyla, yenilikçi bir yöntem önerilmiştir. Önerilen yaklaşım, temsilci anlamsal bilgileri kapsayan davranış birimlerini analiz etmeye odaklanmakta ve davranış analizinin dayanıklılığını artırmaktadır. Davranış birimlerinin genel anlamını ve bağlamsal ilişkilerini anlamak

için çok seviyeli DL modelinden yararlanarak oluşturulan yöntem hem yerel hem de büyük ölçekli davranışları hedefleyen pertürbasyon saldırılarını etkili bir şekilde azaltmaktadır. Ayrıca, API gibi düşük seviyeli ve sistem çağrıları gibi yüksek seviyeli davranış kayıtları da dahil olmak üzere çeşitli türdeki davranış kayıtlarına uygulanabilirliğini göstermektedir. Yüksek düzeyde koruma modelleri nedeniyle saldırganın sınırlı bilgiye erişebileceği varsayılmıştır. Bu bağlamda, saldırganın davranışları değiştirmesini dikkate alan bir tehdit modellemesi kullanılmıştır. İlk olarak, kritik davranışsal eylemleri belirlemek için davranış dizileri tanımlanmıştır. Ardından, ilişkili davranış parçalarını çıkarmak için en uzun ortak alt dizi (LCS) algoritması kullanılarak modelin dayanıklılığı artırılmıştır. Son olarak, davranışların özelliklerini çıkarmak için çok katmanlı dönüştürücü modeller uygulanmıştır. Sistem dizisinin anormal mi yoksa normal mi olduğunu belirlemek için davranış sınıflandırması kullanılmıştır. Yapılan deneyler sonucunda elde edilen verilere göre atak enjeksiyon oranı %5-%40 oranında artmış, iki farklı veri kümesi için F1-score değerleri ise sırasıyla 0.719 ve 0.729 olarak elde edilmiştir.

Ahn vd. (2022) tarafından yapılan çalışmada, program davranışının bireysel giriş değerleri yerine tek bir karakteristik değer (CV) ile temsil edilebileceği yeni bir yaklaşım önerilmiştir. Bu CV dizisi, sinir ağlarına giriş olarak kullanılarak, program davranışının modellenmesinde artırılmış verimlilik gösterilmiştir. Bu, çoklu uzun kısa süreli bellek- tekrarlayan sinir ağları (LSTM-RNN) modelleri kullanılarak ve sinir ağının giriş alanını azaltılarak başarılmıştır. Bu metodoloji, program davranışının modellenmesinde güncel teknolojiye dayalı bir ilerlemeyi temsil ederek, daha sade ve etkili bir yaklaşım sunmaktadır. Önerilen modelin ana odak noktası, CV kullanarak program davranışını modellemektir. Program davranış modeli, CV ve fonksiyon dizileri kullanılarak program yürütme adımından sonra uygulanmıştır. Daha sonra, oluşturulan CV modellemesi dikkate alınarak program davranış anormalliği tespit aşaması gerçekleştirilmiştir. Özel bir veri kümesi kullanılarak, önerilen model mevcut olanlarla tespit doğruluğu açısından karşılaştırılmıştır. Yapılan deneylerden alınan sonuçlar, CV değerlerine göre yapılan modelleme dikkate alındığında önerilen metodun %99 doğruluk oranına sahip olduğunu ve önerilen diğer metodların tamamından daha iyi bir performans gösterdiğini ortaya koymuştur.

Chen vd. (2017) tarafından yapılan çalışmada, biyolojik sistemlerden ilham alan ve olasılıksal çıkarımlar kullanan nöromorfik anomali tespiti (AnRAD) isimli yenilikçi

bir anomali tanıma ve tespit çerçevesi önerilmiştir. Bu çerçeve, özellik bağımlılıklarını araştırır ve etiketlenmemiş verilerden verimli bir konfabülasyon (confabulation) ağı öğrenen, kendi kendini yapılandıran bir yaklaşım sunmaktadır. Bu ağ, sürekli olarak akan bilgi kümesini filtreleyerek daha hızlı artımlı öğrenmeyi sağlamaktadır. Mevcut anomali tespit yöntemleri ile yapılan karşılaştırmalı analizler, önerilen yapının anomali tespit doğruluğunun oldukça yüksek olduğunu göstermektedir. Ayrıca, AnRAD çerçevesi paralel işleme yeteneklerinden yararlanarak, grafik işleme birimleri (GPU) ve Xeon Phi yardımcı işlemciler üzerinde uygulandığında standart mikroişlemcilerdeki ardışık yürütmeye kıyasla önemli hız artışları sağlamaktadır. Çerçevenin çok yönlülüğü, çeşitli bilgi alanlarında eş zamanlı veri akışlarının gerçek zamanlı işlenmesini mümkün kılarak, birden çok yerel örüntü ile karakterize edilen büyük ölçekli sorunlara uygulanabilir hale getirmektedir. Önerilen metodoloji, belirli uygulamalara uyarlanmış esnek ağ yapılandırmalarını mümkün kılan hiyerarşik bir bilişsel mimari içinde konfabülasyon teorisinin prensiplerini içermektedir. GPU'lar ve Xeon Phi işlemcilerinin hesaplama gücünden yararlanarak farklı tür paralelleştirme yöntemleri ile kayda değer hız artışları elde edilmiştir. Yapılan deneyler sonucunda elde edilen verilere göre, önerilen model 3 farklı veri kümesi için eğri altındaki alan (AUC) değeri sırasıyla; DARPA için 0.810, ADFA-LD clean 0.888 ve ADFA-LD compromised 0.872 olarak ölçülmüştür.

Kötü niyetli işlemler, VM üzerindeki güvenlik mekanizmasında davranışlarını gizleme yeteneğine sahiptir. Bu tür saldırıların sıklığının artmasıyla birlikte, IDS'ler, ML ve DL kullanılarak geliştirilmelidir. Chaudhari vd. (2023) tarafından yapılan çalışmada, sistem çağrısı dizisi analizi kullanarak hem bilinen hem de bilinmeyen saldırıları tespit edebilen yenilikçi bir saldırı tespit çerçevesi önerilmiştir. Çerçeve, VM'lerin sistem çağrısı dizilerini, sistem çağrısı sıklığına dayanan anormallik tespit teknikleri ile LSTM modellerini birleştiren hibrit bir model kullanarak analiz etmektedir. Önerilen çerçevenin etkinliği, ADFA-LD veri kümesi kullanılarak değerlendirilmiştir. Sistem çağrısı izleri, çevrimdışı eğitim için bir veri kümesi olarak saklanmaktadır. Bu nedenle, işlenmemiş izlerden ek bilgileri ortadan kaldırarak yalnızca sistem çağrısı dizilerini oluşturmak için ön işleme gereklidir. Bu işlem sona, tüm izlerde yalnızca sistem çağrısı adları kalır. Daha sonra, tüm sistem çağrılarını benzersiz sistem çağrısı numaraları ile kodlanarak normal ve kötü niyetli izler olarak etiketlenir. İzlerin ön işleme tabi tutulmasının ardından, n-gram torbaları gibi sistem

çağrısı sıklığına dayalı yaklaşımlar kullanılarak özellik vektör matrisi (FVM) oluşturulmuş ve üretilmiştir. Toplanan izler, her izdeki farklı n-gramların oluşum sıklıklarını temsil eden sayısal vektörlerin depolandığı standart bir özellik vektörü veri tabanı dosyasına kaydedilen bir dizi n-gram özellik vektörü oluşturmak için kullanılmıştır. Deneysel sonuçlara göre, önerilen çerçevenin mevcut çerçevelere kıyasla daha yüksek doğruluk ve daha düşük yanlış pozitif oranı elde ettiği belirtilmiştir. Yapılan deneyler sonucunda, çalışmada önerilen modelin doğruluk oranı %97.2, FPR değeri ise %2.4 olarak ölçülmüştür.

Kohyarnejadford vd. (2021) tarafından yapılan çalışmada, geliştiricilere verilerdeki performans problemleri hakkında rehberlik etmek amacıyla sistem çağrıları tabanlı anomali vurgulama ve tespit çerçevesi önerilmiştir. Verileri, Linux çekirdeğinden, uygulamalardan ve kütüphanelerden toplamak için Linux izleme aracı yeni nesil (LTTng) izleme aracı kullanılmıştır. Büyük miktarda etiketlenmiş veriyi yönetmek için denetimli bir öğrenme yöntemi kullanılmıştır. Ayrıca, öğrenme yöntemi yarı-denetimli olarak geliştirilmiş ve modifiye edilmiştir. Daha sonra, en önemli çağrı dizilerinin süreleri dikkate alınarak özellik vektörü oluşturulmuştur. Tespit aşamasında, otomatik bir anomali tespit yöntemi modül bazında uygulanmıştır. Önerilen yöntemin, büyük ölçekli sistemlerde yüksek doğruluk ve verimlilik sağladığı, merkezi işlem birimi (CPU) ve bellek yetersizliklerini ayırt edebildiği ve normal davranışı tespit edebildiği belirtilmiştir. Yapılan deneyler sonucunda elde edilen verilere göre, çalışmada önerilen modelin doğruluk oranı %92 olarak tespit edilmiştir.

Ergen ve Kozat (2019) tarafından yapılan çalışmada, LSTM sinir ağlarına dayalı denetimsiz bir anomali tespiti modeli ve yenilikçi algoritmalar geliştirilmiştir. Önerilen yapı, değişken uzunluktaki veri dizilerinin LSTM tabanlı bir yapı aracılığıyla işlenmesiyle başlatılmış ve sabit uzunlukta diziler elde edilmiştir. Anomali tespiti için karar fonksiyonları, daha sonra tek sınıflı destek vektör makineleri (OC-SVM) veya destek vektör veri tanımlama (SVDD) algoritmaları kullanılarak türetilmiştir. Önerilen çalışmanın ana katkısı, LSTM mimari parametrelerinin yanı sıra OC-SVM veya SVDD parametrelerinin eşzamanlı eğitim ve optimizasyonunun, oldukça etkili gradyan ve kuadratik programlama tabanlı eğitim yöntemleri ile kolaylaştırıldığı belirtilmiştir. Önerilen anomali tespit algoritmaları, değişken uzunluktaki veri dizilerini, özellikle de zaman serisi verilerini işlemede daha iyi sonuçlar vermiştir ve geleneksel yöntemlere kıyasla daha yüksek performansla çalıştığı ortaya konmuştur.

Yaklaşımın çok yönlü olduğu, kapılı tekrarlayan birim (GRU) mimarilerine uygulanmasıyla kanıtlanmış ve yapılan deneylerde kayda değer performans iyileştirmeleri sergilenmiştir.

Mora-Gimeno vd. (2021) tarafından yapılan çalışmada, uygulamaların genel davranışını elde etmeyi amaçlayan, çeşitli tespit tekniklerini tek bir sistemde entegre eden bir saldırı tespit sistemi modeli önerilmiştir. Çalışma, çeşitli tekniklerden gelen bilgileri tek bir veri yapısı içinde entegre etmek ve temsil etmek için tasarlanmış yenilikçi ve özelleştirilmiş sistem çağruları grafiği önermektedir. Sistem, genel modelde kullanılan farklı tespit tekniklerinin sonuçlarını birleştirmek için derin sinir ağı kullanmaktadır. Önerilen yaklaşım, farklı karmaşıklık seviyelerine sahip üç veri kümesi kullanılarak doğrulanmıştır. Bu çalışmanın odak noktası, birden fazla saldırı tespit tekniğinin tek bir sistemde birleştirilmesi olup, bu entegrasyon tespit doğruluğunu artırmaktadır. Önerilen sistemin mimarisinde ilk aşamada, tespit süreci gerçekleştirilecek olup ve ikinci aşamada, birinci aşamada elde edilen sonuçlar birleştirilecektir. Tespit aşamasının ana özelliği, birden fazla saldırı tespit tekniğini aynı anda kullanabilme yeteneğidir. Farklı karmaşıklık seviyelerine sahip olmaları nedeniyle savunma ileri araştırma projeleri ajansı (DARPA), University of New Mexico (UNM) ve ADFA-LD veri kümeleri seçilmiştir. Çalışmada elde edilen sonuçlar, bireysel tekniklerin kullanımına kıyasla tespit sonuçlarında iyileşme sağlandığını göstermektedir. Önerilen model, daha yüksek tespit oranları ve daha düşük yanlış pozitif değerleri elde etmektedir. Yapılan deneyler sonucunda elde edilen verilere göre, karmaşık bir veri kümesi ile yapılan deneylerde doğruluk oranı %98.8, daha basit bir veri kümesi ile yapılan deneylerde ise doğruluk oranı %100 olarak ölçülmüştür.

Ezeme vd. (2019) tarafından yapılan çalışmada, siber-fiziksel sistemlerde çevrimiçi anomali tespiti zorluklarını ele almak için iz içi (intra-trace) ve izler arası (inter-trace) bağlam vektörlerini LSTM ağları ile kullanan bir anomali tespit modeli önerilmiştir. Yaklaşım, sistem çağrısı yürütme dizilerindeki zaman bilgilerini kullanarak anormallikleri tespit etmek için derin tekrarlamalı dikkat modeli (DReAM) kullanan dinamik bir anomali tespit şeması önermektedir. DReAM, izleri sıralı demetler (sequential tuples) olarak işlemekte ve dizilerin hem kısa hem de uzun vadeli bağlamlarını oluşturabilen bir profil oluşturarak zamansal kaymayı dikkate almaktadır. Ayrıca hem çevrimiçi hem de çevrimdışı çalışabilmektedir. Bu dinamik yaklaşım hem

tanımlanmış hem de daha önce karşılaşılmamış anormallikleri ele almayı ve çekirdek yürütme bağlamlarını hem yatay hem de dikey olarak anlamayı geliştirmektedir. Yarı yapılandırılmış dizilerde, özellikle sistem çağrılarını veya çekirdek olaylarına odaklanarak anomali tespiti için derin bağlam farkındalığına sahip bir mimari tanıtılmıştır. Ayrıca sistem kayıtları dizilerinde bulunan doğrusal olmayan yüksek boyutlu ilişkileri tanımlamaya yardımcı olan, bağlama dayalı dikkat katmanının kullanılmasının önemi vurgulanmıştır. Simülasyonda, mevcut çalışmalar ile oluşturulan özel bir veri kümesi kullanılmıştır. Modelin karmaşıklığını analiz etmek için iki temel parametre dikkate alınmıştır. Sonuç olarak, önerilen yaklaşım, RNN kullanarak çevrimiçi yürütme izi analizleri ile sistemin davranışını karakterize etmektedir. Amaç, dizilerin oluşturulma sürecini yakalamak için izleri ardışık olarak analiz etmektir. Deneyler, önerilen modelin sistem çağrısı dizilerini kullanarak anomali tespitinde etkili ve doğru sonuçlar sağladığını göstermektedir. Yapılan deneyler sonucu elde edilen verilere göre, FPR değerinde üç farklı senaryo (full, fifo ve sporadic) için sırasıyla %25, %50 ve %71 iyileşme oranları gözlemlenmiştir.

Haider vd. (2017) tarafından yapılan çalışmada, bulut sunucularının dinamik kullanım davranışlarını modellemek ve anormallikleri tespit etmek amacıyla durum özetleme ve iç içe geçmiş gizli yarı-Markov modeli (NAHSMM) önerilmiştir. Model, sistem çağrısı dizilerinin yayılımını ve daha az kullanım geçişlerini kontrol etmek için tasarlanmıştır. Ayrıca, önerilen tespit algoritması, NAHSMM ve durum özetlemesinin entegrasyonu ile oluşturulmuştur. Farklı uzunluklardaki sistem çağrılarını bu yöntemler kullanılarak özetlenmiş ve NAHSMM zaman dizilerini uyarlamak için kullanılmıştır. Temel olarak, önerilen sistem matematiksel bir model olarak yapılandırılmıştır. Ana konsept, sistem çağrısı tanımlayıcılarının dizilerinin davranışını daha iyi anlamak için durum özetlemelerini içeren yapılandırılmış sayısal modeller kullanmaktır. Eğitim ve test aşamalarında, IXIA Perfect Storm platformu veri toplamak için kullanılmıştır. Sonuç olarak, önerilen modelin makine işletim sistemleri içinde anormallikleri doğru bir şekilde tespit etmedeki etkinliği vurgulanmıştır. Tanımlayıcı özellikler ve optimize edilmiş bir yapısal çerçeve kullanarak, model daha az parametre ile bunu sağlamakta olup, hesaplama karmaşıklığını ve depolama ihtiyaçlarını önemli ölçüde azaltmaktadır. Odak noktası, bulut ortamlarındaki sunuculardan gelen sistem çağrısı tanımlayıcı dizilerini modellemek olsa da yöntemin ağ trafiğini sınıflandırma ve normalin dışında insan davranışlarını tanımlama uygulamaları için de umut verici

olduğu görülmüştür. Yapılan deneyler sonucunda elde edilen verilere göre, önerilen model %86 tespit doğruluğu ile karşılaştırıldığı modellerden çok daha yüksek tespit oranı elde ederken, aynı zamanda %7 FAR oranı ile karşılaştırıldığı diğer modellerden çok daha düşük bir FAR oranı elde etmiştir.

Shin ve Kim (2020) tarafından yapılan çalışmada, normal ve saldırı verilerini sınıflandırmak için çeşitli ML algoritmalarını kullanarak anomali tabanlı saldırı tespitinin doğruluğunu artırmaya yönelik bir yöntem önerilmiştir. Önerilen modellerin etkinliği, modern işletim sistemlerine yönelik saldırılar için sistem çağrısı izlerini içeren ADFA-LD veri kümesi kullanılarak doğrulanmıştır. Ayrıca, modeller geliştirilmiş ve ML algoritmalarına dayalı sunucu tabanlı saldırı tespit sistemlerinin etkinliğini doğrulamak için Arena simülasyon aracı kullanılarak simülasyonlar gerçekleştirilmiştir. Verilerin ön işlenmesinde N-gram yöntemi iki boyutlu matris olarak kullanılmıştır. Daha sonra, işlenmiş verileri eğitmek için SVM, lojistik regresyon (LR) ve KNN gibi ML algoritmaları kullanılmıştır. DeneySEL sonuçlara göre, önerilen metodoloji, normal ve saldırı verilerini etkili bir şekilde tespit ederek saldırıları sınıflandırır ve ML algoritmalarını kullanarak anomali örüntülerini ve özelliklerini çıkararak ve uygulayarak sunucu tabanlı saldırı tespit sistemlerinin doğruluğunu artırmaktadır. Yapılan deneyler sonucunda önerilen modelin doğruluk ve F1-score değeri %95 ve üzeri olarak ölçülmüştür.

Akagic ve Džafić (2024) tarafından yapılan çalışmada, durum tahmini sürecinde anomali tespit ve tanımlama modülü (ADIM) olarak bilinen yenilikçi bir bileşen önerilmiştir. Bu modül, durum tahmini süreci başlamadan önce anormallikleri veya hatalı veri noktalarını tespit etmeyi amaçlamaktadır ve sürekli ölçüm akışları içindeki anormallikleri tespit etmede yüksek doğruluk gösteren DL tabanlı bir yöntem sunmaktadır. ADIM, anormallikleri etkili bir şekilde tespit edip tanımlayarak kötü veri tespit bileşenine olan bağımlılığı önemli ölçüde azaltmakta ve durum tahmini sürecinin genel tepki süresini artırmaktadır. Yöntem, ölçüm kümeleri içindeki anormallikleri hassas bir şekilde tanımlamakta ve konumlarını belirlemektedir. Önerilen bileşen, üç farklı modülden oluşmaktadır: veri kümesi oluşturma, ADIM ve değerlendirme. İlk olarak, veri kümesi oluşturma modülü, seçilen güç sistemi için uygun veri kümesinin oluşturulmasını sağlamaktadır. İkinci olarak, verilerden öğrenmeyi kolaylaştırmak için bir DL mimarisi oluşturulmuştur. Son olarak, önerilen sistem, uygun değerlendirme metrikleri ile bir ölçüm kümesi kullanılarak

değerlendirilmiştir. On farklı tam bağlı katmandan oluşan ve her katmanda değişen sayıda nöron bulunan bir DL modeli geliştirilmiştir. Bir sinir ağı mimarisindeki katman ve nöron sayısının belirlenmesi genellikle yinelemeli bir süreci içermektedir. Bu yinelemeli süreç, sinir ağının optimal etkinlik için özelleştirilmesini ve yanlış sınıflandırmayı minimize etmesini sağlamaktadır. Deneyler, IEEE 6-bus, 14-bus ve 30-bus güç sistemleri ile gerçekleştirilmiştir. Benzetimlerde, durum tahmininden önce anomali tespiti için işlevsel bir sistem oluşturulmuş ve test edilmiştir. Kapsamlı deneyler sonucunda, önerilen DL mimarisinin anormallikleri etkili bir şekilde tespit edebildiği ve tatmin edici sonuçlar elde ettiği gösterilmiştir. Önerilen ADIM yöntemi, ölçüm sayısının fazla olduğu durumlarda dahi doğruluk, güvenilirlik ve sınıflandırma performansı açısından önemli faydalar sağlamaktadır. Ayrıca, gerçek zamanlı operasyon sırasında kontrol merkezinin genel güvenilirliği de artmaktadır. Üç farklı veri kümesi ile yapılan deneyler sonucunda elde edilen sonuçlar şu şekildedir; 6-bus veri kümesi için doğruluk oranı: %90.05, 14-bus veri kümesi için: %99.99, 30-bus veri kümesi için ise %99.99 olarak ölçülmüştür. F1-score değeri ise 6-bus veri kümesi için: %99.99, 14-bus veri kümesi için: %99.99, 30-bus veri kümesi için ise %99.99 olarak ölçülmüştür. Kesinlik değeri ise değişmemiştir.

