MARMARA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

NESNELERİN İNTERNETİ İÇİN DERİN ÖĞRENME İLE VERİ ODAKLI AĞ SALDIRI SINIFLANDIRMA SİSTEMİ

UĞUR ÇEKMEZ

DOKTORA PROGRAMI
Bilgisayar Mühendisliği Anabilim Dalı
Bilgisayar Mühendisliği Doktora Programı

DANIŞMAN
Prof. Dr. Ali BULDU

EŞ-DANIŞMAN
Prof. Dr. Özgür Koray ŞAHİNGÖZ

İSTANBUL, 2022

MARMARA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

NESNELERİN İNTERNETİ İÇİN DERİN ÖĞRENME İLE VERİ ODAKLI AĞ SALDIRI SINIFLANDIRMA SİSTEMİ

UĞUR ÇEKMEZ
(723616831)

DOKTORA PROGRAMI
Bilgisayar Mühendisliği Anabilim Dalı
Bilgisayar Mühendisliği Doktora Programı

DANIŞMAN
Prof. Dr. Ali BULDU

EŞ-DANIŞMAN
Prof. Dr. Özgür Koray ŞAHİNGÖZ

İSTANBUL, 2022

ÖNSÖZ

Tez çalışmamı hazırlama aşamasında bana her konuda yardımcı olan, her soruma sabırla yanıt veren ve tez çalışmasının eksik yönlerinin ortaya çıkarılması ve giderilmesi kapsamında değerli katkılarda bulunan tez danışmanım Prof. Dr. Ali BULDU ve eş-danışmanım Prof. Dr. Özgür Koray ŞAHİNGÖZ başta olmak üzere, çalışmam süresince büyük ilgi ve fedakârlıklarıyla her zaman yanımda olan aileme, bana bu imkânı sağlayan Marmara Üniversitesi'ne, Türk Silahlı Kuvvetleri'ne ve Hava Harp Okulu'na teşekkürlerimi bir borç bilir, sonsuz minnettarlığımı sunarım.

Eylül, 2022 **Uğur ÇEKMEZ**

İÇİNDEKİLER

ÖZET

NESNELERİN İNTERNETİ İÇİN DERİN ÖĞRENME İLE VERİ ODAKLI AĞ SALDIRI SINIFLANDIRMA SİSTEMİ

Sensör ve iletişim teknolojilerinin hızla gelişmesiyle birlikte, internete ve birbirine bağlı cihazların endüstriyel uygulamalarda kullanımı yaygınlaşmaya başlamıştır. Son dönemlerde maliyetlerin düşmesi ve nesnelerin interneti (Internet of Things - IoT) kavramının tanımlanması, birbirine bağlı bu cihazların uygulama alanını son kullanıcı seviyesine kadar genişletmiştir. Bu da aynı zamanda IoT teknolojisinin çok çeşitli uygulama alternatifleri sunmasına ve günlük yaşamın bir parçası haline gelmesine yol açmıştır. Buna karşın, yeterli seviyede korunamayan bir IoT ağının sürdürülebilir olmadığı ve sadece cihazlar üzerinde değil, aynı zamanda ağa bağlı kullanıcılar üzerinde de olumsuz etkilere neden olabileceği değerlendirilmektedir. Güvenlik açığı oluştuğu durumlarda ise geleneksel saldırı tespit sistemlerini kullanan mekanizmaların yetersiz kaldığı bilinmektedir. Davetsiz misafirlerin uzmanlık seviyesi arttıkça, yeni tür saldırıların tanımlanması ve önlenmesi daha zor hale gelmektedir. Bu nedenle olası güvenlik ihlallerinin üstesinden gelmek için doğal veri akışını öğrenebilen akıllı algoritmalar gereklidir. Tekil saldırı türleri için modeller öneren literatürdeki birçok çalışma, belli bir noktaya kadar başarılı çıktılar gösterse de çoklu ve dengesiz dağılımlı saldırı türlerini tespit etmeyi amaçlayan çalışmaların çoğunun tek bir model ile bu saldırıların tamamını başarılı bir şekilde tespit edemediği görülmektedir. Bu çalışmada çoklu saldırı türlerinin tek bir model ile yüksek başarımla tespit edilebilmesi için neler yapılabileceği üzerine bir araştırma yürütülmüştür. Bu amaçla IoT sistemlerinde görülebilecek çeşitli saldırı tipleri üzerinde yüksek F_1 puanı ile tespit yapabilecek bir derin yapay sinir ağı modeli tasarlanmış ve eğitilmiştir. Karşılaştırılabilir sonuçlar için saldırı verileri düzensiz olarak yayılmış, görece zorlu bir veriseti seçilmiş ve ulaşılan sonuçlar literatürdeki güncel çalışmalar ile kıyaslanmıştır. Çalışmanın ilk fazı ortalama sonuçlar ele alındığında ağırlıklı ortalamada %99.94 F_1 puanına ulaştığı görülürken, örneklem sayısı az olan saldırı tipleri özelinde başarımın diğerlerine göre düşük kaldığı gözlenmiştir. Bu nedenle çalışma kapsamında dengesiz veri probleminin üstesinden gelmek için üç adet farklı sentetik veri üretme tekniği de önerilmiş, ardından sınıf dağılımlarına tepki verebilen adaptif bir eşik fonksiyonu geliştirilmiştir. Deneysel sonuçlar, önerilen yöntemlerin gözlemlenebilen literatürdeki diğer çalışmalar içinde en yüksek verimi sağladığını göstermiştir.

