T.C.

MİLLÎ SAVUNMA ÜNİVERSİTESİ

ATATÜRK STRATEJİK ARAŞTIRMALAR VE LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

OTOMATİK MAKİNE ÖĞRENMESİ İLE ENDÜSTRİYEL NESNELERİN İNTERNETİ TABANLI UÇ SİSTEMLERDE GERÇEK ZAMANLI SALDIRI TESPİT SİSTEMİ GELİŞTİRİLMESİ

Anıl SEZGİN

DOKTORA TEZİ

Bilgisayar Mühendisliği Ana Bilim Dalı

Bilgisayar Mühendisliği Programı

Tez Danışmanı

Doç. Dr. Aytuğ BOYACI

Kasım 2023

T.C.

MİLLÎ SAVUNMA ÜNİVERSİTESİ

ATATÜRK STRATEJİK ARAŞTIRMALAR VE LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

OTOMATİK MAKİNE ÖĞRENMESİ İLE ENDÜSTRİYEL NESNELERİN İNTERNETİ TABANLI UÇ SİSTEMLERDE GERÇEK ZAMANLI SALDIRI TESPİT SİSTEMİ GELİŞTİRİLMESİ

Anıl SEZGİN tarafından hazırlanan tez çalışması 10.11.2023 tarihinde aşağıdaki jüri tarafından Millî Savunma Üniversitesi, Atatürk Stratejik Araştırmalar ve Lisansüstü Eğitim Enstitüsü, Bilgisayar Mühendisliği Ana Bilim Dalı, Bilgisayar Mühendisliği Programı **DOKTORA TEZİ** olarak kabul edilmiştir.

Doç. Dr. Aytuğ BOYACI
Millî Savunma Üniversitesi
Tez Danışmanı

Jüri Üyeleri

Doç. Dr. Aytuğ BOYACI, Tez Danışmanı
Millî Savunma Üniversitesi _____

Doç. Dr. Can EYÜPOĞLU, Üye
Millî Savunma Üniversitesi _____

Dr. Öğr. Üyesi Hüseyin Fehmi Selim BAYRAKLI, Üye
Millî Savunma Üniversitesi _____

Doç. Dr. Mustafa ULAŞ, Üye
Fırat Üniversitesi _____

Doç. Dr. Oğuz ATA, Üye
Altınbaş Üniversitesi _____

Danışmanım Doç. Dr. Aytuğ BOYACI sorumluluğunda tarafımca hazırlanan “Otomatik Makine Öğrenmesi ile Endüstriyel Nesnelerin İnterneti Tabanlı Uç Sistemlerde Gerçek Zamanlı Saldırı Tespit Sistemi Geliştirilmesi” başlıklı çalışmada veri toplama ve kullanımında gerekli yasal izinleri aldığımı, diğer kaynaklardan aldığım bilgileri ana metin ve referanslarda eksiksiz gösterdiğimi, araştırma verilerine ve sonuçlarına ilişkin çarpıtma ve/veya sahtecilik yapmadığımı, çalışmam süresince bilimsel araştırma ve etik ilkelerine uygun davrandığımı beyan ederim. Beyanımın aksinin ispatı halinde her türlü yasal sonucu kabul ederim.

Anıl SEZGİN

Babama…

TEŞEKKÜR

Rehberliği, desteği ve benimle paylaştığı bilgileri ile bu tezin ortaya çıkmasında büyük pay sahibi olan tez danışmanım Doç. Dr. Aytuğ BOYACI'ya, öneri ve destekleriyle çalışmama katkıda bulunan Tez İzleme Komitesi üyeleri Doç. Dr. Can EYÜPOĞLU'na ve Doç. Dr. Mustafa ULAŞ'a, bana değerli zamanlarını ayırdıkları için tez savunma juri üyeleri Doç. Dr. Oğuz ATA'ya ve Dr. Öğr. Üyesi Hüseyin Fehmi Selim BAYRAKLI'ya, canımdan çok sevdiğim anneme, babama, sevgili eşim Büşra'ya ve ailemizin en yeni üyesi olan oğlum Hüseyin Poyraz'a teşekkür ederim.

Anıl SEZGİN

İÇİNDEKİLER

KISALTMA LİSTESİ

AutoML	Automated Machine Learning
CNN	Convolutional Neural Network
DNN	Deep Neural Network
IDS	Intrusion Detection
IoT	Internet of Things
IIoT	Industrial Internet of Things
ML	Machine Learning
UML	Unified Modeling Language

ŞEKİL LİSTESİ

TABLO LİSTESİ

ÖZET

Otomatik Makine Öğrenmesi ile Endüstriyel Nesnelerin İnterneti Tabanlı Uç Sistemlerde Gerçek Zamanlı Saldırı Tespit Sistemi Geliştirilmesi

Anıl SEZGİN

Bilgisayar Mühendisliği Ana Bilim Dalı

Bilgisayar Mühendisliği Programı

Doktora Tezi

Tez Danışmanı: Doç. Dr. Aytuğ BOYACI

Saldırı tespit sistemleri, endüstriyel nesnelerin interneti sistemlerinde ağ güvenliğini sağlamak ve sistemi tehlikeye atabilecek her türlü kötü niyetli faaliyeti tespit etmek ve bunlara yanıt vermek için tercih edilmektedir. Anomali tabanlı saldırı tespit sistemleri olağandışı faaliyetleri tespit etmek için imza veritabanlarına ihtiyaç duymadığından, sıfır gün saldırılarını tespit edebilme avantajına sahiptir. Bu çalışma, veri setleri ve makine öğrenmesi süreçleri üzerindeki kontrolü iyileştirmek amacıyla otomatik makine öğrenmesi yaklaşımını kullanarak saldırı tespit sistemleri için makine öğrenmesi süreçlerini otomatikleştirmeyi önermektedir. Önerilen mimari, saldırı tespit etmek için otomatik öğrenmesi tekniklerini kullanmaktadır. Amaç, ön işleme, özellik seçimi, özellik çıkarımı, model seçimi ve hiperparametre optimizasyonunu otomatikleştirerek saldırı tespiti için uygun bir makine öğrenmesi modeli oluşturmaktır. Geliştirdiğimiz çerçeve, deneysel sonuçların ortaya koyduğu üzere başarılı şekilde uygun bir model sunmaktadır. Saldırı tespit sisteminin potansiyel olarak zararlı faaliyetleri tespit etme ve önleme yeteneğini geliştirmek için makine öğrenmesi teknikleri otomatikleştirilebilir. Bu, endüstriyel nesnelerin interneti üzerinden iletilen verilerin bütünlüğünü, güvenliğini ve

gizliliğini sağlayacaktır. Çerçevemiz, saldırı tespiti için güçlü ve etkili bir araç olup ağ güvenliğini arttırmak için otomatik makine öğrenmesi yaklaşımlarını kullanan kapsamlı bir çözüm sunmaktadır. Eksik verilerle ilgilenmek için ön işleme modülünde üç farklı impütasyon yöntemi uygulanmakta ve saldırı tespit sisteminin eksik veri gerçeğiyle karşı karşıya kalmasında sağlamlığını arttırmaktadır. Özellik seçimi modülü, genetik algoritmayı ve Shapley değerlerini birleştirerek hibrit bir yaklaşım kullanmaktadır. Özellik çıkarımı modülünde Autoencoder, PCA ve LDA yöntemleri kullanılmaktadır. Parametre optimizasyon modülünde yere alan 14 farklı sınıflandırma yöntemi ile her bir algoritmaya özgü parametrelerin kapsamlı bir şekilde incelenmesi ve optimize edilmesi mümkündür. Bu parametrelerin dikkatli bir şekilde ayarlanmasıyla, sistem potansiyel saldırıları tespit etme konusunda hem uyum yeteneğini hem de doğruluk oranını artırmaktadır. Deneysel sonuçlarımıza göre, çerçevemiz saldırı tespiti için geleneksel yöntemleri iyileştirmektedir. Açık veri setlerinde %14.39'a varan bir doğruluk oranı artışıyla saldırıları tespit ederken makine öğrenmesi yöntemlerindeki eğitim ve test için gereken süreyi de azaltmaktadır.

Anahtar Kelimeler: Otomatik makine öğrenmesi, saldırı tespit sistemi, endüstriyel nesnelerin interneti, parametre optimizasyonu

ABSTRACT

Development of a Real-Time Intrusion Detection System in Industrial Internet of Things Based Edge Systems with Automated Machine Learning

Anıl SEZGİN

Department of Computer Engineering

Doctor of Philosophy Thesis

Supervisor: Doç. Dr. Aytuğ BOYACI

Industrial Internet of Things (IIoT) network security depends on the Intrusion Detection System (IDS) to detect and respond to any malicious activity that might compromise the system. Since anomaly-based intrusion detection systems do not rely on signature databases to detect unusual activity, they have advantage of being able to identify zero-day attacks. This study proposes automating the machine learning processes for IDS using automated machine learning (AutoML) technique in order improve control over datasets and the process. Automated machine learning techniques are used by our innovative architecture to detect intrusions. The objective is to identify an appropriate machine learning model for detecting intrusions through the automation of data preprocessing, feature selection, model choice, and the optimization of hyperparameters. Framework successfully provides a suitable model, as experimental results demonstrate. To improve the IDS's ability to detect and prevent potentially harmful activity, machine learning techniques can be automated. This will ensure integrity, security, and confidentiality of data transmitted over IIoT network. Our framework is powerful and effective tool for intrusion detection, providing comprehensive solution that utilizes the most recent developments in automated machine learning approaches to improve network

security. Three imputation methods are implemented in preprocessing module to deal with missing data, ensuring intrusion detection system's robustness in facing reality of missing data. Feature selection module utilizes hybrid approach that combines genetic algorithms and Shapley values. With 14 classification methods included in Parameter Optimization module, it’s possible to comprehensively investigate and optimize the parameters related to each algorithm. By precise optimization of these parameters, framework improves either its adaptability and accuracy in detecting potential intrusions. According to experimental results, framework improves traditional methods for intrusion detection. It detects attacks with an accuracy increase of up to 14.39% on public datasets while reducing time required for training and testing in machine learning methods.

Keywords: Automated machine learning, intrusion detection system, industrial internet of things, parameter optimization

1
GİRİŞ

Nesnelerin İnterneti (Internet of Things, IoT) her şeyin birbirine bağlı olduğu bir dünya sunmaktadır. Nesnelerin İnterneti kavramı, sağlık, savunma, ulaşım, tarım, akıllı şehirler, akıllı evler, giyilebilir teknolojiler dahil olmak üzere birçok alanda kullanılmaktadır. IoT, pratik hedeflere ulaşmak için akıllı cihazların ve yönetim sistemlerinin entegrasyonunu içermektedir. IoT teknolojisinin benimsenmesi hayatımızı önemli ölçüde değiştirmiştir.

Endüstriyel ortamlara IoT'nin uygulanması, "Endüstriyel Nesnelerin İnterneti" (Industrial Internet of Things, IIoT) adı verilen bir kavramın doğmasına yol açmıştır. Cihazların birbirine bağlanması endüstride yeni bir fikir değildir, çünkü etkili bir şekilde çalışmak için koordinasyon ve iş birliği gerektirirler. Cihazlar arasında veri alışverişi ve iletişim yaygındır ancak genellikle bir üretim tesisi veya endüstrinin belirli bir bölgesi ile sınırlıdır. Denetleyici Kontrol ve Veri Toplama (Supervisory Control and Data Acquisition, SCADA) sistemleri uzak sensörlerden ve endüstriyel ekipmanlardan veri toplar ve bunları izleme veya kontrol için merkezi bir konuma iletir [1].

Endüstriyel IoT, kritik altyapıda ölçeklenebilirliği, verimliliği ve birlikte çalışabilirliği arttırmak için sensörlerin, veri analitiğinin ve akıllı makinelerin kullanılması anlamına gelir ve bu da otomasyonun iyileştirimesine ve kurumsal üretkenliğin artmasına yol açar [2]. Verimlilik artışı hedefine ulaşmak için bazı engellerin aşılması gerekmektedir. En yüksek öncelik, endüstriyel altyapının ve bileşenlerin güvenliğidir. Kritik altyapıya yönelik siber saldırılar, endüstriler için önemli mali kayıplara yol açabilir.

1.1 Tezin Amacı

Endüstriyel nesnelerin interneti sistemlerinin hızlı yaygınlaşması, karmaşık endüstriyel süreçlerin gerçek zamanlı olarak izlenmesini, kontrol edilmesini ve optimize edilmesini sağlayarak endüstrilerde devrim yaratmıştır. Ancak bu artan bağlanabilirlik ve entegrasyon, endüstriyel nesnelerin interneti sistemlerini yeni ve gelişen güvenlik tehditlerine de maruz bırakmaktadır. Kötü niyetli aktörler, birbirine bağlı cihazlar, sensörler ve ağlardaki güvenlik açıklarından faydalanarak finansal kayıplar, güvenlik tehlikeleri ve operasyonların tehlikeye girmesi gibi potansiyel olarak ciddi sonuçlara yol açmaktadır. Bu nedenle, özellikle endüstriyel nesnelerin interneti sistemlerinin benzersiz

özelliklerine hitap eden sağlam ve uyarlanabilir saldırı tespit çözümleri geliştirmek büyük önem taşımaktadır.

Geleneksel ağlar için tasarlanan saldırı tespit teknikleri, endüstriyel nesnelerin interneti sistemlerine uygulandığından önemli kısıtlamalarla karşılaşmaktadır. Endüstriyel nesnelerin interneti dağıtımlarının ölçeği ve heterojenliğinin yanı sıra bu ortamların dinamik doğası, IIoT verilerinin, cihaz türlerinin ve iletişim protokollerinin karmaşıklığını etkili bir şekilde ele alabilen özel yaklaşımlar gerektirir. Ayrıca, IIoT cihazlarının kaynak kısıtlı doğası, sınırlı işlem gücü ve bellek kısıtlamaları dahilinde çalışabilen hafif ve verimli saldırı tespit mekanizmaları gerekmektedir.

Bu zorlukların üstesinden gelmek ve IIoT sistemleri için etkili saldırı tespit çözümleri geliştirmek için, bu çalışma otomatik/otomatikleştirilmiş makine öğrenmesi tekniklerinin kullanılmasını önermektedir. Otomatik makine öğrenmesi (Automated Machine Learning, AutoML), makine öğrenmesi hattının çeşitli aşamalarını otomatikleştirmek için güçlü bir yaklaşım olarak ortaya çıkmıştır. Bu çalışmanın hedefi, Endüstriyel Nesnelerin İnterneti (IIoT) sistemlerinde izinsiz giriş tespiti yapabilen otomatik makine öğrenmesi yapısını tasarlamaktır. Çalışma ile önerilen çerçeve, AutoML kullanarak izinsiz giriş tespinin temel bileşenlerini otomatikleştirmeyi ve IIoT operatörlerinin hızlı bir şekilde uyarlanabilir ve sağlam sistemler kurmasını sağlamayı amaçlamaktadır. Çalışma ile önerilen AutoML çerçevesi, veri ön işleme, model seçimi, özellik mühendisliği ve hiperparametre optimizasyonu gibi saldırı tespit sürecinin kritik yönlerini otomatikleştirerek IIoT sistemlerinde saldırı tespinin etkinliğini ve verimliliğini arttırmayı amaçlamaktadır.

1.2 Tezin Önemi ve Özgün Değeri

IIoT sistemlerinde saldırı tespiti için otomatik makine öğrenmesi (Automated Machine Learning, AutoML) çerçevesi oluşturmak, endüstriyel IoT sistemlerinin güvenliğini ve dayanıklılığını arttırma bağlamında büyük önem taşımaktadır. Tezin bu bölümü, IIoT alanında izinsiz giriş tespiti için özel olarak tasarlanmış bir AutoML çerçevesi geliştirmenin önemini ve özgünlüğünü vurgulamayı amaçlamaktadır.

Önerilen çerçeve, AutoML tekniklerinden yararlanarak IIoT uç cihazlarının güvenlik duruşunu geliştirmekte ve kritik endüstriyel süreçlerin bütünlüğünü, gizliliğini ve kullanılabilirliğini sağlayacaktır. Bu çalışma, IIoT sistemlerinin karşılaştığı gelişen

tehditleri azaltabilecek proaktif savunma mekanizmalarının geliştirilmesine katkıda bulunmaktadır.

Elde edilen bulgular ve ortaya çıkan AutoML çerçevesi, IIoT sistemlerinin genel esnekliğine ve güvenilirliğine katkıda bulunuyor. IIoT operatörleri, izinsiz giriş tespit sürecini otomatikleştirerek potansiyel güvenlik olaylarını gerçek zamanlı tespit edip bunlara yanıt verebilir, saldırıların etkisini en aza indirebilir ve kesintisiz operasyonlar sağlayabilir.

Önerilen AutoML çerçevesi, izinsiz giriş tespitinde insan uzmanlığına ve manuel müdahaleye olan bağımlılığı azaltmaktadır. Bu çerçeve, siber güvenlik bilgisine ve makine öğrenmesi bilgisine sahip olmayan IIoT operatörlerinin saldırı tespit sistemlerini etkin bir şekilde kurmalarını ve yönetmelerini sağlayacaktır. Çerçeve, veri ön işleme, model seçimi, özellik mühendisliği ve hiperparametre optimizasyonu gibi karmaşık süreçleri basitleştirerek IIoT ortamlarındaki güvenlik operasyonlarını kolaylaştırır.

IIoT dağıtımlarının katlanarak büyümesiyle birlikte, ölçeklenebilirlik ve uyarlanabilirlik kritik hususlar haline gelmektedir. Bu çalışmada sunulan AutoML çerçevesi, IIoT sistemlerinin ölçeğini ve heterojenliğini idare edecek şekilde tasarlanmıştır. Değişen saldırı modellerine ve sistem dinamiklerine uyum sağlayabilir ve saldırı tespit yeteneklerinin gelişen tehdit ortamıyla birlikte gelişmesini sağlar. Bu çalışma sayesinde saldırı tespitinin, çeşitli IIoT mimarilerindeki uç cihazlara sorunsuz bir şekilde entegre olabilmesi sağlanacaktır.

Bu araştırmanın özgünlüğü, otomatik makine öğrenmesi tekniklerinin IIoT sistemlerinde izinsiz giriş tespiti alanına entegre edilmesinde yatmaktadır. AutoML çeşitli alanlarda dikkat çekmiş olsa da IIoT güvenliğine uygulanması nispeten keşfedilmemiştir. Bu çalışma, AutoML'nin izinsiz giriş tespitinin kritik yönlerini otomatikleştirmek için kullanılmasına öncülük ederek, kaynak kısıtlı IIoT ortamlarında verimli ve etkili tehdit tespiti sağlayacaktır.

1.3 Literatür Özeti

Endüstriyel Nesnelerin İnterneti (IIoT) sistemlerine yönelik siber saldırılar, üretim süreçlerinin durmasına, ekipmanlara zarar verilmesine, fikri mülkiyetin kaybına ve insan güvenliğinin tehdit edilmesi gibi ciddi sonuçlara yol açabilir. Bu tür riskleri minimize etmek amacıyla, Saldırı Tespit Sistemleri (IDS) IIoT'nin güvenlik altyapısında kritik bir

rol oynamaktadır. IDS'ler, siber saldırıları belirleyip bu saldırılara karşılık vermek üzere geliştirilmiştir ve etkin kullanımları, IIoT ağları üzerindeki siber saldırı etkilerini ciddi anlamda düşürebilir. Fakat, IIoT sistemlerinin karmaşık ve değişken yapısı, onlara özgü zorluklar taşıdığından, bu sistemlere yönelik etkin bir IDS geliştirmek oldukça zorlu bir iştir.

Hem ağ hem de ana bilgisayar seviyelerindeki siber saldırıları zamanında ve otomatik bir şekilde tespit edip sınıflandırabilen [3] çalışmasında ölçeklenebilir ve etkili bir IDS geliştirmek için makine öğrenmesi tekniklerinin, özellikle de derin sinir ağlarının (Deep Neural Network, DNN) kullanılması önerilmiştir. Bu çalışma, sürekli evrilen siber saldırılar ve dışa açık kötü niyetli yazılım veri setlerinin düzenli olarak karşılaştırılmasının gerekliliği bağlamında, bu tür verilerin tespitinde makine öğrenmesi tekniklerinin kullanımının karşılaştığı zorlukları ele almaktadır. Çalışma birden fazla veri kümesi üzerinde kapsamlı bir değerlendirme sunmakta ve DNN'lerin klasik makine öğrenmesi sınıflandırılarına göre üstün performansını ortaya koymaktadır. Önerilen DNN modeli, IDS verilerinin soyut ve yüksek boyutlu özellik temsillerini öğrenme yeteneğine sahiptir ve olası siber saldırıları proaktif olarak uyarmak için ağ trafiğinin ve ana bilgisayar düzeyindeki olayların gerçek zamanlı izlenmesi için oldukça ölçeklenebilir ve hibrit bir DNN çerçevesinin önerilmesiyle sonuçlanmaktadır.

Bilgisayar ağları için bir saldırı tespit modeli [4] çalışmasında önerilmiştir. Bu çalışma, özellikler arasında güçlü bir korelasyon elde etmek için Pearson korelasyon algıritmasını benimseyen korelasyon tabanlı bir özellik seçim tekniği önermektedir. Seçilen özellikler, sınıflandırma modellemesi için Random Forest algoritması kullanılarak uygulanmıştır. Sonuçlar, yüksek algılama doğruluğu (high detection accuracy), hassasiyet (precision) ve recall metriklerini içermektedir.

2015 yılında saldırı tespitinde araştırma amacıyla UNSW-NB15 veri kümesi oluşturulmuştur. [5] çalışmasında amaç, makine öğrenmesi tekniklerini kullanarak ve ilgisiz/gereksiz özelliklerin neden olduğu yüksek boyutluluk durumunu azaltarak ağ sistemlerinde saldırı tespitini iyileştirebilecek önemli özellikleri belirlemektir. Çalışma, özellik seçimi yaklaşımının etkinliğini değerlendirmek için önerilen özellik alt kümesini KddCup99 veri kümesinde özellik seçimi üzerinde yapılan önceki çalışmalarla karşılaştırmaktadırç Araştırma amaçlı yayınlanmış veri eksikliği, ağ saldırı tepsiti alanındaki araştırmacılar için önemli bir sorundur ve UNSW-NB15 veri kümesi bu sorunu ele amak için oluşturulmuş bir veri kümesidir.

Yeni bir saldırı tespit sistemi modelinin sunulduğu [6] çalışmasında, meta sınıflandırıcı tabanlı özellik seçimi ve overfitting önlemek için çapraz doğrulama yaparak topluluk öğrenme yöntemi tercih edilmiştir. Önerilen sistem, ağ performansını korurken tehditleri engellemek ve güvenlik açığı istismarlını gerçek zamanlı olarak tespit etmek için tasarlanmış genişletilmiş bir IDS çözümü ağ güvenlik teknolojisi olan saldırı önleme sistemlerinin (intrusion prevention system, IPS) karşılaştığı zorlukları ele almayı amaçlamaktadır. Çalışma, hesaplama maliyeti ve özellik seçimi gibi sorunların, IPS'nin etkinliğini sınırlayabileceğini de belirtmektedir. Önerilen sistem, tüm saldırılar için ortak özellikler kullanmak yerine her saldırı türü için daha iyi bir bilgilendirici özellik alt kümesi seçerek genel sınıflandırma doğruluğunu arttırmakta ve öğrenme/test sürecini hızlandırarak daha yüksek algılama oranları ve daha az yanlış alarm sağlamaktadır. Sistemin çevrimiçi modda çalışma kabiliyeti ve ölçeklenebilirliği de çeşitli deneylerle gösterilmiştir.

Genetik algoritmalar, Yazılım Tanımlı Ağ (Software Defined Network, SDN) sistemlerinde saldırı tespiti ve önlemede kural üretimini otomatikleştirmek için kullanılmaktadır [7]. SDN, ağın merkezi bir şekilde yönetilmesini sağlayarak daha esnek ve özelleştirilebilir bir ağ hizmeti sunmasına rağmen ağ saldırıları hala ciddi bir sorun olmaya devam etmektedir. Saldırı tespit sistemleri çok çeşitli tehditleri tespit edebilir ve tanımlayabilir, ancak yeni saldırıların oluşumunu tahmin etmek ve manuel kural yazmak zordur. Çalışmanın amacı, genetik algoritma kullanarak kural oluşturma sürecini otomatikleştirmek, yeni saldırıları tespit etmek, önlemek ve etkilerini azaltmak için kurallar oluşturmaktır. Çalışma, SDN sistemlerinde yeni ve bilinmeyen ağ saldırılarının yarattığı zorlukları ele alarak, öngörülen durumlar için kural oluşturma sürecini otomatikleştirmek için saldırı tespit sistemlerini genetik algoritma ile ilişkilendirmeye odaklanmaktadır.

Araştırmacılar, istatistiksel ya da makine öğrenmesi yöntemlerine dayalı mevcut çözümleri değerlendirip bu tekniklerin sınırlarını [8] çalışmasında ele almışlardır. Bu çalışma, uygulamalarda karşılaşılan eksik veri sorununu ve bu durumun makine öğrenimi yöntemleriyle elde edilen sonuçların doğruluğunu nasıl etkileyebileceğini ele alıyor. Potansiyel çözüm yolları olarak Oto Kodlayıcı ağlar ve Karşıt Eğitim önerilmiş, ancak mevcut yöntemlerin ağ yapısı ve eğitim stratejileri nedeniyle yetersiz kaldığı belirtilmiştir. Araştırmacılar, ağ yapısını ve optimizasyon stratejilerini dikkatlice seçerek diğer derin öğrenme çözümlerine kıyasla daha iyi performans sergileyebileceklerini iddia

ediyor ve girdilerin stokastik bozulmasının ağ performansı üzerindeki etkisini ortaya koyuyorlar.

Güvenlik riskerini azaltmak için makine öğrenmesi tabanlı saldırı tespit sistemleri önerilmiştir, ancak bunları genellikle IIoT ortamlarına yaygın olarak bulunan dengesiz veri dağılımından olumsuz etkilenmektedir. [9] çalışmasında, yeni bir maliyete bağlı kayıp fonksiyonu ile yığılmış oto kodlayıcılar (Stacked AutoEncoder) ve evrişimli sinir ağlarının hibrit bir modeli olan EvolCostDeep'i tanıtarak bu sorunu ele almak için yeni bir yaklaşım önerilmiştir. Kayıp fonksiyonu evrimsel bir algoritma kullanarak maliyetleri belirleyerek modelin parametrelerini optimize etmektedir. Araştırmacılar ayrıca, büyük IIoT trafik verilerinin ölçeklenebilirliğini ele almak için tasarlanmış DeepIDSFog adlı sis hesaplama özellikli bir çerçeve sunmuştur. Bu çerçeve, hem model düzeyinde hem de görev düzeyinde EvolCostDeep modelini paralelleştirerek sınıf dengesizliği sorununa etkili bir çözüm sağlamaktadır. Araştırmacılar, ToN-IoT ve UNSW-NB15 veri setleri üzerinde çalışmalarını yaparak, önerilen çerçevenin hem doğruluk hem de hız açısından diğer yöntemlerden daha iyi çalıştığını sunmuşlardır. Bu araştırma, kritik endüstri alanlarında siber saldırı ve yetkisiz erişimleri minimize etmeye yönelik bir çözüm önererek Internet of Things (IoT) ortamlarına uygun, etkin ve genişleyebilir saldırı tespit sistemlerinin gelişimine katkı sağlamıştır.

Yapay zeka tekniklerinden, özellikle de makine ve derin öğrenme yöntemlerinden yararlanarak anomali tabanlı saldırı tespit sistemlerinin etkinliğinin artırılabileceği ve IoT ile IIoT ağlarında meydana gelebilecek siber saldırıların minimize edilebileceği [10] çalışmasında belirtilmiştir. Bu bağlamda, araştırmacılar birden fazla tahmin modelinin tahminlerini birleştirerek daha doğru sonuçlar elde eden bir ansambl sınıflandırma modeli geliştirmektedirler. Ayrıca, tüm model ağırlıklarının en uygun performansı sağlayacak şekilde ayarlanmasını sağlayan Adam optimizasyon algoritması ile donatılmış bir yapay sinir ağından da faydalanılmıştır. Bu yöntem bir veri seti üzerinde denenmiş ve yapılan deneyler sonucunda, bu birleşik ansambl modelinin, tek başına çalışan diğer modellere kıyasla daha iyi bir performans gösterdiği ve %99.7 gibi oldukça yüksek bir doğruluk oranına ulaştığı gözlemlenmiştir.

Boyut azaltma, ki-kare özellik seçimi ve çok sınıflı destek vektör kombinasyonunu kullanan bir saldırı tespit modelinin önerildiği [11] çalışmasında araştırmacılar, saldırı tespiti için tek bir sınıflandırıcı algoritma kullanmanın yeterli olmadığını, çünkü işlenmesi gereken büyük miktarda veri göz önüne alındığından, düşük yanlış alarm

oranlarıyla yüksek saldırı tespit oranlarına ulaşamadığını savunmaktadır. Bu problemi çözmek üzere geliştirilen model, veri boyutunu, bilgi kaybına uğramadan en alakalı özelliklerin bir araya getirildiği bir özellik setine indirgeyerek ve sonrasında ki-kare özellik seçimi yöntemiyle en ilişkili özellikleri tespit ederek ele almaktadır. Bu azaltılmış özellik kümesi, çok sınıflı bir destek vektör makinesi sınıflandırıcısını eğitmek için kullanılmaktadır. Ayrıca, destek vektör sınıflandırıcısının parametrelerini, yani gama ve overfitting sabitini optimize etmek için bir parametre optimizasyon tekniği uygulanmıştır. Önerilen metod, NSL-KDD adı verilen ve KddCup99 veri setinin gelişmiş bir versiyonu olan veri seti üzerinde test edilmiştir. Elde edilen sonuçlar, mevcut yöntemlere göre daha yüksek tespit başarısı ve daha az yanlış alarm oranı sağladığını ortaya koymuştur.

IIoT ortamında siber saldırıların tespit edilmesinde akıllı karar verme ihtiyacının artması [12] çalışmasında ele alınmıştır. IIoT'nin yükselişiyle birlikte, optimum endüstriyel operasyonlar için büyük miktarda veri toplanmakta ve işlenmektedir, ancak bu aynı zamanda aktörlerin siber saldırılar gerçekleştirmesi için yeni fırsatlar sunmaktadır. Bununla mücadelete etmek için bu çalışmada, derin öğrenme modellerini, özellikle de izinsiz giriş tespiti ve sınıflandırması için etkili teknikler olan CNN ve uzun kısa süreli bellek (Long Short-Term Memory, LSTM) kombinasyonunu kullanan bir izinsiz giriş tespit sistemi önerilmiştir. Önerilen model, IoT ve IIoT uygulamalarının gerçek bir trafik ağını içeren Edge-IIoTset veri kümesi kullanılarak değerlendirilmiş ve geleneksel makine öğrenmesi modellerinin yanı sıra yeni modellerle karşılaştırılmıştır. Sonuçlar, önerilen CNN-LSTM modelinin doğruluk, kesinlik, yanlış pozitif oranı ve tespit maliyet açısından diğer modellerden daha iyi performans gösterdiğini ve IIoT uygulamaları için siber güvenlik saldırı tespitinde etkinliğini ortaya koyduğunu göstermiştir.

SDN tabanlı endüstriyel siber-fiziksel sistemler (Cyber-Physical System, CPS) için hafif bir saldırı tespit modeli [13] çalışmasında önerilmiştir. Çalışma, SDN tabanlı endüstriyel CPS ortamındaki kapsamlı bağlantı ve veri alışverişinin güvenlik açıklarını, özellikle de siber saldırılar için cazip bir hedef olan SDN denetleyicisini tartışmaktadır. Bu çalışma, düşük gecikme süreleri gerektiren bir IIoT ağı içerisinde, derin öğrenme tabanlı ve yüksek derecede karmaşık saldırı tespit yöntemlerini yönetmenin karşılaştığı zorlukları ele almaktadır. Söz konusu zorlukları aşmak amacıyla, bu çalışma SDN (Yazılım Tabanlı Ağ) teknolojisini kullanarak endüstriyel bir CPS (Siber Fiziksel Sistem) ortamı için daha hafif bir saldırı tespit modeli önermektedir. Önerilen model, SDN ile ilgili açık erişimli bir siber güvenlik veri seti kullanılarak test edilmiş ve mevcut yöntemlere kıyasla daha

yüksek doğruluk, kesinlik, duyarlılık ve daha düşük zaman maliyeti gibi performans avantajları sunmuştur.