Liu vd. (2020a) tarafından yapılan çalışmada, sistem çağrısı izlerinden anormallik tespiti için ilişkisiz özelliklerin çıkarılmasını amaçlayan yeni bir özellik çıkarma yöntemi önerilmiştir. Bu çıkarılan özellikler tarafından temsil edilen örnekler doğrudan çapraz platform uygulamalarında kullanılabilir. Önerilen yöntem, bir izdeki sistem çağrılarını n-gram'ların frekans dizilerine dönüştürerek başlamaktadır. Ardından, bu frekans dizilerine dayalı olarak belirli sayıda istatistiksel özellik araştırılmıştır. Özellik değerleri, doğrudan sistem çağrı dizilerine göre değil, frekans dizilerine dayalı olarak hesaplanmıştır. Eğitim kümesi içindeki yalnızca normal verilerden oluşturulan bu özellik vektörleri, anormallik tespiti için tek sınıf sınıflandırma (OCC) modelini eğitmek için kullanılmıştır. Çalışma, sistem çağrısı izlerindeki anormallik tespiti için özellik çıkarma üzerine odaklanmış olup OCC üzerine yoğunlaşmıştır. Sistem çağrısı izlerinde n-gram'ların frekans dizilerinden hesaplanan istatistiklere dayanan bir yöntem olan istatistiksel örüntü tabanlı özellik çıkarma yöntemi (STP) sunulmaktadır. STP tarafından üretilen özellikler, sistem çağrılarının isimlerinden bağımsızdır ve bu yapı özellikler üzerinde eğitilen anormallik tespit modellerinin farklı platformlarda kullanılmasına olanak tanımaktadır. Deneyler

ADFA-LD, Avustralya Savunma Kuvvetleri Akademisi Windows veri kümesi (ADFA-WD) ve yeni nesil izinsiz giriş tespit sistemi veri kümesi (NGIDS-DS) veri kümelerinde gerçekleştirilmiştir. Dört anormallik tespit algoritması kullanılmıştır. Bunlar: izolasyon ormanı (IF), yerel aykırı faktörü (LOF), OC-SVM ve KNN'dir. Karşılaştırma için, sistem çağrısı çantası, terim frekansı-ters doküman frekansı (TF-IDF), alt dizi frekansı ve n-gram tabanlı teknikler gibi özellik çıkarma yöntemleri kullanılmıştır. Ayrıca OCC için bir filtre tabanlı özellik seçimi algoritması uygulanarak her bir tam özellik kümesinden ilgisiz ve gereksiz özelliklerin çıkarılması amaçlanmıştır. Yöntem, sistem çağrısı izlerinde dört mevcut özellik çıkarma tekniği ile karşılaştırılmıştır. Yapılan çalışmalar sonucunda yöntem aynı platformda her çalıştırmada en yüksek AUC değeri elde edemese de genellikle diğer tüm yöntemlerden daha iyi performans göstermektedir. Ayrıca, çapraz platform verileri üzerinde test edildiğinde, tüm karşılaştırılan teknikler arasında en iyi performansı sergilemektedir. Yapılan deneyler sonucunda elde edilen verilere göre, AUC değerinin 1 olduğu durumda modelin doğruluk oranı %100, diğer durumlarda ise ortalama %64.9 olarak ölçülmüştür.

Liu vd. (2019) tarafından yapılan çalışmada, Word2vec tabanlı bir yaklaşımı benimseyen bir tehdit tespit modeli önerilmiştir. Şüpheli davranış olasılığı, çeşitli güvenlik kayıtlarının bir dizisi üzerinde eğitilen Word2vec modeli kullanılarak değerlendirilmiştir. Yöntem üç ana bileşenden oluşmaktadır: Log2text, Text2corpus ve anormallik tespiti. Log2text bileşeni, farklı güvenlik kayıtlarından olayları standart metin formatına dönüştürmektedir. Bu metinler daha sonra Text2corpus bileşeni tarafından birleştirilmekte, sonrasında kronolojik olarak sıralanmakta ve bir diziye dönüştürülmektedir. Son olarak, anomali tespiti bileşeni, bu dizi üzerinde bir Word2vec modeli eğitmekte ve bir olay verildiğinde belirli bir davranışın olasılığını hesaplamaktadır. Belirli bir eşiği aşan durumlar, şüpheli olarak işaretlenir ve bir kullanıcıyla ilişkilendirilen birden fazla şüpheli olay, potansiyel olarak kötü niyetli içeriden faaliyeti gösterebilmektedir. Veri kümesi daha küçük verilere bölünmüş ve belirli güvenlik kayıtları seçilmiştir. Anormallik tespitinin performansını belirlemek için doğru pozitif oranı (TPR) ve FPR başarı metrikleri kullanılarak, maliyet ve karmaşıklık açısından mevcut çalışmalar ile karşılaştırılmıştır.

Wang vd. (2021) tarafından yapılan çalışmada, yeni anomali ve saldırı tespit modelleri tasarlanmıştır. Ağ verileri, örnekler arasındaki ilişki özelliklerini

tanımlamak için bir çizge yapısı olarak temsil edilmiştir. Çizge yapısı üçlü çizge CNN olarak oluşturulmuş ve sistemdeki anomalileri tespit etmek için kullanılmıştır. Ayrıca grafik evrişimsel ağı (GCN) modellenmiş ve modelin performansını izlemek için İletişim Güvenliği Kurumu-Kanada Siber Güvenlik Enstitüsü-İzinsiz Giriş Tespit Sistemi 2018 (CSE-CIC-IDS2018) ve Yeni Güney Galler Üniversitesi- NB15 (UNSW-NB15) veri kümeleri kullanılmıştır. Veri kümesi farklı saldırı türlerini ve alt türlerini içermektedir. Eğitim aşamasında, K komşuluk sayısının değeri optimum tespit doğruluğuna ulaşmak için değiştirilmiş ve öğrenmeyi tamamlamak için KNN modeli kullanılmıştır. Önerilen zamansal grafik evrişimsel ağı (tGCN)-KNN'nin entegrasyon aşamasında küçük bir trafik veri örneği kullanılmıştır. Deneyler, tGCN ve zamansal evrişimsel sinir ağı (tCNN) öğrenme modelleri altında değişen sayıda örnek için gerçekleştirilmiştir. Sonuçlar dikkate alındığında, üç yöntemin karşılaştırılması sonucu, tGCN-KNN'nin doğruluk açısından tCNN ve CNN'den daha iyi performans gösterdiği belirtilmiştir. Yapılan deneyler sonucunda elde edilen verilere göre, önerilen tGCN modeli üç farklı veri kümesi için ortalama %96 doğruluk oranı elde etmektedir. Aynı zamanda modelin çalışma zamanının ise tCNN ve CNN modellerine göre sırasıyla %88.6 ve %81.9 olarak düştüğü gözlemlenmiştir.

Cotroneo vd. (2020) tarafından yapılan çalışmada, hata enjeksiyonu deneyleri için veri analizine yeni bir yaklaşım getirilmiş ve buna hata enjeksiyonu analitiği adı verilmiştir. Yaklaşım, ham hata verilerini toplamak için dağıtık izleme ve enjekte edilen sistemin hata modlarını tanımlamak için denetimsiz makine öğrenimini birleştirmektedir. Hata enjeksiyon deneyleri tarafından üretilen büyük miktarda veri ile başa çıkarken, analistler tarafından hata modlarının tanımlanmasını basitleştirmek amaçlanmıştır. Hata enjeksiyon deneylerinde meydana gelen olağan dışı olayları belirlemek için yeni bir anomali tespit algoritması önerilmiştir. Algoritma, bulut sistemlerinde zamanlama ve olay sıralamasındaki belirsizlikten kaynaklanan gürültüye karşı dayanıklı olacak şekilde tasarlanmıştır. Değişken sıralı bir Markov Modeli kullanılarak dağıtılmış sistem, yalnızca küçük bir hatasız yürütme kümesi kullanılarak verimli bir şekilde eğitilmektedir. Bulut bilişim sistemi, iç işleyişleri hakkında herhangi bir ön bilgiye dayanmadan, kara kutu iletişim bileşenlerinin bir koleksiyonu olarak önerilen yöntem kullanılarak analiz edilmiştir. Sonuç olarak, hata kalıplarını belirlemek için yürütme izleri üzerinde denetimsiz makine öğrenimi kullanılmıştır. Yöntem, dizilerin LCS'sini hesaplayarak diziler arasında paylaşılan sembolleri tespit

etmektedir. LCS, her iki dizide de aynı sırada bulunan sembollerin bir alt kümesidir ve orijinal dizilerden minimum sayıda sembolün çıkarılmasıyla elde edilebilmektedir. Sonuç olarak, yöntemin olasılıksal bir model kullanarak gürültülü yürütme izlerindeki anormallikleri tespit edebildiği, gerçek anormallikleri korurken yanlış alarmları etkili bir şekilde azalttığı görülmektedir. Sonuçlar, kümelemenin çeşitli koşullarda yüksek doğruluk elde edebileceğini göstermektedir. Yapılan deneyler sonucunda elde edilen verilere göre, değişken sıralı Markov model (VMM) ile LCS %90 ve üzerinde doğruluk oranı elde ettiği, gizli Markov model (HMM) ile LCS'nin ise ortalama doğruluk oranının %85 olduğu, FPR değerinin ise %39-%41 arasında değişiklik gösterdiği gözlemlenmiştir.

Kim vd. (2021b) tarafından yapılan çalışmada, Panoptikon'dan ilham alınarak dağıtılmış bir ağ saldırı tespit sistemi (NIDS) olan Panop önerilmiştir. Kötü niyetli ağ paketleri tarafından başlatılan sunucu odaklı saldırıların tespitini geliştirmek için yapay sinir ağlarından (ANN) yararlanarak nesnelerin interneti (IoT) ağları için özel olarak tasarlanmıştır. Hem ağ başlıklarını hem de cihaz davranışlarını izleyerek çalışan ve Panop ANN olarak adlandırılan yapı, ağ ve cihaz davranışları ile kötü amaçlı paketleri tanımlamak üzere kapsamlı bir şekilde eğitilmiştir. IoT ortamlarındaki performansı optimize etmek için paket özelliklerinin önemli bir göstergesi olarak cihaz davranışlarına odaklanılmıştır. Sistem, cihaz özelliklerini çıkarma, bunları ağ özellikleriyle eşleştirme ve yeni bir anormallik algılama yaklaşımı kullanma prosedürlerini içermektedir. IoT ağ koşullarını simüle etmek için Raspberry Pi cihazları kullanılarak yapılan deneylerde önerilen Panop, mevcut NIDS çözümlerine kıyasla minimum çalışma zamanı ek yükü ile gelişmiş algılama doğruluğu göstermiştir. Temel katkılar arasında, IoT ağlarına uygun hafif NIDS'nin geliştirilmesi, verimli cihaz özellik çıkarma yöntemlerinin tasarlanması ve yeni bir özellik eşleştirme şemasının uygulanması yer almaktadır. Geliştirilen yapı ile, Panop IoT ortamlarında güvenliği artırmak için umut verici bir çözüm olarak konumlandırılmış ve önemli performans dezavantajları olmadan gelişmiş algılama yetenekleri sunulmuştur. Yapılan deneyler sonucunda elde edilen verilere göre, çalışmada önerilen modelin %67 doğruluk oranı ile çalıştığı gözlemlenmiştir.

Karamanou vd. (2022) tarafından yapılan çalışmada, Yunanistan'ın resmi açık kamu verilerinin (OGD) portalından alınan trafik verilerine odaklanan bir vaka çalışması aracılığıyla bu verilerin kalitesinin değerlendirilmesi incelenmiştir. Dinamik

veri dağıtımı için bir API'den yararlanılarak, özellikle eksik değerlerin ve anormalliklerin tespit edilmesinde veri kalitesini değerlendirmek için istatistiksel ve makine öğrenimi yöntemleri kullanılmıştır. Anomali tespiti için trafik akış hızı korelasyon analizi, mevsimsel trend ayrıştırma (STL) ve denetimsiz izolasyon ormanı (iForest) gibi bir dizi teknik kullanılmıştır. Araştırma beş adımda ilerlemektedir: veri toplama, veri keşfi, kayıp değer analizi, anomali tespiti ve sınıflandırma ve çoklu değişkenler kullanarak anomali tespiti. Bu adımlar, istatistiksel analizden makine öğrenimi algoritmalarına kadar çeşitli metodolojileri içermektedir. Anomali tespiti ve sınıflandırma aşamasında, trafik verilerindeki anomaliler tespit edilmiş ve kategorize edilmiştir. Anomaliler, beklenen normal davranıştan sapan gözlemler olarak tanımlanmıştır. Analiz, anomalileri alışılmadık trafik koşullarından (örneğin kazalar veya olumsuz hava koşulları) ve sensör hatalarından kaynaklanan anomalilerden ayırmaktadır. Son olarak, hem geleneksel istatistiksel yöntemlerden hem de en yeni makine öğrenimi tekniklerinden yararlanarak veri kalitesini değerlendirmeye yönelik kapsamlı bir yaklaşım sunulmuştur. Yapılan deneyler sonucunda elde edilen verilere göre, oluşturulan model farklı sensör sayılarından alınan trafik verileri dikkate alındığında, iForest yöntemi için ortalama %11.6, Loess kullanarak STL yöntemi için ortalama %12.6 anomali tespit oranı sonuçları ortaya koymuştur.

Lv vd. (2018) tarafından yapılan çalışmada, sistem çağrılarını semantik olarak modelleyerek güçlü bir sistem çağrısı dizisinden bir tahmin modeli oluşturulmuştur. Model, gelecekte yürütülecek bir dizi sistem çağrısını öngörerek sistem durumunun izlenmesini ve saldırı davranışının tahmin edilmesini kolaylaştırmaktadır. Kötü niyetli sürecin yürütülmesi sırasında gerçekleştirilen sistem çağrısı izlerine dayalı olarak çağrılan sonraki sistem çağrılarını tahmin eden bir uçtan uca sistem çağrısı tahmin modeli oluşturulmuştur. Sistem çağrısı izleri programın yürütülmesi sırasında oluşturulmuş ve anlamsal olarak makul dizilerin üretilmesini sağlamak için sistem çağrılarının üretilmesinde RNN modeli uygulanmıştır. RNN'lerin performansı veri kümesini ölçeklendirmek için önemlidir. Modelin hem normal hem de anormal sistem çağrısı dizilerini tahmin edebildiğinden emin olmak için ADFA-LD veri kümesinde eğitim ve test kümeleri oluşturmak üzere saldırı izleri ve normal eğitim izleri kullanılmıştır. Veri kümesi hem normal programların hem de saldırı girişimlerinin yürütülmesi sırasında toplanan sistem çağrısı izlerinden oluşmaktadır. Model, iki dilli değerlendirme ölçütü (BLEU) puanı uygulanarak değerlendirilmiştir. BLEU

genellikle makine çevirisinde kullanılan bir ölçüttür. Ayrıca model çıktısının kodlanmış anlamsal vektörü ile hedef arasındaki Öklid uzaklığı hesaplanmıştır. Tahmin edilen dizi ile hedef arasındaki benzerliği değerlendirmek için TF-IDF algoritması kullanılmıştır. Hem BLEU hem de TF-IDF'nin dezavantajı, sadece metin benzerliğini kontrol etmeleridir. BLEU, iki dizideki N-tuples sayısını dikkate alırken, TF-IDF cümlelerdeki anahtar kelimelerin benzerliğini dikkate almaktadır. Ancak, dizi semantiğinin benzerliğinin değerlendirilmesi iki dizi arasındaki korelasyonu daha doğru bir şekilde belirleyebilir. Sonuçlar dikkate alındığında, eğitilen model kullanılarak, işlem yürütme sırasında çağrılan sistem çağrısı dizisine dayalı güvenilir sistem çağrısı dizisi doğru bir şekilde tahmin edilebilmektedir. Son olarak, bilinen çağrılmış sistem çağrısı izleriyle birleştirildiğinde, tahmin edilen sıra, çeşitli sınıflandırıcılar arasında saldırı tespitinin performansını büyük ölçüde artırmaktadır. Yapılan deneyler sonucunda AUC skorunda ortalama 0.1'lik bir düşüş gözlemlenmiştir. Bu durumda, önerilen modelin en olası sistem çağrısı dizisini yüksek bir doğrulukla tahmin ettiğini ve aynı zamanda dizinin işlevselliği açısından iyi bir tahmin performansı sergilediğini göstermektedir.

Karn vd. (2020) tarafından yapılan çalışmada, bir Kubernetes kümesindeki anormal kapsüller için Linux çekirdeği sistem çağrılarının izlenmesini içeren ML tabanlı bir tespit sisteminin tasarımı ve uygulaması açıklanmaktadır. Bu anomali kapsüllerinde kapsayıcı olarak çeşitli kripto madenciliği (cryptomining) veri imajları kullanılmakta ve çok sayıda sağlıklı bulut iş yükü arasında bu tür kapsülleri tanımlamak için çoklu ML modelleri oluşturulmaktadır. SHapley toplam açıklamaları (SHAP), yerel yorumlanabilir model bağımsız açıklamalar (LIME) ve önerilen ML kripto madenciliği tespit sistemindeki LSTM modelleri için yeni bir otomatik kodlama yaklaşımı ile kolay anlaşılabilir bir yapı ortaya konmuştur. Sistemin açıklanabilir modellerini karşılaştırmak ve farklılaştırmak için yedi değerlendirme ölçütü kullanılmıştır. Konteynerler içindeki uygulamaların dağıtımını, ölçeklendirilmesini ve yönetimini otomatikleştirmek için açık kaynaklı bir platform olan Kubernetes, çalışma için bir konteynerleştirme aracı olarak seçilmiştir. Çalışmanın ana katkısı, Kubernetes kümelerindeki kripto madenciliği veri kapsüllerini tespit etmek için yenilikçi bir otomatik sistem tasarlamak ve dağıtmaktır. Kubernetes podları için gerçek zamanlı sistem çağrısı çıkarma yöntemlerinin oluşturulması ve entegre edilmesi, önerilen yaklaşımın bir başka katkısıdır. Çalışmada karar ağacı (DT), topluluk öğrenimi (EL)

ve geri beslemeli RNN gibi farklı tür ML yöntemleri kullanılmıştır. Sonuç olarak, çalışma Kubernetes kümelerindeki pod anomalilerini tespit etmek için otomatik bir kurulum sunmakta ve özellikle açıklanabilir makine öğrenimi modelleri kullanarak kripto madenciliği uygulamalarını tanımlamaya odaklanmaktadır. Çalışma, sistem yöneticilerinin bilinçli kararlar almasına yardımcı olmak için açıklanabilir ML yöntemlerinin önemini vurgulamaktadır. Çeşitli kripto madenciliği algoritmalarına rağmen, sistem çağrı modellerindeki ortak özellikler, bunların tanımlanmasını kolaylaştırmaktadır. DeneySEL sonuçlar, karar ağacı modelinin en yüksek doğruluğu elde ettiğini göstermektedir. Yapılan deneyler sonucunda elde edilen verilere göre, karar ağacı modeli ile %99.6 doğruluk oranı, topluluk öğrenimi modeli ile %89.3 doğruluk oranı, ileri beslemeli vanilya yapay sinir ağı modeli ile %81.1 doğruluk oranı, geri beslemeli tekrarlayan sinir ağı ile %79.99 doğruluk oranı elde edilmiştir. Kesinlik ve duyarlılık oranları ise 1 ve daha az olarak tespit edilmiştir.

Sistem çağrısı, uygulamanın işletim sistemi ile iletişim kurmasının tek yoludur. Bu nedenle sistem çağrıları, yeni saldırıları tespit etmek üzere modelleri eğitmek için HIDS'lerde kullanılır. Önerilen veri kümeleri çoğunlukla güncel değildir. Bu sorunu ele almak için, Duan vd. (2023) tarafından yapılan çalışmada, özellikle Linux çekirdeğinin anomali tespiti için bir araya getirilen ilk büyük ölçekli veri kümesi ve ayrıca otomatik ve etkili bir çerçeve kullanılarak veri toplama sürecinin kapsamlı bir açıklaması sunulmuştur. Veri kümesi, son beş yıla ait 200'den fazla çekirdek sürümü ve 3600 hata tetikleyici program içermektedir. Buna ek olarak, veri kümesi üzerinde eğitimin diğer ilgili veri kümelerine kıyasla daha iyi genelleme yetenekleriyle sonuçlandığını göstermek için çapraz veri kümesi değerlendirmesi yapılmıştır. Çalışmanın ana katkısı, Linux çekirdeklerindeki anomali tespiti için sistem çağrılarına dayalı kapsamlı bir veri kümesidir. Çerçeve, dağıtık bir mimari içinde büyük miktarda anormal sistem çağrısı verisini etkili bir şekilde toplayabilmektedir. Veriler daha sonra eğitim, doğrulama ve test alt kümelerine bölünerek Linux çekirdeklerindeki anormallikleri otomatik olarak tespit etmek için bir DL modeli eğitilmiştir. Önerilen veri kümesi özellikle Linux çekirdeklerinin anomali tespiti için önerilen sistem çağrılarına dayalı ilk büyük ölçekli veri kümesidir. Deneylerde, çapraz veri kümesi değerlendirmesi için dört veri kümesi seçilmiştir. Performans değerlendirmesi için CNN/RNN, LSTM, WaveNet ve ampirik kümülatif dağılım fonksiyonları (ECOD) yöntemleri incelenmiş ve karşılaştırılmıştır. Deneyler sonrasında iki önemli gözlem

yapılmıştır. İlk olarak, veri kümesi üzerinde eğitilen model en yüksek genelleme puanlarını göstermektedir. İkinci olarak, anormal veriler üzerinde eğitilen model, normal veriler üzerinde eğitilen modele kıyasla daha iyi genelleme yeteneği sergilemektedir. Son olarak, çerçevenin, veri kümesi üretmek için harcanan zamanı önemli ölçüde azalttığı gözlemlenmiştir. Yapılan deneyler sonucunda elde edilen verilere göre, tespit doğruluk oranını da yansıtan F1-score değeri % 84.4 olarak tespit edilmiştir.

Ezeme vd. (2020) tarafından yapılan çalışmada, özellik tabanlı bir anomali tespit çerçevesi olan PESKEA tanıtılmaktadır. PESKEA, izleri sınıflandırmak ve ardından anomalileri tespit etmek için gömülü bir işletim sisteminin yürütme izlerinde bulunan istatistiksel değişkenliği kullanmaktadır. Çalışma, gömülü sistemin performansını anlamak için işletim sistemi yürütme izlerini inceleyerek çevrimdışı izleme yönüne odaklanmaktadır. Önerilen çerçeve, sistemin iç işlemleri göz ardı edildiği için kara kutu testi kategorisine girmektedir. Çalışmada iki sebepten dolayı çevrimiçi analiz tercih edilmiştir. Birincisi, izler cihazın rutin çalışmasından kaynaklanmaktadır ve sistem performansına müdahale edebilecek özel araçlar gerektirmeden kolayca erişilebilir durumdadır. İkincisi, kümeler arasındaki sınırlar bulanıksa veya üst üste biniyorsa, bu durum sınıflandırıcının doğruluğunu etkileyebilmektedir. Bu çalışmanın ana katkısı, çekirdek izleme olaylarından elde edilen öznetelikleri kullanarak gömülü sistemler için özellik tabanlı bir anomali tespit sınıflandırıcısı geliştirmektir. PESKEA, bir girdi dizisinin özellik işlemlerinden başlayarak girdi içindeki anormallikleri belirlemeye kadar kapsamlı bir anormallik tespit sürecini temsil etmektedir. Çerçeve, hiyerarşik bir kümeleme tekniği olarak inşa edilmiştir. PESKEA, değişen koşullara sahip iki farklı veri kümesi üzerinde test edilmiştir. Birincisi, sahadan alınan herkese açık bir veri kümesi, ikincisi ise PESKEA'nın performansını ölçmek için bazı saldırı senaryoları tasarlanarak ve veriler kaydedilerek oluşturulan bir veri kümesidir. Sonuç olarak, deneyler gerçekleştirildikten sonra, ilk veri kümesi testinden elde edilen sonuçların, bu yaklaşımın çekirdek olay izlerinin incelenmesi yoluyla olay tetiklemeli bir gömülü işletim sistemi tarafından yönetilen bir kontrol uygulamasının çalışmasındaki anormallikleri analiz etmek ve tespit etmek için yetkin olduğu görülmektedir. Öte yandan, ikinci veri kümesinin testlerinde, çerçevenin saldırı senaryolarının karmaşıklığı arttıkça performansın düştüğü gözlemlenmiştir. Doğru negatif oranındaki (TNR) yüksek performans, gömülü sistemlerdeki anormallikleri

tespit etmek için geçerli bir seçenek olduğunu kanıtlamaktadır. Yapılan deneyler sonucunda elde edilen verilere göre, önerilen modelin TPR oranında %3-%6 oranında iyileşme sağladığı ve TPR oranının %100 olarak tespit edildiği, TNR oranının ise %88.37-%100 arasında değişkenlik gösterdiği gözlemlenmiştir.

Kim vd. (2021a) yapılan çalışmada, anormal davranışları tespit etmek için log gömme algoritmaları aracılığıyla sıralı bilgi korumayı ML odaklı anomali tespit algoritmalarıyla entegre eden uçtan uca bir yaklaşım önerilmektedir. Log verilerinden anlamlı özellikler çıkarmak için alan uzmanlarına dayanan sistem anomalisi tespitine yönelik diğer ML modellerinin aksine, önerilen yaklaşım işlenmemiş log verilerini orijinal uzunluklarından bağımsız olarak sabit boyutlu sürekli vektörlere dönüştürmektedir. Bu vektörler daha sonra anomali tespit modellerini eğitmek için kullanılmaktadır. Bu çalışmada, Linux ortamları için özel olarak oluşturulan ve anomali tespit algoritmalarının yanı sıra, sıralı bilgiyi koruyan log gömme tekniklerinden yararlanan sağlam bir saldırı tespit modeli önerilmektedir. Farklı uzunluklardaki sistem çağrısı izlerini sabit boyutlu gerçek değerli vektörlere dönüştürmek için Doc2vec'in yanı sıra sıralı bilgileri koruyan RNN tabanlı otomatik kodlayıcı (RNN-AE) ve RNN tabanlı gürültü giderici otomatik kodlayıcı (RNN-DAE) yöntemleri kullanılmaktadır. Tespit modelini doğrulamak için Linux sistem çağrısı izlerini içeren ADFA-LD veri kümesi kullanılarak bir deney gerçekleştirilmiştir. Anomali tespit modellerinin performans değerlendirmesi için üç değerlendirme ölçütü kullanılmıştır. Bu metrikler; alıcı işletim karakteristik eğrisi altındaki alan (AUROC), eşit hata oranı (EER) ve isabet oranıdır. Alıcı işletim karakteristiği (ROC) eğrisi her bir modelin performansını tespit etmek için kullanılmıştır. Deneylerde ilk olarak denetimsiz öğrenmeye dayalı anomali tespit algoritmalarının performansı farklı saldırı türlerinde değerlendirilmiştir. Daha sonra yeterli miktarda etiketli veri toplanarak denetimli bir sınıflandırma modeli eğitilerek performansı anomali tespit algoritmalarıyla karşılaştırılmıştır. Sonuç olarak, model eğitimi için etiketli veriye ihtiyaç duymadan çalışan denetimsiz ML tabanlı bir sistem anomalisi tespit çerçevesi önermektedir. Deneysel sonuçlar, önerilen log kayıtlarını gömme yönteminin denetimsiz bir anomali tespit modeli ile verimli performans sağladığını göstermektedir. Yapılan deneyler sonucunda çalışmada önerilen modelin doğruluk oranı %88.28 olarak tespit edilmiştir.

Geleneksel Markov zinciri modelinin üzerine, Sha vd. (2015) tarafından yapılan çalışmada, anomali tespiti için çok sıralı Markov zinciri tabanlı bir çerçeve önerilmektedir. Yaklaşım, istatistiksel öğrenme çerçevesinde özetlenen bir eğitim süreci ile birlikte algoritmalarda açıklanan bir çerçeve oluşturmak için hem yüksek dereceli Markov zincirlerini hem de çok değişkenli zaman serilerini kullanmaktadır. Zaman ve alan karmaşıklığını azaltmak için algoritmalar, başlangıç ve geçiş matrislerinde sıfır olmayan değer tabloları ve logaritmik değerler kullanılarak hazırlanmış ve yürütülmüştür. Yaklaşımı doğrulamak için, iyi bilinen DARPA saldırı tespit değerlendirme veri kümesinden bir dizi sistem çağrısı ve bunlara karşılık gelen geri dönüş değerleri çıkarılmış ve iki boyutlu bir test girdi kümesi oluşturulmuştur. Çalışma, ML tabanlı yaklaşımlara ve istatistiksel yaklaşım kategorilerine odaklanmaktadır. Çalışmanın en belirgin katkısı, karma sıraya sahip Markov zincirlerinin kombinasyonunun, daha düşük maliyetle herhangi bir tek sıralı zincire göre önemli derecede daha yüksek performans göstereceğini ortaya koyan çok sıralı Markov zincirlerine dayalı yaklaşımdır. Bu yaklaşım, karışık sıralı Markov zincirlerinin kullanılmasının, tek sıralı olanlara kıyasla önemli iyileştirmelere sağladığını ve bu iyileştirmenin düşük bir maliyetle gerçekleştiğini göstermektedir. Sonuç olarak, farklı derecelerdeki modeller tarafından üretilen çıktılar arasındaki karşılaştırmalı ilişkileri izleyerek, anormallikleri tespit etmek için yeni ve etkili bir ölçüt ortaya konmuştur. Deneysel sonuçlar, çok dereceli tekniğin daha etkili sonuçlar üretebildiğini göstermektedir.

Van Leeuwen ve Koole (2023) tarafından yapılan çalışmada, tek değişkenli zaman serilerinde anomali tespiti ve yeni bir aktif öğrenme yaklaşımı kullanılarak bir iş hedefine göre optimizasyon önerilmiştir. Önerilen yöntem, parametreleri aktif bir öğrenme bileşeni aracılığıyla ince ayarlanmış bir tahmin aralığı ile birleştirilmiş hareketli ortalamalara dayanmaktadır. Tespit edici beyaz kutu niteliğindedir, bu nedenle tahmin aralıkları kullanılarak sonuçlar alan uzmanlar için kolayca yorumlanabilmektedir. Alan uzmanlarından gelen ek açıklamalar şeklindeki girdiler, Gauss sürecini kullanan Bayes optimizasyonu yoluyla parametre elde etmek için kullanılmaktadır. Anomali tespit sürecini etkileyen dört yapılandırılabilir parametre belirlenmiştir ve bunlara ek açıklamalara dayalı olarak ince ayar yapılabilmektedir. Yöntem, ek açıklamalardan hızlı bir şekilde geri bildirim almayı ve bunun sonucunda gerekli yüksek kaliteli ek açıklamaları elde etmeyi amaçlamaktadır. Tahmin aralığının

(PI) hesaplanması ya kayan pencerenin veri noktalarındaki bilgiye ya da mevcut veri noktalarının toplam sayısına dayanmaktadır. PI, parametrik ve parametrik olmayan olmak üzere iki varsayım üzerinden oluşturulabilir. Tek bir parametre yerine, hareketli ortalamanın (MA) üstünde veya altında davranış toleransı iki ayrı parametreye bölünmüştür. Bu çerçevede, anomalilerin tespitini etkileyen özel ihtiyaçlara göre MA'nın belirli bir tarafındaki toleransın ince ayarının yapılmasını sağlamaktadır. Parametrik yaklaşım, MA içindeki veri noktalarının belirli bir dağılıma uyduğu varsayımına dayanmaktadır. Parametrik olmayan yaklaşım varsayımlara dayanmamaktadır. Diğer bir ifadeyle MA içindeki veri noktalarının belirli bir dağılıma uyduğu varsayılmamaktadır. Kararlı bir sonuç elde etmek için bu yaklaşım, parametrik yaklaşıma kıyasla kayan pencere içinde daha fazla sayıda veri noktası gerektirmektedir. Yeterli veri noktası elde edildiğinde, parametrik olmayan yaklaşıma geçiş başlatılabilmektedir. Bu yaklaşım, belirtilen her hiper parametre kombinasyonu için bir model oluşturmakta ve değerlendirmektedir. Teorik olarak, bu yöntem en uygun çözümü bulmaktadır çünkü her parametrenin tanımlanmış alt ve üst sınırları vardır ve ayrık bir ızgara tanımına izin vermektedir. Kapsamlı aramadan kaçınmak için Gauss sürecini kullanan Bayes optimizasyonu, amaç fonksiyonunu en aza indirmek için hiper parametre uzayını araştırmaktadır. Bu süreç, hiper parametreleri modelin verilere uyumunun bir göstergesi ile eşleştirmektedir. Bu gösterge, Gauss süreci regresyonu kullanılarak eşlenen değerler arasındaki güven sınırlarını hesaplamak için kullanılmaktadır. Bayesian optimizasyonu, daha az iterasyon gerektiren avantajları nedeniyle hiper parametreleri elde etmek için tercih edilen yöntem olarak seçilmiştir. Tespit cihazı yaygın olarak kullanılan kara kutu modelleriyle karşılaştırılmıştır. Numeta gibi önde gelen algoritmalar, tüm uygulama profillerinde önemli ölçüde üstün performans göstermektedir.

Tablo 3.1. Literatür çalışmalarının genel özeti

Referans	Katkı	Yöntem
Akagic ve Dzafic (2024)	• Güç ağı sistemlerindeki anormallikleri tespit etmek için derin bir öğrenme yaklaşımı • Güç ölçüm kümeleri için anormallik tespiti • Özelleştirilmiş modül kullanılarak güç sistemi verilerinin üretilmesi • Anomali tespitinde dengesiz veri kümesi sorunlarını çözme.	• Özelleştirilmiş, gerçek zamanlı kümesi oluşturma • DNN kullanarak anormallik tespiti ve tanımlanması
Duan, vd. (2023)	• Anormallik tespiti için otomatik sistem çağrısı toplayıcı • Linux çekirdeklerinde sistem çağrılarını içeren büyük ölçekli bir veri kümesi • CNN, LSTM gibi çeşitli modelleri kullanan derin öğrenme tabanlı anormallik tespiti	• Linux sisteminden gerçek zamanlı çağrı kayıtlarının toplanması • CNN/RNN, LSTM, WaveNet ve ECOD ile anomali tespiti
Rocha, vd. (2023)	• Açık kaynaklı uygulamalarda anomali tespit ve uyarı çerçevesi • Sistem çağrılarındaki farklı parametreleri dikkate alarak anomali tespiti • Makine öğrenimi yöntemleri ile konteyner platformlara uygulanan IDS çerçevesi	• Elasticsearch API ile verinin okunması ve yazılması • Kubernetes, CNN, birleşik hesaplama mimarisi derin sinir ağı (cuDNN) ile anomali tespiti
Chaudhari, vd. (2023)	• Bulut sistemleri için frekans ve LSTM odaklı anomali tespit çerçevesi • Sistem çağrıları kullanılarak LSTM mimarisi ile anomali tespiti • Geçmiş çağrı dizilerinin frekansına dayalı saldırı tabanlı sınıflandırma	• Saldırı sınıflandırması için LSTM tabanlı mimari • TF-IDF ile özellik seçilmesi ve NN kullanarak saldırı sınıflandırması
Zhan, vd. (2023)	• Davranış birimlerinin analiz edilmesi ile anomali tespiti • Kritik davranışlara odaklanarak sistem çağrıları ve API düzeyinde tehdit analizi	• Çok seviyeli DL modeli ile anomali tespiti • Çok seviyeli transformatör tabanlı model ile saldırı önleme
Osamor ve Wellman (2022)	• Derin öğrenme tabanlı anomali tespiti	• LSTM ve CNN ile hiperparametre optimizasyonu ve anomali tespiti

- İyileştirilmiş sinir mimarisi ile eğitim süresi verimliliği

Tablo 3.1. Literatür çalışmalarının genel özeti (Devamı)

Referans	Katkı	Yöntem
Ahn, vd. (2022)	• Program davranışlarının CV üzerinden analizi • Fonksiyon parametrelerinin CV dizisi ile değiştirilerek program davranışı modeli • Kelime gömme yöntemi ile sinir ağı vektörü oluşturma ve prototip tasarımı	• LSTM-RNN ile CV değerleri üzerinden öğrenme yöntemi
Karamanou, vd. (2022)	• API'ler kullanarak gerçek zamanı veri kümesinin analizi • API'ler üzerinden, trafik akışı dikkate alınarak makine öğrenmesi ile anomali tespiti	• STL ve iForest ile anomali tespiti ve sınıflandırması
Cotroneo, vd. (2020)	• Bulut sistemleri için anomali tespit algoritması tasarımı • Hatalı durumların tespiti için özelleştirilmiş sınıflandırma metodu • Farklı alt sistemler için tasarlanmış senaryolar üzerinden durum senaryosu tasarımı	• Durum olasılığı tahmini için Markov modeli • Kısmi eşleme ile tahmin-metod C (PPM-C) ve VMM ile modelleme
Wang, vd. (2021)	• Durum verilerini modellemek için çizge modeli • Ağ trafik ve sistem durum verileri ile GCN modellemesi	• Hibrit tGCN-KNN yöntemi ile izinsiz giriş ve anomali tespiti
Karn, vd. (2021)	• Kubernetes kümesindeki anormal pod'ların ML tabanlı bir tespit sisteminin tasarımı • Linux sistem çekirdeği çağrılarını kullanarak sistem modellenmesi	• SHAM ve LIME ile sınıflandırma açıklaması • Karar ağaçları, ANN ve EL ile makine öğrenmesi implementasyonu
Kim, vd. (2021a)	• Linux ortamı için sistem çağrılarını üzerinden anomali tespiti • Makine öğrenimi odaklı uçtan uca sistem kayıtları ile anomali sınıflandırması	• Doc2Vec, RNN-AE ve RNN-DAE ile özellik vektörü dönüşümü • IF, LOF ve 1-SVM ile anomali tespiti

Ezeme, vd. (2020)	• Özellik tabanlı anomali tespiti ve sınıflandırma yöntemi	• Hiyerarşik kümeleme algoritmaları ile anomali tespiti
Mora-Gimeno, vd. (2021)	• Uygulama davranışlarından yola çıkarak çoklu anomali tespit sistemi • Sistem çağrıları numaraları ve frekansları üzerinden ağırlıklı çizge temsili • Sistem çağrılarını kullanarak derin sinir ağı (DNN) ile IDS modelleme	• Dinamik Çok Katmanlı Algılayıcı (D-MLP) ile sinir ağı için üç bileşenli özellik vektörü yöntemi • Sıralı zaman gecikmeli gömme (STIDE), metin sınıflandırma ve sistem çağrı çizgesi ile anomali tespiti

Tablo 3.1. Literatür çalışmalarının genel özeti (Devamı)

Referans	Katkı	Yöntem
Kohyarnjadfard, vd. (2021)	• Geliştiriciler için sorun giderme süresini kısaltarak anomali tespiti • Linux ortamında işlem esnasında sistem çağrılarının toplanması	• Toolkit Next Generation ile sistem çağrılarının toplanması • Çok sınıflı SVM ile sınıflandırma • K-Ortalama (K-means) ve gürültülü uygulamaların yoğunluk tabanlı mekansal kümelenmesi (DbSCAN) ile kümeleme
Kim, vd. (2021b)	• IoT ağları için NIDS yöntemi özelleştirmesi ile Panop isimli anomali tespiti • Ağ ile ilgili cihaz davranışlarından yola çıkarak özellik çıkarımı • Raspberry Pi cihazı kullanarak gerçek zamanlı senaryo temsili	• Kitsune ve ANN ile anomali tespiti
Shin and Kim (2020)	• Normal ve saldırı verileri kullanarak anomali tespit doğruluğu için ML tabanlı yöntem	• Özellik vektörü oluşturmak için N-gram tekniği • SVM, LR ve KNN yöntemleri ile sınıflandırma ve tahmin
Liu, vd. (2020a)	• Sistem çağrılarının isimlerinden bağımsız özellikleri kullanarak özellik çıkarma yöntemi • Farklı platformlarda kullanılabilecek şekilde tasarlanmış düşük maliyetli özellik çıkarma yöntemi	• Sistem çağrılarını frekans dizilerine dönüştürmek için N-gram tekniği • IF, LOC, OC-SVM ve KNN ile sınıflandırma

Ergen ve Kozat (2019)	• LSTM çerçevesi ile anomali tespiti • LSTM ve danışmansız öğrenme çerçevesi ile hibrit tespit modeli	• OC-SVM, LSTM ve SVDD ile sınıflandırma
Liu, vd. (2019)	• Sistem güvenlik kayıtları kullanarak esnek anomali tespit sistemi • Sistem içerisinde yer alan tehditlerden anlamsal özellik çıkarımı ve tehdit davranış modellemesi	• Word2vec ile model eğitimi

Tablo 3.2. Literatür çalışmalarının yöntem analizi

Referans	ML Yöntem	Platform	Başarı Metriği
Akagic ve Dzafic (2024)	DNN	Genel	Doğruluk, kayıp oranı, TNR, kesinlik, F1-değeri, FPR, duyarlılık oranı, yanlış negatif oranı (FNR), G-ortalama
Duan, vd. (2023)	CNN/RNN, LSTM, WaveNet, ECOD	Linux Kernel	FPR, F1-değeri zaman verimliliği, AUC, TPR
Rocha, vd. (2023)	DNN, CNN	Container Platforms (Kubernetes cluster)	Doğruluk, negatif tahmin değeri (NPV), kapsama, hassasiyet, FPR, F1-değeri, ROC
Chaudhari, vd. (2023)	LSTM	Linux, Cloud	Doğruluk, kayıp oranı, FPR, F1-değeri, duyarlılık oranı
Zhan, vd. (2023)	Çok katmanlı DL	Android	Kesinlik, F1-değeri, ROC
Osamor ve Wellman (2022)	LSTM, CNN	Linux	Doğruluk, kayıp oranı, tespit oranı, yanlış alarm oranı (FAR), eğitim süresi
Ahn, vd. (2022)	LSTM-RNN	Genel	Doğruluk, tespit oranı, AUROC, doğruluk-duyarlılık eğrisi altındaki alan (AUPR), CPU çevrim sayısı,

				karmaşıklık, bellek kullanımı
Karamanou, vd. (2022)	iForest	Sensör cihazlar içeren akıllı trafik sistemi		Tespit oranı
Cotroneo, vd. (2020)	Verilmemiş	Bulut Sistemler		FAR, zaman verimliliği, TPR, hesaplama maliyeti
Wang, vd. (2021)	tGCN-KNN	Genel		Doğruluk, kesinlik
Karn, vd. (2021)	DT, ANN	Container Clouds		Kayıp oranı, F1-değeri, kesinlik, geri dönüş oranı
Kim, vd. (2021a)	1-SVM, LOF, İForest	Linux		AUROC, AUC
Ezeme, vd. (2020)	Hiyerarşik kümeleme	Kernel Events, işletim sistemi (OS)		Kesinlik, FPR, karmaşıklık, FNR
Mora-Gimeno, vd. (2021)	D-MLP	Linux		Tespit oranı, FPR, ROC
Kohyarnjadfard, vd. (2021)	SVM, K-means, Dbscan	Linux		Doğruluk, zaman verimliliği
Kim, vd. (2021b)	ANN	Genel, IOT		Doğruluk, FPR, TPR
Shin ve Kim (2020)	SVM, LR, KNN	Linux		Doğruluk, AUC, ROC
Liu, vd. (2020a)	IF, LOF, OC-SVM, KNN	Linux, Windows		FPR, hesaplama maliyeti, TPR
Ergen ve Kozat (2019)	OC-SVM, LSTM, SVDD	Genel		AUC, ROC
Liu, vd. (2019)	Word2vec	Genel		FPR, TPR, hesaplama maliyeti

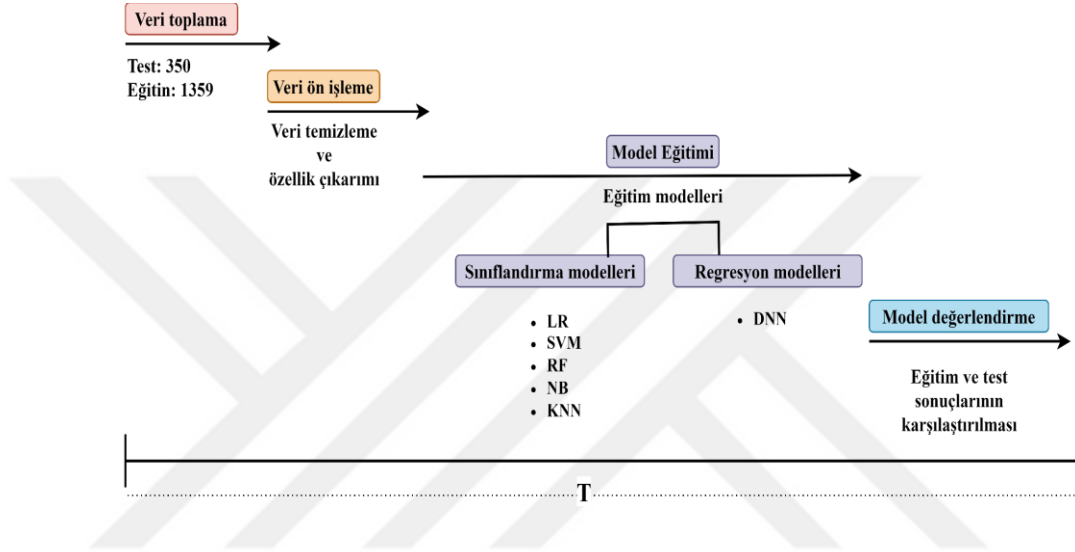
Tablo 3.3. Literatür çalışmalarının simülasyon bakımından karşılaştırılması

Referans	Veri Kümesi	Veri Sayısı	Platform
Akagic ve Dzafic (2024)	Sentetik	946.000 - 3.800.000	Verilmemiş
Duan, vd. (2023)	Firefox-SD, ADFA-LD, PLAID ve Sentetik	18.966	Intel Xeon E5-2640 V4, DDR4 ECC 2400 MHz, 128G bellek, 4 GPU
Rocha, vd. (2023)	Sentetik	1.551.006	Linux OS, GeForce RTX 3060
Chaudhari, vd. (2023)	ADFA-LD	Verilmemiş	Ubuntu 18.04 OS, i5 işlemci, 8 GB RAM, 500 GB HDD ve Xen 4.6
Zhan, vd. (2023)	AndroCT	35,974+	Windows 10 OS, AMD Ryzen 7 5800 8-Core İşlemci, 3.40 GHz, NVIDIA 3060 ve 32.0 GB RAM.,
Osamor ve Wellman (2022)	ADFA-LD	Verilmemiş	Verilmemiş
Ahn, vd. (2022)	Sentetik	Verilmemiş	Verilmemiş
Karamanou, vd. (2022)	Sentetik	5748-15.000.000+	Verilmemiş
Cotroneo, vd. (2020)	Verilmemiş	Verilmemiş	Intel Xeon sunucu (E5-2630L v3 @ 1.80 GHz), 16 GB RAM, 150 GB, ve Linux CentOS v7.0

Wang, vd. (2021)	CSE-CIC-IDS2018 ve UNSW-NB15, sentetik	257.673	Intel Xeon (Cascade Lake) Platinum 8269 2.5GHz/3.2GHz 4-core CPU, 8GB RAM
Karn, vd. (2021)	Sentetik	Verilmemiş	Verilmemiş
Kim, vd. (2021a)	ADFA-LD	5206	Intel (R) Core (TM) i7-7700K CPU 4.20GHz, NVIDIA GTX 1080Ti 11GB RAM, RAM:32G
Ezeme, vd. (2020)	Sentetik	Verilmemiş	Intel Core i5-6200U CPU 2.3 GHz ve 2.4 Ghz, 8GB RAM
Mora-Gimeno, vd. (2021)	DARPA, UNM, ve ADFA_LD	UNM için 2766, diğerleri oturum tabanlı değişken	Intel Core i7-4790 CPUs, 16 GB RAM
Kohyarnjadfard, vd. (2021)	Sentetik	Verilmemiş	Intel Core i7 4 GHz, 8 CPU, 32 GB
Kim, vd. (2021b)	Sentetik	50.000	Verilmemiş
Shin ve Kim (2020)	ADFA-LD	Verilmemiş	Verilmemiş
Liu, vd. (2020a)	ADFA-LD, ADFA-WD ve NGIDS-DS	ADFA-LD 5951, ADDFA-WD 7725 NGIDS-DS 3070	Verilmemiş
Ergen ve Kozat (2019)	Occupancy, HHE, Http ve Alcoa	Verilmemiş	i5-6400, 2.7 GHz CPU, 16 GB RAM
Liu, vd. (2019)	CMU CERT v6.2	200.000	Verilmemiş

4. API ÇAĞRILARI KULLANILAN MAKİNE ÖĞRENİMİ MODELLERİ İLE ANOMALİ TESPİTİ

Bu bölümde, makine/derin öğrenme modellerinin eğitim ve test aşamalarında kullanılan veri kümesinden bahsedilmektedir. Ardından, veri kümesi üzerinde gerçekleştirilen ön işleme aşaması açıklanmıştır. Daha sonra modellerin eğitim ve test aşamaları detaylandırılmıştır. Ön işleme, eğitim ve test aşamaları arasındaki ilişki Şekil 4.1'de gösterilmiştir.