Siber dünyanın hızla gelişen tehditleri, siber saldırılara karşı etkili bir izleme ve savunma sistemi olan IDS, günümüzde büyük önem kazanmıştır. IDS'nin karşılaştığı bazı temel zorluklar, yanlış alarmlar, düşük doğruluk ve hassasiyet, yüksek boyutlu ve karmaşık veri yapıları ve uzun hesaplama süreleri olarak öne çıkmaktadır. [14] numaralı çalışma, IDS (Intrusion Detection System) tespit motorunun işlemesi gereken veri miktarını düşürerek bu zorluklarla başa çıkabilmenin yollarını araştırmaktadır. Bu bağlamda, veri setinden en etkili özellikleri seçmek amacıyla korelasyon tabanlı özellik seçimi yöntemi (Correlation-based Feature Selection, CFS) kullanılmasını tavsiye etmektedir. Önerilen CFS tabanlı IDS sistemi, CIC-IDS2018 veri seti üzerinde yapılan testler sonucunda yüksek doğruluk, duyarlılık, kesinlik, F1-skoru, gerçek pozitif oranı ve gerçek negatif oranı gibi performans metriklerinde üstün sonuçlar elde etmiştir. Deneysel sonuçlar, önerilen CFS tabanlı IDS'nin %99.9995 doğruluk, %100 recall, %99.9985 özgüllük, %99.9992 kesinlik, %99,9996 F1-skoru, %99,9992 gerçek pozitif oranı ve %100 gerçek negatif oranı ile optimum performansa ulaştığını göstermektedir.

Saldırı tespit sistemlerinin performansını arttırmak için [15] çalışmasında SPIDER adı verilen yeni bir ağ anomali tespit modeli önerilmiştir. Çalışma, Tekrarlayan Sinir Ağlarının (Recurrent Neural Network, RNN) dört güncellenmiş versiyonunu birleştirmektedir: Bi-LSTM, LSTMM, Bi-GRU ve GRU. SPIDER modeli ayrıca verilerin boyutlarını azaltmak için Temel Bileşen Analizini (Principal Component Analysis, PCA) kullanmakta ve bu sayede yüksek boyutluluk sorununun üstesinden gelmeye yardımcı olmaktadır. Çalışmanın ana hedefi, ağ güvenliği kapsamında yetkisiz erişimleri saptayabilen IDS sistemlerinin doğruluk ve verimlilik seviyelerini yükseltmektir. SPIDER adı verilen önerilen model, yetkisiz giriş saptama konusunda ne derece sağlam ve etkili olduğunu göstermek amacıyla NSL-KDD ve UNSW-NB15 adlı veri setleri üzerinde test edilmiştir. Elde edilen bulgular, SPIDER modelinin yetkisiz erişimleri belirleme konusunda mevcut sistemlerden daha iyi bir performans gösterdiğini ve bu alanda daha etkin çözümler üretebilme yönünde önemli bir potansiyel taşıdığını göstermektedir.

IIoT ortamları için Derin Rastgele Sinir Ağı (Deep Random Neural Network, DRaNN) tabanlı saldırı tespit modeli [16] çalışmasında sunulmuştur. IIoT teknolojisi, akıllı sensörleri ve cihazları internet ile entegre ederek etkili siber güvenlik mekanizmaları

geliştirmek için akıllı işleme teknikleri gerektiren büyük miktarda veri üretmektedir. Bu çalışmada, bir saldırı tespit sistemi geliştirmek için derin öğrenme yöntemleri kullanılmıştır. Önerilen DraNN, yapay sinir ağının daha iyi genelleme yeteneklerine ve oldukça dağıtık bir yapıya sahip çeşididir. Daha yüksek saldırı tespit doğruluğu elde etmek için önerien DraNN, optimum hiperparametreleri dahil etmek için sıralı kuadratik programlama (Sequential Quadratic Programming, SQP) ile hibrit parçacık sürüsü optimizasyonu (Particle Swarm Optimization, PSO) kullanılarak eğitilmiştir. Önerilen planın etkinliği hem ikili hem de çok sınıflı konfigürasyonlarda üç yeni IIoT veri seti üzerinde kapsamlı deneyler yapılarak analiz edilmiştir. Sonuçlar, önerilen DRaNN tabanlı modelin IIoT ortamlarında saldırı tespiti için diğer yöntemlerden daha iyi performans gösterdiğini ortaya koymuştur.

Anomali tabanlı saldırı tespit sistemlerinin doğruluğunu ve tespit oranını iyileştiren ve yanlış pozitif oranlarını en aza indiren hibrit bir yöntem [17] çalışmasında önerilmiştir. Bu teknik, özelliklerin seçimi için Yapay Arı Kolonisi (ABC) algoritmasını kullanırken, özelliklerin değerlendirilmesi ve sınıflandırılmasında AdaBoost algoritmasından yararlanmaktadır. Önerilen yöntem NSL-KDD ve ISCXIDS2012 veri setleri üzerinde test edilmiş ve farklı saldırı senaryolarında doğruluk ve tespit oranı açısından diğer IDS yöntemlerinden önemli ölçüde daha iyi performans gösterdiği belirtilmiştir. Çalışma, önerilen hibrit yöntemin A-NIDS'in performansını etkili bir şekilde arttırabileceği ve yüksek hacimli ağ verileriyle ilişkili yanlış alarm ve tespit doğruluğu sorununu azaltabileceği sonucuna varmaktadır.

Özellik seçimi için Parçacık Sürü Optimizasyonu (PSO) tabanlı bir teknik ve sınıflandırma için ise SVM kullanan [18] çalışmasında, IoT için bir saldırı tespit sistemi geliştirilmiştir. Bu çalışmada, Parçacık Sürü Optimizasyonu (PSO) metodu Light Gradient Boosting Machine (LightGBM) algoritmasını temel alarak kullanılmıştır. Araştırma ekibi, geliştirdikleri sistemin performansını ölçmek için UNSW-NB15 veri setini kullanmışlar ve modelin doğruluğunu (accuracy) ile yanlış pozitif oranını (false alarm rate, FAR) temel alarak değerlendirme yapmışlardır. Yapılan testler sonucunda, Parçacık Sürü Optimizasyonu (PSO) yöntemiyle iyileştirilmiş LightGBM modelinin %86.68 gibi oldukça yüksek bir doğruluk oranı ve %10.62 gibi bir yanlış alarm oranı ile sonuçlandığı gözlemlenmiştir.

Endüstriyel büyük veri sistemleri için bir değişken LSTM (VLSTM) saldırı tespit sistemi [19] çalışmasında uygulanmıştır. VLSTM öznitelik seçimi ve özniteliklerin yeniden

oluşturulmuş görüntülenmesine dayalı bir teknik ile birlikte uygulanmıştır. Araştırmacılar bu çalışmada, yüksek boyutlu veri setlerinden daha anlamlı ve düşük boyutlu özellik çıkarabilmek amacıyla Auto-Encoder tipinde bir yapay sinir ağı metodunu tercih etmişlerdir. Modelin etkinliğini değerlendirmek üzere UNSW-NB15 veri seti kullanılmıştır. Performansın değerlendirilmesi aşamasında yanlış alarm oranı, eğrinin altındaki alan, kesinlik, duyarlılık ve F1-Skor gibi ölçütler dikkate alınmıştır. Yapılan deneyler sonucunda elde edilen veriler, kullanılan modelin %89.5 oranında bir duyarlılık ve %90.7 F1-Skoru başarısına ulaştığını göstermektedir. Bu sonuçlar, bazı mevcut yöntemlere kıyasla oldukça iyi olmakla birlikte, araştırmacılar veri setinin dengesiz yapısı nedeniyle daha fazla analiz ve deney yapılmasının gerekliliğini vurgulamışlardır.

Özellik seçimi için uyarlanabilir bir temel bileşen analizi ve sınıflandırma işlemleri için artımsal bir aşırı öğrenme makinesi (incremental extreme learning machine, IELM) algoritması kullanan bir makine öğrenmesi temelli bir saldırı tespit sistemi [20] çalışmasında gelişirilmiştir. Araştırmacılar, bu çerçevenin verimliliğini değerlendirmek amacıyla hem NSL-KDD hem de UNSW-NB15 veri setlerini testlerinde kullanmışlardır. Her iki veri seti için de çok sınıflı sınıflandırma yöntemi tercih edilmiştir. Bu çalışmada kullanılan ana performans metriği doğruluktur (accuracy). NSL-KDD veri seti kullanıldığında %81.22 doğruluk elde edilmiştir. UNSW-NB15 veri seti için ise %70.51 doğruluk elde edilmiştir. Araştırmacılar elde edilen sonuçların mevcut sistemler tarafından elde edilenlerden daha üstün olduğu iddia edilse de endüstriyel kontrol sistemlerine uyarlamak için daha fazla araştırma yapılması gerektiğini belirtmişlerdir.

Yeni saldırı tiplerini hızlı ve etkili bir şekilde tespit edebilen esnek ve dayanıklı bir derin sinir ağı (DNN) tabanlı saldırı tespit sistemi [3] çalışmasında tasarlanmıştır. Araştırmacılar, sunulan sistemin performansını test etmek için KDD-Cup99, UNSW-NB15, NSL-KDD, Kyoto, WSN-DS ve CICIDS 2017 gibi çeşitli veri setlerini kullanmışlardır. Bu veri setleri, sistemin farklı türdeki saldırıları ne kadar etkin bir şekilde tespit edebileceğini değerlendirmek için seçilmiştir. UNSW-NB15'e odaklanan testler, DNN'in ikili modelleme işlemi için %76.1 doğruluk, %95.1 precision, %96.3 recall ve %79.7 F1-Skoru elde ettiğini göstermiştir. Buna karşılık DNN, çok sınıflı modelleme prosedürü için %65.1 accuracy, %75.6 F1-Skoru, %59.7 precision ve %65.1 recall elde etmiştir.

IoT ağları için bir saldırı tespit sistemi sunan [21] çalışmasında yapay sinir ağları kullanılmıştır. Bu sistem, IoT ağlarındaki önemli bir problem olan güvenlik sorunlarına

çözüm bulmak amacıyla geliştirilmiştir. IoT cihazlarının genellikle güvenlik için üst düzey hesaplama yapma kapasitesinden yoksun olduğu gerçeği göz önüne alındığında, araştırmacılar ilk savunma hattı olarak makine öğrenmesi tabanlı bir saldırı tespit sistemi kullanmayı amaçlamışlardır. UNSW-NB15 veri seti, önerilen metodolojinin etkinliğini ölçmek amacıyla kullanılmıştır. Deneyler neticesinde elde edilen sonuçlar, Yapay Sinir Ağı tabanlı İzinsiz Giriş Tespit Sistemi (ANN-IDS) kullanılarak yapılan ikili sınıflandırma işlemi sırasında %84.00'lük bir kesinlik oranına ulaşıldığını belirtmektedir. Fakat araştırmacılar, Yapay Sinir Ağı'nın hiperparametre ayarlarının nasıl yapıldığı konusunda yeterince bilgi vermemişlerdir. Ayrıca, bu çalışmada herhangi bir özellik seçim metodu kullanılmamıştır, bu da sonuçların doğruluk ve genel geçerliliği açısından ele alınması gereken önemli bir noktadır.

UNSW-NB15 ve KDD99 veri setlerinin karmaşıklıklarını karşılaştırmak üzere bir analiz çalışması gerçekleştirilmiştir [22]. Araştırmacılar bu karşılaştırmayı yapabilmek için çeşitli yöntemlere başvurmuşlar, bu yöntemler arasında beklenti-maksimizasyon (EM) kümeleme algoritması ve Yapay Sinir Ağları (YSA) bulunmaktadır. Çalışmada modellerin etkinliğini değerlendirebilmek için yanlış alarm oranı ve doğruluk oranı gibi kriterlerden faydalanılmıştır. KDD99 veri seti üzerinde yapılan testlerde EM kümeleme yöntemi %78.06 doğruluk ve %23.79 yanlış alarm oranına ulaşırken, UNSW-NB15 veri seti ile yapılan testlerde %78.47 doğruluk ve yine %23.79 yanlış alarm oranı gözlenmiştir. YSA yöntemi ise UNSW-NB15 veri seti üzerinde %81.34 doğruluk ve %21.13 yanlış alarm oranı ile daha iyi bir performans sergilemiştir. Bu bulgular, UNSW-NB15 veri setinin KDD99'a kıyasla daha karmaşık olduğunu göstermiştir.

YSA kullanılan bir saldırı tespit sistemi [23] çalışmasında önerilmiştir. Bu sistem, modern iletişim sistemlerinin (5G ağları, IIoT ağlar vb.) güvenliğini sağlamaya yöneliktir. Bu araştırmada sunulan YSA-IDS iki aşamada tasarlanmıştır. İlk aşamada, veriden özellik çıkarmak için istatistiksel analiz teknikleri kullanılmıştır. Bu işlemin ardından ikinci aşama olarak sınıflandırma süreci gerçekleştirilmiştir. Araştırmacılar, UNSW-NB15 veri seti üzerinde ikili sınıflandırma çalışmaları yapmışlardır. Yapay Sinir Ağı modellerinin performanslarını değerlendirmek için seçilen ana ölçüt doğruluk oranı olmuştur. Deneysel sonuçlar, test edilen modeller arasında %83.9'luk doğruluk oranı ile en yüksek başarıyı gösteren modelin olduğunu belirtmektedir.

J48 karar ağacı tabanlı sınıflandırıcı ve destek vektör makineleri (SVM) kullanılarak geliştirilen bir saldırı tespit çerçevesi [24] çalışmasında sunulmuştur. Özellik seçimi için

genetik algoritma (GA), ateş böceği algoritması (FFA) ve gri kurt optimizasyonu (GWO) gibi çeşitli yöntemlerden yararlanılmıştır. Araştırmacılar, modelin performansını ölçmek için UNSW-NB15 veri setini kullanmayı tercih etmişlerdir. GA-J48, GWO-J48 ve FFA-J48 modelleri sırasıyla %86.874, %85.676 ve %86.037 doğruluk oranlarına ulaşmıştır. Aynı zamanda, GA-SVM, GWO-SVM ve FFA-SVM modelleri için elde edilen doğruluk oranları ise sırasıyla %86.387, %84.485 ve %85.429 olarak rapor edilmiştir.

Yeni bir özellik seçim yöntemi olan Tabu Arama-Random Forest (TS-RF) [25] çalışmasında tanıtılmıştır. Bu yöntem, Tabu Arama algoritmasının özellik seçimini yaptığı ve öğrenme sürecinde Random Forest tekniğinin kullanıldığı bir özellik çıkarma metodudur. Araştırmacılar, modelin performansını test etmek için UNSW-NB15 veri setini kullanmışlardır. Ana performans kriterleri, doğruluk ve yanlış pozitif oranı (FPR) olarak belirlenmiştir. Elde edilen sonuçlara göre, TS-RF yöntemi Random Forest sınıflandırıcısı ile %83.12 doğruluk ve %3.7 FPR sağlamıştır. Bununla birlikte, çalışmada UNSW-NB15 veri setindeki sınıf dengesizliğinin göz ardı edildiği belirtilmiştir.

IDS için iki aşamalı (Two-Stage, TS) bir model [26] çalışmasında önerilmiştir. Bu metodoloji, iki aşamalı bir yaklaşımı benimsemiştir; birinci aşama azınlıkta olan saldırı sınıflarını belirlemek için, ikinci aşama ise çoğunlukta olan saldırı sınıflarını tespit etmek için kullanılmıştır. Bu çalışmada tercih edilen makine öğrenimi algoritması Random Forest olup, özellik çıkarma işlemi için bilgi kazancı yöntemi kullanılmıştır. Bilgi kazancı tabanlı iki aşamalı sınıflandırma modeli (IG-TS IDS), performans değerlendirmesi için UNSW-NB15 veri seti ile test edilmiştir. Performans ölçütleri olarak doğruluk oranı ve yanlış alarm oranı ele alınmıştır. Araştırmacılar, deneylerinde temel olarak ikili sınıflandırma şemasını kullanmayı tercih etmişlerdir. Elde edilen deneysel sonuçlar, IG-TS modelinin %85.78'lik bir doğruluk oranına ve %15.64'lük bir yanlış alarm oranına ulaştığını göstermektedir.

Genetik Algoritma (GA) ve Lojistik Regresyon'u özellik seçimi için kullanan bir makine öğrenimi tabanlı saldırı tespit sistemi [27] çalışmasında önerilmiştir. Bu sistem, ikili sınıflandırma işlemleri için C4.5 algoritmasına dayanan ağaç tabanlı bir sınıflandırıcı kullanmaktadır. Araştırmada, sistemin performansını değerlendirmek üzere UNSW-NB15 veri seti tercih edilmiş ve performans analizi için çeşitli kriterler dikkate alınmıştır. Ancak, temel performans metriği olarak doğruluk oranı kullanılmıştır. Yürütülen testler sonucunda, GA ve Lojistik Regresyon ile desteklenen Karar Ağacı (GA-LR-DT) yönteminin %81.42 doğruluk oranı elde ettiği belirlenmiştir.

Birçok makine öğrenmesi yöntemiyle birlikte XGBoost tabanlı özellik çıkarma yöntemini kullanan bir saldırı tespit sistemi [28] çalışmasında önerilmiştir. Bu araştırmada, UNSW-NB15'teki öznitelik sayısını azaltmak için topluluk ağacı tabanlı bir algoritma olan XGBoost kullanılmıştır. Bu çalışmada kullanılan sınıflandırıcılardan biri de LR yöntemidir. Deneysel çalışmaların sonucunda, XGBoost ile entegre edilmiş Lojistik Regresyon (XGBoost-LR) modelinin, ikili ve çok sınıflı sınıflandırma senaryolarında sırasıyla %75.51 ve %72.52 oranlarında doğruluk sağladığı tespit edilmiştir. Araştırmacılar, UNSW-NB15 veri setinin içerisinde bulunan sınıf dengesizliği sorununa çözüm bulmak amacıyla, aşırı örnekleme tekniklerinin (oversampling) kullanılmasını tavsiye etmişlerdir. Bu teknikler, az temsil edilen sınıflardan daha fazla örnek üreterek veri setinin dengelenmesine yardımcı olabilir ve böylece sınıflandırma modelinin performansını artırabilir.

1.4 Tezin Yöntemi

İmza tabanlı ve kural tabanlı sistemler gibi geleneksel saldırı tespit yöntemleri, hızla gelişen siber tehdit ortamına ayak uydurmakta çoğu zaman zorlanmaktadır. Bu yöntemler önceden tanımlanmış kalıplara ve kurallara dayandığından yeni ve sofistike saldırı tekniklerine karşı savunmasızdır. Önemli sayıda yanlış pozitif üretebilir veya yeni ve gelişmekte olan tehditleri tamamen gözden kaçırabilirler.

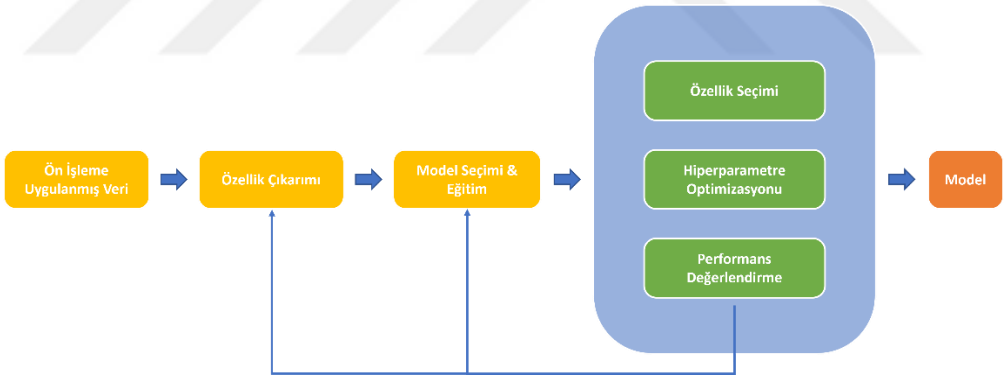
Bu sorunun ele alınması ve saldırı tespit sistemlerinin hassasiyetinin ve etkinliğinin arttırılması çeşitli faydalar sağlayabilir. Bir kuruluşun genel güvenlik duruşunu iyileştirerek başarılı saldırı ve potansiyel zarar tehlikesini azaltabilir. Ayrıca, tespit edilen ihlallere karşı daha hızlı ve proaktif tepkiler verilmesini sağlayarak firmaların bir ihlalin etkisini azaltmasına ve kapsamını kısıtlamasına olanak tanıyabilir. Kuruluşlar verimliliği arttırmak, daha az kaynak kullanmak ve maliyeti düşürmek için eğitim ve test sürelerini optimize edebilir.

Saldırı tespit modellerinin geliştirilmesi ve dağıtımında verimliliği arttırılması AutoML kullanımının bir avantajıdır. Bu çerçeve, veri ön işleme, model seçimi, hiperparametre ve özellik mühendisliğini süreçlerini otomatikleştirir. Otomatikleştirilmiş süreçler sayesinde, kullanılmak istenen veri setlerinin eksik verileri tamamlanır, hatalı verileri elenir, veriler ölçeklenir ve modellerin oluşturulması ve dağıtımı daha hızlı gerçekleşir. Çünkü insan uzmanların çok fazla manuel çaba göstermesi gerekmez. Bu sayede

kuruluşlar yeni tehditlere daha hızlı tepki verebilir ve ağ izinsiz girişlerinin hızlı bir şekilde tanımlanmasını ve kontrol altına alınmasını sağlayabilir.

Saldırı tespitinde geliştirilmiş doğruluk AutoML yaklaşımları kullanılarak elde edilir. AutoML, sofistike algoritmalar ve arama teknikleri kullanara eldeki görev için en uygun olanı belirlemek amacıyla çeşitli modelleri ve yapılandırmaları analiz eder. Geleneksel yaklaşımlar en iyi çözümleri gözden kaçırabilir, ancak AutoML bunları belirleyerek daha hassas saldırı tespit modellerine yol açabilir. Doğruluktaki bu artış, yanlış pozitiflerin ve yanlış negatiflerin azalmasına katkıda bulunarak saldırı tespit sisteminin genel verimliliğini arttırır.

AutoML, büyük ve çok yönlü bir optimizasyon problemi olarak açıklanabilir. İnsan müdahalesi olmadan belirli bir hesaplama sürecine göre bir veri kümesi için tahminler üretebilen bir çözüm uzayı olarak ifade edilebilir. Automl, bir veri kümesi üzerinde optimum performans elde etmek için makine öğrenmesi süreçlerinin otomatik olarak oluşturulmasını sağlar. Veri ön işleme, model seçimi ve hiperparametre optimizasyonu gibi zor ve zaman alıcı görevleri otomatikleştirebilir. AutoML sürecinin adımları en genel haliyle Şekil 1.1'de gösterilmektedir.



Şekil 1.1 AutoML adımları

Hiperparametre optimizasyonu makine öğrenmesi sürecinin önemli bir problemidir [29]. Çünkü hiperparametre optimizasyonu, farklı veri kümelerinde en iyi şekilde çalışabilecek hiperparametre konfigürasyonlarını keşfetmek ve yaygın makine öğrenmesi kütüphanelerinde varsayılan olarak sunulanların geliştirilmesine katkıda bulunmak içiin kullanılır [30] [31]. Ağ saldırı tespitinin kritik doğası ve mevcut yaklaşımların sınırlamaları göz önüne alındığında bu çalışma, önerilen çerçeveyi kullanarak daha yüksek doğruluk seviyelerine ulaşmayı ve eğitim/test sürelerini azaltmayı amaçlamaktadır.

1.5 Teze Genel Bakış

Bu çalışmada IIoT sistemler için otomatik makine öğrenmesi kullanan bir IDS geliştirilmesi amaçlanmaktadır. Makine öğrenmesi tabanlı IDS'ler genellikle en güncel saldırı tespit veri kümeleri kullanılarak eğitilir. Bununla birlikte, modern veri kümelerinin çoğu hem özellik alanı boyutunda hem de ağ izlerinin sayısı bakımından büyüktür. Bir veri setinde fazla miktarda öznitelik bulunması, makine öğrenimi algoritmalarının eğitim sürecini olumsuz yönde etkileyebilir. Genellikle, öznitelik sayısının artmasıyla makine öğrenimi yöntemlerinin performansında bir düşüş gözlemlenir. Başka bir deyişle bir veri setinde öznitelik sayısı arttıkça öğrenme sürecini gerçekleştirmek daha zordur [32]. Bu nedenle, öznitelik vektörünün boyutunun gerekli özniteliklerin optimal sayısına indirgenmesini garanti etmek için bir öznitelik seçimi veya çıkarma işlemi gerçekleştirmek oldukça önemlidir [33].

Çalışmanın birinci bölümü teze giriş bölümüdür, bu bölümde endüstriyel sistemlerde saldırı tespitine duyulan ihtiyaç vurgulanarak çalışmaya zemin hazırlanmaktadır. Araştırma bağlamı ve motivasyonu tanıtılmakta, endüstriyel ağların kendine özgü zorlukları ve otomatik makine öğrenmesinin bir çözüm olarak potansiyeli vurgulanmaktadır.

İkinci bölümde endüstriyel ortamlarda saldırı tespitinin kendine özgü incelikleri ve zorlukları ele alınmaktadır. Endüstriyel ağların özelliklerini ve siber tehditlere karşı kırılganlıklarını araştırmakta ve endüstriyel varlıkların korunmasının önemini vurgulamaktadır.

Üçüncü bölüm saldırı tespiti kapsamında otomatik makine öğrenmesinden yararlanmak için benimsenen sistematik ve metodolojik yaklaşımı ortaya koymaktadır. Araştırma metodolojisinin ayrıntılarını ortaya koyan temel kavramlar ve teknikler incelenmiştir. Bu bölüm veri ön işleme, model seçimi ve değerlendirme tekniklerini kapsayan uygulanan sistematik stratejilerei detaylandırmaktadır.

Çalışmanın dördüncü kısmı, otomatik makine öğrenmesi çerçevesinin tasarım ve geliştirilme süreçlerini kapsamaktadır. Bu kısım, teorik kavramların pratik uygulamalara nasıl dönüştürüleceğini gösterdiği için araştırmanın temel noktalarını yansıtmaktadır. Ayrıca, otomatik makine öğrenimi çerçevesinin yapısı, parçaları ve işlevsellikleri bu bölümde detaylandırılmaktadır. Çerçevenin oluşturulmasında uygulanan yenilikçi teknoloji seçimleri ve metodolojileri bu bölümde ortaya koyulmuştur. Çerçevenin

tasarımı ve geliştirilmesi, araştırmanın teknik becerisini ortaya koymakta ve endüstriyel sistemlerde saldırı tespitinin gerekliliklerini ele almada önemli bir araç olarak sunulmaktadır.

Beşinci bölüm otomatik makine öğrenmesi sisteminin test edildiği bölümdür. Kullanılan veri setleri, değerlendirilen performans ölçütleri ve otomatik makine öğrenmesi çerçevesinin gerçek dünya endüstriyel ağ verilerine uygulanmasının sonuçları bu bölümde sunulmaktadır. Bölüm, araştırmanın pratikliğini ve endüstriyel sistemlerde siber güvenliğin arttırılması üzerinde yaratabileceği somut etkiyi göstermektedir.

Altıncı bölümde çalışmanın içeriği ve önemi açıklanmaktadır. Sonuç bölümünün temel noktaları ve amacı vurgulanmakta, çalışmanın bulgularını özetleme, daha geniş çıkarımlarını tartışma ve endüstriyel siber güvenlik alanında gelecekteeki potansiyel yönelimlere işaret etme rolü vurgulanmaktadır.

2
ENDÜSTRİYEL SİSTEMLERDE SALDIRI TESPİTİ

Nesnelerin İnterneti'nin (IoT) yaygınlaşmasıyla dijitalleşme endüstriyel sahada da kendini göstermektedir. Kaliteyi iyileştirmek için, makine ile makine arası iletişim (M2M) ve yapay zeka (AI) destekli veri analizi üzerine kurulu endüstriyel IoT (IIoT), farklı üreticilerin cihazları arasında verimli, üretken ve güvenli bir iletişim kurulmasını sağlamaktadır. Pek çok IIoT uygulaması, eski sistemlerin yeni IoT teknolojileriyle entegre edildiği kahverengi alan yaklaşımını benimsemiştir. Bu sistemlerin IoT teknolojileriyle uyumlu çalışmasını ve entegrasyonunu sağlamak amacıyla, eski operasyonel teknoloji ile yeni IoT sistemleri arasında köprü vazifesi gören uç ağ geçitleri gibi yeni nesil cihazlar devreye alınmıştır. Ancak bu gelişim, operasyonel teknoloji ile IoT'nin sıkı entegrasyonu siber güvenlik risklerini de beraberinde getirmekte, yeni giriş noktaları bu önemli cihaz ve sistemlerin hedef alındığı karmaşık siber saldırılar için potansiyel tehditler oluşturmaktadır [34].

IIoT ağlarında güvenlik en büyük zorluklardan biridir. IIoT cihazlarının geliştirilmesine yönelik genel bir uzlaşı olmadığı için, IoT tehditlerini engellemek güçleşmektedir. Çeşitli iletişim protokolleri ise, bir IIoT altyapısını kullanırken ekstra bir karmaşıklık unsuru olarak ortaya çıkmaktadı [35]. Çok çeşitli IoT protokolleriyle ilgili zorluklar, IIoT ağları için güvenilir ve tek tip bir siber güvenlik yaklaşımı sağlamayı zorlaştırmaktadır. Saldırganlar, IIoT protokollerinde bulunan birçok güvenlik açığı sayesinde IIoT ağlarına saldırabilir ve güvenliğini tehlikeye sokabilir. Cihaz arızaları, veri çalınması, servis kesintileri ve Man-in-the-Middle (Ortadaki Adam) saldırıları, bu tür tehditlere örnektir. [36]. Kritik altyapılara yönelik tehditlerin artışı, bu önemli sistemlerin korunmasını sağlamak amacıyla önceden önlem alabilen güvenlik teknolojilerinin geliştirilmesini zorunlu kılmaktadır. Ağ güvenliği çözümleri arasında, saldırı tespit sistemi (IDS) etkili bir reaktif araç olarak geniş bir kabul görmüştür. IIoT ekosistemlerinde kritik bir rol oynayan IDS, ağ üzerinden iletilen verilerin doğruluğunu, emniyetini ve gizliliğini temin eder. Bu sistemler, IIoT ağları için tehlike arz edebilecek herhangi bir saldırıyı veya zararlı eylemi engelleme, saptama, karşılık verme ve bu olayları kayıt altına alma kapasitesine sahiptir [37].

Ağ izinsiz giriş tespiti, ağ üzerindeki çeşitli faaliyetleri analiz ederek güvenliğin sağlanmasını hedefler. Bu süreç, ağdaki farklı veri noktalarını inceleyerek gerçekleştirilir. Güvenlik sistemleri genellikle belirli bir eşiğe, önceden tanımlanmış imzalara, sezgisel yaklaşımlara veya istatistiksel yöntemlere dayalıdır. Bu teknikler, daha önce tanımlanmış ve bilinen tehditlerin tespitinde etkilidir fakat yeni ve önceden tanımlanmamış tehditleri saptamak konusunda yetersiz kalabilirler. Bu sistemler sıklıkla uzman bilgisi gerektirir ve düzenli güncellemelere ihtiyaç duyarlar. İmza tabanlı sistemler, yeni siber tehditleri algılayamama ve manuel olarak güncellenmesi gereken imza veritabanına olan bağımlılık nedeniyle sınırlı bir etkinlik gösterirler.

2.1 Geleneksel Saldırı Tespit Sistemleri

Çoğu mevcut girişim tespit sistemi (IDS), kablosuz sensör ağlarına veya standart bilişim teknolojisi altyapılarına hizmet verecek biçimde geliştirilmiştir; fakat bu sistemler, yeni nesil IoT cihazlarının gereksinimlerini karşılayacak yapıda değildir. Kablosuz sensör ağları (WSN), IoT'nin atası olarak görülse de aralarında önemli yapısal farklılıklar bulunmaktadır, bu sebepten ötürü bu IDS çözümlerinin IoT ekosistemlerinde etkin bir kullanımı mümkün olmamaktadır [38]. Ayrıca, IoT ortamının büyüklüğü, çeşitliliği, kullanım senaryoları ve cihaz sınırlamaları gibi özellikleri göz önünde bulundurmayan geleneksel bilişim teknolojisi sistemleri için tasarlanmış IDS'ler de bu yeni ortama uyum sağlayamaz. Geleneksel bilişim güvenlik ekosistemi, IoT'nin ihtiyaçlarını karşılayamayan sabit ağ savunma araçları (firewall vb.) ve merkezi bilgisayar savunmaları (antivirüs vb.) üzerine kurulmuştur [39]. IoT cihazlarının ve üreticilerinin çeşitliliği göz önüne alındığında, saldırı imzalarını belirlemeye yönelik geleneksel metotların yetersiz kalabileceği veya ölçeklendirilemeyeceği açıktır [39]. SNORT ve Bro gibi popüler geleneksel IDS'ler, sadece geleneksel IP ağlarında etkin olup, uyum sağlamakta güçlük çeker ve yalnızca tek bir platforma ya da protokole hizmet ederler. Bu sebeple, bu çeşitlilik ve ölçekteki heterojen ekosistemler için imzaları uyarlayıp öğrenme yeteneğine sahip yeni güvenlik mekanizmalarının geliştirilmesine ihtiyaç duyulmaktadır.