Şekil 4.1. Ön işleme, eğitim ve test aşamaları arasındaki ilişki

4.1. Veri Kümesi

Uzun süredir devam eden sistem veya uygulama API çağrıları, sistemin veya uygulamanın davranışını temsil edebilir. Dolayısıyla API çağrılarından çıkarılan çeşitli özellikler ile sistem veya uygulama davranışında bir anormallik olup olmadığı tespit edilebilir.

Bu çalışmada, makine/derin öğrenme modellerini geliştirmek ve test etmek için (Kaggle, 2022)'deki veri kümesinin çeşitli API çağrı özelliklerini içeren bir dosyası kullanılmıştır. Veri kümesi ikisi "CSV" ve ikisi "JSON" formatında olmak üzere dört dosya içermektedir. Bu dosyalardan "supervised dataset.csv" dosyası model geliştirmek ve test etmek amacıyla veri kümesi olarak kullanılmaktadır. "supervised dataset.csv" dosyasında 1106'sı normal, 592'si anomali olmak üzere toplam 1699 örnek bulunmaktadır. Bununla birlikte, her bir örneği temsil eden API çağrılarının özelliğinin tanımları Tablo 4.1'de verilmiştir.

Tablo 4.1. Özelliklerin tanımı

Özellik	Tanım	Örnek Değer
inter_api_access_duration	Bir oturumdaki ardışık API çağrıları arasındaki medyan zaman aralığı	0.011626192826587695
api_access_uniqueness	Bir oturumdaki farklı API sayısının toplam API çağrısına oranı	0.029285949325436
sequence_length	Bir oturumdaki ortalama toplam API çağrısı sayısı	60.78
vsession_duration	Bir pencere içerisindeki oturumun dakika cinsinden zaman değeri	4240
ip_type	IP erişim türü	varsayılan
num_sessions	Oturum sayısı	107.0
num_users	Aynı API çağrı dizilerine sahip kullanıcı sayısı	100.0
num_unique_apis	Farklı API çağrılarınin sayısı	178.0
source	Veri kaynağı	E
classification	Sınıf etiketi	normal

4.2. Ön İşleme Aşaması

Ön işleme aşamasında API çağrılarının makine öğrenimi ve derin öğrenme modellerinde girdi olarak kullanılacak özellikleri belirlenmektedir. Bu doğrultuda veri kümesi incelendiğinde "ip type" özelliğinin %91 oranında varsayılan (default) değere sahip olduğu görülmektedir. Öte yandan "kaynak" özelliği için 'E' değerine sahip örneklerin oranı %88'dir. Dolayısıyla bu iki özelliğin eğitim ve test aşamalarında herhangi bir ayırt edici özelliğe sahip olmadığı sonucuna varılmaktadır. Bu nedenle modellerin eğitim ve test aşamalarında bu iki özellik kullanılmazken diğer tüm özellikler kullanılmıştır.

4.3. Eğitim Aşaması

Her modelin eğitim aşamasında, ön işleme aşamasından geçmiş olan aynı veri kümesi kullanılmıştır. Eğitim aşamasından önce, veri kümesi rastgele olarak %80 eğitim %20 test verisi olacak şekilde iki bölüme ayrılmıştır.. Eğitim veri kümesi 877 normal örnek ve 482 anomali örneği olmak üzere toplam 1359 örnek içermektedir. Öte yandan eğitim aşamasının gerçekleştirilme şekli modellere göre farklılık göstermektedir. Modellerin eğitim aşamalarında kullanılan parametreler ve nasıl uygulandıkları aşağıda sırasıyla açıklanmıştır.

1. LR: Bu modelin eğitim aşamasında tüm parametreler Scikit-learn kütüphanesindeki varsayılan değerleriyle kullanılmıştır.
2. SVM: Bu modelin eğitim aşamasında "kernel" parametresinin tipi "linear" olarak seçilmiştir. "Kernel" parametresi dışındaki tüm parametreler Scikit-learn kütüphanesindeki varsayılan değerleri ile bırakılmıştır.
3. NB: Gauss dağılımını kullanan bu modelin eğitiminde tüm parametreler Scikit-learn kütüphanesindeki varsayılan değerleriyle kullanılmıştır.
4. RF: Bu modelin eğitim aşamasında "n_estimators" parametresinin değeri "10" olarak ayarlanmıştır. "criterion" parametresinde ise "entropy" kalite metriği seçilmiştir. Bu parametreler dışındaki tüm parametreler Scikit-learn kütüphanesindeki varsayılan değerlerdir.
5. KNN: Bu modelin eğitim aşamasında "n_neighbors" ve "p" parametrelerine sırasıyla "5" ve "2" değerleri atanmıştır. "Metric" parametresinde ise "Minkowski" uzaklık metriği ayarlanmıştır. Bunlar dışındaki tüm parametreler Scikit-learn kütüphanesindeki varsayılan değerlerdir.
6. DNN: Bu model girdi, gizli ve çıktı katmanları olmak üzere üç katman ile oluşturulmuştur. Girdi ve gizli katmanlarda 16 düğüm bulunurken çıktı katmanında sadece bir düğüm bulunmaktadır. Giriş ve gizli katmanda kullanılan aktivasyon fonksiyonu "relu", çıkış katmanının aktivasyon fonksiyonu ise "sigmoid" olarak ayarlanmıştır. DNN modelinin eğitim aşamasında, "optimizer" parametresinde "learning_rate = 0.001" ile "RMSprop" optimizasyon algoritması kullanılır. Ayrıca, "kayıp" (loss) parametresi için "binary_crossentropy" kayıp fonksiyonu seçilmiştir. Öte yandan, eğitime başlamadan önce "epochs" ve "batch_size" parametrelerine "20" ve "256" değerleri verilmiştir. Bu parametreler dışındaki tüm parametrelerde Tensorflow kütüphanesindeki varsayılan değerleri kullanılmıştır.

4.4. Test Aşaması

Test aşamasında ise eğitim aşaması sonucunda oluşturulan modellerin anomali örneklerini tespit etmedeki performansı ölçülmüştür. Ayrıca normal örneklerin ve anomali örneklerini sınıflandırılma yetenekleri değerlendirilmiştir. Bu aşamada veri

kümesinin %20'si test veri kümesi olarak kullanılmıştır. Test veri kümesi 229'u normal, 111'i anomali olmak üzere toplam 340 örnekten oluşmaktadır.

4.5. Deney Sonuçları

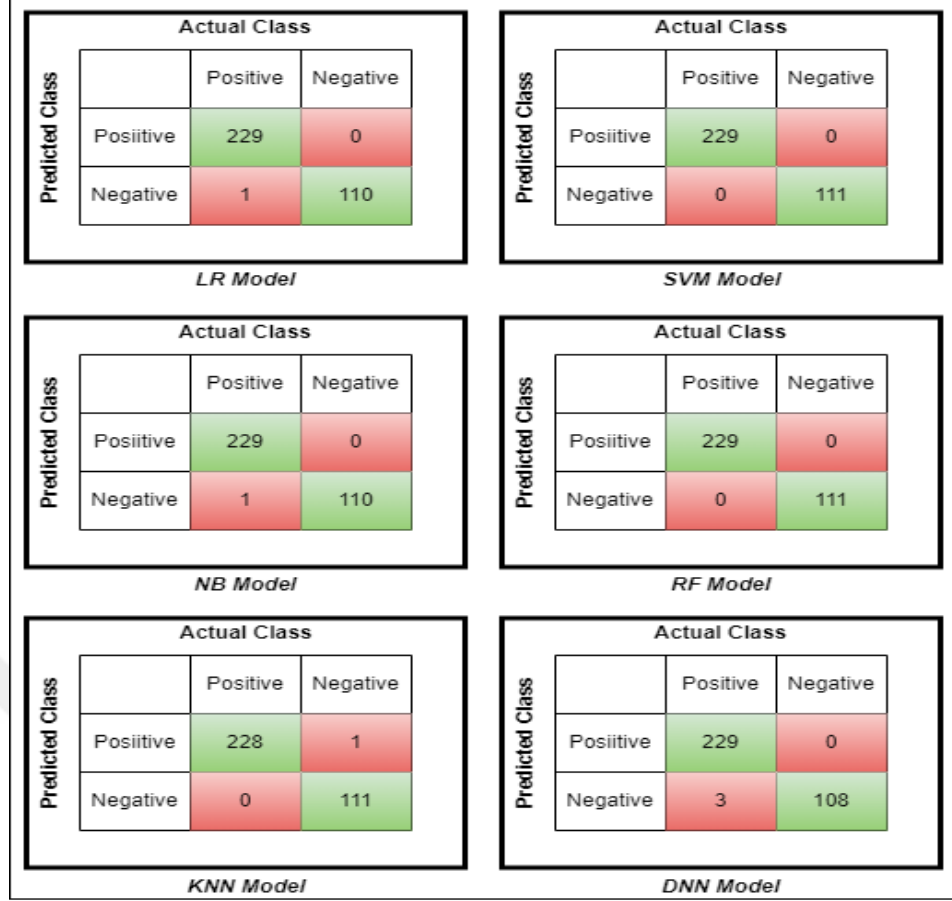
Bu bölümde öncelikle modellerin geliştirildiği ve test edildiği ortamın özelliklerinden bahsedilmektedir. Ardından modeller, deneyler sonucunda ulaştıkları performans değerlerine göre değerlendirilmektedir.

4.6. Deney Ortamı

Modellerin eğitim ve test aşamaları Intel Core i7-10750H 2.60 GHz ve 16 GB RAM ile donatılmış bir bilgisayarda gerçekleştirilmiştir. Modeller Python 3.8.5 programlama dili ve Scikit-learn 0.23.2 ve TensorFlow 2.7.0 kütüphaneleri kullanılarak geliştirilmiştir.

4.7. Sonuçlar

Yapılan deneyler sonucunda modellerin sınıflandırma performanslarını temsil eden karmaşıklık matrisleri Şekil 2'de gösterilmiştir. Karmaşıklık matrislerinde normal örnekler "Positive", anomali örnekleri ise "Negative" etiketi ile temsil edilmektedir. Şekil 4.2'deki karmaşıklık matrislerine dayalı olarak hesaplanan modellerin anomali tespit performansları Tablo 4.2'de verilmiştir.



Şekil 4.2. Modellerin karışıklık matrisleri

Tablo 4.2. Deneyisel sonuçlar

Model	PRC	REC	F1	ACC
LR	0.99	1.00	1.00	1.00
SVM	1.00	1.00	1.00	1.00
NB	0.99	1.00	1.00	1.00
RF	1.00	1.00	1.00	1.00
KNN	1.00	0.99	1.00	1.00
DNN	0.97	1.00	0.99	0.99

Modellerin deneyisel sonuçlar çerçevesinde elde edilen performans karşılaştırmaları Tablo 4.2’de gösterilmiştir. Tablodaki sonuçlara göre SVM ve RF, tüm metriklerde çok iyi sonuçlar elde etmiştir. LR ve NB ise pozitif örnekleri neredeyse her zaman doğru tanımlarken, çok az sayıda yanlış pozitif ile SVM ve RF modellerine yakın performans sergilemektedir. KNN modeli ise diğer modellere

oranla daha düşük REC skoruna sahiptir. Bu da az sayıda pozitif örneği gözden kaçırabileceğini göstermektedir. DNN modelinin ise daha düşük hassasiyet ve doğruluğa sahip olduğu, birkaç yanlış pozitif ve genel olarak biraz daha yüksek hata oranının bulunduğu görülmektedir.



5. ÖNERİLEN MODEL

Bu bölümde, tez kapsamında uygulanmış olan makine öğrenimi yöntemleri ile önerilmiş olan mimari açıklanmıştır.

5.1. Veri Kümesi (Čeponis ve Goranin, 2018; Čeponis ve Goranin, 2019)

Araştırmacılar, sızma girişimi tespit etme probleminde başarı oranını artırmak için ML metodları kullanmaktadırlar. Bu ML metodlarının uygulanabilmesi için eğitim ve test verileri gerekmektedir. Özellikle son zamanlarda yapılan çalışmalarda kullanılan veri kümelerinin birçoğu 1998-1999 yıllarında üretilen DARPA, KDD CUP 99 gibi eski veri kümeleridir. Bu veri kümelerinin eski olması ve daha yeni veri kümelerindeki verilerin yalnızca sistem çağrısı kimliği, sistem dinamik bağlantı kitaplığı dosyası adı ve çağrılan işlev adı gibi sınırlı özellikleri içermesinden dolayı, literatürde daha kapsamlı bir veri kümesine ihtiyaç duyulmuştur. Čeponis ve Goranin, (2018) tarafından bu ihtiyacı karşılamak üzere Windows işletim sistemi için saldırı kaynaklı Windows sistem çağrıları izleri veri kümesi (AWCSTD) adında daha geniş bir veri kümesi üretilmiştir. Bu veri kümesi, daha fazla kötücül yazılım örneği içermekte (12110) ve daha fazla sistem çağrısı dizilerini (112.56 milyon) toplamaktadır. Daha da önemlisi, bu veri kümesi 3145 iyi huylu yazılım tarafından üretilen 16.3 milyon sistem çağrısını da içermektedir (Čeponis ve Goranin, 2019).

Veri kümesinin oluşturulma aşamasında bazı işlevsel olmayan metodlar formüle edilmiştir. Bunlar: sistemin esnek olması; sistemin gelecekte yeni veri toplama adaptasyonuna izin vermesi, kolay yapılandırılabilir olması; sistem parametrelerini değiştirmek için özel bir araca ihtiyaç duyulmaması ve açık kaynak olmasıdır. En yaygın kullanılan işletim sistemi olması dolayısıyla kötücül aktivite derlemesi için seçilen işletim sistemi Windows'tur. Basitlik ve kavram kanıtı (proof of concept) için yalnızca açık kaynaklı kötücül yazılım örnekleri kullanılmıştır. Veri kümesi oluşturulması için kullanılan yöntem kolayca otomatikleştirilebilme, ve herhangi bir kötü amaçlı yazılım örneğinin kolayca indirilip, gerektiği gibi hazırlanarak kullanılmasına olanak tanımaktadır.

Veri kümesi geliştirilme aşaması için üç ana unsur hedeflenmiştir. Bu unsurlar aşağıdaki gibidir:

- Gelecekte veri tabanında yapılan değişikliklerde veri kümesinin kolay bir şekilde yenilenmesi ve sonuçların bağımsız olarak doğrulanmasını sağlamak amacıyla kötü amaçlı yazılım araçları açık kaynak olmalıdır.
- Veri kümesi, sistem çağrısı hakkındaki mümkün olan tüm bilgilere, kötü amaçlı yazılım saldırısı/aracı tarafından etkilenen dosyaların listesine ve ağ trafiğine erişebilmelidir.
- Veri kümesi çok çeşitli kötü amaçlı yazılım saldırıları/araçları tarafından üretilen yüksek sayıda sistem çağrısına dayanmalıdır.

Veri kümesi oluşturmak için VirusShare tarafından sağlanan toplam 12.110 çalıştırılabilir kötü amaçlı yazılım örneği, parola korumalı arşivler şeklinde hazırlanarak kullanılmıştır. Bu örneklerden bazılarının yeterince yüksek bir pozitif tespit oranına sahip olmaması dolayısıyla veri kümesi için sadece pozitif tespit oranı 15 veya daha yüksek olan örnekler seçilmiştir. Bu sebeple 10820 kötü amaçlı yazılım örneği arasından 10276'sı seçilmiştir.

Tablo 5.1. Veri kümesi üretilirken kullanılan en yaygın kötücül yazılım kategorileri ve aileleri

Kötücül Yazılım Kategorisi	Sayı	Kötücül Yazılım Ailesi	Sayı
AdWare	5139	AdWare.Win32	4655
Trojan	2353	Trojan.Win32	2326
Downloader	853	Downloader.Win32	830
WebToolbar	659	WebToolbar.Win32	654
DangerousObject	137	AdWare.MSIL	412
ct Trojan-	101	DangerousObject.M	137
Ransom	79	ulti	96
Backdoor	55	Trojan-	75
RiskTool	45	Ransom.Win32	71
Trojan-	37	Backdoor.Win32	46
Downloader	34	AdWare.NSIS	44
Trojan-Spy	17	RiskTool.Win32	34
Packed	16	Trojan-	34
Virus	15	Downloader.Win32	19
Trojan-PSW	5	Packed.Win32	18

Veri kümesi üretilirken seçilen kötü amaçlı yazılımların dağılımları Tablo 5.1’de verilmiştir. Tablodaki kötü amaçlı yazılım kategori ve aile bilgisi VirusTotal.com sınıflandırmasına dayanmaktadır.

Tablo 5.2’de tez kapsamında kullanılan AWCSTD veri kümesinin alt veri kümelerinin varyasyonları ve etiketleri görülmektedir.

Tablo 5.2. Kullanılan veri kümeleri ve özellikleri

Alt Veri Kümesi	Dizi Çeşitliliği	Veri Özelliği ve Sınıf Sayısı
AllMalware	10, 20, 40, 60, 80, 100, 200, 400, 600, 800, 1000	Sadece zararlı veriler 5 sınıflı
AllMalwarePlusClean	10, 20, 40, 60, 80, 100, 200, 400, 600, 800, 1000	Zararlı ve iyi huylu veriler 5 saldırı, 1 temiz - toplam 6 sınıf
AllMalwarePlusClean2	10, 20, 40, 60, 80, 100, 200, 400, 600, 800, 1000	Zararlı ve iyi huylu veriler 5 saldırı, 1 temiz - toplam 6 sınıf

5.2. Ön İşleme Aşaması

Tez kapsamında veriler önerilen modellere verilmeden önce ön işleme adımından geçirilmiştir. Ön işleme adımı için kullanılan yöntem Byte Pair Encoding (BPE) yöntemidir. BPE yöntemi; bir dizide en sık bulunan bayt çiftini tek bir kullanılmayan bayt ile yinelemeli olarak değiştiren basit ama güçlü bir veri sıkıştırma tekniğidir (Gage, 1994; Sennrich vd., 2015). Bu teknik daha sık bir bayt çifti bulunamayınca kadar tekrarlanmakta ve verilerin boyutu etkili bir şekilde azaltılarak devam etmektedir. Verilerin boyutunu önemli ölçüde azaltabileceği için bu yöntem özellikle metin sıkıştırma işlemi için oldukça kullanışlıdır. Sık karakterleri veya kelime çiftlerini yeni birer sembol olarak kodlayan BPE, orijinal dizinin daha kompakt bir şeklini elde etmektedir.

BPE, alt kelime birimlerini tanımlamak için yaygın olarak kullanılan bir tekniktir. Aslen bir sıkıştırma algoritması olan BPE'nin makine çevirisi ve diğer çeşitli alt görevlerde (özellikle NLP alanında) faydalı olduğu görülmüştür (Sennrich vd., 2015). BPE, örüntü değiştirme temeline dayanan bir yöntemdir. Örneğin bir metin sıkıştırma işleminde yapılan temel uygulama, metinde görünmeyen bir karakterin, metinde sıkça tekrarlanan ardışık iki karakter çifti ile değiştirilmesidir. Bu işlem tüm karakterler bitene kadar veya ardışık iki karakter çifti sıkça görünmeyene kadar tekrarlanmaktadır. Böylece sıkıştırılmış metin iki bölümden oluşur: değiştirme tablosu

ve değiştirilmiş metin. Açma işlemi ise küçük bir çalışma alanı gerektiren oldukça hızlı bir işlemdir. Sıkıştırma işlemi sadece değiştirmeye bağlı olduğu için kısmi olarak açma mümkündür. Bu durum, BPE'yi uyarlanabilir sözlük yöntemlerine göre avantajlı hale getirmektedir (Shibata vd., 1999).