IDS'ler, hangi ağ ortamı için tasarlandıklarına bakılmaksızın (IoT, standart bilişim teknolojileri, WSN vb.) ortak bir prensip çerçevesinde oluşturulmuşlardır [40]. Bir veri toplama ünitesi, ağdaki verileri toplar; bir analiz birimi, bu toplanan veriler üzerinde işlem yaparak saldırıları saptamaya çalışır ve son etapta bir saldırı bildirim sistemi, ağ yöneticilerine durum hakkında bilgi verir. Farklı IDS sistemleri arasındaki asıl ayrım,

kullanılan uygulama stratejileri ve bu stratejilere bağlı tekniklerde ortaya çıkar. Bu farklı strateji ve tekniklerden bazıları şunlardır: (i) tespit yöntemi olarak imza tabanlı veya anomaliye dayalı yaklaşımlar ve (ii) yapı olarak merkezileştirilmiş, dağıtılmış veya hibrit sistemler [41].

2.1.1 İmza Tabanlı Saldırı Tespit Sistemleri

İmza tabanlı saldırı tespit sistemleri, daha önceden bilinen saldırıları belirlemek amacıyla model eşleştirme metodlarına güvenir ve bu sistemler genellikle bilgi tabanlı ya da kötüye kullanım tespiti sistemleri olarak adlandırılır. Bu sistemler, geçmişte gerçekleşen izinsiz erişim olaylarını tanımlamak için eşleştirme tekniklerini kullanır. Yani, bir ağa yapılan izinsiz bir girişin örüntüsü, imza veritabanında daha önceden kaydedilmiş bir olayın örüntüsüyle örtüşürse, bir uyarı mekanizması devreye girer. Bu yöntem, bilinen zararlı yazılım aktivitelerini veya eylemleri saptamak için ana bilgisayar kayıtlarını inceleyerek çalışır ve literatürde bilgiye dayalı tespit veya kötüye kullanım algılama olarak da isimlendirilir [42].

Bu tür bir güvenlik sisteminin temel prensibi, izinsiz girişlerin örüntülerini barındıran bir veritabanı oluşturmak, gerçekleşen faaliyetleri bu veritabanındaki örüntülerle kıyaslamak ve eşleşen bir durum söz konusu olduğunda uyarı sinyali yaymaktır. İmza tabanlı sistem, bilinen izinsiz giriş vakalarını saptamada oldukça yüksek bir doğruluk oranı sunar [43]. Fakat, yeni bir saldırı türünün örüntüsü tanımlanıp veritabanına eklenene kadar bu saldırıları algılamada zorlanır, bu da sıfır gün saldırılarına karşı koyabilme yeteneğini sınırlar. SNORT ve NetSTAT gibi yaygın güvenlik araçları bu sistemden faydalanmaktadır.

Geleneksel imza tabanlı sistemlerin metodolojisi, ağ üzerinden geçen paketleri detaylı bir şekilde inceleyip, bu paketlerin içerdiği bilgileri önceden oluşturulmuş bir imza veritabanı ile kıyaslamaktır. Ancak bu yöntemler, birden fazla paket boyunca gerçekleşen ve yayılan saldırıları tespit etme konusunda yetersiz kalabilir. Günümüzdeki kötü amaçlı yazılımlar daha sofistike hale geldiği için, imza bilgisini çıkarmak için çoklu paketler üzerinden analiz yapılması gerekebilir. Bu durum, bir IDS sisteminin geçmişteki paketlerin içeriklerini hatırlamasını ve bunları sürekli olarak karşılaştırmasını zorunlu kılar. Bu durumlara karşı imza veritabanı oluşturabilmek için, durum makinelerini [44], biçimsel dil dizisi kalıplarını veya anlamsal koşulları [45] içeren bir dizi yöntem kullanılmıştır. Sıfır gün saldırılarının artan oranı bu tür saldırılar için önceden imza

bulunmadığından imza tabanlı sistemin tekniklerini giderek daha az etkili hale getirdi. Kötü amaçlı yazılımın polimorfik varyantları ve artan sayıda hedefli saldırı, bu geleneksel paradigmanın yeterliliğini daha zayıflatmaktadır. Bu sorun için olası çözümlerden biri, anormal olandan ziyade kabul edilebilir bir davranışın profilini çıkararak işleyen anomali tabanlı sistemlerdir.

2.1.2 Anomali Tabanlı Saldırı Tespit Sistemleri

Anomali tabanlı tespit sistemleri, imza tabanlı sistemlerin sınırlılıklarını aşabilme yetenekleri sayesinde pek çok araştırmacının odak noktası haline gelmiştir. Bu sistemler, bir bilgisayarın normal davranış profilini oluşturmak için makine öğrenmesi, istatistiksel, veya bilgi tabanlı teknikler kullanır. Normal profil ile gözlenen davranış arasında belirgin farklılıklar tespit edildiğinde, bu durum bir güvenlik ihlali olabilecek bir anomali olarak değerlendirilir. Bu tekniklerin temel prensibi, zararlı aktivitenin normal kullanıcı eylemlerinden ayırt edilebilir olmasıdır. Normalden sapma gösteren davranışlar, izinsiz giriş olarak sınıflandırılır. Anomali tabanlı sistemin kurulumu iki temel evreden meydana gelir: eğitim ve test evresi. Eğitim evresinde, sistem normal davranış örneklerini öğrenmek üzere standart ağ trafiğini inceler; test evresindeyse, sistem daha önce karşılaşmadığı tehditlere karşı ne kadar iyi genelleme yapabileceği test edilir. Kullanılan eğitim metodolojisine göre anomali tabanlı sistemler, istatistiksel, bilgi tabanlı, veya makine öğrenmesi gibi farklı alt kategorilere ayrılabilir [46].

Anomali tabanlı tespit sistemlerinin temel avantajı, imza tabanlı veritabanlarına ihtiyaç duymadan kullanıcı etkinliklerindeki anormallikleri saptayabilmesidir. Bu özellik, sistemlerin bilinmeyen yani sıfır gün saldırılarını tanıma kabiliyetini beraberinde getirir [47]. Sistem, incelenen etkinliklerin normale göre farklı olması durumunda bir güvenlik uyarısı başlatır. İç tehditlere karşı da duyarlı olan bu sistemler, normal kullanıcı aktiviteleri içinde gizlenmeye çalışan yetkisiz eylemleri fark edebilir. Eğer bir saldırgan, normalde fark edilmeyen bir kullanıcının kimlik bilgilerini ele geçirip harekete geçerse, sistem bunu anlayıp alarm verebilir. Sistem, kullanıcıya özel profiller oluşturarak bir saldırganın normal kullanıcı davranışını taklit etmesini zorlaştırır. Bu özelleştirme, saldırganın sessizce hareket etmesini engeller.

Anomali tabanlı saldırı tespit teknikleri, yaklaşımlarına ve metodolojilerine göre farklı türlerde sınıflandırılabilir. Yaygın anomali tabanlı saldırı tespit türleri şunlardır:

- İstatiksel anomali tespiti: İstatiksel anomali tespit teknikleri, normal davranıştan sapmaları belirlemek için sistem veya ağ faaliyetlerinin istatistiksel özelliklerini analiz eder. Bu yöntemler normal davranışın bir temelini oluşturmak için ortalama, medyan, standart sapma veya olasılık dağılımları gibi istatistiksel modelleri kullanır. Belirlenen istatistiksel normlardan önemli ölçüde sapan gözlemlenen herhangi bir davranış anomali olarak işaretlenir. İstatistiksel anomali tespiti, sürekli veya ayrık değişkenlerdeki sapmaları tespit etmek için uygundur ve daha önce görülmemiş saldırıları belirlemede etkili olabilir.
- Makine öğrenmesi tabanlı anomali tespiti: Makine öğrenmesi tabanlı anomali tespit teknikleri, anomalileri tespit etmek için denetimli veya denetimsiz makine öğrenmesi algoritmalarından yararlanır. Denetimli öğrenmede, bir model hem normal hem de saldırı örnekleri içereb etiketli veriler üzerinden eğitilir. Model, eğitim sırasında öğrenilen kalıplara ve özelliklere dayanarak yeni örnekleri normal veya anormal olarak sınıflandırmayı öğrenir. Öte yandan, denetimsiz öğrenme yaklaşımları, normal ve saldırı örnekleri hakkında önceden bilgi sahibi olmadan etiketsiz verilerdeki örüntüleri ve anormallikleri tanımlamayı amaçlar. Bu yöntemler özellikle bilinmeyen veya yeni saldırı modelleriyle uğraşırken kullanışlıdır.
- Zaman serisi anomali tespiti: Zaman serisi anomali tespiti, veri noktaları arasındaki sıranın ve zamansal bağımlılıkların dikkate alındığı sıralı verilerdeki anomalileri belirlemeye odaklanır. Bu teknik, anomalileri tespit etmek için zaman içindeki kalıpları, eğilimleri ve sapmaları analiz eder. Zamansal örüntüleri yakalamak için otoregresif modeller, hareketli ortalamalar veya Fourier dönüşümleri gibi zaman serisi analiz yöntemleri kullanılabilir. Zaman serisi anomali tespiti, sistem günlüklerindeki, ağ trafiğindeki veya endüstriyel sistemlerde toplanan sensör verilerindeki anormallikleri tespit etmek için çok uygundur.
- Ağ tabanlı anomali tespiti: Ağ tabanlı anomali tespit teknikleri ağ trafiğini izler ve anormal davranışları veya şüpheli faaliyetleri belirlemek için paket düzeyindeki verileri analiz eder. Bu yöntemler, ağ taramaları, bağlantı noktası taramaları veya olağandışı trafik hacimleri gibi anormallikleri tespit etmek için trafik modellerini, paket başlıklarını, protokolleri veya yük içeriğini inceler. Ağ tabanlı anomali tespiti, özellikle ağ altyapısındaki güvenlik açıklarından veya

olağandışı trafik modellerinden yararlanan izinsiz girişlerin tespit edilmesiyle ilgilidir.

- Ana bilgisayar tabanlı (host-based) anomali tespiti: Ana bilgisayar tabanlı anomali tespiti, tek tek ana bilgisayarlarda veya uç noktalarda meydana gelen etkinlikleri ve olayları analiz etmeye odaklanır. Bu yöntemler, normal kalıplardan sapmaları belirlemek için sistem günlüklerini, işlem etkinliklerini, kullanıcı davranışını veya dosya bütünlüğünü inceler. Ana bilgisayar tabanlı anomali tespiti, yetkisiz erişim girişimleri, ayrıcalık yükseltme veya olağandışı sistem kaynağı kullanımı gibi etkinlikleri tespit edebilir.

Bu anomali tabanlı saldırı tespit türleri daha kapsamlı bir saldırı tespit sistemi elde etmek için birleştirilerek hibrit olarak kullanılabilmektedir. Uygun türün seçimi, izlenmekte olan sistemin özelliklerine, mevcut verilere ve saldırı tespit senaryosunun özel gereksinimlerine bağlıdır.

2.2 Endüstriyel Nesnelerin İnternet'inde Saldırı Tespiti Zorlukları ve Yaklaşımlar

Son yıllarda Endüstri 4.0 kavramı ile birlikte Endüstriyel Nesnelerin İnterneti (IIoT) yaygın hale gelmiştir. IoT ailesinin bir üyesi olan IIoT, akıllı fabrikalar, akıllı şebekeler vb. dahil olmak üzere bir dizi kritik uygulamaya sahiptir. Uç tabanlı IIoT (Edge), tanımı gereği, algılama, hesaplama ve depolama için büyük miktarda uç cihaza dayanmaktadır. Ancak merkezi olmayan veri etkileşimi, veri sızıntısı yaşamayı, manipülasyonu ve diğer ağ saldırılarını kolaylaştırır. Bu nedenle uç tabanlı IIoT'nin güvenlik sorunu bu çalışmanın odak noktası haline gelmiştir.

Saldırı tespiti, endüstriyel sistemlerin güvenliğinin ve esnekliğinin sağlanmasında önemli bir rol oynar. Endüstriyel ortamlar benzersiz özelliklere sahiptir ve özel saldırı tespit yaklaşımları gerektiren farklı zorluklarla karşı karşıyadır.

Endüstriyel sistemler, birbirine bağlı çok sayıda cihaz, makine ve sensörü kapsayan büyük bir ölçekte çalışır. Bu sistemler fabrikalar, enerji santralleri veya ulaşım ağları gibi geniş fiziksel alanlara yayılabilir. Endüstriyel sistemlerin ölçeği ve karmaşıklığı, bu sistemler tarafından üretilen büyük miktarda verinin izlenmesi ve analiz edilmesi karmaşık bir görev haline geldiğinden, saldırı tespiti için zorluklar ortaya çıkmaktadır.

Saldırı tespit mekanizmaları, veri hacmini ve çeşitliliğini gerçek zamanlı olarak verimli bir şekilde ele alabilmelidir.

Endüstriyel sistemlerin bir diğer özelliği cihazların ve protokollerin heterojenliğidir. Bu sistemlerde çeşitli üreticilerden ve nesillerden çok çeşitli cihazlar, sensörler ve protokoller içerir. Saldırı tespit mekafXnizmalarının, cihazların ve protokollerin heterojenliğini ele almak için esnek ve uyarlanabilir olması, endüstriyel ekosistem genelinde uyumluluk ve kapsam sağlaması gerekir.

Kritik süreçlerin güvenli ve verimli çalışmasını sağlamak için zamanında yanıt ve karar verme gerektiğinden, gerçek zamanlı gereksinimler endüstriyel sistemlerde kritik öneme sahiptir. Endüstriyel ortamlarda saldırı tespiti neredeyse gerçek zamanlı olarak çalışmalı, sistem operasyonları üzerindeki etkiyi en aza indirmek için potansiyel tehditleri hızlı bir şekilde tanımlamalı ve bunlara yanıt vermelidir. Geciken tespit veya yanıt, üretim kesintileri, ekipman hasarı veya güvenlik tehlikeleri gibi önemli sonuçlara yol açabilir. Bu nedenle, endüstriyel sistemlerdeki saldırı tespit mekanizmaları zaman ve doğruluğa öncelik vermelidir.

Endüstriyel sistemlerde gerçekleşen saldırılar, mali kayıplar ve fikri mülkiyet hırsızlığından tehlikeye atılan güvenlik ve çevresel hasara kadar ciddi sonuçlar doğurabilir. Endüstriyel ortamlar kritik altyapı ve süreçleri kontrol eder, bu da onları kötü niyetli aktörler için cazip hedefler haline getirir. Yetkisiz erişim, kontrol sistemlerinin manipülasyonu veya veri ihlalleri üretim kesintilerine, sistem arızalarına ve hatta insan operatörlerin zarar görmesine neden olabilir. Saldırı tespiti, bu tür olaylarla ilişkili risklerin en aza indirilmesinde ve endüstriyel sistemlerin bütünlüğünün, kullanılabilirliğinin ve gizliliğinin korunmasında önemli bir rol oynar.

Endüstriyel sistemlerde saldırı tespiti, proaktif tehdit azaltma ve olay müdahalesi sağlar. Endüstriyel operatörler anomalileri, şüpheli faaliyetleri veya potansiyel saldırıları gerçek zamanlı olarak tespit ederek riskleri azaltmak, olayların etkisini en aza indirmek ve daha fazla tehlikeyi önlemek için anında harekete geçebilirler. Saldırı tespit mekanizmaları, potansiyel saldırıların erken uyarı işaretlerini sağlayarak hızlı olay müdahalesi, adli analiz ve düzeltmeye olanak tanır. Bu proaktif yaklaşım, endüstriyel sistemlerin esnekliğini ve kurtarma kabiliyetlerini arttırır

Uç tabanlı IIoT'de saldırı tespit sistemleri tasarımı temel 2 yönü dikkate alır: algılama yöntemi ve sistem mimarisi. Biraz daha detaylandırmak gerekirse, algılama yöntemleri

bilgi tabanlı yöntemler ve anomali tabanlı yöntemler olarak sınıflandırılabilir. Burada bilgi tabanlı yöntemler, bilinmeyen saldırıları tanımlanamayan algılama için kullanıma hazır veritabanlarına dayanırken, anomali tabanlı yöntemler tipik olarak trafiği tanımlamak için makine öğrenmesi algoritmalarını kullanır. Sistem mimarisi olarak ise, saldırı tespit sistemlerinin mimarisi üç kategoride sınıflandırılabilir: dağıtılmış, merkezileştirilmiş ve hibrit. Spesifik olarak dağıtılmış saldırı tespit sistemi, ağdaki her bir fiziksel cihaz üzerinde çalışır ve izinsiz giriş tespitlerini bağımsız olarak yürütür. Buna karşılık merkezileştirilmiş saldırı tespit sistemi, merkezi sunucular gibi birkaç merkezi cihazda çalışır. Yalnızca merkezi cihazlar tüm ağdaki kötü niyetli trafiği tespit edebilir. Hibrit sistemler ise, sırasıyla alt katman ağ cihazları ve üst katman ağ cihazları arasında farklı algılama yöntemlerinin çalıştırıldığı yukarıda belirtilen diğer iki mimarinin avantajlarını birleştirir.

3
OTOMATİK MAKİNE ÖĞRENMESİ METODOLOJİK YAKLAŞIMLARI

Makine öğrenmesi, sistemlerin deneyimlerden öğrenme ve insan müdahalesi olmadan karar verme yeteneklerini geliştirip güçlendirmelerini sağlar [48]. Üst düzey makine öğrenmesi yaklaşımları denetimli ve denetimsiz olmak üzere iki kategoride incelenmektedir. Bununla birlikte, granüler düzeyde makine öğrenmesi dört kategoriye ayrılır: denetim, yarı denetimli, denetimsiz ve pekiştirmeli (reinforcement) öğrenme. Denetimli makine öğrenmesi yöntemleri, geleceğe yönelik tahminler yapmak için etiketli bir veri setinden öğrenir. Veriler etiketsiz ise denetimsiz makine öğrenmesi teknikleri kullanılır. Yarı denetimli makine öğrenmesi teknikleri, öğrenme aşamasında açıklamalı ve açıklamasız verilerin bir karışımını kullanır. Pekiştirmeli öğrenme algoritmaları ise tanımlanmış bir ortamdaki etkileşimlere dayalı olarak ödülleri veya cezaları belirler [49].

3.1 Otomatik Makine Öğrenmesi Yöntem ve Teknikleri

AutoML, bireyin tecrübesi doğrultusunda yapacağı birçok deneme yanılma işleminin maliyetini ortadan kaldırarak makine öğrenmesi sürecini sorunsuz bir iş akışı içerisinde kullanmayı hedeflemektedir [50] [51]. AutoML kapsamındaki otomatikleştirilmiş adımlar şunlardır:

- Otomatikleştirilmiş veri hazırlama
    - Öznitelik türlerinin tespiti (kategorik, sayısal vb.)
    - Öznitelik rolünün algılanması
    - Görev tespiti (ikili sınıflandırma, kümeleme vb.)
- Otomatikleştirilmiş özellik mühendisliği
    - Özellik seçimi
    - Özellik çıkarımı
- Otomatikleştirilmiş model seçimi
- Hiperparametre optimizasyonu
- Performans değerlendirme
- Otomatikleştirilmiş sorun tespiti
- Elde edilen sonuçların otomatik olarak değerlendirilmesi
- Makine öğrenmesi süreci için kullanıcı arayüzleri ve görselleştirme

AutoML çalışmalarında sistemin nasıl çalıştığını açıklayabilmek oldukça önemlidir. Çünkü otomatikleştirilmiş süreçler hem kurgulanan modelin ve başarısının sorgulanabilirliğini hem de yapay zekadaki açıklanabilirlik kavramını karşılayabilmelidir. Bu amaçla AutoML çalışmalarında sürecin nasıl ilerlediğini gösteren işlem hattı (pipeline) tanımlanması gerekmektedir. Şekil 3.1'de klasik bir AutoML sürecine ait işlem hattı gösterilmektedir.

Şekil 3.1 AutoML süreçleri

Bugüne kadar farklı AutoML çalışmaları yapılmış, çeşitli çerçeveler geliştirilmiştir ve literatürde otomatik makine öğrenmesi model seçimi üzerine çalışmalar bulunmaktadır. Ancak bu çalışmaların çoğu AutoML pipeline'ının bazı bölümlerine odaklanmıştır. [31] çalışmasında 150 denetimli sınıflandırıcı karşılaştırılmıştır. Kıyaslama sonucunda TPOT'un (Tree-based Pipeline Automation Tool) 21 sınıflandırma görevinde temel bir makine öğrenmesi analizinden daha iyi performans gösterdiği belirtilmiştir. [51] çalışmasında scikit-learn üzerine inşa edilerek sınıflandırma pipeline'ı oluşturulmuş ancak sadece SVM ve k-Nearest Neighbors (kNN) gibi geleneksel makine öğrenmesi yöntemleri kullanılmıştır. Auto-Keras [52], Keras çerçevesine dayalı olarak geliştirilen derin öğrenme modellerini aramaya odaklanan açık kaynaklı bir kütüphanedir. [53] çalışması, hızlı ve öngörücü bir AutoML pipeline'ı olarak önerilen Oracle AutoML platformunun kapsamlı bir incelemesini içermektedir. Bu otomatik makine öğrenmesi modelinde algoritma seçimi, uyarlanabilir örnekleme, özellik seçimi ve ardından hiperparametre optimizasyonu yapılmaktadır. İleri beslemeli bir yaklaşımın, H2O ve Auto-Sklearn gibi son teknoloji ürünü açık kaynaklı AutoML araçlarına kıyasla daha kısa sürede daha üstün sonuçlar ürettiği kaydedilmiştir. Birçok AutoML algoritması, belirli görevleri çözmek için sabit veri setleri üzerinde çalışır. Ancak, bir AutoML sistemi sürekli öğrenme yeteneğine sahip olmalı ve tüm süreci otonom olarak ilerletme kabiliyetine sahip olmalıdır.

3.2 Saldırı Tespitinde Otomatik Makine Öğrenmesi Kullanımı

Otomatik makine öğrenmesi, siber güvenlik alanında, özellikle de saldırı tespit sistemleri bağlamında umut verici bir yaklaşım olarak ortaya çıkmıştır. Saldırı tespit sistemi, ağ

güvenliğinin kritik bir yönüdür ve saldırı tekniklerinin hızlı gelişimi, uyarlanabilir ve verimli çözümler gerektirmektedir. Yapay zekanın bir alt alanı olan otomatik makine öğrenmesi, makine öğrenmesi modelleri geliştirmenin uçtan uca sürecini otomatikleştirmeye odaklanır ve bu sayede saldırı tespiti için değerli bir araç haline gelir.

Endüstriyel nesnelerin interneti sisteminde saldırı tespiti için bir AutoML çerçevesi geliştirmenin sağlayacağı başlıca faydalar şunlardır:

- Artan Verimlilik: AutoML çerçevesi, saldırı tespit modellerinin oluşturulmasını otomatikleştirerek zamanı ve maliyeti azaltır. Hızlı model geliştirme, dağıtım ve izlemeye olanak tanıyarak potansiyel saldırıların belirlenmesinin etkinliğini arttırır.
- Arttırılmış Doğruluk: AutoML çerçevesi, büyük miktarda veriyi değerlendirebilen, karmaşık kalıpları tanımlayabilen ve geçmişten öğrenebilen makine öğrenmesi algoritmalarını içerir. Bu, yanlış pozitifleri ve yanlış negatifleri azaltarak, geleneksel kural tabanlı tekniklere kıyasla saldırı tespit doğruluğunu arttırabilir.,
- Ölçeklenebilirlik: Yüksek hacimlerde veri üreten çok sayıda cihaz ile IIoT ağları çok büyük bir ölçeğe sahip olabilir. Model geliştirme ve eğitim sürecini otomatikleştirerek, hesaplama kaynaklarının etkin kullanımını sağlayarak ve ağ büyüdükçe ölçeklenebilirlik sağlayarak AutoML çerçevesi bu miktarda veriyi yönetebilir.
- Azaltılmış Uzmanlık Gereksinimleri: Geleneksel saldırı tespit sistemleri için kuralların tasarımı ve bakımı genellikle kapsamlı alan bilgisi gerektirir. AutoML çerçevesi teknik karmaşıklığın büyük bir kısmını otomatikleştirerek özel bilgi ihtiyacını azaltır ve uzman olmayan kişilerin de verim saldırı tespit modelleri oluşturmasını mümkün kılar.

4
OTOMATİK MAKİNE ÖĞRENMESİ ÇERÇEVESİNİN TASARIMI VE GELİŞTİRİLMESİ

Bu çalışmada IIoT sistemleri için saldırı tespitinde görev alabilecek bir AutoML sistemi geliştirilmesi amaçlanmıştır. AutoML sisteminin geliştirilmesi sırasında öncelik X-IIoTID veri setine verilmiştir. Sınıflandırma görevi için uygun olan bu veri setinde 59 özellik ve 820.834 örnek bulunmaktadır. Bu veri seti ile net ve kesin bir saldırı sınıflandırması yapılabilmekte ve WebSocket fuzzing, Constrained Application Protocol (CoAP) kaynak keşfi, MQTT malicious subscription, kripto-fidye yazılımı gibi yeni nesil saldırıları içermektedir. Ayrıca Modbus, WebSocket, MQTT, TCP, Adres Çözümleme Protokolü (ARP), Hiper Metin Aktarım Protokolü (HTTP), Güvenli Kabul (SSH), Alan Adı Sistemi (DNS), Basit Posta Aktarım Protokolü (SMTP), Kullanıcı Datagram Protokolü (UDP) gibi protokolleri de kapsamaktadır. IIoT ve IDS için oluşturulan diğer veri setlerine göre daha kapsamlı bir veri setidir [54].

Bu çalışma kapsamında geliştirilen AutoML işlem hattı aşağıdaki aşamalardan oluşmaktadır.

1. Veri Ön İşleme
2. Özellik Seçimi ve Özellik Çıkarımı
3. Performans/Model Karşılaştırma
4. Dağıtım

4.1 Veri Ön İşleme

Veri hazırlama, makine öğrenmesi sürecinin ilk adımıdır. Veri hazırlama temel olarak üç bölümden oluşur: veri toplama, temizleme ve arttırma. Veri toplama, bir veri seti elde etmek veya mevcut bir veri setini genişletmek için gerekli adımlardan biridir. Veri setlerinde gürültü olması doğal olmakla birlikte bu durum model eğitimini olumsuz etkileyebilir. Veri temizleme işleminde model eğitiminde doğruluk oranlarının düşmesine neden olabilecek gürültülü veriler temizlenir. Bu nedenle veri temizliği yapılmalıdır [29]. Veri arttırma işlemi, modelin performansını ve sağlamlığını arttırmak için önemli bir adımdır. Veri arttırımı, mevcut verilerden yeni veriler ürettiği için bir veri toplama aracı olarak da düşünülebilir. Ayrıca modelin overfitting olmasını önlemek için de

kullanılabilir. Verinin boyutu arttıkça, makine öğrenmesine dayalı modellerin doğruluğu ve hızı önemli ölçüde etkilenir. Veri ön işleme, ham verinin dönüştürülmesinden oluşan bir aşamadır, böylece veri setindeki eksiklik ve tutarsızlıklardan kaynaklanan sorunlar giderilir. Bu aşamada daha anlaşılabilir bir veri seti elde edilmesi amaçlanır.

Ön işleme, ham verilerin makine öğrenmesi modelleri oluşturmak için uygun bir formata dönüştürülmesini içerdiğinden otomatik makine öğrenmesinde önemli bir aşamadır. AutoML'de gerçekleştirilmesi gereken veri hazırlama işleminden sonraki üç önemli ön işleme adımı şunlardır: eksik değerlerin işlenmesi, ölçeklendirme ve kodlama. Gerçek dünya senaryolarında toplanan veriler genellikle eksik değerler içerir ve bu da makine öğrenmesi modellerinin performansını olumsuz yönde etkileyebilir. Bu zorluğun üstesinden gelmek için imputasyon veya silme gibi farklı teknikler kullanılabilir. Seçilen imputasyon yöntemi veri türüne, kayıp değerlerin derecesine ve verinin dağılımına bağlıdır. Bazı makine öğrenmesi algoritmalarından, özelliklerin büyüklüğü modeli etkileyebilir. Tüm özelliklerin eşit şekilde dikkate alınmasını sağlamak için, tüm özellikleri aynı büyüklüğe getirmek üzere ölçeklendirme yapılır. Bu genellikle özelliklerin sıfır ortalamaya ve bir standart sapmaya sahip olacak şekilde normalleştirilmesiyle elde edilir. Makine öğrenmesi algoritmaları sayısal verilerle çalışır ve bu nedenle kategorik değişkenlerin sayısal forma dönüştürülmesi gerekir. Yaygın kodlama teknikleri arasında one-hot kodlama, label kodlama ve ordinal kodlama yer almaktadır. Kodlama yönteminin seçimi, kategorik verilerin türüne ve kullanıla makine öğrenmesi algoritmasının özel gereksinimlerine bağlıdır.

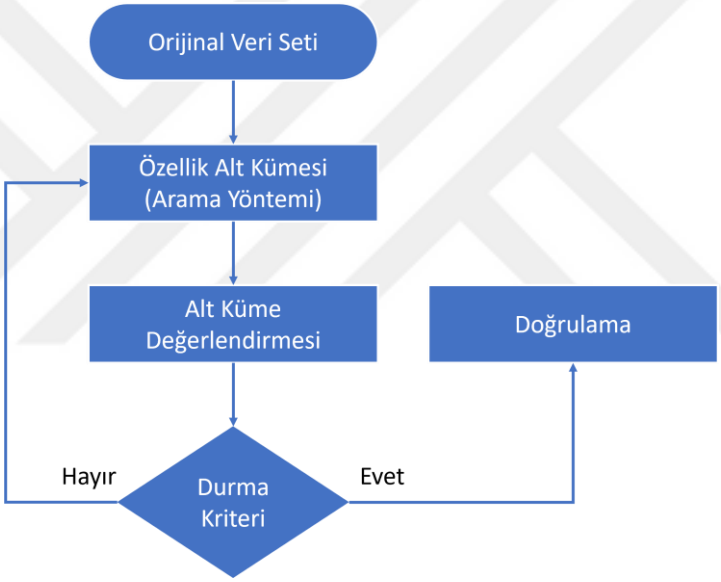
4.2 Özellik Seçimi ve Özellik Çıkarımı

Özellik mühendisliği, ham verilerin algoritmalar ve modeller tarafından kolayca anlaşılabilecek ve analiz edilebilecek bir biçime dönüştürülmesini içeren makine öğrenmesi işlem hattında önemli bir adımdır. Özelliklerin kalitesi bir modelin performansı üzerinde önemli bir etkiye sahip olduğundan, bu adım AutoML işlem hattı için önemlidir.

Özellik seçimi, modelin eğitimini hızlandırmak için gereksiz olanları eleyerek orijinal özelliklerin bir alt kümesini oluşturmayı içerir. Bu süreç, overfitting'i önleyerek model performansını arttırır.

Özellik seçimi, daha basit ve anlaşılır modeller oluşturmada, daha anlaşılır veriler hazırlamada ve sonuç performansını arttırmada fayda sağlar. Özellik seçimi, görüntü tanıma için gürültülü, ilgisiz ve gereksiz özelliklerin elenmesi, doğal dil işleme için kısaltmalar, yanlış yazımlar ve eş anlamlı kelimeler gibi dil zorluklarının giderilmesi, işlem maliyetinin azaltılması, depolamanın en aza indirilmesi ve saldırı tespiti için test verilerinin daha iyi anlaşılmasını sağlamak gibi farklı görevlerde uzun yıllardır kullanılmaktadır.

Özellik seçimi, orijinal özellik setine dayalı bir alt küme oluşturarak alakasız veya gereksiz özellikleri azaltır. Bu şekilde model basitleştirilebilir, overfitting önlenebilir ve model performansı iyileştirilebilir. [21] çalışmasına göre, özellik seçim süreci Şekil 4.1'de gösterildiği gibi dört temel adımdan oluşmaktadır.



Şekil 4.1 Özellik seçimindeki temel adımlar

Özellik seçim süreci, bir arama yöntemi kullanılarak seçilen özelliklerin bir alt kümesinin sistematik olarak değerlendirilmesini içerir. Seçilen alt küme daha sonra geçerliliğini belirlemek için bir doğrulama sürecinden geçirilir. Bu süreç, bir durdurma kriteri karşılanana kadar tekrarlanır.

Özellik seçiminde üç tür arama yöntemi kullanılır: tam arama, sezgisel arama ve rastgele arama. Tam arama, kapsalı ve kapsamlı olmayan arama olarak ikiye ayrılabilir. Kapsamlı olmayan arama, breadth-first, beam search, branch and bound ve best-first search yöntemleri dahil olmak üzere çeşitli tekniklere sahiptir.

Sezgisel arama, belirli yöntemlerin kullanımını içerir. Bu yöntemlerden biri sıralı ileri seçimdir (Sequential Forward Selection, SFS). Bir diğeri sıralı geriye doğru seçim (Sequential Backward Selection, SBS) ve bir diğeri de çift yönlü aramadır (Bidirectional Search, BS). SFS ve SBS yöntemleri, özelliklerin boş bir kümeye eklenmesini veya dolu bir kümeden çıkarılmasını içerir. BS yönteminde, aynı alt küme elde edilene kadar arama yapmak için bu iki algoritma birlikte kullanılır. Yaygın olarak kullanılan rastgele arama yöntemleri genetik algoritmalar ve parçacık sürücü optimizasyonudur.

Özellik seçme yöntemleri üç kategoride sınıflandırılabilir: filtreleme, sarmalayıcı ve gömülü. Filtreleme yöntemleri seçim için yalnızca istatistiksel bilgileri kullanırken, sarmalayıcı yöntemler özelliklerin kendilerine dayanan aramaları kullanır. Öte yandan gömülü yöntemler bölme için en iyi kriteri bulmaya odaklanır [55]. Filtre tabanlı özellik seçimi yönteminde, seçim işlemi veri setinin özelliklerine göre yapılarak optimal özellik kümesini belirlemek için çeşitli istatistiksel testler kullanılır. Bu yöntem hesaplama yükü açısından düşük maliyetli ve etkili olup, hızlı sonuçlar verir. Diğer taraftan, sarmalayıcı tabanlı özellik seçimi yöntemi, bir sınıflandırıcıyı öngörücü olarak kullanarak, en iyi özellik alt kümesini buluncaya kadar çeşitli özellik kombinasyonlarının performansını değerlendirir. Bu yaklaşım daha maliyetli ve zaman alıcıdır, fakat genellikle daha doğru sonuçlar verir. Gömülü tabanlı özellik seçimi yöntemi ise filtre ve sarmalayıcı yöntemlerinin özelliklerini birleştirerek, hibrit bir yaklaşım sunar ve genellikle bu teknik hibrit özellik seçimi olarak da adlandırılır [56] [57] [58].

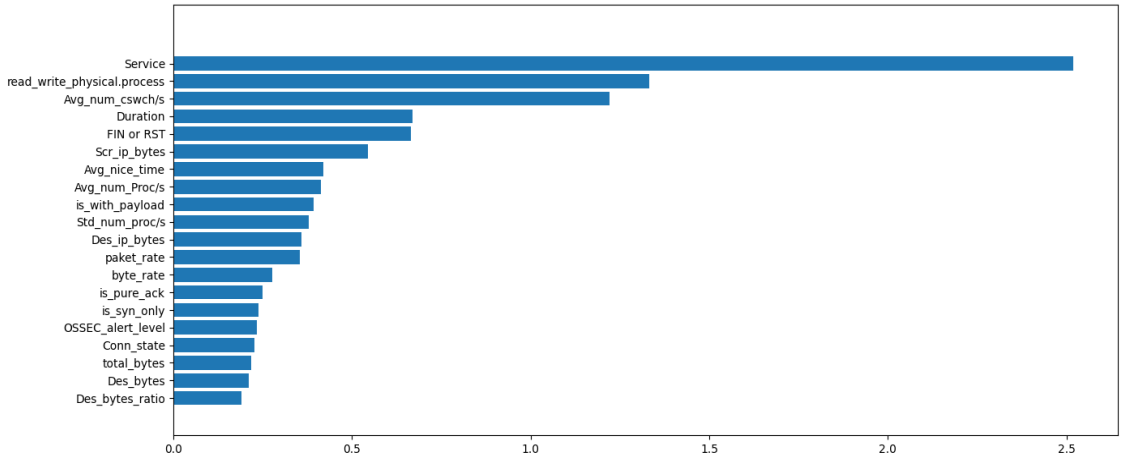
Bu çalışmada özellik seçimi için iki farklı yöntem kullanılmıştır. Bunlardan biri Shapley Additive Explanations (SHAP), diğeri ise genetik algoritmadır.

4.2.1 Shapley Değerleri

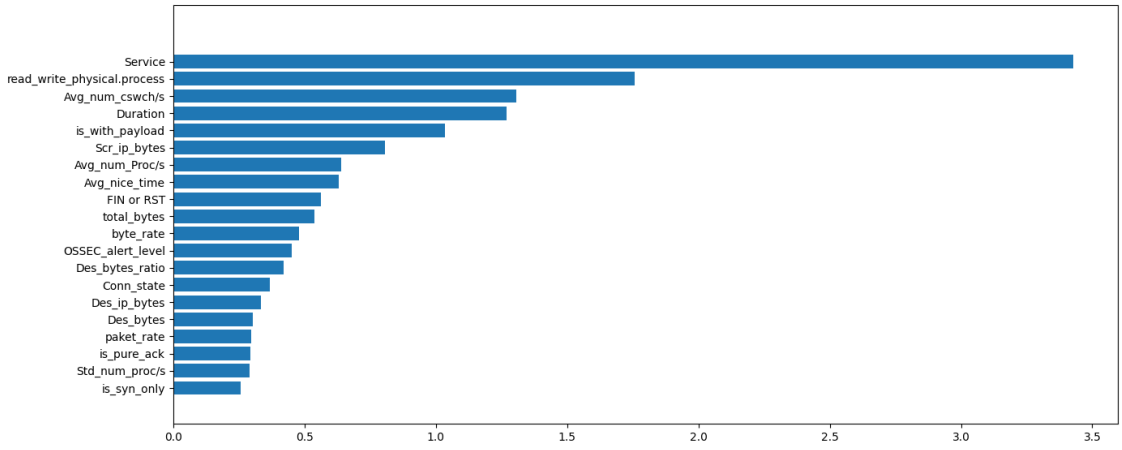
Lloyd Shapley [59] tarafından işbirlikçi oyun teorisi alanında ortaya atılan Shapley Değerleri, her oyuncunun belirli bir oyuna katkısını ölçmek için bir araç sağlar. Oyun teorisi, iki veya daha fazla oyuncunun, sonuçlarının birbirine bağlı olduğu bir durumda nasıl etkileşime girdiğinin stratejik bir analizidir. Shapley Değerleri özellikle oyuncuların katkılarının eşit olmadığı, ancak yine de ortak sonuçlar üretmek için birlikte çalıştıkları durumlarda önemlidir.

Shapley Değerleri yöntemi, SHAP yönteminin oluşturulmasıyla makine öğrenmesi modelleri tarafından yapılan tahminlerin yorumlanmasında kullanılmak üzere uyarlanmıştır. SHAP yöntemi, [24] çalışmasında sunulduğu gibi, analiz altındaki bir

örneğin her bir özelliği için Shapley Değerlerini hesaplayarak her bir özelliğin nihai tahmine katkısı hakkında bilgi verir. Bu, makine öğrenmesi modellerinin karar verme süreçlerinde daha fazla şeffaflık sağlayarak tahminlerine nasıl ulaştıklarının daha iyi anlaşılmasına olanak tanır. Şekil 4.2 ve Şekil 4.3, SHAP ve farklı sınıflandırma modellerini kullanarak X-IIoTID veri setindeki en önemli 20 özelliği göstermektedir.



Şekil 4.2 SHAP ve XGBoost ile özellik seçimi



Şekil 4.3 SHAP ve LightGBM ile özellik seçimi

4.2.2 Genetik Algoritma

Optimizasyon algoritmalarından biri olan genetik algoritma, insan genlerinin nesiller arası geçiş sürecine benzetilmektedir. Genetik algoritma, belirli bir probleme en iyi çözümü bulmak için doğal seleksiyon ve genetik sürecini taklit eden olasılıksal bir optimizasyon tekniğidir.

Genetik algoritma yöntemi, özelliklerin seçilmesi için makine öğrenmesi modelleri alanında sıkça tercih edilen bir optimizasyon stratejisidir. Genetik algoritma kullanarak

belirli bir makine öğrenmesi modeli için önemli veya ilgili özellikleri belirleme süresi genetik algoritma ile özellik seçimi olarak bilinir. Temel amaç, modelin performansını korurken verilerin boyutunu azaltmaktadır. Özellik seçiminde genetik algoritma, optimum özellik kombinasyonunu belirlemek için popülasyon boyutu, çaprazlama ve mutasyon oranları ve seçim kriterleri gibi hiperparametreleri optimize etmek için kullanılır. Genetik algoritma tabanlı özellik seçiminde kullanılan fitness fonksiyonu, belirli bir özellik kümesini kullanarak makine öğrenmesi modelinin performansını değerlendirir ve en iyi performans gösteren özellikler seçilerek yeni bir nesil oluşturmak üzere birleştirilir. Genetik algoritma ile özellik seçimi, uygun bir performans eşiğine ulaşılana veya sonlandırma kriteri karşılanana kadar gerçekleştirilen iteratif bir süreçtir. Bu süreç, overfitting riskini azaltmayı amaçlar ve makine öğrenmesi modelinin ne kadar iyi performans gösterdiği üzerinde önemli bir etkiye sahip olan özellikleri bularak modelin genelleme yeteneğini geliştirir. Popülasyon bilgisi, fitness değeri, ebeveyn seçimi, çaprazlama ve mutasyon algoritmayı oluşturan aşamalardan bazılarıdır. Popülasyon istatistiklerine dayanarak, ilk aşamada veriye uygun bir popülasyon rastgele üretilir. İkinci aşamada her bir çözümün (kromozom) fitness değeri belirlenir. Üçüncü aşamada en iyi kromozomlar ebeveyn olarak seçilir. Sonuç olarak yeni bireyler oluşur ve genler yayılır. Dördüncü aşamada, ebeveynler çaprazlama yoluya yeni bir kromozom kümesi oluşturmak için birleştirilir. Bu yeni oluşan popülasyon daha sonra popülasyon kümesine eklenir. Son aşamada, popülasyon kümesindeki kromozomlara mutasyon uygulanır.

Özellik seçimi için genetik algoritma kullanmanın temel avantajı, sadece tek tek özellikleri değerlendirmek yerine, özellikler arasındaki etkileşimleri göz önünde bulundurarak en uygun özellik alt kümesini arama yeteneğidir. Genetik algoritma çok sayıda özelliği ele alabilir ve özellik seçimi problemine başarılı bir çözüm sağlayabilir. Ancak genetik algoritma, özellikle yüksek boyutlu veri kümeleriyle uğraşırken hesaplama açısından pahalı olabilen çok sayıda fonksiyon değerlendirmesi gerektirir. Bu nedenle, değerlendirme metriğini dikkatlice seçmek ve hesaplama verimliliği ile çözüm kalitesi arasındaki dengeyi sağlamak için uygun sonlandırma kriterlerini belirlemek önemlidir.

Genetik algoritmanın adımları Tablo 4.1'de gösterilirken, Tablo 4.2'de CatBoost sınıflandırma yöntemi kullanılarak genetik algoritmanın özellik seçimindeki uygulaması gösterilmektedir. Algoritma her iterasyonda kullanılan özellikleri seçer ve bu özellikler kullanıldığında elde edilen doğruluk değerlerini hesaplar.

Tablo 4.1 Genetik algoritma ile özellik seçiminin adımları

Genetik Algoritma ile Özellik Seçiminin Adımları
1. Başlangıç: Rastgele popülasyon. 2. Adaptasyon: Her kromozom için uygunluk değeri hesaplanır, uygunluk değerleri dizilerin çözüm kalitesini gösterir, kromozom bir özellik kümesidir. 3. Seçim: Seçim, uygunluk değeri hesaplanan bireylerin belirli bir seçim operatöründen geçirilerek hayatta kalmasına dayanır. 4. Çaprazlama: Çaprazlama operatörleri, seçim yöntemleri yardımıyla belirlenen bireylerden yeni bireyler oluşturmak için kullanılır. 5. Mutasyon: Bu adımda amaç, gelecek nesillerde benzer bireylerin oluşmasını engelleyerek çeşitlilik yaratmak ve arama uzayını genişleterek daha detaylı bir arama ile en iyi ve en güçlü bireylerin ortaya çıkarılmasını sağlamaktır.

Tablo 4.2 Seçilen özelliklerin CatBoost sınıflandırıcısı ile hesaplanan doğruluğu

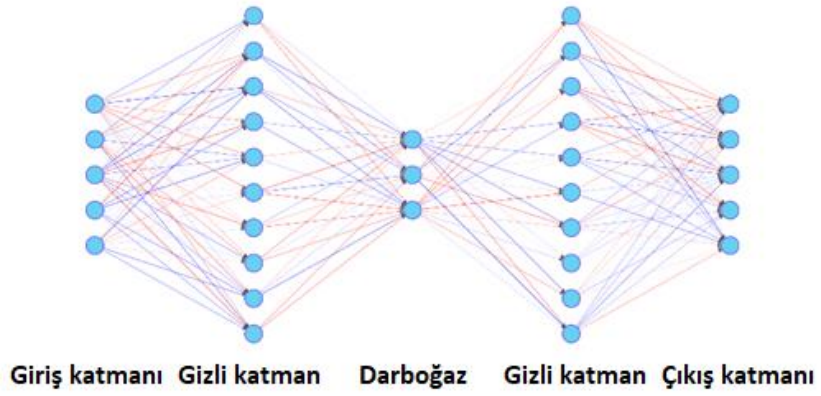
Özellik Numarası	Doğruluk
1, 3, 4, 5, 7, 8, 11, 12, 13 ,14, 16, 18, 19, 20, 23, 24, 26, 27, 28, 29, 31, 34, 35, 36, 37, 39, 41, 43, 45, 47, 50, 51, 52, 53, 54, 57, 58, 59	0,9963
1, 3, 4, 5, 6, 7, 9, 10, 11, 13, 14 ,15 ,16, 17, 18, 19, 20, 21, 24, 25, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 41, 43, 45, 46, 47, 48,49, 50, 51, 52, 54, 55, 58, 59	0,9964
1, 2, 3, 4, 6, 12, 13, 15, 16, 17, 20, 23, 24, 26, 28, 29, 30, 31, 32, 34, 35, 36, 38, 45, 47, 49, 51, 52, 54, 55, 57, 58, 59	0,9965
2, 3, 4, 5, 6, 7, 11, 13, 14, 16, 17, 18, 21, 23, 24, 25, 27, 29, 30, 31, 32, 33, 34, 36, 39, 41, 43, 44, 45, 48, 49, 50, 51, 52, 53, 55, 57, 58, 59	0,9966
1, 2, 3, 4, 6, 7, 11, 12, 15, 16, 17, 18, 19, 21, 22, 24, 25, 26, 28, 29, 30, 33, 34, 38, 41, 43, 44, 48, 49, 50, 51, 52, 53, 55, 57, 58, 59	0,9967

Özellik çıkarımı, özellikle ağ güvenliğinde sınıflandırma sürecinin önemli aşamalarından biridir. Ağ güvenliğinde sınıflandırma işleminin en zor yönü, sürekli büyüyen trafik verisi

hacmidir. Bu nedenle, saldırı tespit sistemlerinde makine öğrenmesi yöntemlerinin doğruluğunu arttırmak için özellik çıkarımı önemli bir konu haline gelmiştir. Bu çalışmada otomatik makine öğrenmesinin özellik çıkarımı aşamasında doğrusal boyut indirgeme teknikleri PCA, LDA ve doğrusal olmayan boyut indirgeme yöntemi Autoencoder uygulanmış ve saldırı tespit sistemleri için oluşturulan veri setlerinde özellik çıkarımı yöntem ve parametre seçimleri otomatikleştirilmiştir. Sistemin otomatik olarak değerlendirip seçeceği yöntem ve parametrelerin performansları, literatürde sıklıkla kullanılan saldırı tespit sistemi veri setleri kullanılarak elde edilmiş, sınıflandırma doğrulukları ve hesaplama süreleri karşılaştırılmıştır.

4.2.3 Autoencoder

Autoencoder, sıkça tercih edilen bir denetimsiz öğrenme algoritmasıdır. Giriş verilerini yeniden oluşturmak için birlikte çalışan kodlayıcı ve kod çözücü olmak üzere iki temel bileşenden oluşur. Giriş verileri kodlayıcı tarafından daha düşük boyutlu bir temsile dönüştürülür ve kod çözücü daha sonra bu temsili orijinal girişe geri eşler. Bir Autoencoder hem giriş hem de çıkış katmanlarında aynı sayıda nörona sahip olması nedeniyle diğer ileri beslemeli sinir ağlarından ayrılmaktadır. Bu durum, Autoencoder'ın girdiyi minimum kayıpla yeniden yapılandırmasını sağlayarak boyut azaltma görevleri için etkili olmasını sağlar. Boyut azaltma, gizli katmanda giriş ve çıkış katmanlarına kıyasla daha az nöron bulunmasıyla elde edilmektedir. Yüksek boyutlu giriş verilerinin daha düşük boyutlarda ifade edilmesini sağlayan Autoencoder'ın çalışma prensibi Şekil 4.4'te gösterilmektedir.



Şekil 4.4 Autoencoder ile boyut indirgeme

Autoencoder'lar genellikle aynı boyutta giriş-çıkış katmanları ve daha küçük ara katmanlardan oluşur. Girdiyi yeniden yapılandırmak üzere eğitilen Autoencoder'lar, bu ara katmanlar sayesinde girdi verisini sıkıştırarak daha düşük boyutlu hale getirebilir. Gizli katmanlı yapay sinir ağının kodlayıcı (4.1) ve kod çözücü (4.2) denklemleri sırasıyla aşağıda gösterilmiştir.

$$Y = f_\theta(X) = s(WX + b_X) \tag{4.1}$$

$$X' = g_{\theta'}(Y) = s(W'Y + b_Y) \tag{4.2}$$

Denklem (4.1) ve (4.2)'deki X verileri, ağırlık parametreleri W ve sapma parametreleri b kullanan kodlanmış bir f(x) ve kodu çözülmüş bir g(Y) fonksiyonu ile dönüştürülmektedir. Kodu çözülmüş g fonksiyonu, gizli Y temsilini X'in yeniden yapılandırılmasına geri eşler. Autoencoder'ın eğitim süreci yeniden yapılandırma kaybını en aza indirmek için θ = (W, bx, by) parametrelerini belirlemektir. Bu fonksiyon (4.3)'te gösterilmektedir.

$$\theta = \min L(X, X') = \min L(X, g(f(x))) \tag{4.3}$$

Doğrusal yeniden yapılandırmada, L1 kayıplı yeniden yapılandırma ortalama kare hatasına göre hesaplanır (4.4):

$$L_1(\theta) = \frac{1}{M} \sum_{i=1}^{M} |x_i - \tilde{x}_i|^2 \tag{4.4}$$

Doğrusal olmayan yeniden yapılandırmada, L2 kayıp yeniden yapılandırması ikili çapraz entropiye dayalı olarak hesaplanır (4.5):

$$L_2(\theta) = -\frac{1}{M} \sum_{i=1}^{M} \sum_{j=1}^{n} [x_{j,i} \log \tilde{x}_{j,i} + (1 - x_{j,i}) \log(1 - \tilde{x}_{j,i})] \tag{4.5}$$

4.2.4 Temel Bileşen Analizi

Bir verinin temel bileşenleri, veri normalize edildikten sonra kovaryans matrisinin özdeğeri ve özvektörü hesaplanarak elde edilir. Bu yöntem, veri kümesindeki özellikler arasındaki ilişkiyi tanımlar, bu ilişkilere dayanarak verileri dönüştürür ve bu ilişkilerin önemini ölçer. Bu şekilde, önemli ilişkiler korunurken diğerlerinin boyutu azaltılır. Temel bileşen analizinde dört temel adım bulunmaktadır:

- N boyutlu verileri NxN boyutlu kovaryans matrisine sahiptir.

- Matrisin N özdeğeri hesaplanır.

- En büyük ilk M özdeğere karşılık gelen M özvektör belirlenir.

- Veriler M özvektör üzerine yansıtılarak N boyuttan M boyuta indirgenir.

Temel bileşen analizinin hesaplanmasında kullanılan denklemler (4.6) - (4.10)'da yer almakta olup sözde kod aşağıda gösterilmiştir.

$$X^T X = \sum_{i=1}^{N} (x_i - \mu)^T (x_i - \mu) \tag{4.6}$$

$$X^T X = V \Lambda V^T \tag{4.7}$$

$$U = X V \Lambda^{-\frac{1}{2}} \tag{4.8}$$

$$U_d = [u_1, \dots, u_d] \tag{4.9}$$

$$Y = U_d{}^T X \tag{4.10}$$

Algoritma 4.1 Temel Bileşen Analizi
1: **prosedür** PCA
2: Nokta çarpım martisini hesaplama (4.6)
3: Eigen analizi (4.7)
4: Eigen vektörleri hesaplama (4.8)
5: Belirli sayıdaki ilk bileşeni kaydetme (4.9)
6: d özelliklerini hesaplama (4.10)

4.2.5 Doğrusal Diskriminant Analizi

Doğrusal Diskriminant Analizi, sınıf bilgisi belli olan verileri indirgemek için kullanılır ve sınıfları en iyi ayrıştıracak vektörleri arar. Bu yöntem, sınıflar arasındaki mesafenin büyük, sınıf içindeki mesafenin ise küçük olmasını gerektirir. PCA, verilerin sınıflarından bağımsız olarak boyut indirgeme işlemi gerçekleştirir. Ancak sınıf örneklerinin birbiriyle örtüştüğü durumlarda sınıflandırma başarısı düşer. Bu gibi durumlarda LDA tercih edilir, çünkü LDA varyans değerlerine ek olarak sınıf bilgisini de kullanarak boyut indirgeme yapar.

Verinin boyutunun azaltılması analiz kolaylığı, eğitim/test sürelerinden azalma ve doğrulukta artış sağlayabilir. Boyut indirgeme yöntemlerinin temel amacı, minimum veri içeriği kaybıyla maksimum boyut azaltımıdır. Bunu başarmak için verilerin temel bileşenlere izdüşümü yaklaşımı kullanılır. Başka bir deyişle, verinin önemli olmayan

bileşenleri tespit edilerek çıkarılır ve böylece boyut azaltılır. Boyut indirgeme ile yüksek boyutlu verilerdeki bilgi daha az sayıda özellik ile temsil edilir. Küçük bir özellik kümesi ile sınıflandırma performanslarının arttırılması hedeflenir.

Boyut azaltma teknikleri özellik mühendisliğinin adımlarından biridir. Bu teknikler arasında PCA ve LDA öne çıkarken, doğrusal olmayan ve bir makine öğrenmesi algoritması olan Autoencoder yöntemi öne çıkmaktadır. LDA yönteminin sözde kodu aşağıda gösterilmiştir. Algoritmada kullanılan denklemler Denklem (4.11) - (4.13)'de yer almaktadır.

$$(I - M)X^T X(I - M) = V_w \Lambda_\omega {V_w}^T \tag{4.11}$$

$$U = X(I - M)V_w {\Lambda_\omega}^{-1} \tag{4.12}$$

$$U = [u_1, \dots, u_{N-C}] \tag{4.13}$$

$$\tilde{X}_b = U^T XM \tag{4.14}$$

Algoritma 4.2 Doğrusal Diskriminant Analizi

1: **prosedür** LDA

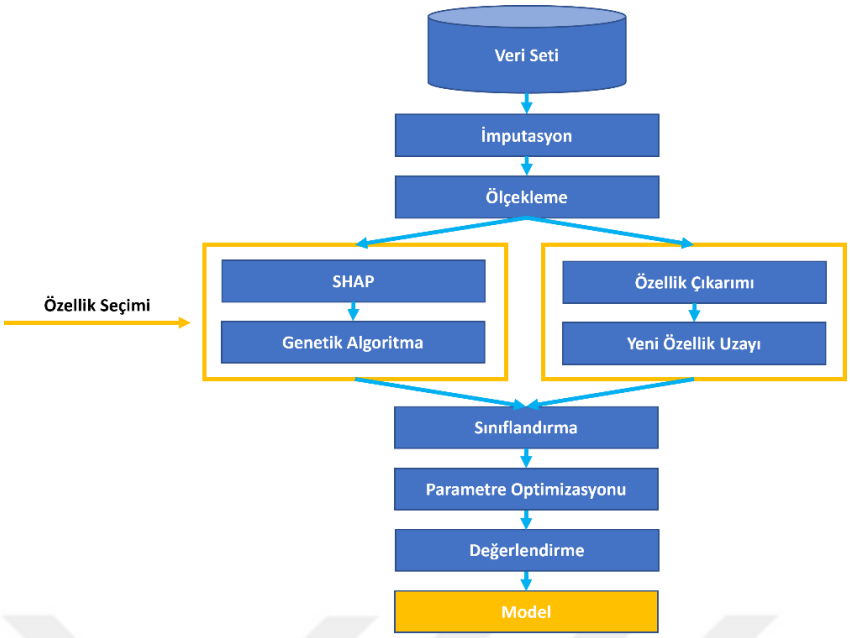
2: Denklem (4.11) üzerine eigen analiz uygulayarak ve (4.12)'yi hesaplayarak S_W'nin sıfır olmayan (4.13) öz değerlerine karşılık gelen özvektörleri belirleme

3: Verilere (4.14) eşitliği uygulayarak daha düşük boyutlu bir uzaya yansıtma

4: Yansıtılan verileri bir sınıflandırıcıya girdi olarak sunma

4.3 Performans/Model Karşılaştırma

Endüstriyel Nesnelerin İnterneti sistemlerinde saldırı tespitine yönelik önerilen otomatik makine öğrenmesi çerçevesinde, her adımda tüm süreçleri ve bunlara karşılık gelen metrikleri izlemek ve kaydetmek için kapsamlı bir kayıt tutma mekanizması geliştirilmiştir. Bu kayıt tutma yaklaşımı, çerçevenin her bir bileşeni için en uygun tekniklerin seçilmesine olanak tanıyarak, sağlam ve yüksek performanslı bir saldırı tespit sisteminin geliştirilmesini sağlamaktadır. Çerçeveye ait akış Şekil 4.5'te gösterilmektedir. Her bir adımda elde edilen en yüksek doğruluğa sahip çıktı bir sonraki adıma girdi olarak verilmektedir.



Şekil 4.5 Otomatik makine öğrenmesi çerçevesi akışı

Veri ön işleme aşamasında, uygulanan ölçeklendirme teknikleri (Min-Max, Standart ve Robust), imputasyon yöntemleri (Mean, Median ve Most Frequent) sonuçları kaydedilmektedir. Ön işleme sonuçları arasında en yüksek sonucu veren yöntem ve konfigürasyonlar bir sonraki adım olan özellik seçimi adımına gönderilmektedir. Birden fazla makine öğrenmesi modelini eğitip değerlendirirken, farklı hiperparametre yapılandırmalarına sahip her model için performans metrikleri kaydedilmektedir. Kayıt işlemi hem Grid Search hem TPE hiperparametre optimizasyon tekniklerinden elde edilen sonuçları içermektedir. Her model ve konfigürasyon için hiperparametre optimizasyonunda geçen süre de kaydedilmektedir. Çapraz doğrulama süreci boyunca, her kattan elde edilen performans metriklerini kaydederek modelin veri kümesinin farklı alt kümeleri üzerindeki performansının ayrıntılı bir görünümü elde edilmektedir. Farklı eğitim ve doğrulama koşullar altında modellerin tutarlılığının ve genellenebilirliğinin değerlendirilmesi sağlanmaktadır.

Otomatik makine öğrenmesi işlem hattının tamamını gerçekleştirdikten ve sonuçları kaydettikten sonra, çerçevede kullanılan farklı tekniklerin karşılaştırmaları yapılmaktadır. Bu karşılaştırma, ölçekleme, imputasyon, özellik seçimi, model ve hiperparametre yapılandırmasının her bir kombinasyonu için kaydedilen performans metriklerini içermektedir. Kaydedilen verilere ve performans metrikleri karşılaştırmalarına dayanarak, çerçevenin her bir bileşeni için en yüksek sonucu veren teknik ve konfigürasyonlar seçilmektedir. Farklı değerlendirme kriterleri arasında tutarlı

bir şekilde en yüksek doğruluk ve performans metriklerini elde eden teknikler otomatik olarak seçilmektedir

4.4 Dağıtım

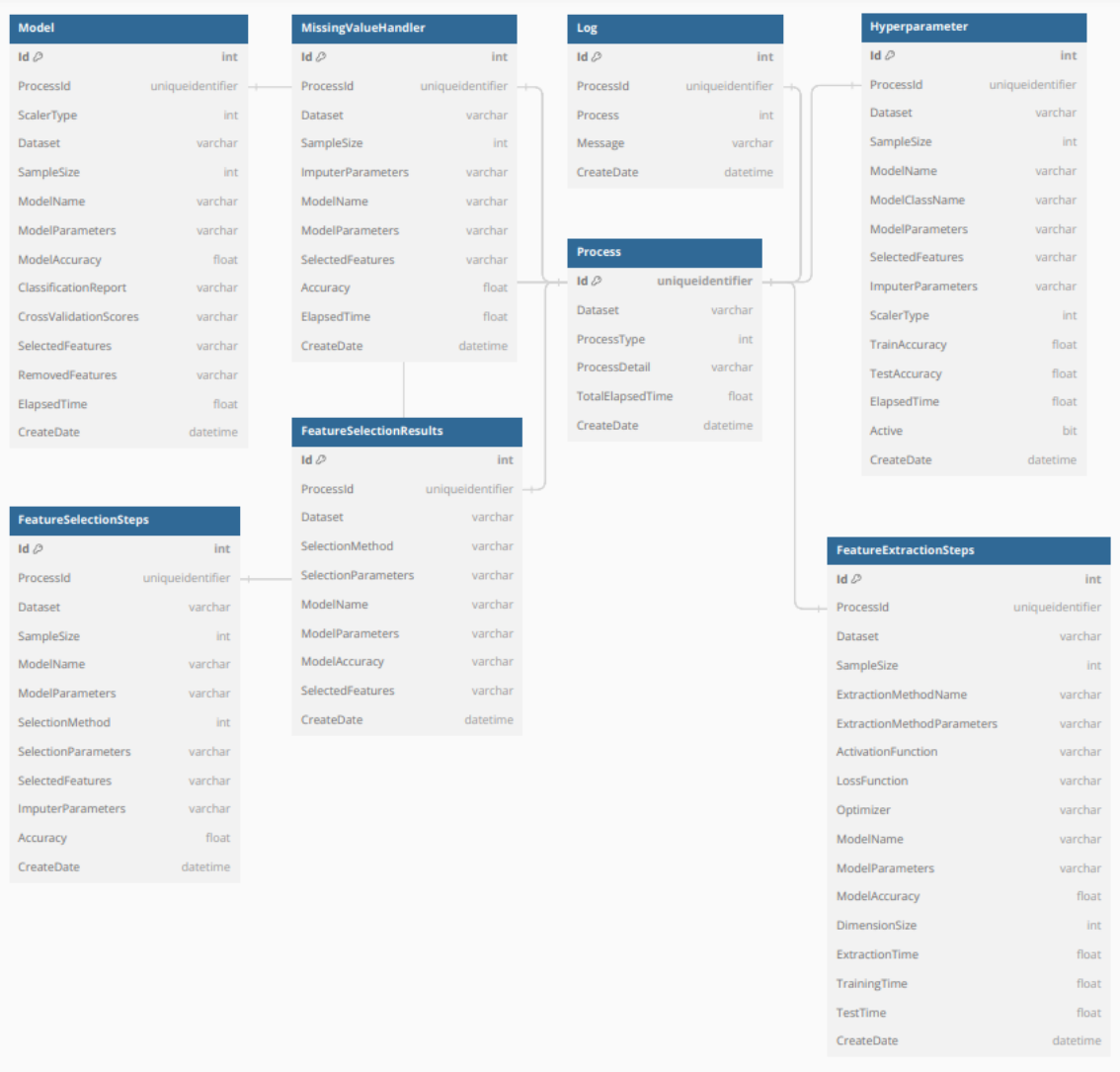
Otomatik makine öğrenmesi çerçevesinde, yoğun kaynak gerektiren eğitim ve test süreçlerini yürütmek için bulut bilişimin gücünden yararlanılmaktadır. Model eğitimi tamamlandıktan sonra, eğitilen model gerekli ölçeklendirme ve kodlama bilgileriyle birlikte IIoT ağındaki uç cihazlara dağıtılmaktadır. Bu uç dağıtım stratejisi, kaynak kısıtlaması olan uç cihazlardan hesaplama yükünün kaldırılmasına olanak tanıyarak, bu cihazların kapsamlı eğitim veya test kaynaklarına ihtiyaç duymadan gerçek zaman saldırı tespini verimli bir şekilde gerçekleştirmelerini sağlamaktadır.