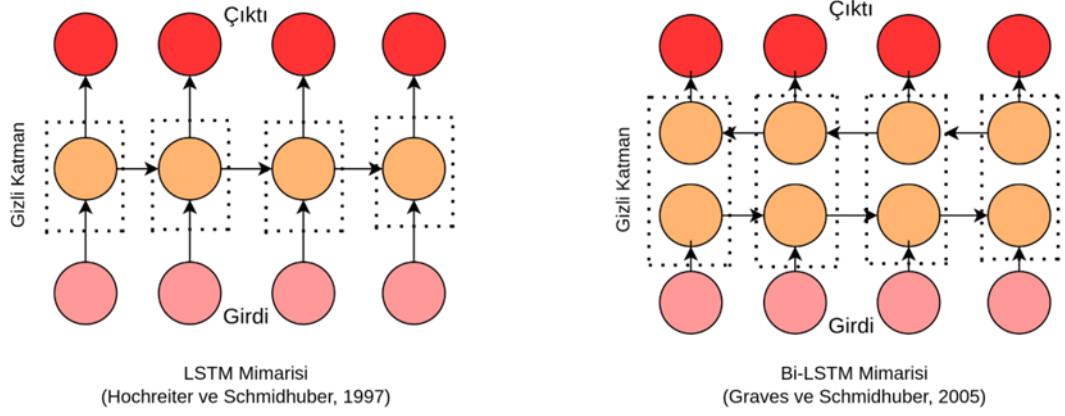
BPE, NLP'de alt kelime belirtimi için yaygın olarak kullanılmaktadır. Verileri sıkıştırmak için gereksiz kalıpları ortaya çıkararak daha sonraki görevlerde ortaya çıkabilecek seyreklik sorunlarını azaltılmasını sağlamaktadır. Birleştirme işlemlerinde tanımlanan alt kelimeler tipik olarak metin sıkıştırma üzerinde büyük etkiye sahiptir (Gutierrez-Vasques vd., 2023). Bu teknik, gereksiz dizeleri veya kalıpları daha kısa referanslara ortak işaretçilerle değiştirerek sıkıştırma sağlayan makro tabanlı algoritmalar kapsamına girmektedir (Storer ve Szymanski, 1978; Gallé, 2019). Algoritma, öncelikle kelimeleri tek tek karakter dizilerine ayırmakta ve her bir karakteri boşluk kullanarak ayrılmış gibi ele almaktadır. İlk işlem sırasında algoritma, ('a', 'b')nin ('ab') haline gelmesi gibi en sık görülen ardışık karakter çiftini tanımlar ve bu tarz durumlarda birleştirme işlemi yapmaktadır. Birleştirilen bu karakterler daha sonra yeni bir sembol oluşturmaktadır. Algoritma sık kullanılan bir sembol çiftini birleştirdiğinde, sembol çiftinin birçok örneğini -ve aralarındaki beyaz boşlukları- tek bir sembolle değiştirerek metni kısaltmaktadır. Birleştirilen karakterler alt kelime birimi olarak kabul edilmektedir. Ek birleştirmeler yapıldıkça, daha uzun alt kelime birimleri oluşturulmakta ve aşamalı olarak tam ortografik kelimeler seviyesine yaklaşmaktadır. Önceden belirlenmiş sayıda birleştirme işlemine ulaşıldığında veya hiçbir ardışık sembol çiftinin frekansı birden büyük olmadığına algoritma durmaktadır.

Veri kümelerinin sınıflandırılması için kullanılan yöntemlerde BPE algoritması ön işleme adımı olarak kullanılmıştır. Veriler modellere verilmeden önce ilk olarak metin formatına dönüştürülmektedir. Devamında BPE algoritması çalıştırılmakta ve buradan elde edilen çıktılar modellere girdi olarak verilmektedir. BPE algoritması uygulaması için Huggingface tokenizers kütüphanesi kullanılmıştır ve parametreler bu kütüphanenin varsayılan parametreleridir.

5.3. Bi-LSTM (Bidirectional Long Short-Term Memory)

Klasik bir RNN çeşidi olan LSTM gradyan kaybolması veya patlaması sorununun önüne geçerek uzun vadeli bağımlılıklarla başa çıkabilmektedir. Zaman içinde düzen kavramını ele alamayan ileri beslemeli ağlar veya RNN yapılarındaki

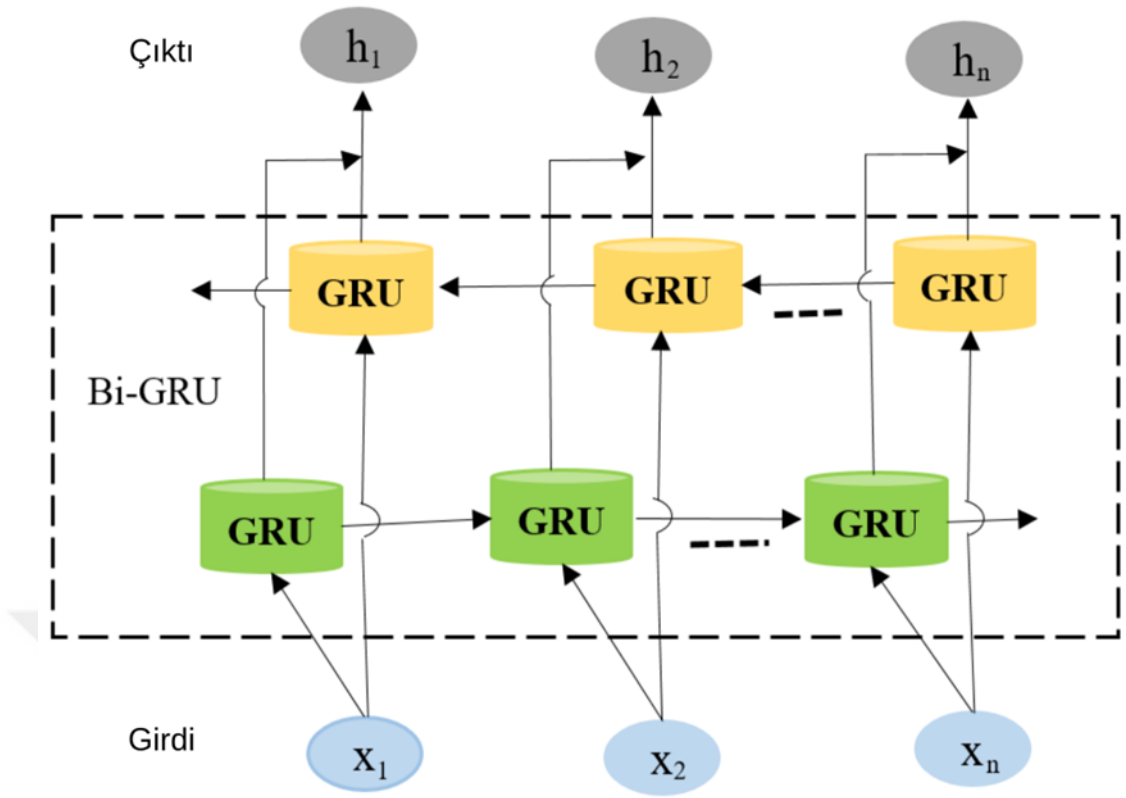
kaybolan gradyan sorunu nedeniyle ortaya çıkan problemler daha sağlam bir alternatif olan LSTM modeli ile aşılabilmektedir. LSTM mimarileri, uzun dönemli bağımlılık elde ederek eğitim süreci sırasında ağırlıkların güncellenmesinin verimsiz olması (kaybolan gradyan) sorunuyla başa çıkmaktadır. LSTM modelleri veriyi genellikle geçmişten geleceğe olacak şekilde tek yönlü olarak işlemektedir. Bu tek yönlü yaklaşım hem geçmiş hem gelecek yönlü verilerin analizinin ve anlaşılmasının gerektiği durumlarda sınırlayıcı olmaktadır. Bu gibi durumların önüne geçmek için Bi-LSTM modeli önerilmiştir. Çift yönlü uzun kısa süreli bellek (Bi-LSTM), veri kümesi üzerindeki tüm ağı öğrenmesini geliştirmek için çift yönlü bilgilerin çıkarılmasıyla doğmuştur. Basit bir ifadeyle, Bi-LSTM, ileri yönlü bir LSTM ağı ve geri yönlü bir LSTM ağının birleşimidir.. LSTM'nin her iki yöndeki durum parametreleri bağımsızdır, bu da ağın zamansal verileri hem ileri hem de geri yönde analiz etmesini sağlar. Bi-LSTM, LSTM'nin veri kullanımını ve tahmin doğruluğunu daha da geliştirmektedir (Song vd., 2024). Bi-LSTM, geleneksel LSTM'nin bir uzantısıdır. LSTM mimarisi, gelecekteki olayları tahmin etmek için zaman adımları ve dizi verileri arasındaki uzun vadeli bağımlılıkları öğrenmektedir. Çift yönlü uzun kısa süreli bellek (Bi-LSTM) ağı, gelecekteki olayları tahmin etmek için zaman adımları ve dizi verileri arasındaki çift yönlü uzun vadeli bağımlılıkları öğrenen yenilikçi bir hesaplama paradigmasıdır. Bi-LSTM, sırasıyla ileri yönlü ve geri yönlü olmak üzere iki adet LSTM eğitir. Bir Bi-LSTM katmanında geçmiş durumları toplamak için negatif zaman yönünden sorumlu ve gelecek durumları toplamak için pozitif zaman yönünden sorumlu iki adet zıt yönlerde LSTM bulunmaktadır. Bi-LSTM ileri ve geri yönde eş zamanlı hesaplama yapabilir. Bu sayede iki ayrı çıktı dizisi oluşturabilir. Bi-LSTM in geçmişten ve gelecekte gelen elde edilen bağlamı tamamen dikkate alarak tanıma doğruluğunda önemli ölçüde gelişme sağlayabileceği kanıtlanmıştır (Guo vd., 2020). Şekil 5.1'de gösterildiği gibi, Bi-LSTM, biri ileri yönde diğeri geri yönde çalışan iki gizli katman ağına ayrıştırılabilir. Bu ileri ve geri yayılımlar, tüm zaman adımları için gizli durumların katlanmamış olması dışında, sıradan ileri ve geri yayılım ağlarına benzerdir. Bu nedenle, Bi-LSTM ile LSTM arasındaki temel fark, LSTM yalnızca geçmişe ait bilgileri korurken, Bi-LSTM yapısında iki gizli durum bir arada kullanılarak hem geçmiş hem de gelecekteki bilgilerin zamanın herhangi bir noktasında korunabilmesidir (Shen vd., 2021).



Şekil 5.1. LSTM ve Bi-LSTM mimarileri (Hochreiter ve Schmidhuber, 1997; Graves ve Schmidhuber, 2005)

5.4. Bi-GRU (Bidirectional Gated Recurrent Unit)

Birçok dizi modelleme çalışması için geçmiş ve gelecek bağlamlara erişmek önemli ölçüde avantaj sağlamaktadır. Bununla birlikte standart GRU ağı gelecekteki bağlamı göz ardı ederek diziyi kronolojik sıraya göre işlemektedir. Çift yönlü bir GRU ağı, ileri ve geri sinir ağlarına sahip bir Bi-LSTM ağına benzer çift katmanlı bir yapıdır. Bu yapının ana fikri, tüm bağlamsal bilgiyi girişten çıkış katmanına aktarmaktır. Çift yönlü bir LSTM yapısına benzer şekilde, bir Bi-GRU'da giriş bilgisi ileri yönde bir sinir ağından ve geri yönde bir sinir ağından geçer. Bu ileri ve geri katmanlardan elde edilen çıktılar nihai çıktıyı sağlamak için birleştirilir (Raj vd., 2021). Daha açık bir ifadeyle Bi-GRU ağları, gizli bağlantıların tam tersi bir kronolojik sırayla aktığı ikinci bir katman ekleyerek tek yönlü GRU ağını genişletmektedir. Girdi dizisini baştan sona doğru işleyen standart GRU ağlarından farklı olarak, bir Bi-GRU ağı, bir tanesi diziyi baştan sona doğru işleyen, diğeri ise sondan başa doğru işleyen eden iki adet GRU katmanı bulundurmaktadır (Chauhan vd., 2024). Bu iki yönlü yaklaşım modelin hem geçmiş durumlardan hem de gelecek durumlardan bağlam yakalamasına olanak tanıyarak verilerdeki bağımlılıkları anlama yeteneğini artırmaktadır (Chen vd., 2022). GRU, daha az parametre kullanarak LSTM ağları ile karşılaştırılabilir performans sunmaktadır. Bu nedenle, gerçek zamanlı tahmin yapabilmek ve tahmin performansını artırmak için düşük hesaplama gereksinimleri ile hareket edebilir. Bi-GRU modelinin yapısı Şekil 5.2'de gösterilmiştir.

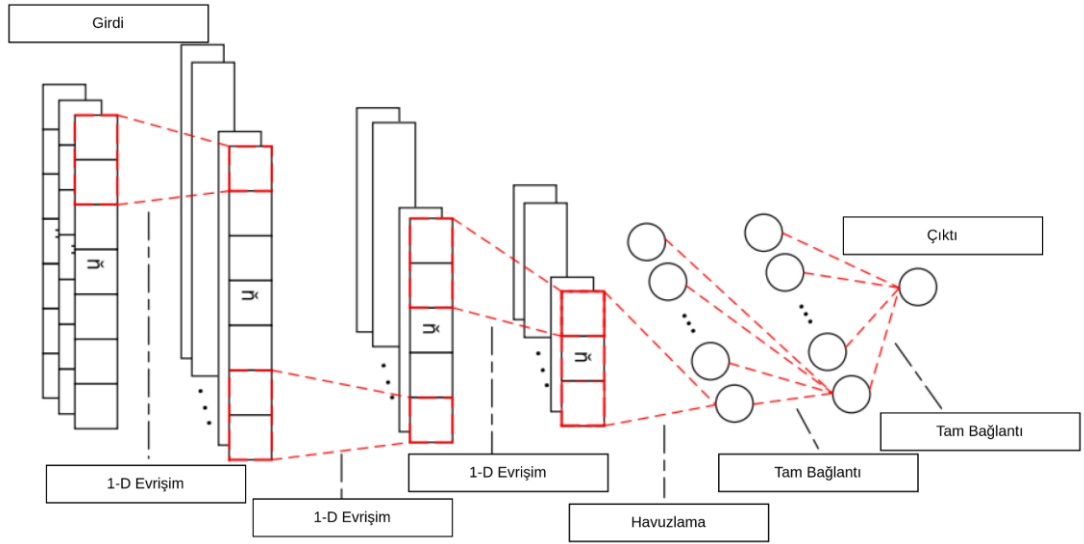


Şekil 5.2. Bi-GRU modelinin mimarisi (Maji ve Swain, 2022)

5.5. 1D-CNN (1-Dimensional Convolutional Neural Network)

Geleneksel CNN'ler, yalnızca görüntüler ve videolar gibi 2 boyutlu veriler üzerinde çalışmak üzere tasarlanmıştır. Bu nedenle genellikle "2D-CNN" olarak adlandırılırlar (Kiranyaz vd., 2018). 2D-CNN yapılarının en büyük avantajı, yüksek performanslarının yanı sıra, geleneksel YSA'ların aksine hem özellik çıkarma hem de sınıflandırma işlevlerini tek bir yapıda birleştirebilmeleridir. Geleneksel ML yöntemleri, veri üzerinde belirli ön işleme adımları uygular ve genellikle yüksek hesaplama karmaşıklığı gerektiren sabit ve elle hazırlanmış özellikler kullanılmaktadır. Buna karşın, CNN tabanlı yöntemler, sınıflandırma doğruluğunu maksimize etmek için "öğrenilen" özellikleri doğrudan ham verilerden çıkarabilir (Kiranyaz vd., 2021). Bu, CNN'leri karmaşık mühendislik uygulamaları için cazip kılan ve sınıflandırma performansını önemli ölçüde artıran temel özelliktir. 2D-CNN'in 1 boyutlu veri işleme uygulaması, genellikle 1 boyuttan 2 boyuta bir dönüşüm gerektirir. Literatürde, erken aşamalarda derin CNN yapılarını 1D veriler üzerinde uygulayan çalışmalar bulunmaktadır. Bu çalışmalarda, 1 boyutlu verilerin 2 boyutlu olarak temsil edilebilmesi için çeşitli dönüşüm teknikleri kullanılmıştır. Bu yöntemler, 1D verilerden ham verileri doğrudan çıkararak sınıflandırma doğruluğunu artırmayı

hedefler. Bu şekilde, hem özellik çıkarma hem de sınıflandırma görevleri aynı yapıda birleştirilir (Lu vd., 2017; Lee vd., 2017). Ancak bu tarz veriler için 2D-CNN kullanımının en önemli dezavantajı yüksek bir hesaplama karmaşıklığı oluşturmaktır. Buna ek olarak, CNN'lerin uygun şekilde eğitilmesi, makul bir genelleme kabiliyeti elde etmek için büyük boyutlu bir veri kümesi gerektirir. Ancak, etiketli verilerin az olduğu birçok pratik 1D uygulaması için bu uygun olmayabilir (Kiranyaz vd., 2021). Bu dezavantajları gidermek amacıyla, 2015 yılında Kiranyaz ve arkadaşları, doğrudan hastaya özgü EKG sinyalleri üzerinde çalışan ilk kompakt ve uyarlanabilir 1D CNN'leri önermişlerdir (Kiranyaz vd., 2015). 1D-CNN'ler, 1D verileri işlemek için geliştirilmiş ve diğer birçok model arasında hesaplama avantajlarına sahip olduğu kanıtlanmış bir makine öğrenimi yöntemidir. Bu modeller, özellikle veri boyutu büyük olduğunda ve etiketli veri sınırlı olduğunda etkili çözümler sunar (Mitiche vd., 2020). 1D-CNN'lerde konvolüsyon işlemi vektör şeklindeki veriler üzerinde gerçekleştirilir ve giriş vektörüne bir filtre vektörü aracılığıyla uygulanır. 1D-CNN modellerinin yapısı, 2D-CNN modellerine çok benzer. Her konvolüsyon katmanını bir havuzlama katmanı takip eder. Daha sonra, tam bağlı katman, önceki katmandan gelen özellik haritası çıktısını girdi olarak alır ve tüm nöron düğümlerini birbirine bağlar. Bu yapı sırasında, tam bağlı katman üzerinde toplu normalizasyon uygulanabilir, bu da daha hızlı eğitim sağlar. Aktivasyon fonksiyonları (Softmax, sigmoid, vb.), ağ eğitimi sırasında çapraz entropi kaybını hesaplamak için tahmin edilen çıktıyı belirler. Son zamanlarda popüler hale gelen 1-boyutlu evrişimsel sinir ağları (1D-CNN), farklı kaynaklardan gelen sinyaller ve sınırlı sayıda etiketli veri ile yapılan uygulamalarda oldukça iyi performans göstermektedir. Görüntü işleme için kullanılan ve iki boyutlu verilerle (yükseklik ve genişlik) çalışan 2 boyutlu evrişimsel sinir ağlarının (2D-CNN) aksine 1D-CNN'ler tek boyutlu veriler üzerinde tek bir uzamsal boyutta çalışırlar. 1D-CNN, özellikle zaman serisi verileri, ses sinyalleri ya da girdinin dizi olarak kabul edilebileceği herhangi bir veri gibi sıralı verileri işlemek için dizayn edilmiş bir evrişimsel sinir ağı türüdür. Bu iki teknik arasındaki temel fark 1D-CNN'in evrişimsel çekirdeğinin tek boyutlu olmasıdır. Bu da tek boyutlu dizilerden özellikler çıkarabileceğini göstermektedir. Başla bir deyişle karmaşık matris işlemleri olmadan yalnızca vektör işlemlerini gerçekleştirerek bellek tüketimini önemli ölçüde azaltmaktadır (Hu vd., 2023). 1D-CNN modelinin yapısı Şekil 5.3'te gösterilmektedir.



Şekil 5.3. 1D-CNN modelinin yapısı

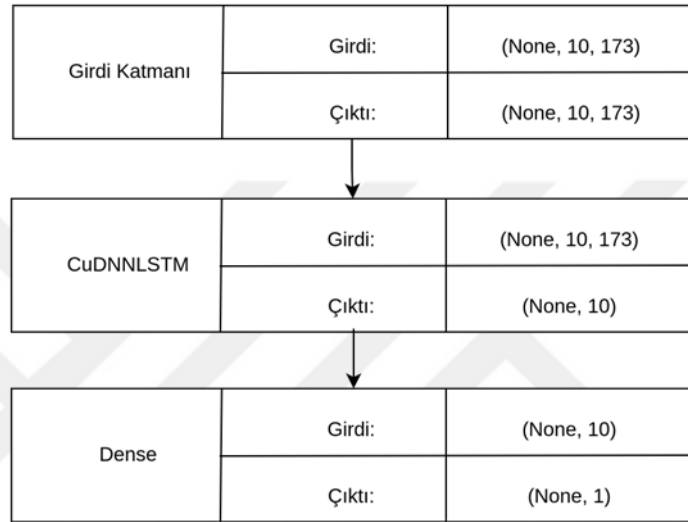
5.6. Benzetimler

Bu bölümde, tez kapsamında önerilen yöntemlerin anomali tespit performansını API çağrılarında oluşan veri kümeleri üzerinde göstermekte ve literatürde yer alan diğer yöntemlerle karşılaştırmaktayız. Ayrıca önerilen yöntemlerin davranışını farklı parametre ve alt yöntemlere göre analiz ediyor ve karşılaştırmalı analiz sunuyoruz. Bu bağlamda AWSCTD isimli, anomali tespit sistemleri için kullanılan veri kümesi ele alınmaktadır (Čeponis ve Goranin, 2018).

2019 yılında Čeponis ve Goranin tarafından yapılan bir çalışmada AWSCTD veri kümesi kullanılarak DL modelleri üzerinde deneyler yapılmıştır. Çalışmada LSTM, GRU, 1D-CNN modelleri AWSCTD veri kümesinin alt kümeleri olan AllMalware, AllMalwarePlusClean, AllMalwarePlusClean2 veri kümeleri kullanılarak eğitilmiş ve elde edilen sonuçlar analiz edilmiştir. Veri kümeleri %80 eğitim, %20 test olacak şekilde parçalara bölünmüştür. Model başarıları beş noktalı çapraz doğrulama yöntemi ile değerlendirilmiştir. Tez kapsamında önerilen Bi-LSTM, Bi-GRU ve 1D-CNN modellerinin her biri eğitilirken Adam optimizasyon yöntemi kullanılmıştır. Adam yöntemi için kullanılan parametreler; öğrenme oranı 0.001, beta_1 0.9 ve beta_2 0.99 şeklindedir. Modeller eğitilirken kullanılan kayıp fonksiyonu ise ikili çapraz entropi fonksiyonudur. Bu parametreler dışındaki tüm parametreler Tensorflow kütüphanesinin varsayılan parametreleridir. Bu bölümün devamında bahsedilen çalışma ile tez kapsamında önerilen modellerin elde ettiği sonuçların yöntem bazlı bir analizi yapılmıştır.

5.6.1. Önerilen Modellerin Literatür ile Karşılaştırılması

Bu bölümde, AWSCTD veri kümesinin alt veri kümeleri olan AllMalware, AllMalwarePlusClean, AllMalwarePlusClean2 veri kümeleri üzerinde, tez kapsamında önerilen Bi-LSTM, Bi-GRU ve 1D-CNN modellerinin, Čeponis ve Goranin tarafından 2019 yılında yapılan çalışmada kullanılan yöntemler olan LSTM, GRU ve CNN modelleri ile karşılaştırmaları ele alınmaktadır. Çalışmada ele alınan LSTM modelinin yapısı Şekil 5.4'te verilmiştir.