İlk eğitim ve test süreçleri, geniş hesaplama kaynaklarından ve ölçeklenebilirliğinden yararlanılarak bulut altyapısı üzerinde gerçekleştirilmektedir. Bu aşamada AutoML çerçevesi ön işleme tekniklerini, özellik seçimini, makine öğrenmesi modellerini ve hiperparametre yapılandırmalarını gerçekleştirmektedir. En uygun yöntem ve parametre kombinasyonunu belirledikten sonra, çerçeve modelin doğruluğunu ve performans metriklerini optimize eder. Bulut tabanlı eğitim ve test tamamlandıktan sonra, çerçeve eğitilmiş modeli, ölçeklendirmeli ve kodlama bilgilerini dağıtım paketinde toplar. Bu paket daha sonra IIoT ağındaki uç cihazlara güvenli bir şekilde iletilir. Model, uç cihazlara yerleştirildikten sonra gelen ağ akış verileri üzerinde gerçek zamanlı çıkarım yapılabilmektedir. Uç cihazlar, gelen verileri önceden işleme için alınan ölçeklendirme ve kodlama bilgilerini kullanır ve model için uygun bir formatta olmasını sağlar. Uç cihazlar eğitilmiş modeli aldıklarından ve herhangi bir yoğun kaynak kullanımı ile eğitim veya test gerçekleştirmeleri gerekmediğinden, çıkarım görevleri için sınırlı hesaplama kaynaklarını verimli bir şekilde kullanabilir. Bu da uç cihazların üzerindeki hesaplama yükünü azaltarak, performanstan ödün vermeden neredeyse gerçek zamanlı saldırı tespiti yapabilmelerini sağlar.

Bulut tabanlı eğitim ve test süreçleri, güncellenmiş veriler üzerinde veya gelişen siber tehditlere yanıt olarak modeli yeniden eğitmek için periyodik olarak planlanabilmektedir. Güncellenen model daha sonra sorunsuz bir şekilde uç cihazlara yeniden dağıtılabilmektedir. Böylece saldırı tespit sisteminin güncel kalması ve IIoT ortamındaki değişikliklere uyum sağlanmaktadır.

4.5 Sistem Mimarisi

Otomatik makine öğrenmesi tabanlı saldırı tespit sisteminin mimarisi, endüstriyel ağları siber tehditlerden korumak için tasarlanmış, dağıtılmış ve ölçeklenebilir bir çerçevedir. Mimari, her biri endüstriyel ağların güvenliğini ve güvenilirliğini sağlamada belirli bir amaca hizmet eden, birbirine bağlı çeşitli bileşenlerden oluşmaktadır. Mimariye ait veri tabanı diyagramı Şekil 4.6'da gösterilmektedir.

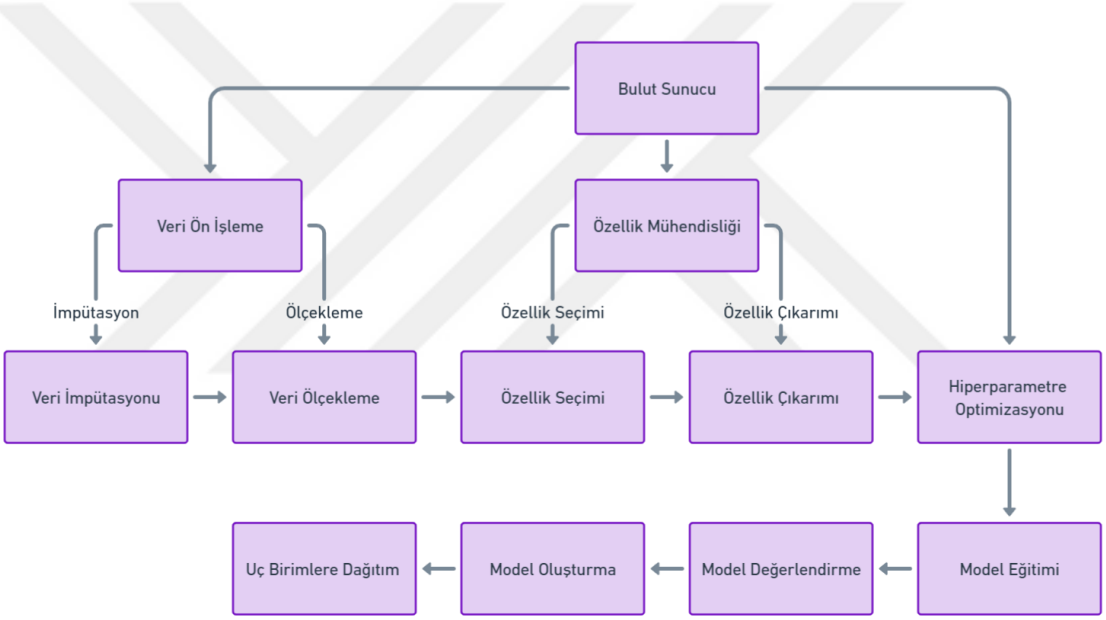


Şekil 4.6 Veri tabanı diyagramı

Uç birim cihazları, endüstriyel ağ içinde konumlandırılan sensörlerden, cihazlardan ve ağ trafiğinden ham verileri yakalayan cihazlardır. Analiz için buluta verileri göndermeden önce gürültüyü ve ilgisiz verileri gidermek üzere verileri ön işleme tabi tutar ve filtreler. Uç birim cihazlarında verileri temizlenir, dönüştürülür, kalite ve uygunluklarını sağlamak

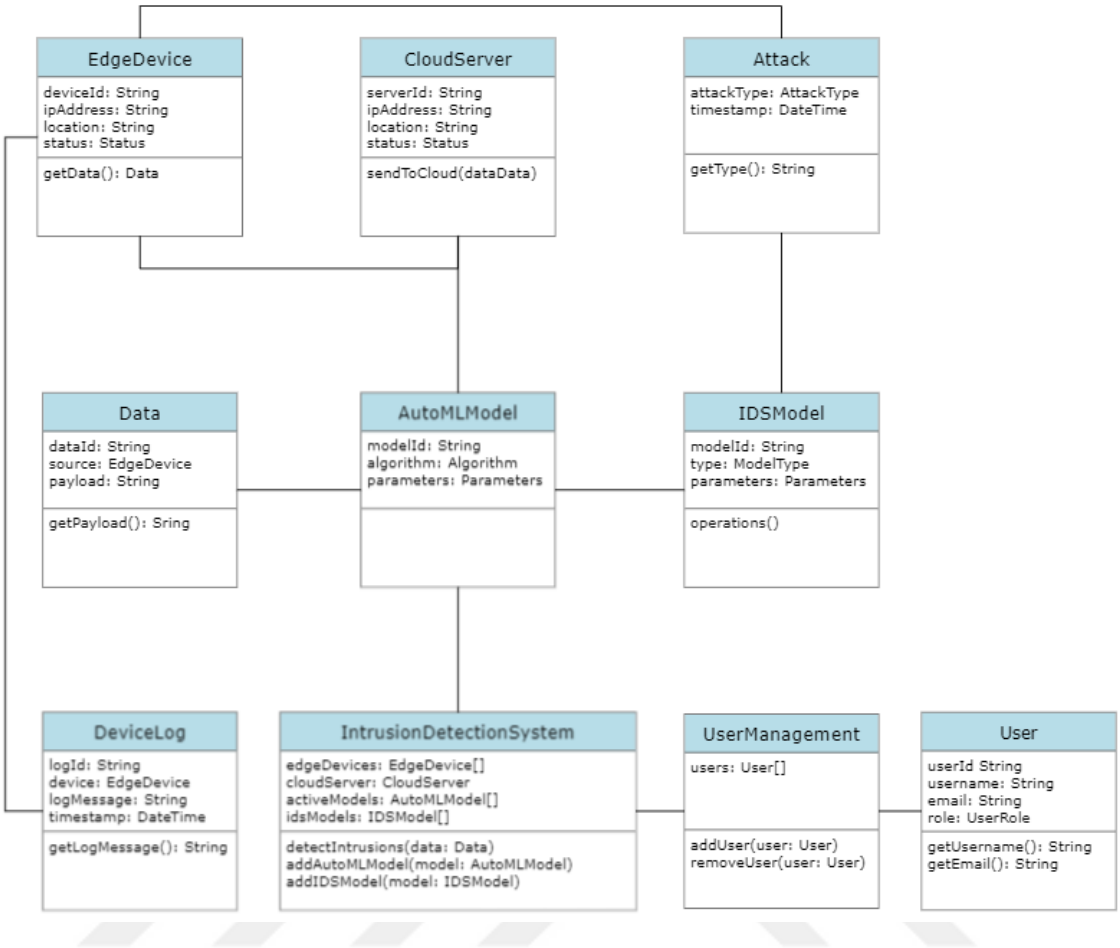
için bir araya getirilir. Bu sayede buluta iletilen veri hacmini azaltarak bant genişliğini ve işleme kaynaklarını optimize eder.

Bulut tabanlı altyapı, IDS mimarimizin merkezi işlem birimini oluşturmaktadır. Bu altyapının temeli, uç birim cihazlardan veri akışlarını almak için bir ağ geçidi görevi gören bulut sunucusudur. Bu sunucu, en uygun makine öğrenmesi algoritmalarını seçmekten, hiperparametreleri optimize etmekten ve önceden işlenmiş verilere dayalı makine öğrenmesi modellerini eğitmekten sorumlu otomatik makine öğrenmesi süreçlerini barındırmaktadır. Bulut sunucusu, eğitilen makine öğrenmesi modellerini depolamak için bir dosya ve mesajlaşma sistemi ile donatılmıştır ve uç birim cihazlarına sorunsuz bir şekilde modellerin dağıtılmasını sağlamaktadır. Geliştirilen sisteme ait akış Şekil 4.7'de gösterilmektedir.



Şekil 4.7 AutoML sistemine ait akış diyagramı

Gerçek zamanlı tehdit algılama aşamasında, eğitilmiş makine öğrenmesi modelleri uç birim cihazlarına dağıtılır. Bu modeller daha sonra gelen veri akışlarını gerçek zamanlı olarak analiz etmek için kullanılır. Tespit edilen herhangi bir anormallik, uç birim cihazında hızlı bir şekilde tanımlanır. Bu hızlı tespit, uyarıları ve bildirimleri tetikleyerek yöneticilere ve kullanıcılara zamanında yanıtlar ve önlemler almaları için bilgi sağlar. Geliştirilen sisteme ait UML sınıf diyagramı Şekil 4.8'de gösterilmektedir.



Şekil 4.8 UML sınıf diyagramı

Ölçeklenebilirlik ve uyarlanabilirlik, IDS mimarisindeki önemli konulardandır. Bulut tabanlı bileşenler yatay ölçeklenebilirlik için tasarlanmıştır ve sistemin değişen veri hacimlerini ve işleme taleplerini bu sayede sorunsuz bir şekilde karşılayabilmektedir. Sistemin doğruluğunu ve etkinliğini geliştirmek için tehdit tespitlerinin sonuçları ve model performans ölçümleri sürekli olarak toplanmakta olup bu sayede saldırı tespit sisteminin yetenekleri zaman içinde yinelemeli olarak geliştirilmektedir.

5
DENEYSEL SONUÇLAR

5.1 Endüstriyel Nesnelerin İnterneti Veri Setleri

Endüstriyel Nesnelerin İnterneti alanında saldırı tespiti, kritik altyapının güvence altına alınmasında ve birbirine bağlı endüstriyel sistemlerin esnekliğinin korunmasında önemli bir rol oynamaktadır. IIoT teknolojileri çeşitli sektörlerde devrim yaratmaya devam ettikçe, potansiyel tehditleri azaltmak ve kötü niyetli faaliyetlere karşı koruma sağlamak için sağlam saldırı tespit mekanizmalarının gerekli olduğu giderek daha belirgin hale gelmektedir. IIoT ortamlarının ortaya çıkardığı benzersiz zorlukları ele almak için araştırmacılar ve uygulayıcılar, endüstriyel ortamlara uyarlanmış saldırı tespit sistemlerini değerlendirmek ve iyileştirmek için oluşturulmuş veri setlerine güvenmektedir.

Bu bölümde IIoT'de saldırı tespiti için yaygın olarak kullanılan veri setlerinin teknik bilgileri ele alınmaktadır. Bu veri setleri, araştırmacılara endüstriyel sistemlerde karşılaşılan ağ trafiği, iletişim modelleri ve potansiyel saldırı senaryolarının çeşitli ve temsili örneklerini sağlayan çok değerli kaynaklar olarak hizmet vermektedir. Boyut, özellikler ve saldırı türleri gibi hususları kapsayan ayrıntılı özellikleri sayesinde bu veri setleri, IIoT dağıtımlarındaki anormallikleri ve potansiyel izinsiz girişleri etkili bir şekilde ayırt edebilen saldırı tespit yöntemlerinin geliştirilmesini ve doğrulanmasını sağlar. Araştırmacılar, bu veri setlerinin teknik yönlerini inceleyerek, gerçek zamanlı çalışabilen ve endüstriyel sistemlerin ölçeğini ve heterojenliğini idare edebilen saldırı tespit teknikleri geliştirmek için önemli olan IIoT ağ trafiğinin karmaşıklığı ve değişkenliği hakkında bilgi edinmektedir. Bu veri setlerinden yararlanmak, IIoT için saldırı tespit alanındaki gelişmeleri teşvik ederek kritik altyapıyı potansiyel tehditlerden proaktif olarak koruyan sofistike ve uyarlanabilir güvenlik mekanizmalarının tasarımını güçlendirir.

Aşağıdaki bölümlerde, IIoT'de saldırı tespiti için kullanılan popüler veri setlerinden bazıları, belirli özellikleri, saldırı senaryoları ve endüstriyel ortamlar için güvenlik alanını ilerletmedeki rollerinin önemi incelenmektedir. Bu veri setleri, araştırmacıların çeşitli saldırı tespit yaklaşımlarını, makine öğrenmesi algoritmalarını ve kural tabanlı yöntemleri denemeleri için temel oluşturmakta ve böyle Endüstriyel Nesnelerin İnternetinin gelişen

ortamı için saldırı tespit sistemlerinin sürekli iyileştirilmesine ve yenilenmesine katkıda bulunmaktadır.

5.1.1 WUSTL-IIoT

Bu veri seti SCADA siber güvenlik araştırması için kullanılan bir veri setidir. Veri seti SCADA ortamı kullanılarak oluşturulmuştur [60]. Oluşturulan test ortamının amacı gerçek dünyadaki endüstriyel sistemlerin benzerini elde etmektir. Test ortamı sayesinde gerçekçi siber saldırılar gerçekleştirilmesine olanak sağlanmıştır. Test ortamına karşı gerçekleştirilen saldırılar Tablo 5.1'de gösterilmektedir [60].

Tablo 5.1 Test ortamına karşı gerçekleştirilen saldırılar [60]

Saldırı Adı	Açıklama
Port Tarama	Bu saldırı, ağdaki genel SCADA protokollerini tanımlamak için kullanılır. Nmap aracı kullanılarak 1 ile 3 saniye arasında değişen aralıklarla paketler hedefe gönderilir. TCP bağlantısı tam olarak kurulmadığı için saldırının kurallar tarafından tespit edilmesi zordur.
Adres Tarama Saldırısı	Bu saldırı, ağ adresini taramak ve Modbus sunucu adresini belirlemek için kullanılır. Her sistemin yalnızca bir Modbus sunucusu vardır ve bu cihazın devre dışı bırakılması tüm SCADA sistemini çökertecektir. Bu saldırı, daha sonraki saldırılarda kullanılmak üzere Modbus sunucusunun unique adresini bulmaya çalışır.
Cihaz Tanıma Saldırısı	Bu saldırı, ağdaki SCADA Modbus slave ID'lerini numaralandırmak ve bulunan ilk slave ID'den üretici ve firmware gibi ek bilgileri toplamak için kullanılır.
Cihaz Tanıma Saldırısı (Agresif Mod)	Bu saldırı önceki saldırıya benzer. Ancak tarama, sistemde bulunan tüm slave ID'ler hakkında ek bilgilerin toplandığı anlamına gelen agresif bir mod kullanır.
Exploit	Exploit, SCADA cihazlarının bobin değerlerini okumak için kullanılır. Bobinler, motorlar, valfler ve sensörler gibi PLC tarafından kontrol edilen cihazların AÇIK/KAPALI durumunu temsil etmektedir.

Bu veri seti, çok çeşitli normal ve kötü niyetli ağ etkinlerini yakalayarak gerçek dünyadaki endüstriyel ortamları simüle etmek için özel olarak tasarlanmıştır. Saldırı tespit tekniklerini, anomali tespit algoritmalarını ve IIoT ortamlarına uyarlanmış güvenlik mekanizmalarını değerlendirmek için bir test ortamı görevi görmektedir. Veri seti,

kontrollü bir IIoT ortamından yakalanan önemli sayıda ağ akışı kaydını içermektedir. Veri seti, kaynak ve hedef IP adresleri, port numaraları, protokol türleri, akış süresi, paket ve bayt sayıları ve IIoT cihazları arasındaki iletişim modelleri gibi çeşitli ağ akışı düzeyinde özellikler içermektedir. Bu özellikler, endüstriyel sistemlerde yaygın olarak karşılaşılan ağ trafiğinin özelliklerini temsil etmektedir. Bu veri seti, IIoT sistemlerinin karşılaştığı gerçek dünya tehditlerini simüle eden çok çeşitli saldırı senaryolarını içermektedir. Saldırılar arasında, Denial-of-Service (DoS) saldırıları, Distributed Denial-of-Service (DDoS) saldırıları, keşif faaliyetleri ve endüstriyel operasyonları potansiyel olarak kesintiye uğratabilecek veya tehlikeye atabilecek diğer kötü niyetli faaliyetler yer almaktadır. Tablo 5.2'de yakalanan ağ trafiğine ilişkin istatistiksel bilgiler bulunmaktadır [60].

Tablo 5.2 Yakalanan ağ trafiğine ilişkin istatistiksel bilgiler [60]

Ölçüm	Değer
Yakalama süresi	25 saat
Veri seti büyüklüğü	627 MB
Gözlem sayısı	7.049.989
Port tarayıcı saldırılarının yüzdesi	%0,0003
Adres tarama saldırılarının yüzdesi	%0,0075
Cihaz tanımlama saldırılarının yüzdesi	%0,0001
Cihaz tanımlama asldırılarının yüzdesi (agresif mod)	%4,9309
İstismar (exploit) saldırılarının yüzdesi	%1,1312
Tüm saldırıların yüzdesi (toplam)	%6,07
Normal trafik yüzdesi	%93,93

5.1.2 X-IIoTID

Bu veri seti, saldırganların kullandıkları güncel taktikleri, teknikleri ve prosedürleri ile endüstriyel kontrol döngü cihazları (sensör, aktüatör, kontrolör vb.), edge, mobil, bulut trafiği ve faaliyetleri (MQTT, CoAP, WebSocket vb.), yeni bağlantı protokollerinin davranışları dahil olmak üzere gerçekçi IIoT sistemlerinin faaliyetlerinin simülasyonunu temsil etmektedir. Bağlantı ve cihazdan bağımsız özelliklere sahip olması, bu veri setini IIoT sistemlerinin heterojen doğası ve birlikte çalışabilirlik için uygun hale getirir. Bu özellikler ağ trafiğinden, sistem günlüklerinden, uygulama günlüklerinden, cihazın kaynaklarından (CPU, I/O, bellek vb.) ve ticari saldırı tespit sistemlerinin (OSSEC ve Zeek/Bro) günlüklerinden çıkarılmıştır.

Sınıflandırma görevi için uygun olan bu veri setinde 64 özellik ve 820.834 örnek bulunmaktadır. Bu veri seti ile net ve kesin bir saldırı sınıflandırması yapılabilmekte ve WebSocket Fuzzing, Constrained Application Protocol (CoAP) kaynak keşfi, MQTT (Message Queuing Telemetry Transport) kötü niyetli subscription, kripto-fidye yazılımı gibi yeni nesil saldırıları içermektedir. Ayrıca Modbus, WebSocket, MQTT, TCP (Transmission Control Protocol), ARP (Address Resolution Protocol), HTTP (Hypertext Transfer Protocol), SSH (Secure Shell), DNS (Domain Name System), SMTP (Simple Mail Transfer Protocol), UDP (User Datagram Protocol) gibi protokolleri de kapsamaktadır. IIoT ve IDS için oluşturulan diğer veri setlerine göre daha kapsamlı bir veri setidir. Network trafiğinden elde edilen özellikler Tablo 5.3'te, Host üzerinden elde edilen özellikler ise Tablo 5.4'te gösterilmektedir [54].

Tablo 5.3 Ağ trafiğinden elde edilen özellikler

Özellik	Veri Tipi	Tanım
Ts	Kesikli	Zaman bilgisi
Src_IP	Kesikli	Kaynak IP adresi
Des_IP	Kesikli	Hedef IP adresi
Src_Port	Kesikli	Kaynak port numarası
Des_Port	Kesikli	Hedef port numarası
Protocol	Kesikli	Bağlantı protokolü

Tablo 5.3 Ağ trafiğinden elde edilen özellikler (devamı)

Özellik	Veri Tipi	Tanım
Service	Kesikli	Hedef portta çalışan uygulama protokolü
Duration	Sürekli	İlk ve son paket arasında geçen süre
Src_bytes	Sürekli	Kaynaktan hedefe giden bayt sayısı
Des_bytes	Sürekli	Hedeften kaynağa giden bayt sayısı
Missed_byte	Sürekli	Kayıp bayt sayısı
Src_pkts	Sürekli	Gönderilen paket sayısı
Des_pkts	Sürekli	Alınan paket sayısı
Src_IP_bytes	Sürekli	IP header toplam uzunluk alanındaki gönderilen bayt sayısı
Des_IP_bytes	Sürekli	IP header toplam uzunluk alanındaki alınan bayt sayısı
Conn_state	Kesikli	Bağlantı Durumu (1 tamamlandı, 2 bekleme, 3 parçalı)
Total_bytes	Sürekli	Kaynak ve hedef arasında iletilen toplam bayt sayısı
Byte_rate	Sürekli	Saniye başına toplam bayt sayısı
Total_Pkts	Sürekli	Kaynak ve hedef arasında iletilen toplam paket sayısı
Pkts_rate	Sürekli	Saniye başına toplam paket sayısı
orig_bytes_ratio	Sürekli	Gönderilen bayt sayısının toplam bayt sayısına oranı
resp_bytes_ratio	Sürekli	Alınan bayt sayısının toplam bayt sayısına oranı
orig_pkts_ratio	Sürekli	Gönderilen paket sayısının toplam paket sayısına oranı
resp_pkts_ratio	Sürkeli	Alınan paket sayısının toplam paket sayısına oranı
SYN	Kesikli	Eğer bağlantıda SYN flag olan paket varsa (0-1)
SYN-ACK	Kesikli	Eğer bağlantıda SYN-ACK flag olan paket varsa (0-1)
Pure ACK	Kesikli	Eğer bağlantıda Pure ACK flag olan paket varsa (0-1)

Tablo 5.3 Ağ trafiğinden elde edilen özellikler (devamı)

Özellik	Veri Tipi	Tanım
Packet with payload	Kesikli	Eğer bağlantıda payload olan paket varsa (0-1)
FIN or RST	Kesikli	Eğer bağlantıda FIN ya da RST flan olan paket varsa (0-1)
Bad checksum	Kesikli	Eğer bağlantıda bad checksum olan paket varsa (0-1)
SYN with RST	Kesikli	Eğer bağlantıda hem SYN hem RST flag olan paket varsa (0-1)

Tablo 5.4 Host üzerinden elde edilen özellikler

Özellik	Veri Tipi	Tanım
Avg_user_time	Sürekli	Son 10 saniyedeki kullanıcı zamanının ortalaması (işlem süresi)
Std_user_time	Sürekli	Son 10 saniyedeki kullanıcı zamanının standart sapması
Avg_nice_time	Sürekli	Sürecin önceliğini tanımlamak için kullanılan zamanın ortalaması
Std_nice_time	Sürekli	Sürecin önceliğini tanımlamak için kullanılan zamanın standart sapması
Avg_system_time	Sürekli	Son 10 saniyedeki sistem zamanının ortalaması (işlemcinin işletim sistemi işlevlerinde çalıştığı süre)
Std_system_time	Sürekli	Son 10 saniyedeki sistem zamanının standart sapması (işlemcinin işletim sistemi işlevlerinde çalıştığı süre)
Avg_IO_wait_time	Sürekli	Son 10 saniyedeki G/Ç bekleme süresinin ortalaması (CPU'nun G/Ç işlemi için boşta beklediği toplam süre)
Std_IO_wait_time	Sürekli	Son 10 saniyedeki G/Ç bekleme süresinin standart sapması (CPU'nun G/Ç işlemi için boşta beklediği toplam süre)
Avg_idle_time	Sürekli	Ortalama boşta kalma süresi (CPU'nun meşgul olmadığı ve son 10 saniye içinde bekleyen bir disk G/Ç isteği olmadığı toplam süre)

Tablo 5.4 Host üzerinden elde edilen özellikler (devamı)

Özellik	Veri Tipi	Tanım
Std_idle_time	Sürekli	Boşta kalma süresinin standart sapması (CPU'nun meşgul olmadığı ve son 10 saniye içinde bekleyen bir disk G/Ç isteği olmadığı toplam süre)
Avg_tps	Sürekli	Son 10 saniye içinde cihaza gönderilen saniye başına transfer isteği sayısının ortalaması
Std_tps	Sürekli	Son 10 saniye içinde cihaza gönderilen saniye başına transfer isteği sayısının standart sapması
Avg_rtps	Sürekli	Son 10 saniye içinde cihaza gönderilen saniye başına okuma işlemi sayısının ortalaması
Std_rtps	Sürekli	Son 10 saniye içinde cihaza gönderilen saniye başına okuma işlemi sayısının standart sapması
Avg_wtps	Sürekli	Son 10 saniye içinde cihaza gönderilen saniye başına yazma işlemi sayısının ortalaması
Std_wtps	Sürekli	Son 10 saniye içinde cihaza gönderilen saniye başına yazma işlemi sayısının standart sapması
Avg_ldavg_l	Sürekli	Pencere zaman boyutu 10 saniye içinde son dakika boyunca ortalama sistem yükünün ortalaması
Std_ldavg_l	Sürekli	Pencere zaman boyutu 10 saniye içinde son dakika boyunca ortalama sistem yükünün standart sapması
Avg_Kbmemused	Sürekli	Son 10 saniyede kilobayt cinsinden kullanılan bellek ortalaması
Std_Kbmemused	Sürekli	Son 10 saniyede kilobayt cinsinden kullanılan bellek standart sapması
Avg_num_proc/s	Sürekli	Son 10 saniye içinde saniye başına oluşturulan görev sayısının ortalaması
Std_num_proc/s	Sürekli	son 10 saniye içinde saniye başına oluşturulan görev sayısının standart sapması

Tablo 5.4 Host üzeriden elde edilen özellikler (devamı)

Özellik	Veri Tipi	Tanım
Avg_num_swch/s	Sürekli	Son 10 saniye içinde saniye başına bağlam değiştirme sayısının ortalaması
Std_num_swch/s	Sürekli	Son 10 saniye içinde saniye başına bağlam değiştirme sayısının standart sapması
Anomaly_Alert	Kesikli	Zeek üzerinden gelen alarm kontrolü (0-1)
OSEEC_Alert	Kesikli	OSEEC üzerinden gelen alarm kontrolü (0-1)
Alert_level	Kesikli	OSSEC alarm seviyesi
R_W_physical	Kesikli	Fiziksel sürece yönelik okuma/yazma faaliyeti kontrolü (0-1)
File_act	Kesikli	Dosya aktivitesi kontrolü (0-1)
Proc_act	Kesikli	Proses çalışma kontrolü (0-1)
Is_privileged	Kesikli	Ayrıcalıklı faaliyet (login, proses ya da dosya faaliyet) kontrolü (0-1)
Login_attmp	Kesikli	Login denemesi kontrolü (0-1)
Succ_login	Kesikli	Başarılı login kontrolü (0-1)

5.1.3 KDD99

KDD99 (KddCup99) veri seti, çeşitli ağ saldırılar ve normal etkinliklerle bil bilgisayar ağı ortamını simüle eden bir ağ trafiği veri setidir. Bu veri seti 1999 yılında düzenlenen Üçüncü Uluslararası Bilgi Keşfi ve Veri Madenciliği Araçları Yarışması'nın (Third International Knowlegde Discovery and Data Mining Tools Competition, KDD Cup) bir parçası olarak oluşturulmuştur. Veri seti, bir ABD Hava Kuvvetleri ağından yakalanan ham ağ trafiği verilerini içeren DARPA Saldırı Tespit Değerlendirme Programı 1998 (Intrusion Detection Evaluation Program, IDEA) verilerinden oluşturulmuştur. KDD99 veri seti, bir eğitim seti ve bir test setine bölünmüş yaklaşık 4.9 milyon ağ bağlantısından oluşmaktadır. Eğitim seti yaklaşık 2.4 milyon bağlantı içerirken, test seti yaklaşık 2.5 milyon bağlantı içermektedir. Veri seti her bir ağ bağlantısı için toplam 41 özellik içermektedir. Bu özellikler arasında protokol türü, hizmet, kaynak ve hedef adresler ve

bağlantı noktaları, bağlantı süresi, aktarılan bayt ve paket sayısı gibi ağ trafiğiyle ilgili özellikler hakkında bilgiler yer almaktadır. Veri seti, dört ana kategoride gruplandırılan çok sayıda ağ saldırısı türünü içermektedir: Denial of Service, Probe, Remote to Local ve User to Root. Veri setindeki saldırılar, ağ güvenliğinde yaygın olarak karşılaşılan çeşitli saldırı girişimlerini ve senaryoları temsil etmektedir.

5.1.4 NSL-KDD

NSL-KDD veri seti, saldırı tespit araştırması amacıyla oluşturulmuş bir ağ trafiği veri setidir. KDD99 veri setindeki bazı eksikliklerin üstesinden gelmek için geliştirilen bir veri setidir. NSL-KDD veri seti, bir eğitim seti ve bir test setine bölünmüş yaklaşık 4 milyon ağ bağlantısı içermektedir. Eğitim seti 125.973 bağlantıdan oluşurken, test seti 25.544 bağlantı içermektedir. Veri seti, tıpkı KDD99 veri setinde olduğu gibi, her bir ağ bağlantısı için toplam 41 özellik içermektedir. KDD99 veri seti, saldırı tespit değerlendirmelerinin doğruluğunu etkileyen fazlalık ve yinelenen kayıtlar içermektedir. Buna karşılık, NSL-KDD veri seti, yinelenen örnekleri kaldırmak ve saldırı tespit tekniklerinin tarafsız değerlendirmesi için daha uygun hale getirek için dikkatli bir veri ön işleme aşamasından geçmiştir. KDD99'da yinelenen kayıtların yaygınlığı nedeniyle, bu veri seti üzerinde eğitilen saldırı tespit sistemleri potansiyel olarak gereksiz kalıplarda overfitting oluşturup hatalı performans ölçümlerine yol açabilmektedir.

5.2 Performans Değerlendirme Metrikleri

Saldırı tespit alanında, tespit sistemlerinin etkinliğini ve verimliliğini değerlendirmek, kritik ağların ve sistemlerin güvenliğini ve esnekliğini sağlamak için önemlidir. Performans değerlendirme metrikleri, saldırı tespit tekniklerini doğruluğunu, sağlamlığını ve genel performansını nicel olarak ölçmede önemli bir rol oynamaktadır. Karmaşıklık matrisleri (Confusion Matrix), modelin hedef kategorilerini doğru tahmin etme yeteneğinin görsel bir gösterimini sunarken, sınıflandırma raporları precision, recall, F1-skor ve destek metriklerini kapsayan modelin performansının daha derinlemesine bir değerlendirmesini sunar. Karmaşıklık matrisi, makine öğrenmesinde yaygın olarak kullanılan bir değerlendirme metriğidir. Bir sınıflandırma modelinin etkinliğinin değerlendirilmesini kolaylaştırır. Tahmin edilen sonuçlara karşı gerçek sonuçların dağılımını gösterir. Karmaşıklık matrisi, dört temel bileşenden oluşur: Doğru Pozitif (TP), Yanlış Pozitif (FP), Yanlış Negatif (FN) ve Doğru Negatif (TN). Doğru Pozitif, pozitif olarak doğru sınıflandırılmış durumları; Yanlış Pozitif, pozitif diye yanlış

sınıflandırılmış durumları; Yanlış Negatif, negatif diye yanlış sınıflandırılmış durumları; Doğru Negatif ise negatif olarak doğru sınıflandırılmış durumları ifade eder.

Makine öğrenmesiyle yapılan saldırı tespiti işlemlerinde, ağ trafiği ya da olayları normal ya da saldırı kategorilerine doğru bir şekilde ayırmak hedeflenir. Bu süreçte, bir sınıflandırma modelinin başarımını ölçmek için kesinlik (precision), duyarlılık (recall), F1 skoru ve destek gibi metrikler kullanılır.

- Kesinlik: Kesinlik, doğru olarak tanımlanan pozitif durumların, pozitif olarak belirlenen tüm durumlar arasındaki oranını ifade eder. Bu, bir modelin pozitif bir sınıfı ne derece doğru tahmin ettiğini gösteren bir doğruluk ölçütüdür. Yüksek bir kesinlik skoru, az sayıda yanlış pozitif sonuç veren ve pozitif durumları doğru bir şekilde belirlemede etkili bir modeli işaret eder.
- Duyarlılık: Duyarlılık, gerçekte pozitif olan örneklerin, model tarafından ne kadar doğru bir şekilde pozitif olarak saptandığının oranını belirtir. Bu değerlendirme ölçütü güvenlik ihlallerinin ne kadar iyi tespit edildiğini gösterir. Yüksek bir duyarlılık değeri, modelin yanlış negatif sonuçların sayısının az olduğunu ve olumsuz durumları iyi bir şekilde belirlemedee başarılı olduğunu ifade eder.
- F1-skoru: F1-skoru, kesinlik ve duyarlılığın harmonik ortalamasını alarak iki ölçüt arasında bir denge sağlar ve bu sayede farklı modellerin performanslarını karşılaştırmak için etkin bir değerlendirme ölçütüdür. Yüksek bir F1-skoru, hem kesinlik hem de duyarlılık açısından dengeli bir performans sergileyen modeli gösterir.
- Destek: Destek, bir test veri setindeki her sınıfın örnek sayısını ifade eder ve bir modelin yeni verilere ne kadar iyi genelleme yaptığının değerlendirilmesinde kullanılır.

Hata (karışıklık) matrisleri ve sınıflandırma raporlarından çıkarılan sonuçlar, otomatik makine öğrenmesi sisteminin performansını ölçmede ve ileride yapılacak iyileştirmeler için yol göstermede kritik bir öneme sahiptir. Denklem (5.1), (5.2) ve (5.3), sırasıyla duyarlılık, doğruluk ve kesinlik ölçütlerini tanımlamaktadır.

$$Duyarlılık = \frac{TP}{TP+FN} \tag{5.1}$$

$$Doğruluk = \frac{TP+FN}{TN+TP+FN+FP} \tag{5.2}$$

$$Kesinlik = \frac{TP}{TP+FP} \tag{5.3}$$

5.3 Deneysel Tasarım ve Metodoloji

Bu çalışmanın amacı, IIoT'de saldırı tespitinin tüm sürecini otomatikleştirebilen ve makine öğrenmesi konusunda çok az deneyimi olan veya hiç deneyimi olmayan kişiler için erişilebilir hale getiren bir makine öğrenmesi işlem hattı geliştirmektir. Uygun parametre değerlerini belirlemek için birden fazla teknik kullanılarak, saldırı tespiti için en verimli sınıflandırma algoritmalarının belirlenmesine odaklanılmaktadır. Bunu başarmak için modeller, en yüksek başarı oranına sahip yaklaşımı belirlemek amacıyla girdi veri seti kullanılarak eğitilir ve test edilir. Endüstriyel IoT sistemlerindeki saldırı tespit etmek için, makine öğrenmesi konusunda önemli bilgi veya teknik yeterlilik gerektirmeyen, basit ve kullanıcı dostu çözüm sunulmalıdır. Bu, Endüstriyel IoT sistemlerinin genel etkinliğini ve güvenliğini arttıracaktır.

Bir saldırı tespit sisteminin geliştirilmesi, bir AutoML işlem hattı kullanılarak otomatikleştirilebilir. İşlem hattı, ön işleme, özellik seçimi, hiper parametre ayarlama ve model eğitimi dahil olmak üzere tüm önemli işlemleri manuel gözlem gerektirmeden tamamlayabilir. İşlem hattı daha sonra, verilen veri seti için özel olarak oluşturulmuş bir model sağlayabilir ve bu da saldırıların tanınması ve sınıflandırılmasında yüksek doğruluk sağlar.

5.3.1 İmputasyon

Bu çalışmada, kayıp verileri ele almak için üç farklı imputasyon yöntemi bulunmaktadır: ortalama imputasyon, medyan imputasyon ve en sık imputasyon. En etkili stratejiyi seçmek için veri setleri bu yöntemlerin her birine göre test edilmekte ve sonuçlar karşılaştırılmaktadır.

Ortalama imputasyon yöntemi, eksik veri noktaları için bir sütundaki eksik olmayan değerlerin ortalamasını kullanır. Bu yöntemin kullanımı kolay olsa da kayıp değerler rastgele dağılmamışsa yanlılığa neden olabilir. Medyan imputasyon yöntemi kullanılırken kayıp değerlerin yerine aynı sütundaki kayıp olmayan değerlerin medyanı kullanılır. Aykırı değerlere karşı savunmasız olan ortalama imputasyona kıyasla medyan imputasyon, verilerin merkezi eğiliminin daha güvenilir bir şekilde değerlendirilmesini sağlar. Bununla birlikte, kayıp değerler rastgele kayıp değilse, bu strateji de yanlılığa neden olmaya devam edebilir. En yaygın imputasyon yöntemi, eksik verilerin yerine aynı sütundaki en sık rastlanan değerin konulmasını içerir. Bu yöntem kategorik verilerle

çalışırken kullanışlıdır, ancak eksik değerler daha az sıklıkta görülen bir kategoriyi temsil ediyorsa bilgi kaybına yol açabilir.

Çalışma kapsamında geliştirilen AutoML çerçevesi, en etkili yaklaşım yöntemini belirlemek için her bir imputasyon tekniğini kullandıktan sonra sonuçları karşılaştırmaktadır. Çerçeve, her bir imputasyon yönteminin performansını doğruluk, kesinlik, duyarlılık ve F1-skoru kriterleriyle değerlendirir.

5.3.2 Ölçeklendirme

Çeşitli makine öğrenmesi algoritmaları, farklı ölçeklerdeki girdi özelliklerinin modelin performansını aşırı etkileme potansiyeline sahiptir. Ölçeklendirme süreci, bir veri setindeki özellikleri standart bir ölçeğe dönüştürerek, algoritmanın yakınsama sürecini hızlandırmak ve tahmin doğruluğunu iyileştirmek için kullanılır. Özellikle, ölçeklendirme, bazı algoritmaların girdi verilerinin boyutlarına duyarlı olduğu durumlarda, modelin performansını ve tahminlerinin hassasiyetini artırabilir. Mesafe tabanlı algoritmalar olan k-en yakın komşu, destek vektör makineleri ve yapay sinir ağları gibi sistemler, veri noktaları arasındaki benzerliği ölçerken mesafe hesaplamalarına dayanır. Eğer özellikler arasındaki ölçek farklılıkları varsa, daha büyük ölçekli özellikler mesafe hesaplamasını baskın hale getirir, bu da hatalı tahminlere ve modelin suboptimal performansına sebep olabilir.

Makine öğrenmesi algoritmalarının performansı, kullanılan veri setlerinin ölçeklenmesi ile doğrudan ilişkilidir. Ölçeklendirme, özellikle yinelemeli optimizasyon tekniklerinin yakınsama hızını etkileyen kritik bir ön işleme adımıdır. İteratif metodlar, düzgün ölçeklenmiş veri setleri ile daha hızlı ve etkin bir şekilde optimum çözümlere ulaşabilir. Bu sürecin hızı, algoritmanın verimliliğini ve eğitim süresini doğrudan etkiler. Ölçeklendirme işlemi, veri setinin özelliklerinin ve algoritmanın ihtiyaçlarına uygun bir şekilde seçilmelidir. Makine öğrenmesi uygulamalarında kullanılan çeşitli ölçekleyiciler ve bunların özellikleri şöyledir:

- Min-Max Ölçekleyici: Min-Max ölçeklendirme, veri özelliklerini 0 ile 1 arasında bir aralığa dönüştüren bir normalizasyon tekniğidir. Bu, özellikle farklı özellikler arasında büyük değer farklılıkları olduğunda ve bazı özelliklerin diğerlerinden daha dominant olabileceği durumlarda yararlıdır. Min-Max ölçeklendirme yöntemi, tüm özellikleri aynı ölçekte standardize ederek, makine öğrenmesi modellerinin veri setinden daha etkili bir şekilde öğrenmesini sağlar. Yöntemin

basit uygulanışı ve veri yapısını koruma avantajları bulunmaktadır. Ancak, bu yöntem aykırı değerlerin varlığından etkilenebilir ve aykırı değerler ölçeklendirmeyi bozabilir. Bu durum, modelin genelleme yeteneği üzerinde olumsuz etkilere neden olabilir. Min-Max ölçeklendirme fonksiyonu Denklem (5.4)'te gösterilmektedir.

$$z_i = \frac{x_i - \min(x)}{\max(x) - \min(x)} \tag{5.4}$$

- Standart Ölçekleyici: Standart ölçeklendirme, her bir özelliği, ortalama değerinin 0 ve standart sapmanın 1 olacak şekilde dönüştürür. Bu yöntem, özelliklerin normal bir dağılım göstermesini sağlar ve makine öğrenmesi algoritmalarının veri üzerinden öğrenmesini kolaylaştırır. Standart ölçeklendirme, modelin yorumlanabilirliğini artırır ve Min-Max ölçeklemeye göre aykırı değerlere karşı daha dirençlidir, ancak çok ekstrem aykırı değerler varsa bu yöntem de etkilenebilir. Standart ölçeklendirme fonksiyonu Denklem (5.5)'te gösterilmektedir.

$$z = \frac{x_i - \mu}{\sigma} \tag{5.5}$$

- Robust Ölçekleyici: Robust ölçeklendirme, veri setinin medyanını ve çeyrekler arası aralığını kullanarak özellikleri ölçeklendirir ve bu sayede aykırı değerlerin özellikler üzerindeki etkisini minimize eder. Aykırı değerlerin model üzerindeki etkisini azaltma avantajına sahip olan bu yöntem, özellikle aykırı değerlerin baskın olduğu veri setlerinde kullanışlıdır. Robust ölçeklendirme, verilerdeki aykırı değerlere karşı sağlamlığı ile öne çıkar ve bu tür verilerde daha güvenilir sonuçlar üretebilir. Ancak, bu yöntemin uygulanması diğer ölçeklendirme tekniklerine göre daha karmaşık olabilir ve veri setinin orijinal yapısını koruma konusunda Min-Max veya Standart ölçekleme kadar etkili olmayabilir. Robust ölçekleme fonksiyonu Denklem (5.6)'da gösterilmektedir.

$$X_{yeni} = \frac{X - X_{medyan}}{IQR} \tag{5.6}$$

5.3.3 Özellik Seçimi

Özellik seçimi adımı için kullanılan yaklaşım, diğer tüm sınıflandırma modellerine kıyasla üstün performans sergileyen bir tekniktir. SHAP (SHapley Additive exPlanations) yöntemi, makine öğrenmesi model tahminlerinin açıklanabilirliğini artırmak için, her bir özelliğin Shapley değerlerini hesaplayarak kullanılır. Öte yandan, genetik algoritma,

geniş bir özellik havuzundan en etkili özellikleri seçebilmek için bir optimizasyon aracı olarak devreye girer. SHAP ve genetik algoritmanın entegrasyonu, özellik seçim sürecini geliştirmekte ve optimizasyonu daha verimli kılmaktadır. Her bir özellik için SHAP değerleri belirlendikten sonra genetik algoritma kullanılarak en yüksek öneme sahip özellikler seçilir. Genetik algoritma, büyük boyutlu verilerin hesaplanmasının üstesinden gelebildiği ve en önemli özellikleri etkili bir şekilde keşfedebildiği için, bu yaklaşımda özellik seçim süreci iyileştirilmiştir.

Önerilen metodoloji, her bir özelliğin önemini değerlendirmek için Shapley değerlerini kullanmakta ve bunun için bir eşik değeri belirlemektedir. Çalışmada eşik değerini belirlemek için iki farklı yöntem kullanılmıştır. İlk yaklaşımda sabit bir eşik değeri seçilmiştir. Sonuç olarak, herhangi bir sınıflandırma algoritması ve veri seti için genel bir eşik değeri elde edilebilmektedir. İkinci yaklaşım ise ortalama mutlak Shapley değerinin hesaplanmasını ve eşik değeri olarak kullanılmasını gerektirmiştir. Özellik önemi için Shapley tabanlı bir ölçü, ortalama mutlak Shapley değeridir. Pozitif veya negatif olabilen Shapley değerleri, her bir özelliğin modelin tahminine katkısını temsil etmektedir. Her bir özelliğin Shapley değerinin mutlak değerinin yanı sıra veri setindeki tüm örneklerin ortalama değeri de hesaplanarak ortalama mutlak Shapley değeri elde edilir. Shapley değerlerinin büyüklüğü ve yönü dikkate alınarak, her bir özelliğin öneminin genel bir ölçüsü sağlanır. Önemi eşiğin altında olan herhangi bir özellik genetik algoritmada kullanılmaz. Özellik seçimi, ilgisiz, gereksiz veya gürültülü özellikleri belirleyip kaldırarak veri setinin boyutunu azaltabilir ve modeli daha sağlam ve yorumlanabilir hale getirebilir. Bu da daha iyi genelleme performansı, daha az overfitting ve daha hızlı eğitim süreleri ile sonuçlanır.

Makine öğrenmesi algoritmalarının en iyi sonuçları verebilmesi için, iyi düzenlenmiş ve yapılandırılmış bir veri setine ihtiyaç vardır. Bir modeli eğitmeden önce verilerin ön işlemesini yapmak, modelin doğruluğunu ve performansını iyileştirmenin kritik bir adımıdır. Bu çalışma kapsamında geliştirilen AutoML çerçevesinin ön işleme aşaması, üç adım içeren bir işlem hattından oluşmaktadır: imputasyon, ölçeklendirme ve özellik seçimi. İşlem hattı, eksik verileri ele almak ve özellikleri makine öğrenmesi algoritmaları için daha uygun hale getirmek üzere tasarlanmıştır. Ön işleme işlem hattında, her adımın çıktısı bir sonraki adım için girdi görevi görerek hattı otomatik hale getirmektedir. Bu otomatikleştirilmiş yapı, manuel müdahale ihtiyacını ortadan kaldırarak tutarlılığı sağlar ve insan hatasını riskini azaltır. Otomatikleştirilmiş işlem hattı, zaman ve kaynak

tasarrufu sağlar ve makine öğrenmesi için veri hazırlama sürecini daha verimli hale getirir. Otomatikleştirilmiş ön işleme hattının adımları Şekil 5.1'de gösterilmektedir.

Şekil 5.1 Otomatikleştirilmiş ön işleme adımları

Çoğu algoritma eksik değerlerle iyi çalışamadığından, eksik verilerin ele alınması makine öğrenmesinde yaygın bir zorluktur. Bu sorunu ele almak için, işlem hattımızdaki ilk adım, eksik değerleri tahmini değerlerle değiştiren imputasyondur. Bu adımda üç imputasyon yöntemi uygulanmakta olup her bir imputasyonun sonucu farklı makine öğrenmesi algoritmalarında değerlendirilmektedir. Bu şekilde, veri seti üzerinde hangi imputasyon yönteminin hangi makine öğrenmesi yöntemiyle en iyi sonucu verdiği belirlenebilmektedir.

İşlem zincirimizin bir sonraki aşaması özelliklerin ölçeklendirilmesidir. İmputasyon aşamasından elde edilen en verimli sonuçlar ölçeklendirme işlemi için temel teşkil eder. Makine öğrenimi algoritmalarının birçoğu, özelliklerin ölçeklendirilmesine hassas olduğundan, bu işlem ön işleme safhasında kritik bir rol oynamaktadır. Ölçeklendirme, farklı ölçeklerdeki özellikleri standart bir ölçeğe dönüştürerek, makine öğrenimi algoritmasının bu özellikleri etkin bir şekilde işlemesini sağlar.

İşlem akışımızın son evresi ise, bir makine öğrenimi modelinde kullanılacak özelliklerin bir alt kümesini belirleyen özellik seçimidir. Özellik seçimi, modelin karmaşıklığını düşürür, aşırı uyuma (overfitting) karşı korur ve modelin anlaşılırlığını artırır. Bu aşamada, Shapley değerlerini ve genetik algoritmaları bir araya getiren yenilikçi bir hibrit yöntem uygulanmıştır. Shapley değerleri ile özellik seçimi Algoritma 5.1'de gösterilmiştir.

Algoritma 5.1 Shapley Değerleri ile Özellik Seçimi
Her özellik için Shapley Değerlerini saklamak üzere boş bir sözlük oluştur. 'S' özelliklerinin her bir alt kümesi için:

‘S’ içindeki her bir ‘i’ özelliğinin marjinal katkısını hesapla:

‘X’ girdisinden ‘S’ içindeki özellikleri seç ve kalan özellikleri X_degil_S olarak ayır.

Modeling hem X_S hem de X_degil_S üzerindeki tahminlerini hesapla.

‘S’ içinde ‘i’ özelliği olan ve olmayan tahminler arasındaki farkı hesapla.

‘X’ içindeki tüm örnekler üzerindeki farkların ortalamasını al

Her bir ‘i’ özelliği için:

Shapley_degeri_i = ‘i’ içeren marjinal katkıların toplamı / ‘i’ içeren alt kümelerin toplam sayısı

Shapley_degeri_i’yi ‘i’ anahtarı ile sözlükte sakla.

Shapley değerleri kullanılarak özelliklerin önemi belirlendikten sonra, belirlenen eşik değerinin üzerinde kalan özellikler genetik algoritma için giriş olarak kullanılır. Genetik algoritma, belirli bir durdurma kriterine ulaşana kadar iterasyonları sürdürür. Seçilen özellik sayısı azaltıldıkça, genetik algoritma daha hızlı işler ve potansiyel olarak yüksek performanslı özellikler ortaya çıkarabilir. Bu yaklaşım, modelimizde, daha etkili bir özellik seçimi için Shapley değerlerini bir filtre olarak kullanmamızın temelini oluşturur. Genetik algoritma ile özellik seçimi Algoritma 5.2’de gösterilmiştir. Shapley değerleri ve genetik algoritma kullanarak hibrit özellik seçimi yaklaşımımızın akış diyagramı Şekil 5.2’de gösterilmiştir.

Algoritma 5.2 Genetik Algoritma ile Özellik Seçimi

Popülasyonu başlatma: Her bireyin veri setindeki toplam özellik sayısına eşit uzunlukta bir ikili özellik vektörü olduğu popülasyon oluşturma

Başlangıç popülasyonunun uygunluğunu değerlendirme

Her nesil için i=1 to maxGenerations

Bir sonraki nesil için ebeveny seçme

Çaprazlama ve mutasyon kullanarak yavrular oluşturma

Yavruların uygunluğunu değerlendirme

En az uygun olan bireyleri yavrularla değiştirme

Çözüm olarak en iyi bireyi seçme

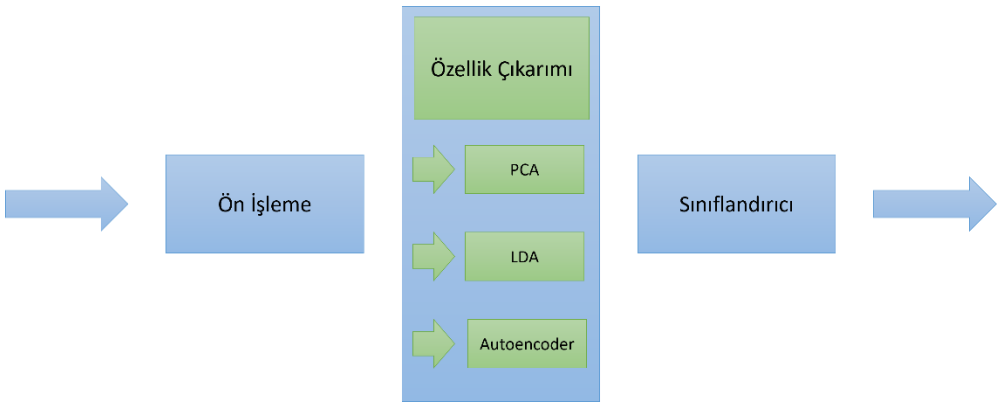
Şekil 5.2 Özellik seçimi akış diyagramı

5.3.4 Özellik Çıkarımı

Otomatik makine öğrenmesi kapsamında önerdiğimiz model, farklı boyut indirgeme yöntemlerinin farklı parametrelerle test edilerek en uygun yöntemin otomatik olarak belirlenmesini içermektedir. Geliştirilen sistemde kullanılan PCA, LDA ve Autoencoder ile boyut indirgeme yapılmakta, her bir yöntem için parametre optimizasyonu otomatik olarak uygulanmakta ve en iyi yöntem ve parametre bilgisi çıktı olarak bir sonraki modüle aktarılmaktadır.

Özellik çıkarımı yöntemlerinin hiperparametre optimizasyonunda, hiperparametreler için belirli aralıklar problem hakkındaki ön bilgilere dayalı olarak tanımlanmaktadır. Bu aralıklar daha sonra hiperparametreler için olası değerlerin listesini oluşturmak üzere örneklenir. Özellik çıkarımındaki en uygun hiperparametrelerin bulunabilmesi için Grid Search yöntemi kullanılmaktadır. Belirtilen aralıklardaki tüm değer kombinasyonları

eğitilmekte ve performans ölçütlerine göre belirlenen en iyi kombinasyon seçilmektedir. Otomatik özellik çıkarma modülü için işlem hattı Şekil 5.3'te gösterilmektedir.



Şekil 5.3 Otomatikleştirilmiş özellik çıkarma işlem hattı

Bu modül, endüstriyel nesnelerin internet için saldırı tespit sistemlerinin eğitiminde/testinde kullanılabilecek makine öğrenmesi tabanlı bir otomatikleştirilmiş özellik çıkarma modülüdür. Modülde, boyut azaltma teknikleri için hiperparametre optimizasyonu gerçekleştirilmektedir. Modül, her veri seti için PCA ve LDA yöntemleri ile farklı sayıda bileşen denemekte ve çıkarılan özellikleri sınıflandırıcılara göndererek farklı makine öğrenmesi yöntemlerindeki doğrulukları hesaplamaktadır. Her adımda denenecek bileşen sayısını veri setindeki toplam girdi sayısına göre belirlemektedir. PCA ve LDA için tüm iterasyonlar tamamlandıktan sonra Autoencoder için hiperparametre optimizasyonuna başlanmaktadır. Bu aşamada gizli katmanlardaki düğüm sayıları değiştirilerek ve Autoencoder'in aktivasyon ve kayıp fonksiyonları ile farklı darboğaz düğüm sayıları test edilerek özellik çıkarımı yapılmaktadır.

Özellik çıkarımının değerlendirme aşamasında şu alanlarda performans iyileştirmeleri gözlemlenmektedir: (i) tespit oranının (5.7) maksimize edilmesi (detection rate), (ii) doğruluğun (5.98) maksimize edilmesi (accuracy), (iii) yanlış alarm oranının (5.9) minimize edilmesi (false alarm rate), (iv) eğitim/test sürelerinin minimize edilmesi (GAIN). Özellik çıkarımı modülünün performansı, saldırı tespit sistemlerinde yaygın olarak kullanılan aşağıdaki metriklerle değerlendirilmektedir [61] [62].

$$Tespit\ Oranı = \frac{TP}{TP+FN} \tag{5.7}$$

$$Doğruluk = \frac{TP+FN}{TN+TP+FN+FP} \tag{5.8}$$

$$Yanlış\ Alarm\ Oranı = \frac{FP}{FP+TN} \tag{5.9}$$

5.3.5 Hiperparametre Optimizasyonu

Makine öğrenmesi, eğitim verilerini girdi olarak alarak tahminler ve kararlar vermek için model oluşturmaya odaklanmaktadır. Hem denetimli hem de denetimsiz öğrenme algoritmaları için eğitim sürecinden önce parametre değerlerinin ayarlanması gerektiği makine öğrenmesinde sık karşılaşılan bir sorundur. Bu parametrelere hiperparametreler denir. Hiperparametreler, belirli bir öğrenme algoritmasının öğrenme oranı, çekirdek parametreleri, ağ mimarisi gibi birçok sonucunu yapılandırmak için kullanılır. Model oluşturulurken yüksek başarı oranına sahip sonuçlar elde etmek için hiperparametreler optimize edilmiş bir şekilde verilmelidir. Hiperparametre optimizasyonu, makine öğrenmesi sürecinde bu parametrelerin dikkatli bir şekilde değerlendirilmesini ve seçilmesini gerektiren önemli bir adımdır. Hiperparametrelerin seçimi modelin performansı üzerinde önemli bir etkiye sahiptir ve öğrenme sürecinin doğruluğunu, genelleme kabiliyetini ve hızını etkileyebilir. Hiperparametre optimizasyonunun temel amacı, modelin en iyi performansıyla sonuçlanan optimum hiperparametre kombinasyonunu bulmaktır. Hiperparametre optimizasyonu algoritmasının adımları Algoritma 5.3'te gösterilmektedir.

Algoritma 5.3 Hiperparametre Optimizasyonu Algoritması

```
Veri setini yükle
Eksik değerleri işleme (ortalama/medyan/en sık)
Etiket kodlayıcı uygulama
Ölçekleyici seçimi (standard/min-max/robust)
    Ölçekleyici dosyasını kaydetme
    SHAP/Genetik algoritma ile özellik seçim (model tabanlı)
    Tüm sınıflandırma modelleri için:
        //Grid search yöntemi kullanılıyor
        Grid içindeki her bir parametre için:
            Parametreleri uygulama ve seçilen özelliklerle doğruluğu hesaplama
            Hesaplanan doğruluğu önceki doğruluklarla karşılaştırma
            En yüksek doğruluk oranını belirlenen parametrelerle birlikte kaydetme
Kaydedilen tüm hiperparametre bilgilerini doğruluk oranları sıralanmış bir şekilde getirme
Ölçekleyici dosyasını kullanma
Seçilen parametrelerle modelleri oluşturma
Model dosyasını kaydetme
```

Geliştirilen AutoML çervesinde kullanılan sınıflandırma algoritmaları şunlardır:

- AdaBoost

- CatBoost
- Decision Tree
- Gaussian Process
- Gradient Boosting
- K-Nearest Neighbor
- LightGBM
- Neural Network
- Perceptron
- Random Forest
- Ridge
- Stochastic Gradient Descent
- SVM
- XGBoost

Hiperparametre optimizasyonunda arama yöntemleri olarak çerçevede uygulanan yöntemler: Grid search (Izgara arama) ve Tree-structured Parzen Estimator (TPE) yöntemleridir. Grid search ve TPE, performansını optimize etmek için bir makine öğrenmesi modelinin hiperparametrelerini ayarlama süreci olan hiperparametre optimizasyonu için iki yöntemdir.

Grid search yöntemi, hiperparametre optimizasyonu sürecinde belirli bir hiperparametre setini tanımlamayı ve bu hiperparametrelerin her bir kombinasyonu için modeli eğitip değerlendirmeyi içeren bir yaklaşımdır. En uygun hiperparametre kombinasyonu, modelin doğrulama veri seti üzerindeki performansına göre seçilir. Grid search basit ve sezgisel bir yöntemdir, ancak hiperparametre değerlerinin her bir kombinasyonu için ayrı bir modelin eğitilmesini ve değerlendirilmesini gerektirdiğinden hesaplama açısından maliyetli olabilmektedir.

Grid search yönteminin avantajları:

- Basit ve sezgisel: Grid search, hiperparametrelerin ve değerlerinin bir grid içinde belirtilmesini ve her değer kombinasyonu için bir modelin eğitilmesini ve değerlendirilmesini içerdiğinden basit ve sezgisel bir yöntemdir. Bu, anlaşılmasını ve uygulanmasını kolaylaştırır ve optimizasyonu algoritmasının derinlemesine anlaşılmasını gerektirmez.

- Kolay paralelleştirme: Grid search, hiperparametre değerlerinin her bir kombinasyonu için bağımsız modellerin eğitilmesini ve değerlendirilmesini içerdiğinden kolayca paralelleştirilebilir. Bu, geniş bir hiperparametre alanı üzerinde arama yapmayı daha hızlı hale getirebilmektedir.
- Kategorik ve sürekli hiperparametreleri işleme: Grid search, her hiperparametre için bir dizi değer belirterek hem kategorik hem de sürekli hiperparametreleri işleyebilir. Bu sayede çok çeşitli hiperparametre kombinasyonlarının keşfedilmesine olanak tanır.

Grid search yönteminin dezavantajları:

- Maliyetli hesaplama: Grid search, hiperparametre değerlerinin her bir kombinasyonu için ayrı bir modelin eğitilmesini ve değerlendirilmesini gerektirdiğinden hesaplama açısından maliyetli olabilir. Bu durum geniş bir hiperparametre alanı üzerinde arama yapmayı veya aramanın birden fazla iterasyonunu gerçekleştirmeyi pratik olmaktan çıkarabilir.
- Global optimum değer bulunamayabilir: Grid search, yalnızca grid içinde belirtilen hiperparametre değerlerini dikkate aldığından hiperparametre uzayının global optimumunu bulamayabilir. Grid içinde olmayan hiperparametrelerin iyi sonuçlar üretebilecek kombinasyonları kaçırılabilir.
- Hiperparametrelerin ölçeklendirmesine duyarlı olma: Modelin performansı hiperparametrelerin göreceli ölçeklerine bağlı olabileceğinden, Grid search hiperparametrelerin ölçeklendirilmesine duyarlıdır. Bu durum hiperparametreler için uygun ölçeklerin seçilmesini zorlaştırabilir ve ayrıca farklı hiperparametre kombinasyonlarının performansını karşılaştırmayı da zorlaştırabilir.
- Kısıtlamalarla başa çıkılamaz: Grid search, hiperparametreler üzerindeki sınırlar veya karşılıklı dışlama (mutual exclusivity) gibi kısıtlamaları işleyemez. Bu durum, geçerli veya uygulanabilir olmayan hiperparametrelerin kombinasyonlarını dikkate alabileceği anlamına gelmektedir.
- Başlatma (initialization) işlemine duyarlıdır: Grid search, modelin performansı parametrelerin başlangıç değerlerine bağlı olabileceğinden modelin başlatılmasına duyarlıdır. Bu durum, farklı hiperparametre kombinasyonlarının performansını karşılaştırmayı zorlaştırabilir ve modelin yakınsamasını etkileyebilir.

TPE yöntemi, hiperparametre değerlerinin optimum kombinasyonunu aramak için Bayes optimizasyonunu kullanan bir yöntemdir. TPE, hiperparametre değerlerinin bir fonksiyonu olarak modelin performansının dağılımını tahmin etmek için olasılıksal bir model kullanır ve bu dağılımı optimum hiperparametrelerin aranmasına rehberlik etmek için kullanır. TPE, her hiperparametre değer kombinasyonu için ayrı bir modelin eğitilmesini ve değerlendirilmesini gerektirmediğinden Grid search yöntemine göre daha verimlidir. Bununla birlikte, verilere olasılıksal bir model uydurmayı içerdiğinden diğer yöntemlere göre daha fazla hesaplama süresi gerektirebilir.

TPE yönteminin avantajları:

- Verimli: TPE, her hiperparametre değeri kombinasyonu için ayrı bir modelin eğitilmesini ve değerlendirilmesini gerektirmediğinden Grid search yöntemine göre daha verimlidir. Bunun yerine, hiperparametre değerlerinin bir fonksiyonu olarak modelin performansının dağılımını tahmin etmek için olasılıksal bir model kullanır. Bu sayede büyük hiperparametre uzayları için Grid search yönteminden daha hızlı çalışabilmektedir.
- Global optimum değeri bulunabilir: TPE, tüm uzayı dikkate aldığı ve sabit bir değer grid'ine dayanmadığı için hiperparametre uzayının global optimumunu bulabilir. Bu yüzden Grid search yöntemine göre optimum kombinasyonu bulma olasılığı daha yüksektir.
- Kısıtlamaları ele alabilir: TPE, hiperparametreler üzerindeki sınırlar veya karşılıklı dışlama gibi kısıtlamaları olasılıksal modele dahil ederek ele alabilir. Bu özellik sayesinde TPE, yalnızca geçerli ve uygulanabilir hiperparametre kombinasyonlarını dikkate almaktadır.
- Kategorik ve sürekli hiperparametreleri işleyebilir: TPE, hiperparametrelerin bir fonksiyonu olarak modelin performansının dağılımını tahmin ederek hem kategorik hem de sürekli hiperparametreleri ele alabilir. Bu sayede çok çeşitli hiperparametre kombinasyonları üzerinde arama yapabilir.

TPE yönteminin dezavantajları:

- Daha fazla hesaplama süresi gerektirebilir: TPE, verilere olasılıksal bir model uydurmayı içerdiğinden diğer yöntemlere göre daha fazla hesaplama süresi gerektirebilir. Bu durum, özellikle büyük veri kümeleri veya karmaşık modeller için TPE'nin diğer yöntemlere göre daha yavaş çalışmasına neden olabilir.

- Daha fazla bellek gerektirir: TPE, olasılıksal modeli bellekte sakladığı için diğer yöntemlere göre daha fazla bellek gerektirir. Bu yüzden karmaşık modeller ya da çok sayıda hiperparametre içeren problemler için kullanılması pratik olmaktan çıkabilir.
- Olasılıksal model seçimine duyarlı olabilir: Modelin performansı modelin varsayımlarına ve parametrelerine bağlı olabileceğinden, TPE olasılıksal modelin seçimine duyarlı olabilir. Bu durum uygun modelin seçilmesini zorlaştırabilir ve modelin doğruluğunu etkileyebilir.