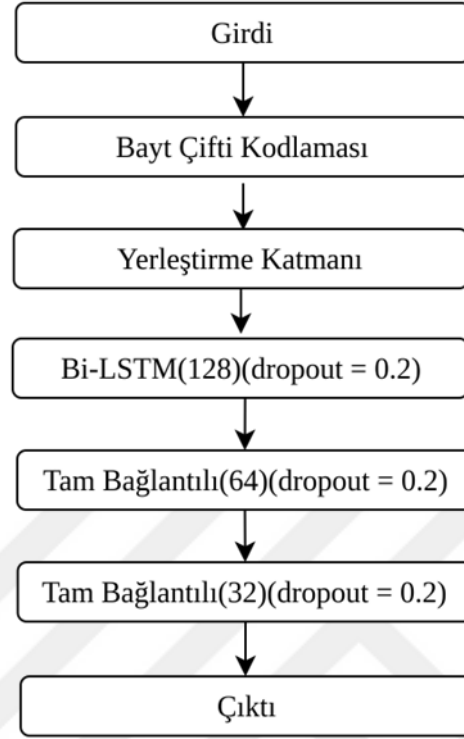


Şekil 5.4. LSTM mimarisi (Čeponis ve Goranin, 2019)

5.6.1.1. Bi-LSTM

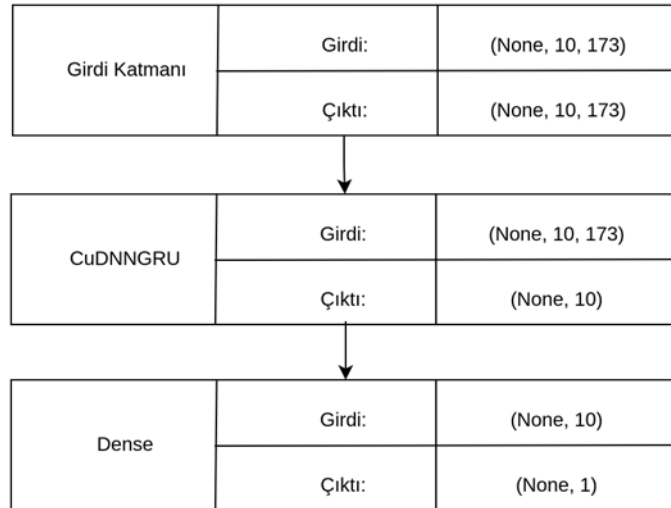
Tez kapsamında Bi-LSTM modeli, dizi verileri arasındaki çift yönlü uzun vadeli bağımlılıkları öğrenebilen bir yapıya sahip olması dolayısıyla LSTM yerine tercih edilmiştir. Veriler, Bi-LSTM modeline beslenmeden önce BPE yöntemi ile ön işleme adımından geçirilmiştir. Devamında, ön işleme adımından geçirilen veriler yerleştirme (embedding) katmanına aktarılmıştır. Bu kısımda kullanılan embedding boyutu 1000x128'dir. Embedding katmanında boyut indirgemesi yapılan veriler Bi-LSTM katmanına geçmektedir. Bi-LSTM katmanında kullanılan hücre sayısı 128'dir. Sonrasında, sırasıyla 64, 32 ve 1 boyutlu olmak üzere tam bağlantılı katmanlar kullanılmaktadır. Bu katmanlar arasında ilk iki katman gizli katmanlar olup ReLu aktivasyon fonksiyonu kullanılmıştır. Son katman ise sınıflandırma katmanıdır ve bu katmanda veri kümesindeki sınıf sayısına göre Sigmoid veya Softmax aktivasyon fonksiyonu kullanılmıştır. Modelin aşırı öğrenmesi ve ezberlemesinin önüne geçmek

için katmanlar arasında 0.2 dropout oranı kullanılmıştır. Önerilen Bi-LSTM modeli Şekil 5.5’te görülmektedir.



Şekil 5.5. Tez kapsamında önerilen Bi-LSTM mimarisi

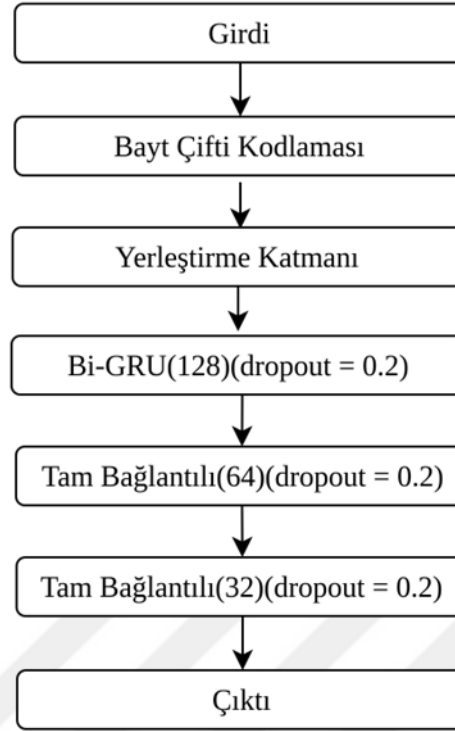
Čeponis ve Goranin tarafından 2019 yılında yapılan çalışmada kullanılan GRU modelinin yapısı Şekil 5.6’da verilmiştir.



Şekil 5.6. GRU (Čeponis ve Goranin, 2019)

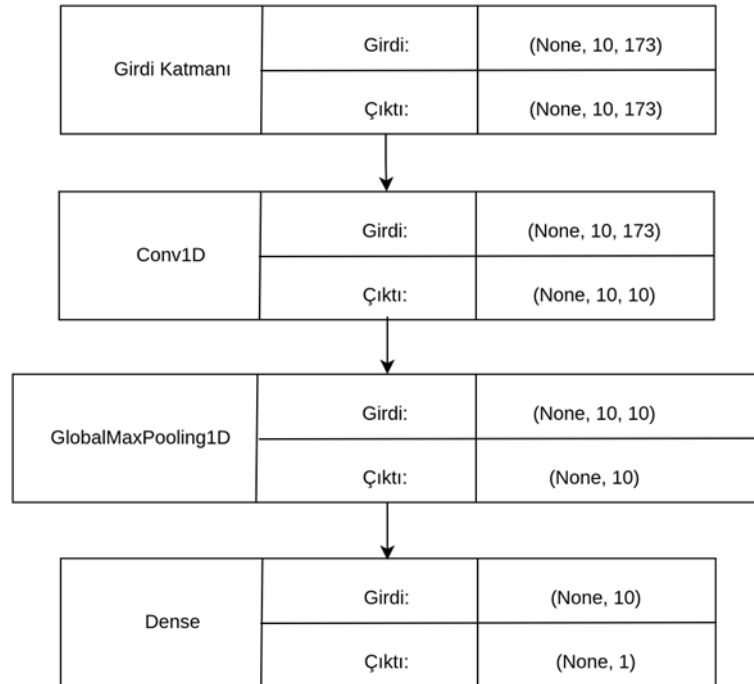
5.6.1.2. Bi-GRU

Tez kapsamında önerilen bir diğer model olan Bi-GRU modelinin tercih edilme sebebi ise, Bi-GRU modelinin dizi verileri arasındaki çift yönlü uzun vadeli bağımlılıkları öğrenebilme yeteneğine sahip olmasıdır. Veriler, Bi-GRU modeline beslenmeden önce BPE yöntemi ile kapsamlı bir ön işleme sürecinden geçirilmiştir. Bu ön işleme adımının ardından, veriler yerleştirme (embedding) katmanına aktarılmıştır. Embedding katmanında kullanılan boyut 1000x128 olarak belirlenmiştir. Bu katmanda, boyut indirilmesi yapılan veriler, Bi-GRU katmanına geçmektedir. Bi-GRU katmanında kullanılan hücre sayısının 128'dir. Bu katmandan sonra, sırasıyla 64, 32 ve 1 boyutlu olmak üzere tam bağlantılı (dense) katmanlar yer almaktadır. Bu tam bağlantılı katmanlardan ilk iki tanesi gizli katmanlar olup, ReLU aktivasyon fonksiyonu kullanılmıştır. Son katman ise sınıflandırma katmanıdır ve bu katmanda veri kümesindeki sınıf sayısına göre Sigmoid veya Softmax aktivasyon fonksiyonu kullanılmıştır. Modelin aşırı öğrenmesini ve ezberlemesini önlemek amacıyla, her katman arasında 0.2 oranında dropout uygulanmıştır. Bu şekilde modelin daha sağlam bir performans sergilemesi hedeflenmiştir. Önerilen Bi-GRU modelinin yapısı Şekil 5.7'de gösterilmektedir.



Şekil 5.7. Tez kapsamında önerilen Bi-GRU mimarisi

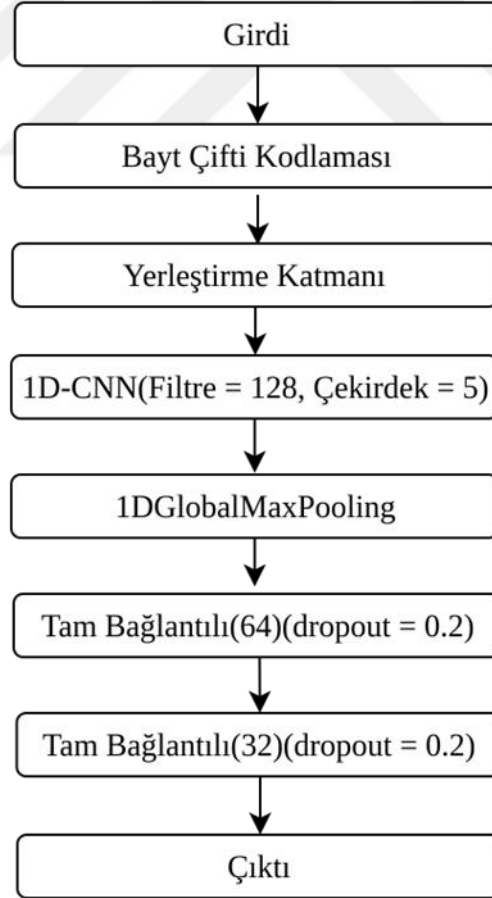
Čeponis ve Goranin tarafından 2019 yılında yapılan çalışmada kullanılan 1D-CNN modelinin yapısı Şekil 5.8’de verilmiştir.



Şekil 5.8. 1D-CNN (Čeponis ve Goranin, 2019)

5.6.1.3. 1D-CNN

Tez kapsamında ele alınan bir diğer model ise 1D-CNN modelidir. Veri kümelerindeki verilere BPE işlemi uygulandıktan sonra elde edilen çıktılar 1000x128 boyutlarında bir yerleştirme katmanından geçirilmektedir. Devamında bu katmandan elde edilen çıktılar convolution1D katmanına girdi olarak verilmiştir. Bu katmanda kullanılan parametreler; filtre sayısı 128, çekirdek boyutu 5, stride 1 ve padding = “valid” şeklindedir. Bu katmandan elde edilen çıktılar 1DMaxPooling katmanına girdi olarak verilmektedir. Devamında sırasıyla 64 ve 32 boyutlu tam bağlantılı katmanlar tanımlanmıştır. Son olarak tek bir nörondan oluşan sınıflandırma katmanı bulunmaktadır. Konvolüsyon, 64 ve 32 nöronluk tam bağlantılı katmanlarda modelin ezberlemesini engellemek adına 0.2 Dropout oranı ve aktivasyon fonksiyonu olarak ReLu kullanılmıştır. Son katman ise sınıflandırma katmanıdır ve bu katmanda veri kümesindeki sınıf sayısına göre Sigmoid veya Softmax aktivasyon fonksiyonu kullanılmıştır. Önerilen 1D-CNN modelinin yapısı Şekil 5.9’daki gibidir.



Şekil 5.9. Tez kapsamında önerilen 1D-CNN mimarisi

Önerilen Bi-LSTM, Bi-GRU ve 1D-CNN modellerinin, Čeponis ve Goranin, (2019) tarafından önerilen modeller ile karşılaştırıldığında elde ettiği sonuçlar; AllMalware, AllMalwarePlusClean, AllMalwarePlusClean2 veri kümeleri için tablolar halinde gösterilmektedir.

5.6.2. Oluşturulan Modellerin Elde Ettiği Sonuçlar

Bu bölümde, tez kapsamında ele alınan Bi-LSTM, Bi-GRU ve 1D-CNN modellerinin AllMalware, AllMalwarePlusClean, AllMalwarePlusClean2 veri kümeleri kullanılarak eğitilmesi sonucu elde edilen sonuçlar ve Čeponis ve Goranin, (2019) çalışmasında önerilen LSTM, GRU ve 1D-CNN modellerinden elde edilen sonuçlar karşılaştırılmıştır. Tablo 5.3, Tablo 5.4 ve Tablo 5.5'te oluşturulan modellerin ve karşılaştırma yapılan modelin veri kümeleri üzerinde elde ettikleri doğruluk değerleri verilmiştir. Tablo 5.6, Tablo 5.7 ve Tablo 5.8'de ise oluşturulan modellerin bu doğruluk metriği destekleyen precision, recall ve fl -puanı metrikleri paylaşılmıştır. Önerilen modellerin Čeponis ve Goranin (2019) çalışmasındaki modellere göre genel olarak daha iyi veya benzer performans sergilediği görülmüştür.

Veri kümelerindeki veriler modellere verilmeden önce metin formatına dönüştürüldüğünden burada uğraşılacak problem aslında dizi sınıflandırma problemine dönüşmektedir. BPE yönteminin metin verilerini alt birimlerine ayırma mantığı, sayısal veya kategorik verilerde de özelliklerin daha küçük ve anlamlı alt birimlere dönüştürülmesi şeklinde uygulanmıştır. Veri kümelerinde 1000 tane eşsiz sistem çağrısı olduğu düşünüldüğünde normal bir makine öğrenimi modeli eğitimi gerçekleştirildiğinde 1000 boyutlu bir girdi katmanının kullanılması gerekmektedir. Ancak BPE yöntemi aynı zamanda bir veri sıkıştırma yöntemi olduğundan, kullanımıyla birlikte modele beslenen girdinin boyutu düşürülmektedir. Buna ek olarak, BPE'nin metin verilerinde bulunan örüntüleri tanıma ve kelimeleri alt birimlere ayırma yeteneği, sayısal verilerde kullanılmıştır. Eldeki veriler sadece rakamlardan oluştuğundan sözlük boyutu NLP uygulamalarına göre çok daha düşüktür ve bu durum eğitim işleminin daha verimli olmasını sağlamaktadır.

Önerilen modellerin başarılı sonuçlar elde etmesinin bir diğer sebebi de LSTM ve GRU modellerinin çift yönlü olarak kullanımlarıdır. Çift yönlü LSTM ve GRU modelleri, dizinin hem geçmiş hem de gelecek bilgilerini aynı anda işlemektedir. Bu yapıları sayesinde eldeki dizi sınıflandırma probleminde bağlamdan tam olarak yararlanılmaktadır. Hem ileri hem de geri yönlü bilgi akışının birleştirilmesi, daha

zengin ve güçlü veri temsiller oluşturulmasını sağlamaktadır. Ek olarak, çift yönlü modeller verilerdeki uzun menzilli bağımlılıkları ve karmaşık ilişkileri daha iyi yakalayabilmektedir. Bu özelliklerinden dolayı veri boyutu arttıkça sınıflandırma performansında bir düşüş olmamıştır aksine bir yükseliş gözlemlenmiştir. Çift yönlü modeller, veri içindeki gürültüye ve hatalara karşı daha dayanıklıdır. İleri ve geri yönlü bilgi akışlarının birleştirilmesi, tek yönlü bir modelin kullanamayacağı bazı bilgilerin kullanılma olasılığını artırmaktadır.

Yukarıda açıklanan bu sebeplere bağlı olarak, BPE yöntemi ile veriler üzerinden elde edilen daha güçlü temsil ve çift yönlü modellerin (Bi-GRU, Bi-LSTM) verileri çift yönlü işleme kapasitesi sayesinde dizi boyutunun yüksek olduğu durumlarda bile karşılaştırma yapılan modellere oranla çok daha yüksek başarı oranı elde edildiği görülmektedir.

Önerilen modellerin sınıflandırma başarıları değerlendirilirken 5 noktalı çapraz doğrulama yöntemi kullanılmıştır. Veriler bu aşamada %80 eğitim %20 test olacak şekilde parçalara ayrılmaktadır. Veri kümeleri üzerinden bu şekilde toplamda 5 farklı alt küme oluşturulmaktadır ve her alt küme kendi içerisinde eğitilip test edilmektedir. Bu sayede veri kümesindeki bütün veriler eğitim ve test aşamalarında bulunmuştur. Son aşamada bu alt kümelerden elde edilen değerlerin ortalamaları alınmaktadır ve nihai sonuç ortaya çıkmaktadır. Bu yöntem ile birlikte elde edilen sonuçların güvenilirliğinin artırılması hedeflenmiştir.

Tablo 5.3. Tez kapsamında önerilen modeller ve Čeponis ve Goranin, (2019) çalışmasındaki modellerin AllMalware Subdataset için karşılaştırma sonuçları

AllMalware Subdataset						
Accuracy Metriği						
Count	Çalışmadaki Bi-LSTM	LSTM (Čeponis ve Goranin, 2019)	Çalışmadaki Bi-GRU	GRU (Čeponis ve Goranin, 2019)	Çalışmadaki 1D-CNN	1D-CNN (Čeponis ve Goranin, 2019)
10	88.4	88.1	88.4	88.3	88.5	87.3
20	88.1	88.8	88.3	88.1	88.5	89.0
40	90.4	89.1	89.9	90.6	89.9	91.2
60	91.5	88.2	90.2	90.5	89.8	91.2
80	92.8	91.8	91.6	91.6	91.3	92.3
100	92.6	91.6	91.9	91.9	91.7	92.8
200	92.6	90.4	91.9	91.5	91.8	92.7
400	92.6	87.6	92.2	90.3	91.6	93.0
600	92.0	87.4	92.0	91.4	92.0	93.1
800	92.6	82.1	91.5	88.5	91.5	93.0
1000	92.5	75.5	91.9	89.6	91.8	93.1

Tablo 5.4. Tez kapsamında önerilen modeller ve Čeponis ve Goranin, (2019) alışmasındaki modellerin AllMalwarePlusClean Subdataset iin karřılařtırma sonuçları

AllMalwarePlusClean Subdataset						
Accuracy Metrięi						
Count	alışmadaki Bi-LSTM	LSTM (Čeponis ve Goranin, 2019)	alışmadaki Bi-GRU	GRU (Čeponis ve Goranin, 2019)	alışmadaki 1D-CNN	1D-CNN (Čeponis ve Goranin, 2019)
10	80.0	77.2	78.5	78.3	78.9	78.7
20	83.6	83.0	83.5	82.3	82.3	83.4
40	86.4	84.1	84.8	84.1	83.6	85.1
60	86.8	85.0	87.2	85.8	86.5	86.1
80	89.5	87.1	87.2	86.8	87.6	88.5
100	89.9	87.1	88.2	86.5	87.7	88.2
200	88.8	85.8	87.9	87.5	86.9	89.4
400	88.7	80.9	88.5	88.1	87.3	89.9
600	92.5	79.8	90.9	89.1	92.6	93.2
800	92.1	68.3	91.0	85.3	91.7	93.5
1000	91.8	77.1	91.4	89.1	91.9	93.6

Tablo 5.5. Tez kapsamında önerilen modeller ve Čeponis ve Goranin, (2019) alışmasındaki modellerin AllMalwarePlusClean2 Subdataset iin karřılařtırma sonuçları

AllMalwarePlusClean2 Subdataset						
Accuracy Metrięi						
Count	alışmadaki Bi-LSTM	LSTM (Čeponis ve Goranin, 2019)	alışmadaki Bi-GRU	GRU (Čeponis ve Goranin, 2019)	alışmadaki 1D-CNN	1D-CNN (Čeponis ve Goranin, 2019)
10	81.9	80.1	80.2	79.4	81.0	80.1
20	86.4	85.8	86.2	85.6	86.2	86.9
40	87.8	87.3	88.9	87.6	89.0	88.8
60	87.6	87.5	87.4	86.4	87.3	88.5
80	88.6	87.5	88.1	88.0	87.9	89.2
100	87.9	87.7	88.4	87.9	87.8	88.8
200	87.8	86.2	88.6	87.3	88.3	89.6
400	88.4	80.3	89.5	87.3	89.4	89.6
600	91.3	77.6	91.8	89.4	93.0	93.5
800	92.8	41.5	91.7	92.3	92.9	93.6
1000	91.9	64.0	92.3	84.1	92.8	93.9

Tez kapsamında ele alınan Bi-LSTM, Bi-GRU ve 1D-CNN modellerinin AllMalware, AllMalwarePlusClean, AllMalwarePlusClean2 veri kümeleri kullanılarak eğitilmesi sonucu elde edilen ve modellerin performansını

değerlendirmek için kullanılan PRC, REC, F1-score ve ACC metrikleri Tablo 5.3, Tablo 5.4 ve Tablo 5.5’te verilmiştir.

Tablo 5.6. Modellerin AllMalware Subdataset veri kümesi ile eğitilmesi sonucunda elde edilen PRC, REC, F1-score ve ACC değerleri

	AllMalware Subdataset											
	Bi-LSTM				Bi-GRU				1D-CNN			
Count	PRC	REC	F1	ACC	PRC	REC	F1	ACC	PRC	REC	F1	ACC
10	86.9	88.4	87.4	88.4	86.8	88.4	87.5	88.4	86.9	88.5	87.5	88.5
20	86.5	88.1	87.1	88.1	86.6	88.3	87.2	88.3	86.9	88.5	87.4	88.5
40	89.4	90.4	89.9	90.4	88.6	89.9	89.1	89.9	88.5	89.9	89.1	89.9
60	90.3	91.5	90.8	91.5	88.8	90.2	89.4	90.2	88.3	89.8	88.8	89.8
80	91.7	92.8	92.3	92.8	90.5	91.6	91.0	91.6	90.1	91.3	90.7	91.3
100	91.6	92.6	92.1	92.6	90.9	91.9	91.4	91.9	90.4	91.7	91.0	91.7
200	91.6	92.6	92.1	92.6	90.7	91.9	91.3	91.9	90.7	91.8	91.2	91.8
400	91.9	92.6	92.2	92.6	90.9	92.2	91.5	92.2	91.3	91.6	91.2	91.6
600	91.1	92.0	91.5	92.0	90.9	92.0	91.4	92.0	91.2	92.0	91.5	92.0
800	91.5	92.6	91.9	92.6	90.2	91.5	90.8	91.5	90.6	91.5	90.9	91.5
1000	91.4	92.5	91.9	92.5	90.9	91.9	91.4	91.9	90.9	91.8	91.3	91.8

Tabloda, üç modelin de AllMalware Subdataset veri kümesi ile yapılan eğitimler sonucunda benzer performanslar sergilediği, ancak özellikle PRC ve REC metriklerinde Bi-LSTM ve Bi-GRU modellerinin oldukça iyi sonuçlar elde ettiği görülmektedir. 1D-CNN modeli de aynı şekilde bazı metriklerde diğer modellere yakın performans sergilemiştir. Bi-LSTM modeli farklı metriklerde istikrarlı ve

yüksek performans sergileyerek en iyi sonuçları elde ederken, Bi-GRU modeli de Bi-LSTM modeline yakın bir performans göstermektedir. 1D-CNN modeli ise diğer iki modele oranla daha düşük performans göstermesine rağmen yüksek ACC ve F1-score değerleri elde etmektedir.