5.4 Deneysel Sonuçların Yorumlanması

Bu çalışmanın amacı, endüstriyel nesnelerin interneti sistemlerinde saldırı tespiti için otomatik makine öğrenmesi çerçevesi geliştirmektir.

Özellik seçimi, ilgisiz veya gürültülü özellikleri belirleyip kaldırarak veri setinin boyutunu azaltabilir ve modelin daha sağlam ve yorumlanabilir hale getirebilir. X_IIoTID verisetinin çzellik seçiminin etkinliği Tablo 5.5'te görülmektedir. Burada özellik seçiminden sonra eğitim ve test için geçen sürenin önemli ölçüde azaldığı ve modelin doğruluğunun arttığı gözlemlenmektedir. Eğitim için geçen süreyi hesaplamak için modelleme aşamaları 10 kez tekrarlanmış ve ortalaması alınmıştır.

Tablo 5.5 Seçilen özellikler ile model doğruluğu ve geçen süre

Model Adı	Doğruluk (Özellik Seçimi Öncesi)	Doğruluk (Özellik Seçimi Sonrası)	Eğitim için Toplam Süre (Özellik Seçimi Öncesi) (saniye)	Eğitim için Toplam Süre (Özellik Seçimi Sonrası) (saniye)	Silinen Özellik Sayısı
AdaBoost	0,9691	0,9792	801,82	695,48	18
CatBoost	0,9963	0,9969	367,04	332,25	17
Decision Tree	0,9962	0,9966	118,2	82,88	16
Gradient Boosting	0,9966	0,9968	15751,3	11728,44	16

Tablo 5.5 Seçilen özellikler ile model doğruluğu ve geçen süre (devamı)

Model Adı	Doğruluk (Özellik Seçimi Öncesi)	Doğruluk (Özellik Seçimi Sonrası)	Eğitim için Toplam Süre (Özellik Seçimi Öncesi) (saniye)	Eğitim için Toplam Süre (Özellik Seçimi Sonrası) (saniye)	Silinen Özellik Sayısı
Ridge	0,9342	0,9519	**5,04**	**4,01**	17
Stochastic Gradient Descent	0,9403	0,9488	11,23	8,74	17
SVM	0,9828	0,9904	37612,4	32274,54	15
XGBoost	**0,9970**	0,9972	194,67	143,29	16
kNN	0,9901	0,9920	5811,32	5114,28	17
LightGBM	0,9960	0,9968	42,48	38,22	15
Naïve Bayes	0,5783	0,8625	5,59	4,97	17
Neural Network	0,9878	0,9927	7197,74	5293,49	18
Perceptron	0,9296	0,9385	8,9	6,4	16
Random Forest	**0,9970**	**0,9975**	1088,63	722,49	**19**

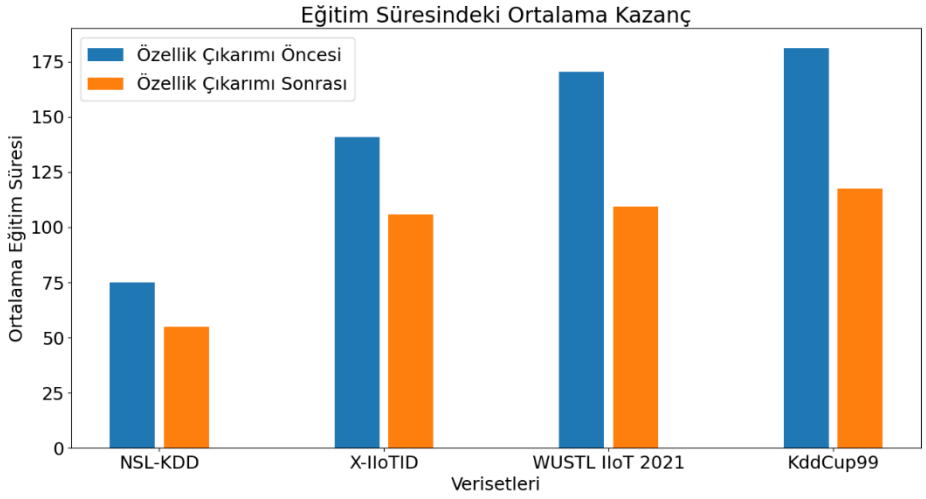
Saldırı tespitinde doğru pozitif (TP) model tarafından doğru şekilde pozitif olarak sınıflandırılan örnekleri ifade ederken, doğru negatif (TN), doğru şekilde negatif olarak sınıflandırılan örnekleri ifade eder. TP, gerçek saldırı olan ve model tarafından bu şekilde sınıflandırılan örnekleri ifade ederken TN, doğru sınıflandırılan saldırı olmayan örneklerin sayısını temsil etmektedir. Saldırı olmayan örneklerin saldırı olarak yanlış sınıflandırılması yanlış pozitif (FP) olarak adlandırılır. Yanlış negatif (FN) saldırı olmayan durumların saldırı olarak sınıflandırılmasını ifade eder. Farklı veri setleri üzerindeki özellik çıkarma modülünün sonuçları, tespit oranı, yanlış alarm oranı ve

doğruluk dahil olmak üzere Tablo 5.6'da gösterilmektedir. Ayrıca Tablo 5.6, sonuçlardan elde edilen en yüksek doğruluğu göstermektedir.

Tablo 5.6 En yüksek doğruluğa dayalı olarak tüm veri setlerinin tespit oranı, yanlış alarm oranı, doğruluğu ve eğitim süresindeki kazanç

Veri Seti	Tespit Oranı	Yanlış Alarm Oranı	Doğruluk	Özellik Çıkarma Yöntemi	Kazanç (%)
KddCup99	0,9993	0,324	0,9995	Autoencoder	38,17
NSL-KDD	0,9982	0,347	0,9983	Autoencoder	25,72
WUSTL IIoT 2021	0,9995	0,294	0,9999	Autoencoder	39,89
X-IIoTID	0,9934	0,349	0,9948	Autoencoder	22,58

Geliştirilen otomatikleştirilmiş özellik çıkarımı modülü ile elde edilen en yüksek doğruluk oranlarına sahip sınıflandırmaların eğitim sürelerindeki kazanç Şekil 5.4'te gösterilmektedir.



Şekil 5.4 Eğitim süresindeki ortalama kazancın karşılaştırılması

Tablo 5.7 ve Tablo 5.8, otomatik özellik çıkarımı yaklaşımımızın KddCup99 veri seti üzerindeki PCA ve LDA yöntemlerinin sonuçlarını göstermektedir. Bu yaklaşım ile farklı parametreler otomatik olarak test edilmekte ve sonuçlar kaydedilmektedir. Tablo 5.7'de PCA yöntemi ile kullanılan bileşen sayısı ile doğruluk oranının ve eğitim süresinin arttığı

gözlemlenmiştir. Bileşen sayısı 15 ve üzeri olduğunda doğruluk oranının değişmediği, eğitim süresinin ise artmaya devam ettiği görülmektedir.

Tablo 5.7 PCA yöntemi ile KddCup99 veri setinin özellik çıkarma sonuçları

Boyut Büyüklüğü	Doğruluk	Çıkarma Süresi (s)	Eğitim Süresi (s)	Test Süresi (s)
1	0,989474	1,84	84,02	0,29
5	0,999281	2,07	84,53	0,21
10	0,999352	2,79	87,92	0,17
15	0,999413	3,81	111,68	0,16
20	0,999413	4,57	113,84	0,17
25	0,999413	5,38	140,33	0,16
30	0,999413	6,21	146,04	0,17
35	0,999433	6,45	176,08	0,21

Tablo 5.8 KddCup99 veri setinin LDA yöntemi ile özellik çıkarma sonuçları

Boyut Büyüklüğü	Doğruluk	Çıkarma Süresi (s)	Eğitim Süresi (s)	Test Süresi (s)
1	0,956182	0,82	3,97	0,01

Tablo 5.9 ve Tablo 5.10'da NSL-KDD veri seti üzerinde PCA ve LDA yöntemleri ile özellik çıkarma sonuçlarını içermektedir. PCA sonuçlarının yer aldığı Tablo 5.9'da görüldüğü üzere bileşen sayısı 15 ve üzerinde doğruluk oranı çok değişmemekle birlikte eğitim süresi önemli ölçüde artmıştır.

Tablo 5.9 NSL-KDD veri setinin PCA yöntemi ile özellik çıkarma sonuçları

Boyut Büyüklüğü	Doğruluk	Çıkarma Süresi (s)	Eğitim Süresi (s)	Test Süresi (s)
1	0,90387	0,41	3,44	0,01
5	0,995912	0,46	3,85	0,01
10	0,997341	0,65	4,39	0,01
15	0,997301	0,87	7,19	0,01
20	0,997380	1,08	7,77	0,01
25	0,997420	1,26	10,06	0,01
30	0,997976	1,42	10,66	0,01
35	0,997998	1,45	11,65	0,01

Tablo 5.10 NSL-KDD veri setinin LDA yöntemi ile özellik çıkarma sonuçları

Boyut Büyüklüğü	Doğruluk	Çıkarma Süresi (s)	Eğitim Süresi (s)	Test Süresi (s)
1	0,991549	3,78	95,22	0,33

Tablo 5.11 ve Tablo 5.12, PCA ve LDA yöntemlerinin WUSTL IIoT 2021 veri seti üzerindeki özellik çıkarma sonuçlarını göstermektedir. Bileşen sayısı 25'e ulaştığında maksimum doğruluk oranı gözlemlenmektedir.

Tablo 5.11 WUSTL IIoT 2021 veri setinin PCA yöntemi ile özellik çıkarma sonuçları

Boyut Büyüklüğü	Doğruluk	Çıkarma Süresi (s)	Eğitim Süresi (s)	Test Süresi (s)
1	0,979598	6,17	46,93	0,1
5	0,999862	5,95	50,96	0,1
10	0,999904	7,37	57,03	0,11
15	0,999895	10,09	98,24	0,12
20	0,999904	13,56	102,41	0,14
25	0,999941	15,25	131,76	0,09
30	0,999941	17,66	152,52	0,1
35	0,999941	15,78	155,11	0,09

Tablo 5.12 WUSTL IIoT 2021 veri setinin LDA yöntemi ile özellik çıkarma sonuçları

Boyut Büyüklüğü	Doğruluk	Çıkarma Süresi (s)	Eğitim Süresi (s)	Test Süresi (s)
1	0,994709	11,53	45,26	0,09

Tablo 5.13 ve Tablo 5.14, otomatik özellik çıkarımı yaklaşımımızın X-IIoTID veri seti üzerindeki PCA ve LDA yöntemlerinin sonuçlarını göstermektedir.

Tablo 5.13 PCA yöntemi ile X-IIoTID veri setinin özellik çıkarma sonuçları

Boyut Büyüklüğü	Doğruluk	Çıkarma Süresi (s)	Eğitim Süresi (s)	Test Süresi (s)
1	0,760599	6,42	396,81	0,89

Tablo 5.13 PCA yöntemi ile X-IIoTID veri setinin özellik çıkarma sonuçları (devamı)

Boyut Büyüklüğü	Doğruluk	Çıkarma Süresi (s)	Eğitim Süresi (s)	Test Süresi (s)
5	0,965495	7,37	591,33	0,87
10	0,983479	8,84	636,85	0,96
15	0,986089	10,43	896,61	0,72
20	0,987626	9,22	1030,81	0,89
25	0,990389	8,04	1163,15	0,81
30	0,990584	11,66	1204,96	0,71
35	0,991749	12,97	1414,70	0,71
40	0,992657	12,62	1479,92	0,71
45	0,994029	11,60	1705,99	0,72
50	0,995639	7,54	1741,37	0,73
55	0,996985	7,50	1794,45	0,73

Tablo 5.14 X-IIoTID veri setinin LDA yöntemi ile özellik çıkarma sonuçları

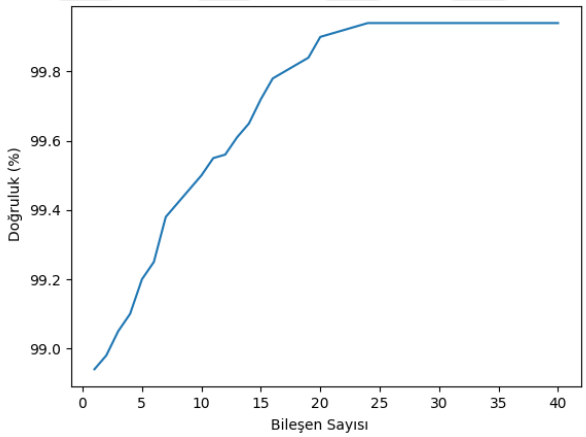
Boyut Büyüklüğü	Doğruluk	Çıkarma Süresi (s)	Eğitim Süresi (s)	Test Süresi (s)
1	0,8979	13,29	38,53	0,83

Tablo 5.15'te bir makine öğrenmesi yöntemi olan Autoencoder ile dört farklı veri seti üzerinde özellik çıkarma sonuçları gösterilmektedir. En yüksek doğruluk oranlarının elde edildiği aktivasyon ve kayıp fonksiyonları ile darboğaz katmanında kullanılan düğüm sayıları bu tabloda gösterilmektedir.

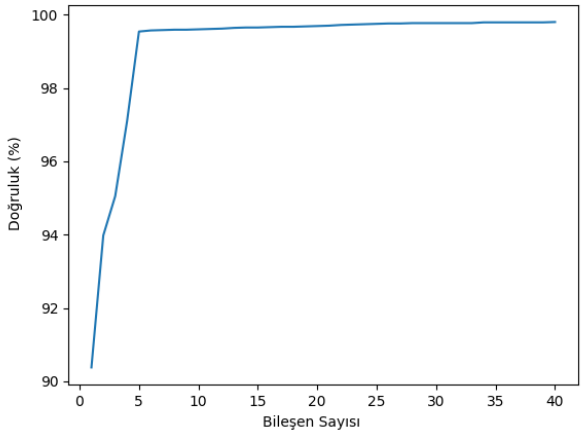
Tablo 5.15 Autoencoder yöntemi ile üç veri setinin en yüksek doğruluk tabanlı özellik çıkarma sonuçları

Darboğaz	Veri Seti	Aktivasyon Fonksiyonu	Kayıp Fonksiyonu	Doğruluk	Çıkarma Süresi (s)	Eğitim Süresi (s)	Test Süresi (s)
20	KddCup99	sigmoid	mse	0,999555	784,6	181,53	0,17
30	NSL-KDD	softsign	mse	0,998373	203,22	13,0	0,01
20	WUSTL IIoT 2021	linear	mse	0,999967	1866,32	113,25	0,09
40	X-IIoTID	linear	mse	0,9948	1744,58	128,43	0,32

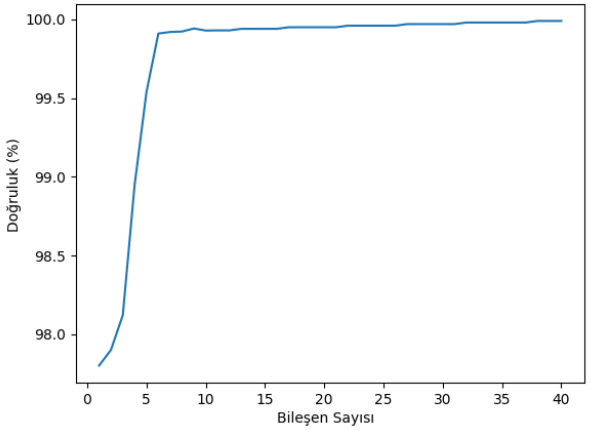
KddCup99, NSL-KDD, WUSTL IIoT 2021 ve X-IIoTID üzerinde PCA yöntemi ile yapılan özellik çıkarımlarının bileşen sayısı ve doğruluk değişimleri sırasıyla Şekil 5.5, 5.6, 5.7 ve 5.8'de gösterilmiştir.



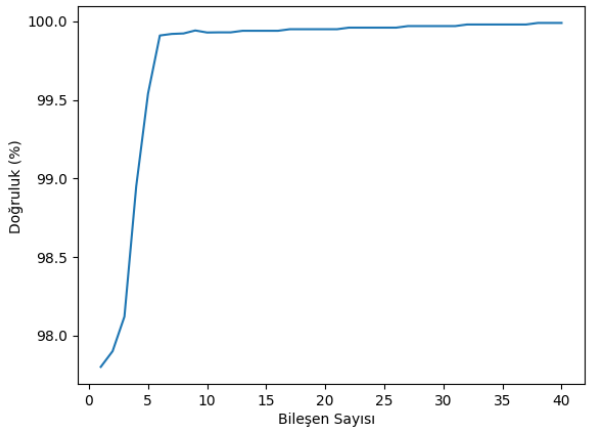
Şekil 5.5 KddCup99 - PCA ile özellik çıkarımı



Şekil 5.6 NSL-KDD - PCA ile özellik çıkarımı

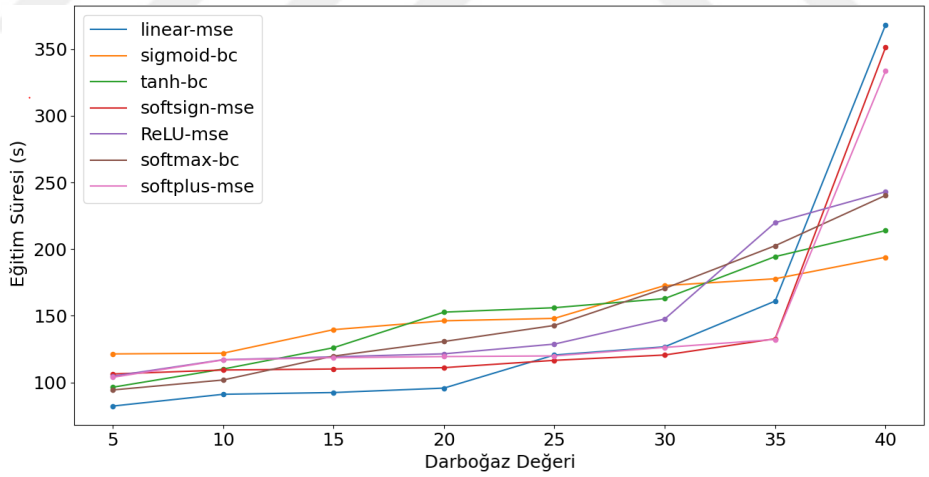


Şekil 5.7 WUSTL IIoT 2021 - PCA ile özellik çıkarımı

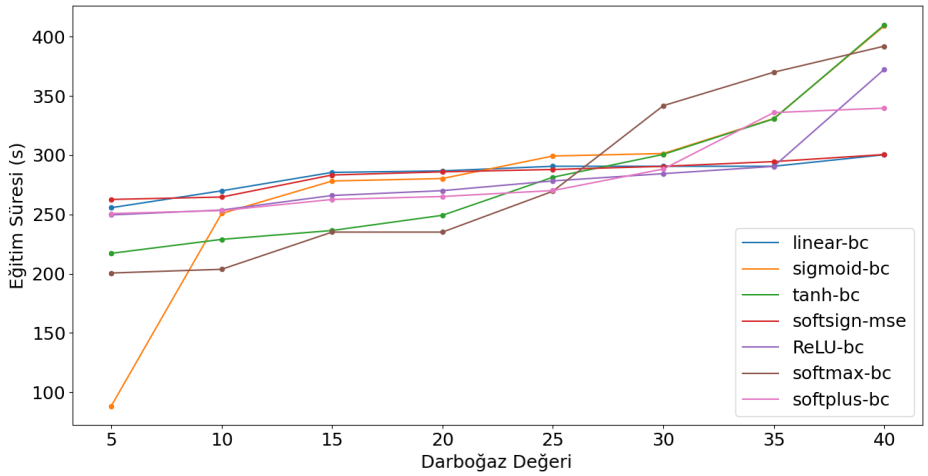


Şekil 5.8 X-IIoTID - PCA ile özellik çıkarımı

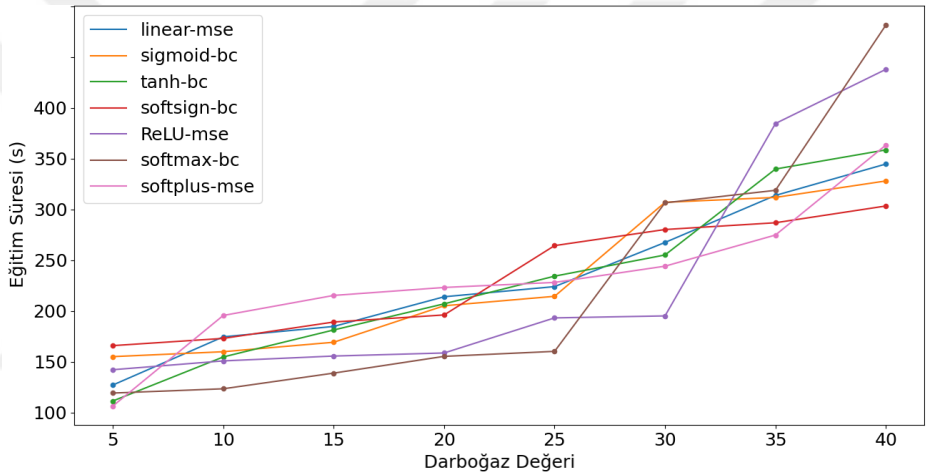
Özellik çıkarma yöntemlerinde parametre optimizasyonu için Grid Search yöntemi kullanılmaktadır. PCA ve LDA yöntemlerinde en önemli parametreler bileşen ve sınıf sayısı olacağından birden başlayarak toplam girdi ve sınıf sayısına kadar olan aralıktaki sayılar parametre olarak uygulanmakta ve her bir sonuç sınıflandırıcılara gönderilerek doğruluk oranları hesaplanmaktadır. Autoencoder yönteminde sinir ağı mimarisine ek olarak çok fazla parametre olduğu için uygun parametrelerin bulunması önemlidir. Özellik çıkarımı modülünde kullanılan parametre setinde aktivasyon fonksiyonu olarak lineer, ReLU, sigmoid, softmax, softplus, softsign ve tanh fonksiyonları bulunmaktadır. Kayıp fonksiyonu olarak ise ortalama kare hata, ikili çapraz entropi fonksiyonları kullanılmaktadır. Ayrıca, tıpkı PCA yönteminde farklı bileşen sayılarının test edilmesi gibi, Autoencoder ile özellik çıkarımında da farklı düğüm sayılarına sahip bir darboğaz katmanı uygulanmaktadır. Şekil 5.9, 5.10, 5.11 ve 5.12'de dört farklı veri setine uygulanan parametre optimizasyonları ve eğitim süreleri gösterilmektedir. Farklı aktivasyon, kayıp fonksiyonları ve farklı sayıda darboğaz düğümü uygulandığında eğitim süreleri gösterilmiştir. Şekil 5.13, 5.14, 5.15 ve 5.6'da, Autoencoder için KddCup99, NSL-KDD, WUSTL IIoT 2021 ve X-IIoTID veri setlerine farklı parametreler uygulandığında doğruluk oranlarını göstermektedir.



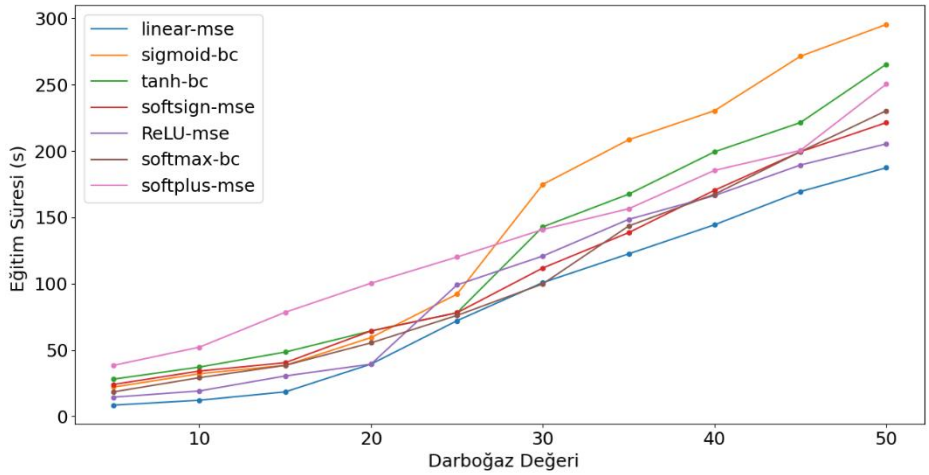
Şekil 5.9 KddCup99 - farklı parametrelerle eğitim süresi



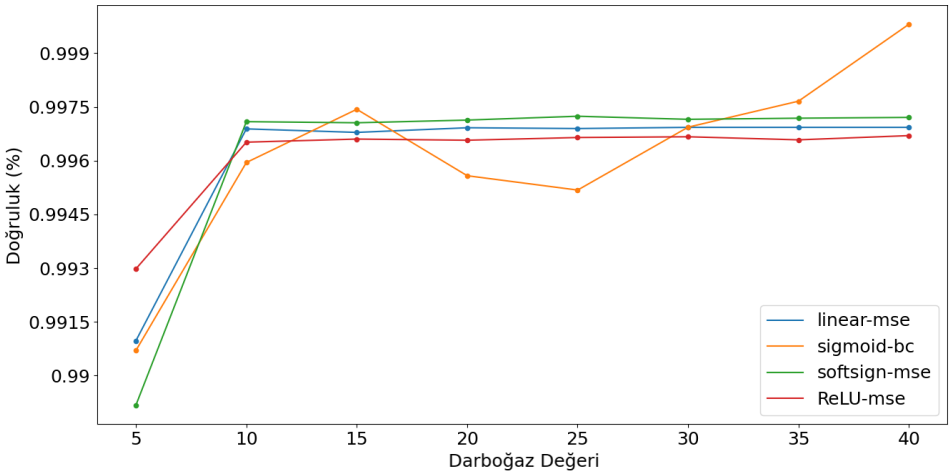
Şekil 5.10 NSL-KDD - farklı parametrelerle eğitim süresi



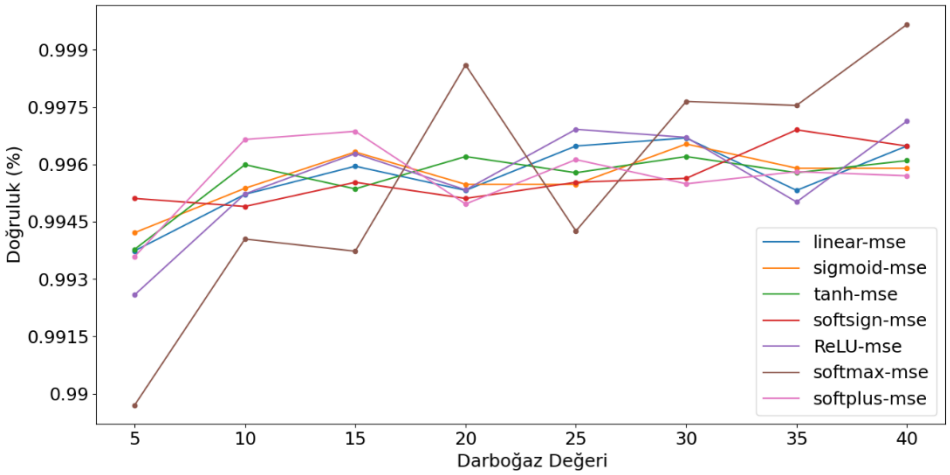
Şekil 5.11 WUSTL IIoT 2021 - farklı parametrelerle eğitim süresi



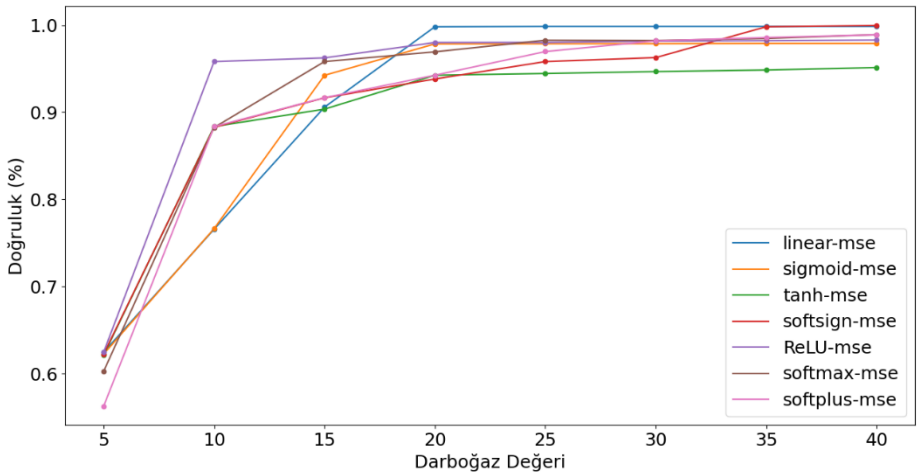
Şekil 5.12 X-IIoTID - farklı parametrelerle eğitim süresi



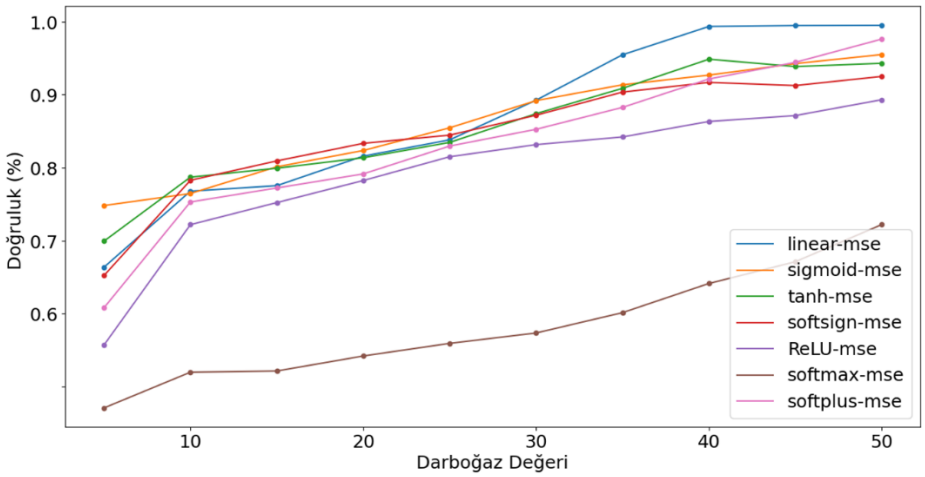
Şekil 5.13 KddCup99 - farklı parametrelerle doğruluk



Şekil 5.14 NSL-KDD - farklı parametrelerle doğruluk



Şekil 5.15 WUSTL IIoT 2021 - farklı parametrelerle doğruluk



Şekil 5.16 X-IIoTID - farklı parametrelerle doğruluk

Tablo 5.16, ikili sınıflandırmada hiperparametre optimizasyonundan önce ve sonra model doğruluğunun karşılaştırmasını göstermektedir. Hiperparametreleri dikkatli bir şekilde seçerek, modelin doğruluğunu önemli bir ölçüde arttırmak mümkündür ve bu da test verilerinden daha iyi sonuçlar elde edilmesini sağlar.

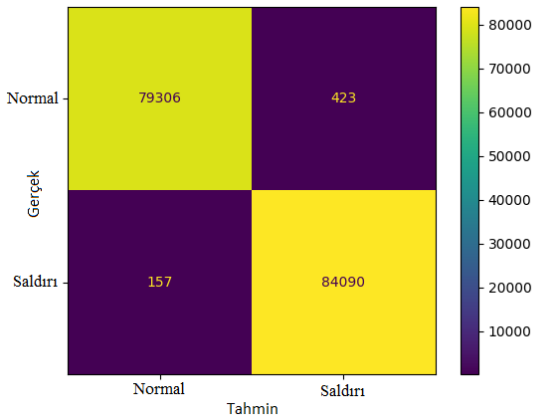
Tablo 5.16 Hiperparametre optimizasyonu öncesi ve sonrası model doğruluğu

Model Adı	Doğruluk (Optimizasyon Öncesi)	Doğruluk (Optimizasyon Sonrası)
AdaBoost	0,9691	0,9849
CatBoost	0,9963	0,9981
Decision Tree	0,9962	0,9967
Gradient Boosting	0,9966	0,9971
kNN	0,9901	0,9930
LightGBM	0,9960	0,9972
Naïve Bayes	0,5783	0,8710

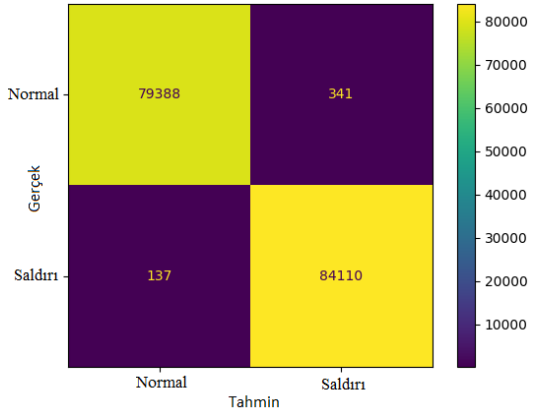
Tablo 5.16 Hiperparametre optimizasyonu öncesi ve sonrası model doğruluğu (devamı)

Model Adı	Doğruluk (Optimizasyon Öncesi)	Doğruluk (Optimizasyon Sonrası)
Neural Network	0,9878	0,9954
Perceptron	0,9296	0,9753
Random Forest	0,9970	0,9986
Ridge	0,9342	0,9588
Stochastic Gradient Descent	0,9403	0,9537
SVM	0,9828	0,9970
XGBoost	0,9970	0,9982

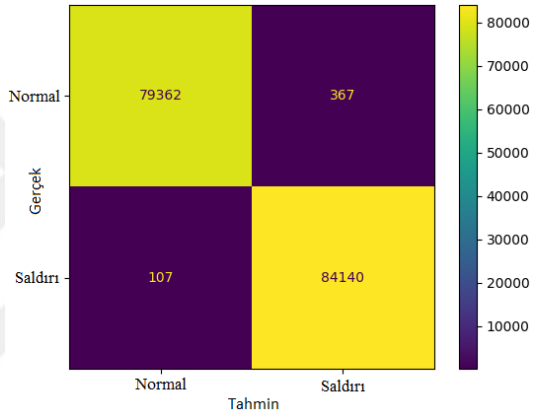
Modellerin sonuçları, doğruluklarını ve performanslarını değerlendirmek için karmaşıklık matrisleri ve sınıflandırma raporları kullanılarak analiz edilmektedir. Geliştirilen çerçeve bu sonuçları AutoML sisteminde kullanılan tüm modeller için hesaplamıştır. Şekil 5.17-5.20, kullanılan yöntemlerden dördünün karmaşıklık matrislerini ve Tablo 5.17-5.20 ilgili sınıflandırma raporlarını göstermektedir. Bu sonuçlar herhangi bir hiperparametre ayarı yapılmadan önce elde edilmiştir.