Tablo 5.7. Modellerin AllMalwarePlusClean Subdataset veri kümesi ile eğitilmesi sonucunda elde edilen PRC, REC, F1-score ve ACC değerleri

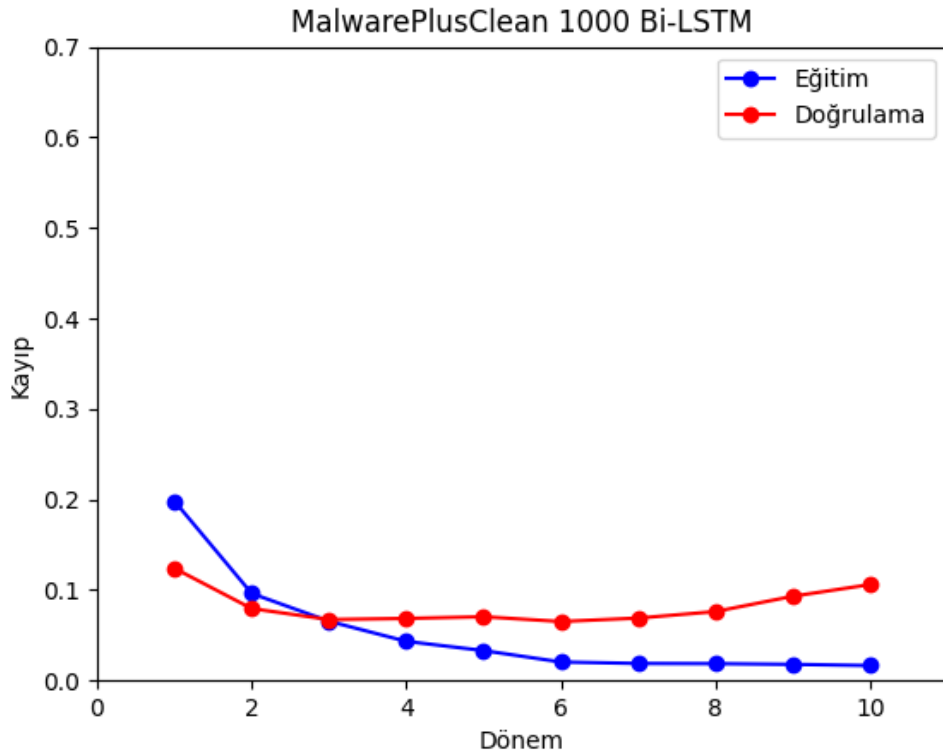
AllMalwarePlusClean Subdataset												
Count	Bi-LSTM				Bi-GRU				1D-CNN			
	PRC	REC	F1	ACC	PRC	REC	F1	ACC	PRC	REC	F1	ACC
10	80.1	80.0	78.3	80.0	78.5	78.5	76.9	78.5	78.8	78.9	77.4	78.9
20	82.7	83.6	82.5	83.6	82.4	83.5	82.7	83.5	81.9	82.3	81.6	82.3
40	85.6	86.4	85.8	86.4	83.9	84.8	84.1	84.8	82.7	83.6	82.7	83.6
60	86.0	86.8	86.2	86.8	86.4	87.2	86.7	87.2	85.9	86.5	85.9	86.5
80	88.9	89.5	89.1	89.5	86.5	87.2	86.7	87.2	86.7	87.6	87.1	87.6
100	89.2	89.9	89.5	89.9	87.3	88.2	87.6	88.2	86.7	87.7	87.1	87.7
200	88.6	88.8	88.5	88.8	87.1	87.9	87.5	87.9	86.9	86.9	86.6	86.9
400	87.7	88.7	88.1	88.7	87.4	88.5	87.9	88.5	87.5	87.3	86.9	87.3
600	91.6	92.5	92.0	92.5	89.9	90.9	90.4	90.9	91.9	92.6	92.2	92.6
800	91.1	92.1	91.5	92.1	90.1	91.0	90.6	91.0	91.0	91.7	91.4	91.7
1000	90.8	91.8	91.3	91.8	90.6	91.4	90.9	91.4	91.2	91.9	91.5	91.9

Tablo 5.8. Modellerin AllMalwarePlusClean2 Subdataset veri kümesi ile eğitilmesi sonucunda elde edilen PRC, REC, F1-score ve ACC değerleri

AllMalwarePlusClean2 Subdataset												
Count	Bi-LSTM				Bi-GRU				1D-CNN			
	PRC	REC	F1	ACC	PRC	REC	F1	ACC	PRC	REC	F1	ACC
10	81.3	81.9	80.8	81.9	79.6	80.2	79.7	80.2	80.2	81.0	80.3	81.0
20	85.6	86.4	85.8	86.4	85.3	86.2	85.7	86.2	85.3	86.2	85.7	86.2
40	86.8	87.8	87.1	87.8	88.2	88.9	88.4	88.9	88.1	89.0	88.4	89.0
60	86.9	87.6	86.9	87.6	86.6	87.4	86.8	87.4	86.5	87.3	86.8	87.3
80	88.1	88.6	88.1	88.6	87.4	88.1	87.6	88.1	87.4	87.9	87.5	87.9
100	87.2	87.9	87.2	87.9	87.8	88.4	87.9	88.4	87.3	87.8	87.4	87.8
200	87.5	87.8	87.2	87.8	87.4	88.6	87.8	88.6	86.9	88.3	87.5	88.3
400	87.1	88.4	87.7	88.4	88.9	89.5	88.9	89.5	88.7	89.4	88.9	89.4
600	90.2	91.3	90.7	91.3	91.0	91.8	91.4	91.8	92.4	93.0	92.7	93.0
800	92.4	92.8	92.5	92.8	90.7	91.8	91.2	91.7	91.9	92.9	92.3	92.9
1000	91.2	91.9	91.6	91.9	91.3	92.3	91.7	92.3	91.9	92.8	92.3	92.8

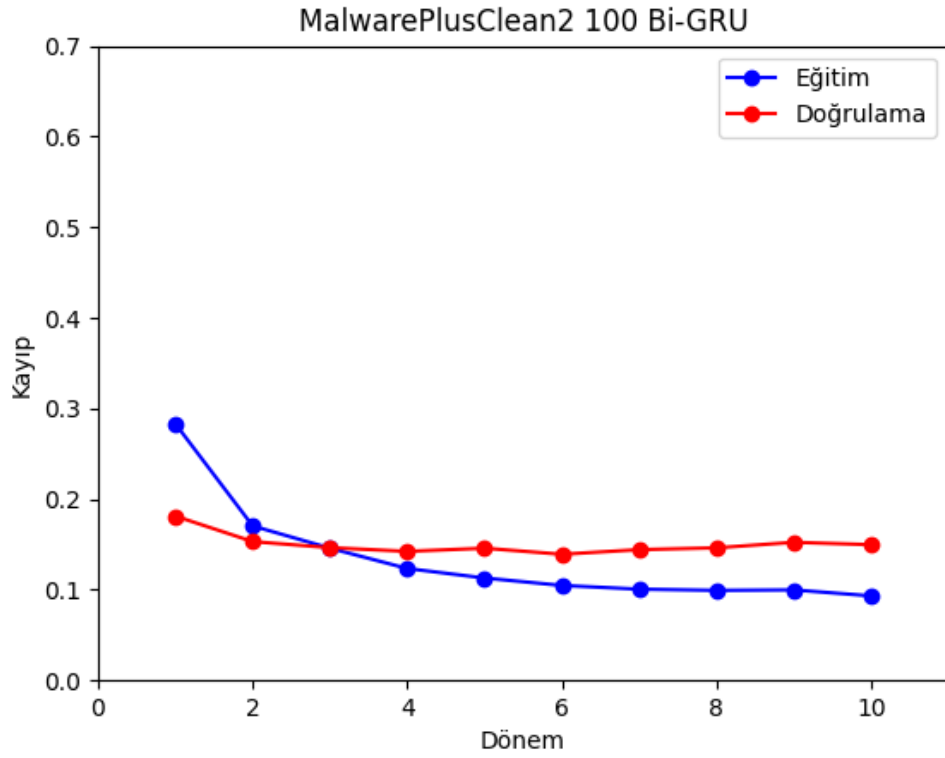
Makine öğrenimi modellerinin eğitim ve doğrulama aşamalarında elde ettikleri kayıp değerlerinin karşılaştırılması ile model performansı hakkında bilgi sahibi olunmaktadır. Bu doğrultuda oluşturulan modellerin ezberleme (overfitting) veya öğrenememe (underfitting) durumlarının incelenmesi için bu kayıp değerleri kullanılmıştır. Çalışma kapsamında çok sayıda veri kümesi bulunduğu ve yine çok sayıda model oluşturulduğu için bu kısımda yapılan inceleme her model için farklı özellik sayısına sahip farklı bir veri kümesi kullanılacak şekilde yapılmıştır. Aşağıda

Bi-LSTM, Bi-GRU ve 1D-CNN modellerinin bazı veri kümeleri ile 10 dönem eğitilmesi sonucunda elde edilen kayıp grafikleri bulunmaktadır. Şekil 5.10'daki grafik incelendiğinde Bi-LSTM modelinin MalwarePlusClean veri kümesi üzerinde 1000 özellikli veriler ile eğitilmesi sonucunda 9. dönem ve sonrasında ezberleme eğilimi göstermekte olduğu görülmektedir. Doğrulama eğrisi incelendiğinde 8.dönemden itibaren kaybın yükseldiği, ancak eğitim eğrisinin düşüş göstermeye devam ettiği görülmektedir.



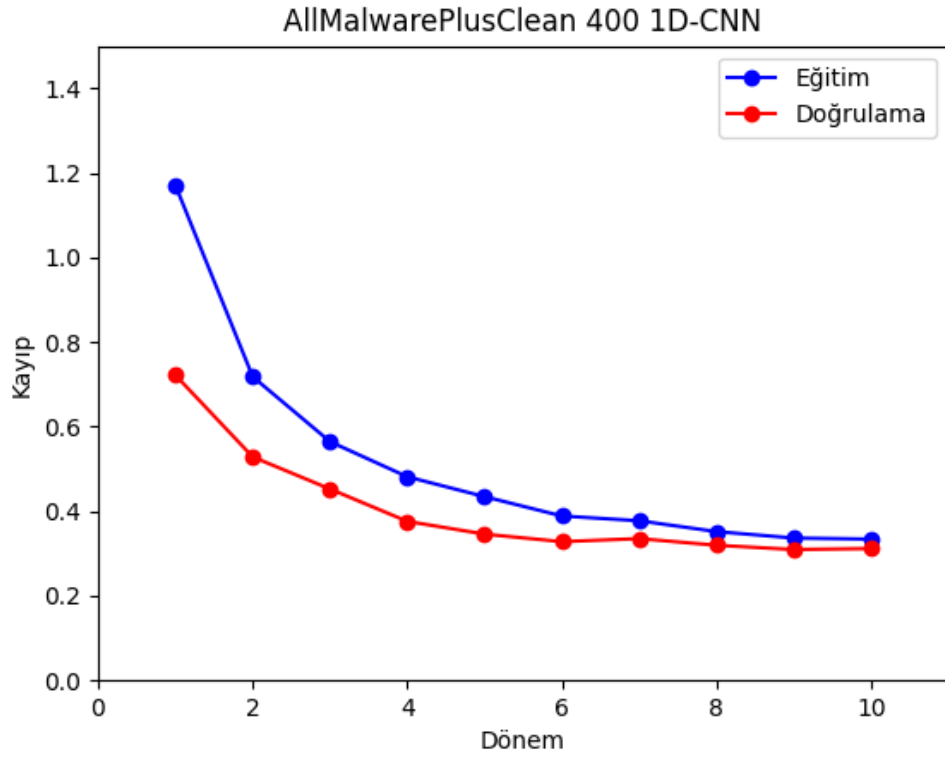
Şekil 5.10. Bi-LSTM modelinin MalwarePlusClean veri kümesi üzerinde 1000 özellikli verilerle 10 dönem eğitilmesi sonucu elde edilen kayıp grafiği

Şekil 5.11'deki grafik incelendiğinde Bi-GRU modelinin MalwarePlusClean2 veri kümesi üzerinde 100 özellikli veriler ile eğitilmesi sonucunda yine 9. dönem ve sonrasında ezberleme eğilimi göstermekte olduğu görülmektedir. Doğrulama ve eğitim eğrileri incelendiğinde eğitim kaybının sürekli bir düşüş gösterdiği, doğrulama kaybının ise küçük değerler ile de olsa yükseliş eğiliminde olduğu görülmektedir.



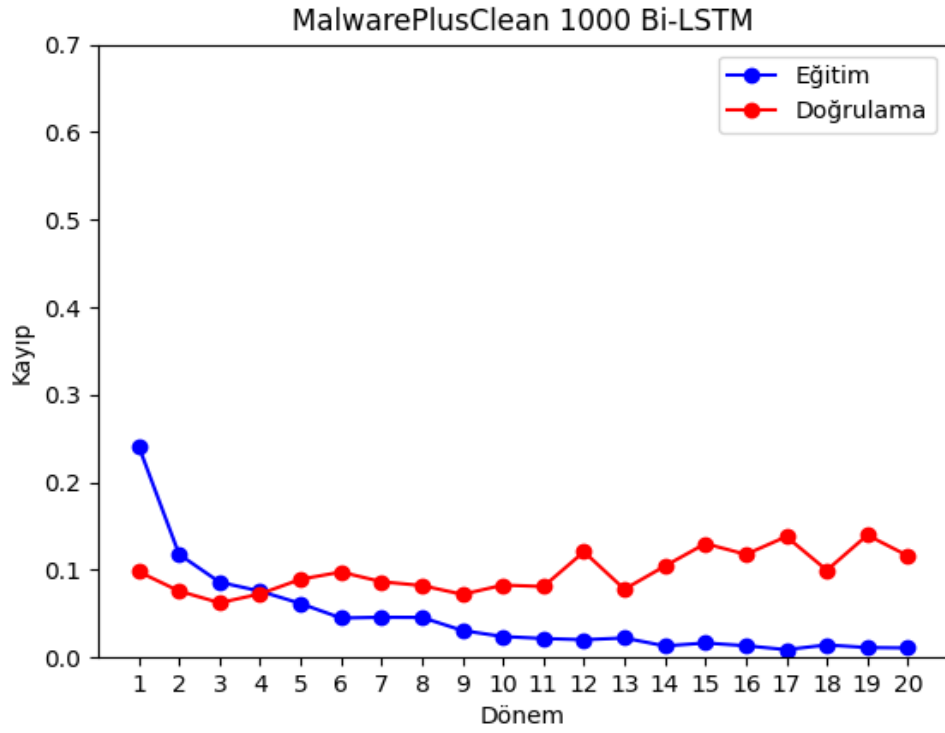
Şekil 5.11. Bi-GRU modelinin MalwarePlusClean2 veri kümesi üzerinde 100 özellikli verilerle 10 dönem eğitilmesi sonucu elde edilen kayıp grafiği

Şekil 5.12'deki grafik incelendiğinde 1D-CNN modelinin AllMalwarePlusClean veri kümesi üzerinde 400 özellikli veriler ile eğitilmesi sonucunda 10 dönem boyunca herhangi bir ezberleme belirtisi göstermediği görülmektedir. Diğer iki model ile karşılaştırma yapabilmek adına bu model hiçbir ezberleme belirtisi göstermemesine rağmen eğitim 10 dönem ile sınırlandırılmıştır.

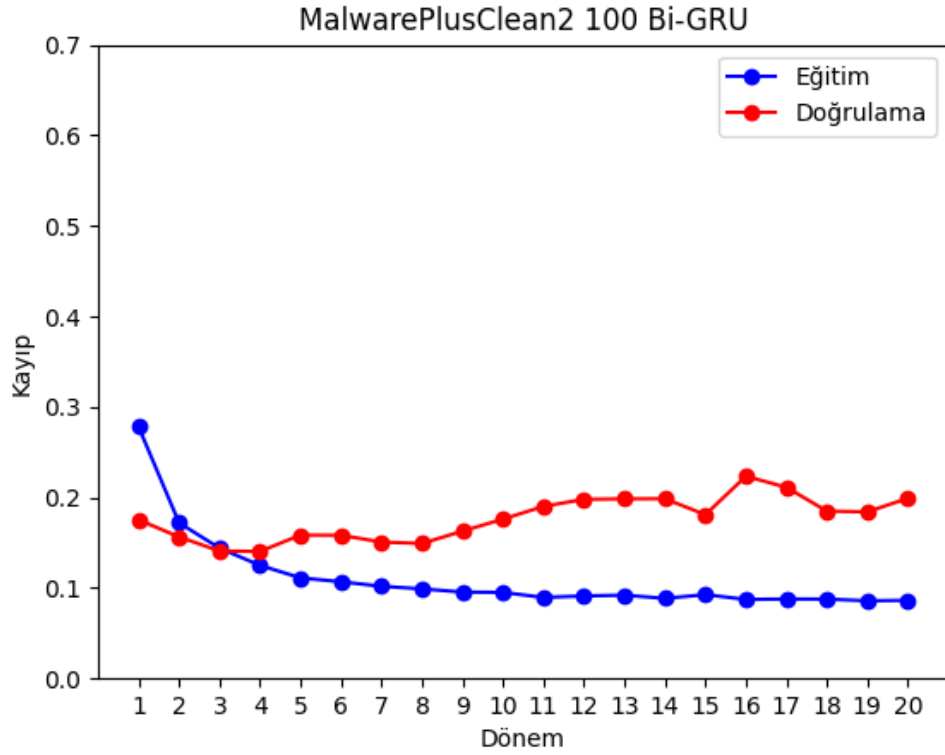


Şekil 5.12. 1D-CNN modelinin AllMalwarePlusClean veri kümesi üzerinde 400 özellikli verilerle 10 dönem eğitilmesi sonucu elde edilen kayıp grafiği

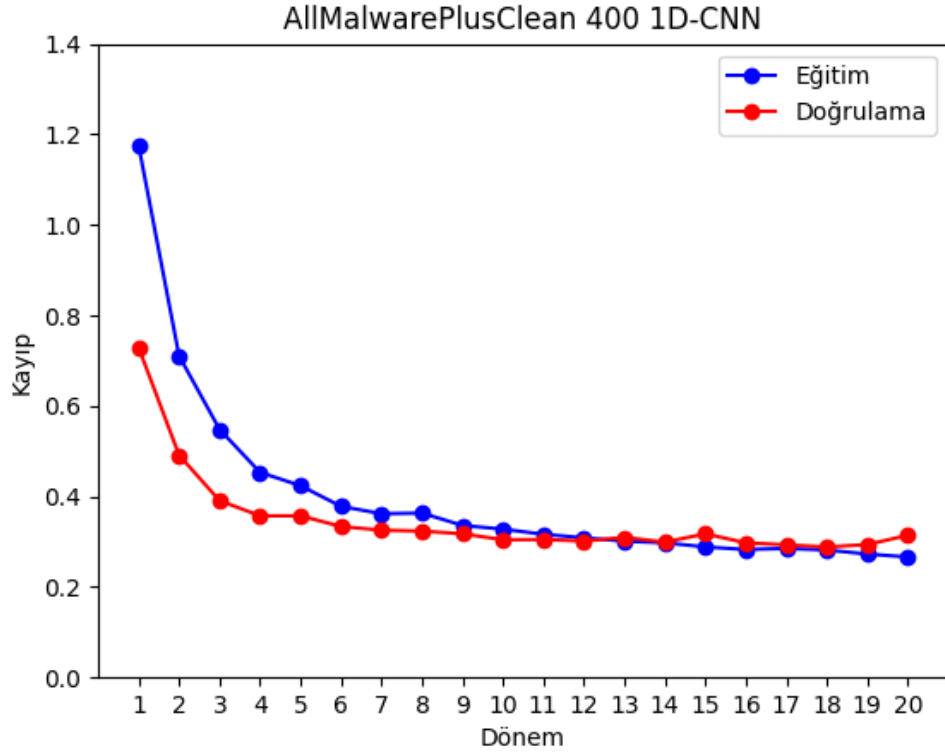
Aşağıda Şekil 5.13, Şekil 5.14 ve Şekil 5.15'te sırasıyla Bi-LSTM, Bi-GRU ve 1D-CNN modellerinin 20 dönem eğitilmesi sonucunda elde edilen kayıp grafikleri görülmektedir. Bu kısımda yapılan deneylerin bir önceki aşamada gerçekleştirilen deneyler ile farkı modellerin eğitildiği dönem sayısıdır. Bu deneylerin amacı, modellerin daha fazla eğitildiklerinde ezberleme veya öğrenememe açısından nasıl performans gösterdiklerinin incelenmesidir.



Şekil 5.13. Bi-LSTM modelinin MalwarePlusClean veri kümesi üzerinde 1000 özellikli verilerle 20 dönem eğitilmesi sonucu elde edilen kayıp grafiği



Şekil 5.14. Bi-GRU modelinin MalwarePlusClean2 veri kümesi üzerinde 100 özellikli verilerle 20 dönem eğitilmesi sonucu elde edilen kayıp grafiği



Şekil 5.15. 1D-CNN modelinin AllMalwarePlusClean veri kümesi üzerinde 400 özellikli verilerle 20 dönem eğitilmesi sonucu elde edilen kayıp grafiği

Bu grafikler incelendiğinde, önceki aşamada elde edilen çıktılara benzer olarak, Bi-LSTM ve Bi-GRU modellerinin 10. dönemden itibaren ezberlemeye yönelik davrandıkları görülmektedir. Çalışmanın 10 dönem ile sınırlandırılmasının sebebi bu modellerde görülen ezberleme eğilimidir. Ancak 1D-CNN modelinde 20 dönem eğitilmesine rağmen herhangi bir ezberleme veya öğrenememe durumu görülmemektedir.

6. SONUÇ VE DEĞERLENDİRME

Bu bölümde, tezin geneline ait bir değerlendirme sunulacak, tezin ana katkıları verilerek ve gelecek çalışmalar için öneriler verilecektir.

Tez kapsamında, API çağrılarını kullanarak makine öğrenimi yöntemleri ile anomali tespiti çalışması gerçekleştirilmiştir. Literatür çalışmasının yanı sıra, tezin önemli bir kısmı ML altyapısının sunulmasına ve Bi-LSTM, Bi-GRU ve 1D-CNN modellerinin eğitime odaklanmıştır. Bu modeller, AllMalwarePlusClean2 Subdataset, AllMalwarePlusClean Subdataset ve AllMalware Subdataset veri kümeleri üzerinde eğitilmiş ve test edilmiştir. Elde edilen sonuçlar, literatürde yer alan LSTM, CNN ve GRU modelleriyle karşılaştırılmış ve doğruluk, precision, recall ve F1 skoru gibi metrikler açısından önemli iyileştirmeler sağlanmıştır. Bu bakımdan, değerlendirme sonuçları aşağıdaki şekilde özetlenebilir.

- Bi-LSTM ve Bi-GRU modelleri, literatürde yaygın olarak kullanılan LSTM ve GRU modellerine göre daha üstün performans göstermiştir. Veriyi çift yönlü kullanmaları sayesinde, zaman serisi verilerindeki karmaşık bağımlılıkları daha iyi yakalamış ve anomali tespitinde daha yüksek doğruluk oranları elde etmişlerdir. Özellikle, Bi-LSTM ve Bi-GRU modellerinin yüksek doğruluk oranları, anomalilerin daha doğru ve eksiksiz bir şekilde tespit edildiğini göstermektedir.
- AllMalwarePlusClean2, AllMalwarePlusClean ve AllMalware veri kümeleri üzerinde yapılan deneyler, Bi-LSTM ve Bi-GRU modellerinin geniş bir yelpazede veri çeşitliliği ile başarılı olduğunu ortaya koymuştur. Bu veri kümeleri, zararlı yazılım ve temiz veri örnekleri içermekte olup, anomali tespiti senaryolarında gerçek dünyadaki uygulamalar için uygun bir test ortamı sunmuştur.
- Literatürdeki LSTM ve GRU modelleri ile yapılan karşılaştırmalar, Bi-LSTM ve Bi-GRU modellerinin belirgin iyileştirmeler sağladığını göstermiştir. Bu modeller, özellikle doğruluk metriklerinde üstünlük sağlamış, bu da anomalilerin daha doğru ve eksiksiz bir şekilde tespit edildiğini göstermektedir. Doğruluk metriklerine ek olarak kullanılan

precision, recall ve f1-puanı metrikleri oluşturulan modellerin sınıflandırma başarıları desteklemektedir.

Bu bağlamda, tezin genel katkılarını aşağıdaki şekilde sıralayabiliriz:

- Kullanılan veri kümesi üzerinde yapılan deneyler Bi-LSTM ve Bi-GRU modelleri kullanılarak gerçekleştirilmiştir. Bu modeller, veri kümesindeki zaman serisi ve ardışık bağımlılıkları yakalamada etkin oldukları için tercih edilmiştir. Bu çalışma, çift yönlü öğrenme kapasitelerinin anomali tespiti üzerindeki etkisini vurgulamış ve bu modellerin bu tür uygulamalar için uygunluğunu göstermiştir.
- Veri kümeleri üzerinde ön işleme aşamasında Bit Çifti Kodlaması (BPE) algoritması kullanılmıştır. BPE algoritması, veri kümesindeki kelimeleri veya karakter dizilerini sıkıştırarak daha kompakt ve yönetilebilir bir temsil oluşturmaktadır. Bu, özellikle büyük veri kümelerinde veri boyutunu azaltarak işleme süresini ve hesaplama maliyetini düşürmektedir. Bu ön işleme aşaması, modelin eğitim ve test süreçlerinde daha hızlı ve etkili sonuçlar elde edilmesine katkıda bulunmuştur.
- Literatürde önceki çalışmalarda elde edilen sonuçlar ile karşılaştırma yapılmıştır. Bu karşılaştırmalarda, literatürde yaygın olarak kullanılan LSTM ve GRU modelleri temel alınmıştır. Yapılan deneyler sonucunda, Bi-LSTM ve Bi-GRU modellerinin literatürdeki LSTM ve GRU modellerine kıyasla daha iyi sonuçlar verdiği görülmüştür. Bu modeller, çift yönlü öğrenme kapasiteleri sayesinde zaman serisi verilerindeki bağımlılıkları daha etkili bir şekilde yakalamış ve anomali tespitinde daha yüksek başarı oranları elde etmiştir. Bu bağlamda, Bi-LSTM ve Bi-GRU modellerinin anomali tespiti gibi uygulama alanlarında kullanılabilirliğini göstermiştir.

Tezin sonuçları ve elde edilen bulgular ışığında, gelecek çalışmalar için aşağıdaki öneriler sunulmuştur:

- **Diziden Diziye (Seq2Seq) Modellerin Kullanımı:** Diziden diziye (Seq2Seq) modeller, doğal dil işleme ve zaman serisi analizlerinde yaygın olarak kullanılmakta olup, anomali tespiti alanında da potansiyel olarak değerlendirilebilir. Bu modeller, veri kümelerindeki ardışık bağımlılıkları

ve örüntüleri daha etkili bir şekilde yakalayarak, anomali tespitinde daha yüksek performans sağlayabilir. Seq2Seq modellerinin bu veri kümesinde denenmesi, zaman serisi verilerinin daha derinlemesine analiz edilmesine ve daha doğru anomali tespiti yapılmasına olanak tanıyabilir.

- **Ön Eğitimli Modellerin İncelenmesi:** BERT, RoBERTa, DistilBERT gibi ön eğitimli modeller, geniş veri kümeleri üzerinde önceden eğitilmiş olup, belirli bir görev için ince ayar yapılarak yüksek performans sergileyebilirler. Bu tür modellerin anomali tespiti için incelenmesi ve uygulanması, özellikle büyük ve çeşitli veri kümelerinde önemli avantajlar sağlayabilir. Ön eğitimli modeller, verilerdeki karmaşık örüntüleri ve anomalileri daha etkili bir şekilde öğrenerek, tespit süreçlerinde daha yüksek doğruluk oranlarına ulaşılmasını sağlayabilir.
- **Transformer Yapısının İncelenmesi:** Transformer yapıları, son yıllarda doğal dil işleme ve diğer alanlarda büyük başarılar elde etmiştir. Bu yapıların anomali tespiti alanında kullanılması, özellikle uzun vadeli bağımlılıkların ve karmaşık örüntülerin daha iyi yakalanmasına yardımcı olabilir. Transformer modelleri, paralel işlem yetenekleri ve verimli öğrenme kapasiteleri sayesinde, büyük veri kümeleri üzerinde hızlı ve etkili anomali tespiti yapabilir. Bu yapının incelenmesi ve adaptasyonu, mevcut modellerin performansını daha da artırabilir.
- **Farklı Veri Kümelerinin Kullanımı:** Mevcut çalışmada kullanılan veri kümelerinin yanı sıra, farklı veri kümelerinin kullanımı da modellerin genelleme yeteneklerini ve performanslarını değerlendirmek için önemli olabilir. Farklı sektörlerden ve uygulama alanlarından elde edilen veri kümeleri ile yapılan testler, modellerin çeşitli senaryolarda ne kadar başarılı olduğunu göstermeye yardımcı olabilir. Bu, anomali tespit modellerinin geniş bir yelpazede uygulanabilirliğini ve güvenilirliğini artırabilir.

KAYNAKLAR

- Ahmed, M., Mahmood, A. N., & Islam, M. R. (2016). A survey of anomaly detection techniques in financial domain. *Future Generation Computer Systems*, 55, 278-288.
- Ahn, S., Yi, H., Bae, H., Yoon, S., & Paek, Y. (2022). Data Embedding Scheme for Efficient Program Behavior Modeling With Neural Networks. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 6(4), 982-993.
- Akagic, A., & Džafić, I. (2024). Enhancing smart grid resilience with deep learning anomaly detection prior to state estimation. *Engineering Applications of Artificial Intelligence*, 127, 107368.
- Almaleh, A., Almushabb, R., & Ogran, R. (2023). Malware API Calls Detection Using Hybrid Logistic Regression and RNN Model. *Applied Sciences*, 13(9), 5439.
- Alsaman, D. (2024). A Comparative Study of Anomaly Detection Techniques for IoT Security using AMoT (Adaptive Machine Learning for IoT Threats). *IEEE Access*.
- AWS. 2024. Amazon Web Server. <https://aws.amazon.com/tr/what-is/api/> (Erişim Tarihi: 18.04.2024).
- Bernal, J., Kushibar, K., Asfaw, D. S., Valverde, S., Oliver, A., Martí, R., & Lladó, X. (2019). Deep convolutional neural networks for brain image analysis on magnetic resonance imaging: a review. *Artificial intelligence in medicine*, 95, 64-81.
- Beyer, K., Goldstein, J., Ramakrishnan, R., & Shaft, U. (1999). When is “nearest neighbor” meaningful?. In *Database Theory—ICDT’99: 7th International Conference Jerusalem, Israel, January 10–12, 1999 Proceedings 7* (pp. 217-235). Springer Berlin Heidelberg.
- Buskirk, T. D., Kirchner, A., Eck, A., & Signorino, C. S. (2018). An introduction to machine learning methods for survey researchers. *Survey Practice*, 11(1).
- Čeponis, D., & Goranin, N. (2018). Towards a robust method of dataset generation of malicious activity for anomaly-based HIDS training and presentation of AWSCTD dataset. *Baltic Journal of Modern Computing*, 6(3), 217-234.
- Čeponis, D., & Goranin, N. (2019). Evaluation of deep learning methods efficiency for malicious and benign system calls classification on the AWSCTD. *Security and Communication Networks*, 2019(1), 2317976.
- Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection: A survey. *ACM computing surveys (CSUR)*, 41(3), 1-58.
- Chaudhari, A., Gohil, B., & Rao, U. P. (2023). A novel hybrid framework for cloud intrusion detection system using system call sequence analysis. *Cluster Computing*, 1-17.
- Chauhan, N. S., Kumar, N., & Eskandarian, A. (2024). A Novel Confined Attention Mechanism Driven Bi-GRU Model for Traffic Flow Prediction. *IEEE Transactions on Intelligent Transportation Systems*.
- Chen, D., Yongchareon, S., Lai, E. M. K., Yu, J., Sheng, Q. Z., & Li, Y. (2022). Transformer with bidirectional GRU for nonintrusive, sensor-based activity recognition in a multiresident environment. *IEEE Internet of Things Journal*, 9(23), 23716-23727.
- Chen, Q., Luley, R., Wu, Q., Bishop, M., Linderman, R. W., & Qiu, Q. (2017). AnRAD: A neuromorphic anomaly detection framework for massive concurrent data streams. *IEEE transactions on neural networks and learning systems*, 29(5), 1622-1636.
- Cotroneo, D., De Simone, L., Liguori, P., & Natella, R. (2020). Fault injection analytics: A novel approach to discover failure modes in cloud-computing systems. *IEEE transactions on dependable and secure computing*, 19(3), 1476-1491.

- Duan, G., Fu, Y., Cai, M., Chen, H., & Sun, J. (2023). DongTing: A large-scale dataset for anomaly detection of the Linux kernel. *Journal of Systems and Software*, 203, 111745.
- Ergen, T., & Kozat, S. S. (2019). Unsupervised anomaly detection with LSTM neural networks. *IEEE transactions on neural networks and learning systems*, 31(8), 3127-3141.
- Ezeme, O. M., Azim, A., & Mahmoud, Q. H. (2020b). Peskea: Anomaly detection framework for profiling kernel event attributes in embedded systems. *IEEE Transactions on Emerging Topics in Computing*, 9(2), 957-971.
- Ezeme, O. M., Mahmoud, Q. H., & Azim, A. (2019). Dream: deep recursive attentive model for anomaly detection in kernel events. *IEEE Access*, 7, 18860-18870.
- Gage, P. (1994). A new algorithm for data compression. *The C Users Journal*, 12(2), 23-38.
- Garg, S., & Batra, S. (2017). A novel ensembled technique for anomaly detection. *International Journal of Communication Systems*, 30(11), e3248.
- Guntur, R.: Api security: Access behavior anomaly dataset. Kaggle, <https://www.kaggle.com/datasets/tangodelta/api-access-behaviour-anomaly-dataset>
- Guo, S., Fang, C., Lin, J., & Wang, Z. (2020, September). A configurable FPGA accelerator of Bi-LSTM inference with structured sparsity. In *2020 IEEE 33rd International System-on-Chip Conference (SOCC)* (pp. 174-179). IEEE.
- Gutierrez-Vasques, X., Bentz, C., & Samardžić, T. (2023). Languages through the looking glass of bpe compression. *Computational Linguistics*, 49(4), 943-1001.
- Haider, W., Hu, J., Xie, Y., Yu, X., & Wu, Q. (2017). Detecting anomalous behavior in cloud servers by nested-arc hidden semi-Markov model with state summarization. *IEEE Transactions on Big Data*, 5(3), 305-316.
- Hu, M., Pan, Y., Zhang, N., & Xu, X. (2023, February). A Review on Zernike Coefficient-Solving Algorithms (CSAs) Used for Integrated Optomechanical Analysis (IOA). In *Photonics* (Vol. 10, No. 2, p. 177). MDPI.
- Jin, B., Sahni, S., Shevat, A. (2018). *Designing Web APIs: Building APIs That Developers Love*. United States: O'Reilly Media.
- Karamanou, A., Brimos, P., Kalampokis, E., & Tarabanis, K. (2022). Exploring the quality of dynamic open government data using statistical and machine learning methods. *Sensors*, 22(24), 9684.
- Karn, R. R., Kudva, P., Huang, H., Suneja, S., & Elfadel, I. M. (2020). Cryptomining detection in container clouds using system calls and explainable machine learning. *IEEE transactions on parallel and distributed systems*, 32(3), 674-691.
- Kim, C., Jang, M., Seo, S., Park, K., & Kang, P. (2021a). Intrusion detection based on sequential information preserving log embedding methods and anomaly detection algorithms. *IEEE Access*, 9, 58088-58101.
- Kim, H., Ahn, S., Ha, W. R., Kang, H., Kim, D. S., Kim, H. K., & Paek, Y. (2021b). Panop: Mimicry-resistant ann-based distributed nids for iot networks. *IEEE Access*, 9, 111853-111864.
- Kiranyaz, S., Avci, O., Abdeljaber, O., Ince, T., Gabbouj, M., & Inman, D. J. (2021). 1D convolutional neural networks and applications: A survey. *Mechanical systems and signal processing*, 151, 107398.
- Kiranyaz, S., Gastli, A., Ben-Brahim, L., Al-Emadi, N., & Gabbouj, M. (2018). Real-time fault detection and identification for MMC using 1-D convolutional neural networks. *IEEE Transactions on Industrial Electronics*, 66(11), 8760-8771.

- Kiranyaz, S., Ince, T., Hamila, R., & Gabbouj, M. (2015, August). Convolutional neural networks for patient-specific ECG classification. In 2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC) (pp. 2608-2611). IEEE.
- Kishore, P., Nayak, G., & Barisal, S. K. (2022). Applying unsupervised system-call based software security techniques for anomaly detection. *Journal of Information and Optimization Sciences*, 43(5), 915-922.
- Kohyarnejadfard, I., Aloise, D., Dagenais, M. R., & Shakeri, M. (2021). A framework for detecting system performance anomalies using tracing data analysis. *Entropy*, 23(8), 1011.
- LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), 2278-2324.
- Lee, K. B., Cheon, S., & Kim, C. O. (2017). A convolutional neural network for fault classification and diagnosis in semiconductor manufacturing processes. *IEEE Transactions on Semiconductor Manufacturing*, 30(2), 135-142.
- Li, Y., Kang, F., Shu, H., Xiong, X., Zhao, Y., & Sun, R. (2023). APIASO: A Novel API Call Obfuscation Technique Based on Address Space Obscurity. *Applied Sciences*, 13(16), 9056.
- Liu, H., & Lang, B. (2019). Machine learning and deep learning methods for intrusion detection systems: A survey. *applied sciences*, 9(20), 4396.
- Liu, L., Chen, C., Zhang, J., De Vel, O., & Xiang, Y. (2019). Insider threat identification using the simultaneous neural learning of multi-source logs. *IEEE Access*, 7, 183162-183176.
- Liu, W., Wang, Q., Zhu, Y., & Chen, H. (2020c). GRU: optimization of NPI performance. *The Journal of Supercomputing*, 76(5), 3542-3554.
- Liu, Y., Hao, X., Zhang, B., & Zhang, Y. (2020b). Simplified long short-term memory model for robust and fast prediction. *Pattern Recognition Letters*, 136, 81-86.
- Liu, Z., Japkowicz, N., Wang, R., Cai, Y., Tang, D., & Cai, X. (2020a). A statistical pattern based feature extraction method on system call traces for anomaly detection. *Information and Software Technology*, 126, 106348.
- Lu, C., Wang, Z., & Zhou, B. (2017). Intelligent fault diagnosis of rolling bearing using hierarchical convolutional network based health state classification. *Advanced Engineering Informatics*, 32, 139-151.
- Lv, S., Wang, J., Yang, Y., & Liu, J. (2018). Intrusion prediction with system-call sequence-to-sequence model. *IEEE Access*, 6, 71413-71421.
- Mahjoub, S., Chrifi-Alaoui, L., Marhic, B., & Delahoche, L. (2022). Predicting energy consumption using LSTM, multi-layer GRU and drop-GRU neural networks. *Sensors*, 22(11), 4062.
- Maji, B., & Swain, M. (2022). Advanced fusion-based speech emotion recognition system using a dual-attention mechanism with conv-caps and bi-gru features. *Electronics*, 11(9), 1328.
- Manakitsa, N., Maraslidis, G. S., Moysis, L., & Fragulis, G. F. (2024). A Review of Machine Learning and Deep Learning for Object Detection, Semantic Segmentation, and Human Action Recognition in Machine and Robotic Vision. *Technologies*, 12(2), 15.
- Min, E., Long, J., Liu, Q., Cui, J., & Chen, W. (2018). TR-IDS: Anomaly-based intrusion detection through text-convolutional neural network and random forest. *Security and Communication Networks*, 2018.

- Mitiche, I., Nesbitt, A., Conner, S., Boreham, P., & Morison, G. (2020). 1D-CNN based real-time fault detection system for power asset diagnostics. *IET Generation, Transmission & Distribution*, 14(24), 5766-5773.
- Mora-Gimeno, F. J., Mora-Mora, H., Volckaert, B., & Atrey, A. (2021). Intrusion detection system based on integrated system calls graph and neural networks. *IEEE Access*, 9, 9822-9833.
- Muhammad, I., & Yan, Z. (2015). Supervised Machine Learning Approaches: A Survey. *Ictact Journal on Soft Computing*, 5(3).
- Osamor, F., & Wellman, B. (2022). Deep learning-based hybrid model for efficient anomaly detection. *International Journal of Advanced Computer Science and Applications*, 13(4).
- Peddoju, S. K., Upadhyay, H., Soni, J., & Prabakar, N. (2020). Natural language processing based anomalous system call sequences detection with virtual memory introspection. *International Journal of Advanced Computer Science and Applications*, 11(5).
- Raj, C., Agarwal, A., Bharathy, G., Narayan, B., & Prasad, M. (2021). Cyberbullying detection: Hybrid models based on machine learning and natural language processing techniques. *Electronics*, 10(22), 2810.
- Ranshous, S., Shen, S., Koutra, D., Harenberg, S., Faloutsos, C., & Samatova, N. F. (2015). Anomaly detection in dynamic networks: a survey. *Wiley Interdisciplinary Reviews: Computational Statistics*, 7(3), 223-247.
- Rish, I. (2001, August). An empirical study of the naive Bayes classifier. In *IJCAI 2001 workshop on empirical methods in artificial intelligence* (Vol. 3, No. 22, pp. 41-46).
- Rocha, S. L., de Mendonca, F. L. L., Puttini, R. S., Nunes, R. R., & Nze, G. D. A. (2023). DCIDS—Distributed Container IDS. *Applied Sciences*, 13(9301), 9301.
- Sennrich, R., Haddow, B., & Birch, A. (2015). Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*.
- Sha, W., Zhu, Y., Chen, M., & Huang, T. (2015). Statistical learning for anomaly detection in cloud server systems: A multi-order Markov chain framework. *IEEE transactions on cloud computing*, 6(2), 401-413.
- Shanthamallu, U. S., Spanias, A., Tepedelenlioglu, C., & Stanley, M. (2017, August). A brief survey of machine learning methods and their sensor and IoT applications. In *2017 8th International Conference on Information, Intelligence, Systems & Applications (IISA)* (pp. 1-8). IEEE.
- Shen, S. L., Atangana Njock, P. G., Zhou, A., & Lyu, H. M. (2021). Dynamic prediction of jet grouted column diameter in soft soil using Bi-LSTM deep learning. *Acta Geotechnica*, 16(1), 303-315.
- Shibata, Y., Kida, T., Fukamachi, S., Takeda, M., Shinohara, A., Shinohara, T., & Arikawa, S. (1999). Byte pair encoding: A text compression scheme that accelerates pattern matching.
- Shin, Y., & Kim, K. (2020). Comparison of anomaly detection accuracy of host-based intrusion detection systems based on different machine learning algorithms. *International Journal of Advanced Computer Science and Applications*, 11(2).
- Song, C., Zhou, Z., Yu, Y., Shi, M., & Zhang, J. (2024). An improved Bi-LSTM method based on heterogeneous features fusion and attention mechanism for ECG recognition. *Computers in Biology and Medicine*, 169, 107903.

- van Leeuwen, R., & Koole, G. (2023). Anomaly detection in univariate time series incorporating active learning. *Journal of Computational Mathematics and Data Science*, 6, 100072.
- Wang, Y., Jiang, Y., & Lan, J. (2021). Intrusion detection using few-shot learning based on triplet graph convolutional network. *Journal of Web Engineering*, 20(5), 1527-1552.
- Xie, H., Randall, M., & Chau, K. W. (2022). Green roof hydrological modelling with GRU and LSTM networks. *Water Resources Management*, 36(3), 1107-1122.
- Yamashita, R., Nishio, M., Do, R. K. G., & Togashi, K. (2018). Convolutional neural networks: an overview and application in radiology. *Insights into imaging*, 9, 611-629.
- Yang, S., Jin, A., Nie, W., Liu, C., & Li, Y. (2022). Research on SSA-LSTM-based slope monitoring and early warning model. *Sustainability*, 14(16), 10246.
- Yao, G., Lei, T., & Zhong, J. (2019). A review of convolutional-neural-network-based action recognition. *Pattern Recognition Letters*, 118, 14-22.
- Zhan, D., Tan, K., Ye, L., Yu, X., Zhang, H., & He, Z. (2023). An Adversarial Robust Behavior Sequence Anomaly Detection Approach Based on Critical Behavior Unit Learning. *IEEE Transactions on Computers*.

ÖZ GEÇMİŞ

Varol ŞAHİN, Sivas Yenişehir Lisesi’ni bitirdikten sonra 2013 yılında Ondokuz Mayıs Üniversitesi Yabancı Diller Yüksekokulu’nda hazırlık sınıfından başarıyla mezun olarak 2014 yılında Ondokuz Mayıs Üniversitesi Bilgisayar Mühendisliği bölümünde lisans eğitimine başladı. 2015-2016 Eğitim Öğretim döneminde Erasmus programı kapsamında 1 yıl süre ile University “Instituto Superior Miguel Torga”. Coimbra, Portekiz üniversitesinde lisans eğitimine devam etti. 12/11/2018 - 15/03/2022 tarihleri arasında Datassist Bilgi Teknolojileri isimli şirkette Java Yazılım Geliştirici pozisyonunda çalıştı. Ocak 2020’de bilgisayar mühendisliği bölümünden mezun oldu. 15/03/2022 tarihinde Ondokuz Mayıs Üniversitesi Bilgisayar Mühendisliği bölümünde araştırma görevlisi olarak çalışmaya başlamış olup görevine halen devam etmektedir.

İletişim Bilgileri

ORCID ID : 0009-0000-3000-9899

Yayınlar:

1. Sahin, V., Satilmis, H., Yazar, B.K., Akleylek, S. (2024). Anomaly Detection with Machine Learning Models Using API Calls. 2nd International Conference on Information Technologies and Their Applications. ITTA 2024. pp.1-13. April 23-25, 2024, Bakü, Azerbaycan.
2. Sahin, V., Arat, F., Akleylek, S. (2024). Anomaly Detection with API Calls By Using Machine Learning: Systematic Literature Review. Current Trends in Computing.
3. Yazar, B.K., Satilmis, H., Sahin, V., Akleylek, S. (2024). Detection of Malware and Benign Samples in AWSCTD Dataset with Kolmogorov-Arnold Networks. ICAIAME 2024: 6th International Conference on Artificial Intelligence and Applied Mathematics in Engineering.