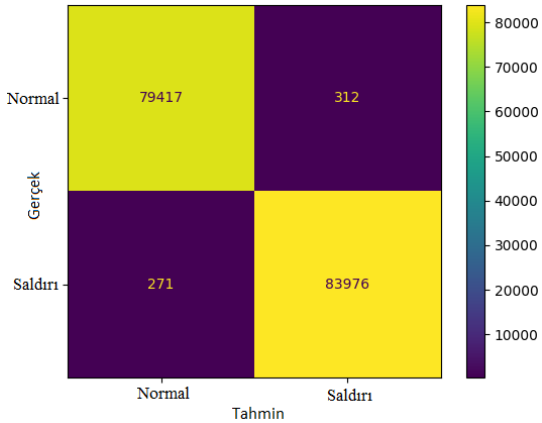
Şekil 5.17 CatBoost karmaşıklık matrisi



Şekil 5.18 XGBoost karmaşıklık matrisi



Şekil 5.19 Random Forest karmaşıklık matrisi



Şekil 5.20 Decision Tree karmaşıklık matrisi

Tablo 5.17 CatBoost yönteminin sınıflandırma raporu

	Kesinlik	Duyarlılık	F1-Skoru	Destek
Normal	0,9980	0,9947	0,9964	79729
Saldırı	0,9950	0,9981	0,9966	84247
Doğruluk				0,9965

Tablo 5.18 XGBoost yönteminin sınıflandırma raporu

	Kesinlik	Duyarlılık	F1-Skoru	Destek
Normal	0,9983	0,9957	0,9970	79729
Saldırı	0,9960	0,9984	0,9972	84247
Doğruluk				0,9971

Tablo 5.19 Random Forest yönteminin sınıflandırma raporu

	Kesinlik	Duyarlılık	F1-Skoru	Destek
Normal	0,9987	0,9954	0,9970	79729
Saldırı	0,9955	0,9987	0,9972	84247
Doğruluk				0,9971

Tablo 5.20 Decision Tree yönteminin sınıflandırma raporu

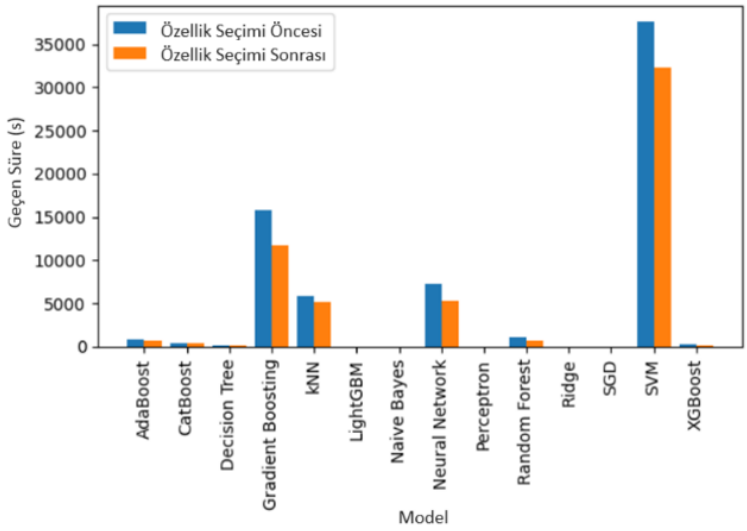
	Kesinlik	Duyarlılık	F1-Skoru	Destek
Normal	0.9966	0.9961	0.9963	79729
Saldırı	0.9963	0.9968	0.9965	84247
Doğruluk				0.9964

Önerilen sistemin performansı X-IIoTID veri setinin oluşturulduğu ve kullanıldığı çalışmalarla karşılaştırılmıştır. Bu çalışmalarda farklı yöntemlerle sınıflandırma yapılmış ve [63] çalışmasında en yüksek başarı oranı 98,23 iken [54] çalışmasında en yüksek başarı oranı 99,54 ile karar ağacı ile elde edilmiştir. Tablo 5.20'de görüldüğü üzere hiperparametre optimizasyonu yapılmayan karar ağacı yönteminin doğruluğu mevcut çalışmalara göre daha yüksektir. Aynı yöntemle daha yüksek başarı oranı elde edilmesinin temel nedeni ön işleme adımlarının ve özellik seçiminin otomatikleştirilmesidir. Tablo 5.19 incelendiğinde X-IIoTID veri setinin en yüksek başarı oranının (0,9986) hiperparametre optimizasyonu yapılarak Random Forest yöntemi ile elde edildiği görülmektedir. Tüm bu işlemler ikili sınıflandırma (saldırı/normal) için gerçekleştirilmiştir ve aynı işlemler önerilen model tarafından çok sınıflı sınıflandırma için de çalıştırılabilir. Çok sınıflı sınıflandırma sonuçları AutoML modelinin saldırı türlerine göre doğruluk oranlarını göstermektedir. Tablo 5.21'de X-IIoTID veri setindeki 18 farklı saldırı türünün dördünde önerilen AutoML modeli tarafından optimize edilen sınıflandırma yöntemlerinden üçünün tespit oranları gösterilmektedir. Karar ağacı yönteminin sonuçları incelenecek olursa, Bruteforce saldırılarını tespit etmede Command&Control saldırılarına göre çok daha yüksek doğruluğa sahiptir.

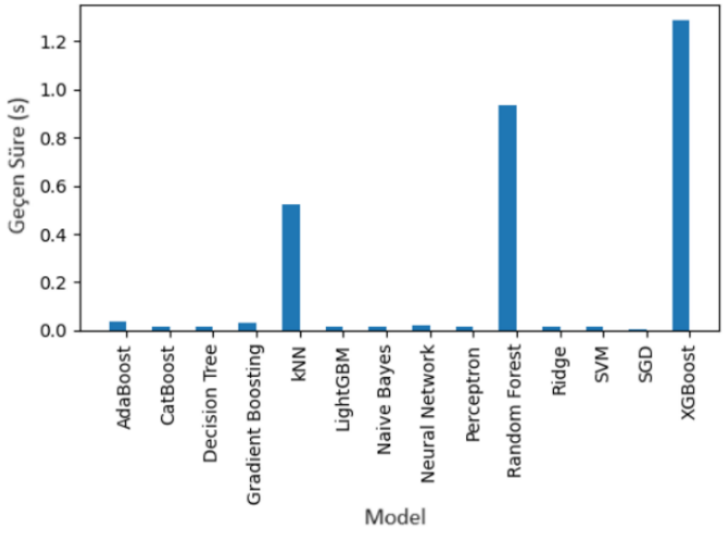
Tablo 5.21 Farklı saldırı tipleri için tespit oranları

Model	Bruteforce	C&C	Ransomware	Dictionary
Decision Tree	0,9999	0,9961	0,9999	0,9984
Random Forest	0,9999	0,9955	0,9999	0,9997
XGBoost	0,9999	0,9974	0,9999	0,9999

Model performans metriklerinin yanı sıra, eğitim ve tahmin için gereken süre de önemlidir. Söz konusu bir IDS ve IIoT olduğundan, buradaki gecikme süresinin çok düşük olması gerekmektedir. Geliştirilen çerçeve sayesinde AutoML süreçlerinin her bir adımında harcanan zaman hesaplanabilmektedir. Tablo 5.5'te incelenen model eğitim sürecinde harcanan zamanın grafiği Şekil 5.21'de gösterilmiştir. Ayrıca Şekil 5.22'de her bir modele gelen bir talebin ne kadar sürede tahmin edildiği gösterilmektedir.



Şekil 5.21 Model eğitimlerinde geçen süre



Şekil 5.22 Model tahminlerinde geçen süre

Şekil 5.22'de görüldüğü üzere XGBoost ile talebin tahmin başarısı yüksek olmasına rağmen model üzerinden tahmin işlemi çok uzun sürmektedir.

Makine öğrenmesi yöntemleri ile saldırı tespit sistemi geliştirme çalışmalarında kullanılan eski ama popüler veri setleri de bulunmaktadır. Bu veri setlerine örnek olarak KDD99, NSL-KDD ve UNSW-NB15 veri setleri verilebilir. Bu çalışmada geliştirilen model bir AutoML olduğu için farklı veri setlerinde eğitim ve tahmin süreçlerinde başarılı sonuçlar vermesi beklenmektedir. Önerilen AutoML modeli yukarıda belirtilen veri setlerine uygulandığında sonuçların başarılı olduğu gözlemlenmiştir. Bu karşılaştırmaların sonucu Tablo 5.22'de gösterilmektedir.

Tablo 5.22 KDD99, NSL-KDD, UNSW-NB15 veri setleri üzerinde önerilen modelin önceki yaklaşımlarla karşılaştırılması

Çalışma	Yaklaşım	Veri Seti	Doğruluk
[64]	Naïve Bayes	KDD99	0,950
[65]	K-Means and ANN	KDD99	0,975
[66]	Convolutional Neural Network	KDD99	0,9984
[67]	Meanshift Clustering Algorithm	KDD99	0,8120
[68]	Multi-agent System	KDD99	0,9582
AID4I	**Önerilen Model: CatBoost, İmpütasyon: Medyan, Ölçekleyici: Min-Max**	**KDD99**	**0,9998**
[69]	C4.5, Naïve Bayes, Random Forest	NSL-KDD	0,9965
[70]	k-NN, K-Means	NSL-KDD	0,9943
[71]	Random Forest	NSL-KDD	0,9870
AID4I	**Önerilen Model: XGBoost, İmpütasyon: Mean, Ölçekleyici: Standart**	**NSL-KDD**	**0,9993**

Tablo 5.22 KDD99, NSL-KDD, UNSW-NB15 veri setleri üzerinde önerilen modelin önceki yaklaşımlarla karşılaştırılması (devamı)

Çalışma	Yaklaşım	Veri Seti	Doğruluk
[22]	Decision Tree, Naïve Bayes, ANN, Logistic Regression, EM Clustering	UNSW-NB15	0,8556
[72]	MSCNN-LSTM	UNSW-NB15	0,9560
[73]	Fuzzy C-Means	UNSW-NB15	0,9890
AID4I	**Önerilen Model: Decision Tree, İmpütasyon: Mean, Ölçekleyici: Min-Max**	**UNSW-NB15**	**0,9995**
[54]	Decision Tree	X-IIoTID	0,9954
AID4I	**Önerilen Model: Random Forest, Impütasyon: Mean, Ölçekleyici: Standart**	**X-IIoTID**	**0,9986**

6
SONUÇ

Makine öğrenmesi alanında talebin giderek arttığı bir dünyada, bu alanda istenilen başarının elde edilebilmesinin alandaki nitelikli insan sayısına bağlı olması önemli bir sorundur. AutoML kavramı bu soruna çözüm bulmak için geliştirilmiştir. Çünkü gerçek dünya problemi ne olursa olsun, veriyi içeren problemin makine öğrenmesi sürecini bir optimizasyon problemi olarak ele alarak iteratif veya lineer bir işlem hattı kurgusu ile yaklaşmak önemli bir avantajdır.

Veri boyutu arttıkça ön işleme, özellik seçimi, makine öğrenmesi yöntemi seçimi ve parametre optimizasyonu daha önemli hale gelmektedir. Bu çalışmada modele bağlı ön işleme adımları, özellik seçimi işlemleri ve hiperparametrelerin optimizasyonu otomatize edilmiştir. İlgili algoritmaların özetlenmesinin yanı sıra, otomasyon süreci için kullanılan yöntemler, artıları, eksileri ile belirtilmiştir.

Bu çalışmanın amacı, ön işleme adımları, özellik seçimi, hiperparametre optimizasyonu ve makine öğrenmesi tekniği seçimi dahil olmak üzere siber güvenlikle ilgili temel süreçleri otomatikleştirmektir. Otomatikleştirme sürecinde kullanılan yöntemler, faydaları ve dezavantajları ile kapsamlı şekilde açıklanmıştır. Daha karmaşık modeller için kullanılan algoritmaların uygulanabilirliği ve hesaplama kaynaklarının kullanımı da bu çalışmada karşılaştırılmış ve değerlendirilmiştir. AutoML, saldırı tespit modelleri oluşturma sürecini otomatikleştirerek güvenlik uzmanlarının ve bilgi teknolojileri çalışanlarının iş yükünü azaltmaya yardımcı olabilir. AutoML kullanımı ile odak noktası, modellerin manuel olarak optimize edilmesinden sonuçların değerlendirilmesine ve IIoT sisteminin genel güvenlik duruşunun geliştirilmesine kaydırılabilir. Ek olarak AutoML, IIoT sistemlerine yönelik siber saldırıları tanımada saldırı tespit sistemlerinin güvenilirliğini ve doğruluğunu arttırabilir. AutoML, büyük miktarda veriyi analiz ederek daha önce tanımlanmamış tehditlerin belirlenmesine ve yanlış pozitif alarmların miktarının azaltılmasına yardımcı olabilir. Bu sistemlerde siber güvenliğe yaklaşımımız, IIoT için saldırı tespit sistemlerinde AutoML'in kullanılmasıyla tamamen değişebilir. Model oluşturma sürecinin otomatikleştirilmesiyle güvenlik uzmanlarının iş yükü azaltılabilir ve IIoT sistemlerinin güvenlik duruşu iyileştirilebilir.

KAYNAKÇA

[1] A. Bagri, R. Netto ve D. Jhaveri, "Supervisory Control and Data Acquisition," International Journal of Computer Applications, Vol. 102, No. 10, pp. 1-5, 2014.

[2] A. Hassanzadeh, S. Modi ve S. Mulchandani, "Towards effective security control assignment in the Industrial Internet of Things," World Forum on Internet of Things (WF-IoT), Milan, Italy, 2015.

[3] R. Vinayakumar, M. Alazab, K. Soman, P. Poornachandran, A. Al-Nemrat ve S. Venkatraman, "Deep learning approach for intelligent intrusion detection system," IEEE Access, vol. 7, pp. 41525-41550, 2019.

[4] D. P. Hostiadi, Y. P. Atmojo, R. R. Huizen, I. M. D. Susila, G. A. Pradipta ve I. M. Liandana, "A New Approach Feature Selection for Intrusion Detection System Using Correlation Analysis," International Conference on Cybernetics and Intelligent System (ICORIS), Prapat, Indonesia, 2022.

[5] T. Janarthanan ve S. Zargari, "Feature selection in UNSW-NB15 and KDDCUP'99 datasets," International Symposium on Industrial Electronics (ISIE), Edinburgh, UK, 2017.

[6] A. Saber, M. Abbas ve B. Fergani, "Two-dimensional Intrusion Detection System: A New Feature Selection Technique," International Workshop on Human-Centric Smart Environments for Health and Well-being (IHSH), Boumerdes, Algeria, 2021.

[7] N. Sampath, M. A. Jerlin, L. B. Kritkika ve A. Anitha, "Intrusion Detection in Software Defined Networking using Genetic Algorithm," International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE), Vellore, India, 2020.

[8] H. Aiods ve P. Tomás, "Neighborhood-aware autoencoder for missing value imputation," European Signal Processing Conference (EUSIPCO), Amsterdam, Netherlands, 2020.

[9] A. Telikani, J. Shen, J. Yang ve P. Wang, "Industrial IoT Intrusion Detection via Evolutionary Cost-Sensitive Learning and Fog Computing," IEEE Internet of Things Journal, cilt 9, no. 2, pp. 23260-23271, 2022.

[10] G. A. Da Silva Oliveira, P. S. Silva Lima, F. Kon ve R. Terada, "A Stacked Ensemble Classifier for an Intrusion Detection System in the Edge of IoT and IIoT Networks," IEEE Latin-American Conference on Communications (LATINCOM), Rio de Janeiro, Brazil, 2022.

[11] S. T. Ikram ve A. K. Cherukuri, "Intrusion detection model using fusion of chi-square feature selection and multi class SVM," Journal of King Saud University - Computer and Information Sciences, cilt 29, no. 4, pp. 462-472, 2017.

[12] A. Khacha, R. Saadouni, Y. Harbi ve Z. Aliouat, "Hybrid Deep Learning-based Intrusion Detection System for Industrial Internet of Things," International Symposium on Informatics and its Applications (ISIA), M'sila, Algeria, 2022.

[13] A. Zainudin, R. Akter, D. S. Kim ve J. M. Lee, "Towards Lightweight Intrusion Identification in SDN-based Industrial Cyber-Physical Systems," Asia Pacific Conference on Communications (APCC), Jeju Island, Korea, Republic of, 2022.

[14] A. Heryanto, D. Stiawan, M. Y. Bin Idris, M. R. Bahari, A. A. Hafizin ve R. Budiarto, "Cyberattack Feature Selection using Correlation-Based Feature Selection Method in an Intrusion Detection System," International Conference on Electrical Engineering, Computer Science and Informatics (EECSI), Jakarta, Indonesia, 2022.

[15] P. B. Udas, E. Karim ve K. S. Roy, "SPIDER: A shallow PCA based network intrusion detection system with enhanced recurrent neural networks," Journal of King Saud University - Computer and Information Sciences, cilt 34, no. 10, pp. 10246-10272, 2022.

[16] J. Ahmad, S. A. Shah, S. Latif, F. Ahmed, Z. Zou ve N. Pitropakis, "DRaNN_PSO: A deep random neural network with particle swarm optimization for intrusion detection in the industrial internet of things," Journal of King Saud University - Computer and Information Sciences, cilt 34, no. 10, pp. 8112-8121, 2022.

[17] M. Mazini, B. Shirazi ve I. Mahdavi, "Anomaly network-based intrusion detection system using a reliable hybrid artificial bee colony and AdaBoost algorithms," Journal of King Saud University - Computer and Information Sciences, cilt 31, no. 4, pp. 541-553, 2019.

[18] J. Liu, D. Yang, M. Lian ve M. Li, "Research on intrusion detection based on particle swarm optimization in IoT," IEEE Access, vol. 9, pp. 38254-38268, 2021.

[19] X. Zhou, Y. Hu, W. Liang, J. Ma ve Q. Jin, "Variational LSTM Enhanced Anomaly Detection for Industrial Big Data," IEEE Transactions on Industrial Informatics, vol. 17, no. 5, pp. 3469-3477, 2021.

[20] J. Gao, S. Chai, B. Zhang ve Y. Xia, "Research on network intrusion detection based on incremental extreme learning machine and adaptive principal component analysis," Energies, vol. 12, no. 7, p. 1223, 2019.

[21] S. Hanif, T. Ilyas ve M. Zeeshan, "Intrusion detection in IoT using artifical neural networks on UNSW-15 dataset," Smart Cities: Improving Quality of Life Using ICT & IoT (HONET-ICT), 2019.

[22] N. Moustafa ve J. Slay, "The evaluation of network anomaly detection systems: Statistical analysis of the UNSW-NB15 data set and the comparison with KDD99 data set," Information Security Journal: A Global Perspective, vol. 25, no. 1-3, pp. 18-31, 2016.

[23] E. Ketzaki, A. Drosou, S. Papadopoulos ve D. Tzovaras, "A light-weighted ANN architecture for the classification of cyber-threats in modern communication networks," International Conference on Network of the Future (NOF), 2019.

[24] O. Almomani, "A feature selection model for network intrusion detection system based on PSO, GWO, FFA and GA algorithms," Symmetry, vol. 12, no. 6, p. 1046, 2020.

[25] A. Nazir ve R. A. Khan, "A novel combinatorial optimization based feature selection method for network intrusion detection," Computers & Security, vol. 102, art. no. 102164, 2021.

[26] W. Zong, Y. W. Chow ve W. Susilo, "A two-stage classifier approach for network intrusion detection," Information Security Practice and Experience, 2018.

[27] C. Khammassi ve S. Krichen, "A GA-LR wrapper approach for feature selection in network intrusion detection," Computers & Security, vol. 70, pp. 255-277, 2017.

[28] S. M. Kasongo ve Y. Sun, "Performance analysis of intrusion detection systems using a feature selection method on the UNSW-NB15 dataset," Big Data, vol. 7, no. 1, pp. 1-20, 2020.

[29] Y. Bengio ve J. Bergstra, "Random search for hyper-parameter optimization," Journal of Machine Learning Research, pp. 281-305, 2012.

[30] S. Sanders ve C. Giraud-Carrier, "Informing the Use of Hyperparameter Optimization Through Metalearning," International Conference on Data Mining (ICDM), 2017.

[31] R. S. Olson ve J. H. Moore, "TPOT: A Tree-Based Pipeline Optimization Tool for Automating Machine Learning," Automated Machine Learning, Springer, Cham, 2019, pp. 151-160.

[32] T. Janarthanan ve S. Zargari, "Feature selection in UNSW-NB15 and KDDCUP'99 datasets," International Symposium on Industrial Electronics, ISIE, 2017.

[33] H. Gharaee ve H. Hosseinvand, "A new feature selection IDS based on genetic algorithm and SVM," International Symposium on Telecommunications, IST, 2016.

[34] T. Qiu, J. Chi, X. Zhou, Z. Ning, M. Atiquzzaman ve D. O. Wu, "Edge computing in industrial internet of things: Architecture, advances and challenges," IEEE Communications Surveys & Tutorials, vol 22, no 4, pp. 2472-2488, 2020.

[35] C. Meffert, D. Clark, I. Baggili ve F. Breitinger, "Forensic state acquisition from Internet of Things (FSAIoT): A general framework and practical approach for IoT forensics through IoT device state acquisition," International Conference on Availability, Reliability and Security, 2017.

[36] S. N. Firdous, Z. Baig, C. Valli ve A. Ibrahim, "Modelling and evaluation of malicious attacks against the IoT MQTT protocol," IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CpsCom) and IEEE Smart Data (SmartData), 2017.

[37] A. Khraisat, I. Gondal, P. Vamplew ve J. Kamruzzaman, "Survey of intrusion detection systems: Techniques, datasets and challenges," Cyber-security, vol. 2, no. 1, pp. 1-22, 2019.

[38] A. H. Farooqi ve F. A. Khan, "Intrusion Detection Systems for Wireless Sensor Networks.," Communication and Networking, pp. 234-241, 2009.

[39] T. Yu, V. Sekar, S. Seshan, Y. Agarwal ve C. Xu, "Handling a trillion (unfixable) flaws on a billion devices: Rethinking network security for the internet-of-things," ACM Workshop on Hot Topics in Networks, 2015.

[40] F. Sabahi ve A. Movaghar, "Intrusion detection: A survey," Systems and Networks Communications, 2008.

[41] D. Midi, A. Rullo, A. Mudgerikar ve E. Bertino, "Kalisa system for knowledge-driven adaptable intrusion detection for the internet of things," Distributed Computing Systems (ICDCS), 2017.

[42] C. Modi, D. Patel, B. Borisaniya, H. Patel, A. Patel ve M. Rajarajan, "A survey of intrusion detection techniques in cloud," Journal of Network and Computer Applications, vol. 36, no. 1, pp. 42-57, 2013.

[43] C. Kreibich ve J. Crowcroft, "Honeycomb: creating intrusion detector signatures using honeypots," SIGCOMM Computer Communication Review, no. 34, pp. 51-56, 2004.

[44] C. R. Meiner, J. Patel, E. Norige, E. Torng ve A. X. Liu, "Fast regular expression matching using small TCAMs for network intrusion detection and prevention systems," USENIX Security Symposium, Washington, DC, 2010.

[45] P. L. Lin, Y. R. Lin ve Y. Lai, "A Hybrid Algorithm of Backward Hashing and Automaton Tracking for Virus Scanning," IEEE Transactions on Computers, vol. 60, no. 4, pp. 594-601, 2011.

[46] İ. Bütün, S. D. Morgera ve R. Sankar, "A Survey of Intrusion Detection Systems in Wireless Sensor Networks," IEEE Communications Surveys & Tutorials, vol. 16, pp. 266-282, 2013.

[47] A. Alazab, M. Hobbs, J. Abawajy ve M. Alazab, "Using feature selection for intrusion detection system," Communications and information technologies (ISCIT), 2012.

[48] X. Zhang, A Matrix Algebra Approach to Artificial Intelligence, Springer, Singapore, 2020.

[49] M. Mohammed, M. B. Khan ve E. B. M. Bashier, Machine Learning: Algorithms and Applications, Boca Raton, FL, USA: CRC Press, 2016.

[50] L. Kotthoff, C. Thornton, H. H. Hoos, F. Hutter ve K. Leyton-Brown, "Auto-WEKA 2.0: Automatic model selection and hyperparameter optimization in WEKA," Journal of Machine Learning Research, 2017.

[51] M. Feurer, A. Klein, K. Eggensperger, J. T. Springenberg, M. Blum ve F. Hutter, "Efficient and robust automated machine learning," International Conference on Neural Information Processing Systems, vol. 2, 2015.

[52] H. Jin, Q. Song ve X. Hu, "Auto-keras: An efficient neural architecture search system," International Conference on Knowledge Discovery & Data Mining, ACM SIGKDD, 2019.

[53] A. Yakovlev, H. F. Moghadam, A. Moharrer, J. Cai, N. Chavoshi, V. Varadarajan, S. R. Agrawal, S. Idicula, T. Karnagel, S. Jinturkar ve N. Agarwal, "Oracle AutoML: a fast and predictive AutoML pipeline," Proceedings of the VLDB Endowment, Vol. 13, Issue 12, pp. 3166-3180, 2020.

[54] M. Al-Hawawreh, E. Sitnikova ve N. Aboutorab, "X-IIoTID: A Connectivity-Agnostic and Device-Agnostic Intrusion Data Set for Industrial Internet of Things," IEEE Internet of Things Journal, Vol. 9, No. 5, pp. 3962-3977, 2022.

[55] Y. Saeys, I. Inza ve P. Larranaga, "A review of feature selection techniques in bioinformatics," Bioinformatics, Vol. 23, Issue 19, pp. 2507-2517, 2007.

[56] R. Wald, T. M. Khoshgoftaar ve A. Napolitano, "Stability of filter and wrapped based feature subset selection," Tools With Artificial Intelligence, 2013.

[57] M. Shafiq, Z. Tian, A. K. Bashir, X. Du ve M. Guizani, "IoT malicious traffice identification using wrapper-based feature selection mechanisms," Computer Security, vol. 49, 2020.

[58] M. A. Siddiqi ve W. Pak, "Optimizing filter-based feature selection method flow for intrusion detection system," Electronics, vol. 9, no. 12, p. 2114, 2020.

[59] L. S. Shapley, "A Value for n-Person Games," Contributions to the Theory of Games (AM-28), Vol. 2, Princeton, Princeton University Press, 2016, pp. 307-318.

[60] M. A. Teixeira, T. Salman, M. Zolanvari, R. Jain, N. Meskin ve M. Samaka, "SCADA System Testbed for Cybersecurity Research Using Machine Learning Approach," Future Internet, vol. 10, no. 8, 2018.

[61] Y. Li, R. Ma ve R. Jiao, "A Hybrid Malicious Code Detection Method based on Deep Learning," International Journal of Software Engineering and Its Applications, cilt 9, no. 5, pp. 205-216, 2015.

[62] N. Gao, L. Gao, Q. Gao ve H. Wang, "An intrusion detection model based on deep belief networks," International Conference on Advanced Cloud and Big Data, 2014.

[63] M. Al-Hawawreh, E. Sitnikova ve N. Aboutorab, "Asynchronous Peer-to-Peer Federated Capability-Based Targeted Ransomware Detection Model for Industrial IoT," IEEE Access, Vol. 9, pp. 148738-148755, 2021.

[64] M. Panda ve M. R. Patra, "Network Intrusion Detection Using Naive Bayes," International Journal of Computer Science and Network Security, Vol. 7, No: 12, pp. 258-263, 2007.

[65] B. Ren, M. Hu, H. Yan ve P. Yu, "Classification and Prediction of Network Abnormal Data Based on Machine Learning," International Conference on Robots & Intelligent System (ICRIS), 2019.

[66] N. Ding, Y. Liu, Y. Fan ve D. Jie, "Network Attack Detection Methods Based on Convolutional Neural Network," Chinese Intelligent Systems Conference, 2019.

[67] A. Kumar, W. Glisson ve R. Benton, "Network Attack Detection Using an Unsupervised Machine Learning Algorithm," Hawaii International Conference on System Sciences, 2020.

[68] M. A. Riyad, I. Ahmed ve R. Khan, "An Adaptive Distributed Intrusion Detection System Architecture Using Multi Agents," International Journal of Electrical and Computer Engineering (IJECE), pp. 4951-4960, 2019.

[69] C. Khammassi ve S. Krichen, "A NSGA2-LR wrapper approach for feature selection in network intrusion detection," Computer Networks, Vol. 178, 2020.

[70] H. Wang, B. Han, J. Su ve X. Wang, "A High-Performance Intrusion Detection Method Based on Combining Supervised and Unsupervised Learning," SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI), 2018.

[71] Z. E. Mrabet, H. E. Ghazi ve N. Kaabouch, "A Performance Comparison of Data Mining Algorithms Based Intrusion Detection System for Smart Grid," International Conference on Electro Information Technology (EIT), 2019.

[72] J. Zhang, Y. Ling, X. Fu, X. Yang, G. Xiong ve R. Zhang, "Model of the intrusion detection system based on the integration of spatial-temporal features," Computers & Security, Vol. 89, 2020.

[73] V. Hajisalem ve S. Babaie, "A hybrid intrusion detection system based on ABC-AFS algorithm for misuse and anomaly detection," Computer Networks, Vol. 136, pp. 37-50, 2018.

TEZDEN ÜRETİLMİŞ YAYINLAR

Konferans Bildirileri

1. Sezgin, A., Boyacı, A. (2023). A Survey of Privacy and Security Challenges in Industrial Settings, 2023 11th International Symposium on Digital Forensics and Security (ISDFS), Chattanooga, TN, USA, 2023, pp. 1-7, doi: 10.1109/ISDFS58141.2023.10131858.

2. Sezgin, A., Boyaci, A. (2023). Securing the Skies: Exploring Privacy and Security Challenges in Internet of Drones, 10th International Conference on Recent Advances in Air and Space Technologies (RAST), Istanbul, Turkiye, 2023, pp. 1-6, doi: 10.1109/RAST57548.2023.10197987.

Makaleler

1. Sezgin, A., Boyacı, A. (2023). AID4I: An Intrusion Detection Framework for Industrial Internet of Things Using Automated Machine Learning. Computers, Materials & Continue 76(2), 2121-2143. https://doi.org/10.32604/cmc.2023.040287

2. Sezgin, A., Boyacı, A. (2023). Enhancing Intrusion Detection in Industrial Intenet of Things through Automated Preprocssing. Advances in Science and Technology Research Journal 17(2), 120-135. https://doi.org/10.12913/22998624/162004

3. Sezgin, A., Boyacı, A. (2023). Açık Kaynaklardan Test Otomasyon Araçlarıyla Siber Tehdit İstihbaratı Çıkarılması . Fırat Üniversitesi Mühendislik Bilimleri Dergisi , 35 (1) , 283-290 . DOI: 10.35234/fumbd.1217219

Projeler

1. Endüstriyel ağlarda dağıtık mimari ve AutoML tabanlı yeni nesil saldırı tespit sistemi, TÜBİTAK 1501.

2. Tarım alanları için bulut tabanlı yapay zeka destekli insansız hava aracı sürü kontrol ve analiz platformu, TÜBİTAK 1512