ABSTRACT

DATA-ORIENTED NETWORK ATTACK CLASSIFICATION SYSTEM WITH DEEP LEARNING FOR THE INTERNET OF THINGS

With the rapid development of sensor and communication technologies, the use of internet and interconnected devices in industrial applications has become widespread. Recently, the decrease in costs and the practical use cases of the Internet of Things (IoT) have extended the application area of these interconnected devices to the end user level. This has also led IoT technology to offer a wide variety of application alternatives and become a part of daily life. On the other hand, it is considered that an IoT network that is not adequately protected is not sustainable and may cause adverse effects not only on devices but also on users connected to the network. It is known that traditional intrusion detection systems are insufficient in cases where novel security vulnerabilities occur. As the level of expertise of intruders increases, new types of attacks become more difficult to identify and prevent. Therefore, intelligent algorithms that can learn the natural data flow are required to overcome potential security breaches. Although many studies in the literature suggesting models for single attack types show successful outputs up to a certain point, it is seen that most of the studies aiming to detect multiple and unevenly distributed attack types cannot successfully detect all of these attacks with a single model. In this study, a research was conducted on what can be done to detect multiple attack types with a single model with high performance. For this purpose, a deep artificial neural network model that can detect various attack types that can be seen in IoT systems with a high F_1 score has been designed and trained. For comparable results, the attack data is unevenly spread, a relatively difficult dataset is selected, and the results are compared with current studies in the literature. Although the first phase of the study gave a relatively good output when the average results were considered, it was observed that the performance was lower than the others in the attack types with a small number of samples. Therefore, within the scope of the study, a number of synthetic data generation techniques are also proposed to overcome the unbalanced data problem, and then an adaptive threshold value function that can respond to class distributions is developed. Experimental results showed that the proposed methods provided the highest efficiency among other studies in the observed literature.

YENİLİK BEYANI

NESNELERİN İNTERNETİ İÇİN DERİN ÖĞRENME İLE VERİ ODAKLI AĞ SALDIRI SINIFLANDIRMA SİSTEMİ

Bu tez çalışmasında IoT ağlarına yönelik yapılan farklı türden ağ saldırılarının tek bir Derin Öğrenme modeli ile tespit edilebilmesi için genel amaçlı bir yöntem dizisi geliştirilmiştir. Literatürde çoğunlukla benzer örüntüleri ihtiva eden ve eşit dağılımlı sınıflara yönelik çözüm çalışmaları bulunurken, farklı örüntülere ve farklı sınıflar için dengesiz bir veri yoğunluğuna sahip karmaşık sınıf tiplerini içeren problemlerde sınıflandırma başarımının $\%22$ F_1 puanına kadar düştüğü gözlenmiştir. Bu kapsamda, verisetleri içerisinde dengesiz dağılıma sahip olan saldırı tiplerinin de yüksek başarımla tespit edilebilmesi için farklı konseptlerde sentetik veri artırma modülleri önerilmiş ve bu modüllerin eşik değerlerinin sınıf dağılımlarına göre adaptif olarak belirlenebilmesi için yöntemler geliştirilmiştir. Adaptif eşik değeri ile veri artırmanın model başarımına etkisi gözlenmiş ve gelecek çalışmalar için yol haritaları çizilmiştir.

Eylül, 2022 **Prof. Dr. Ali BULDU** **Uğur ÇEKMEZ**

KISALTMALAR

ALU : Arithmetic Logic Unit

CAN : Controller Area Network

CNN : Convolutional Neural Network - Evrişimli Yapay Sinir Ağları

CPU : Central Processing Unit - Merkezi İşlem Birimi

CSV : Comma-Separated Values

CTGAN : Conditional Tabular Generative Adversarial Network

CUDA : Compute Unified Device Architecture

DBN : Deep Belief Networks - Derin İnanç Ağları

DDoS : Distributed Denial of Service - Dağıtık Hizmet Engelleme

DL : Deep Learning - Derin Öğrenme

DNN : Deep Neural Network - Derin Yapay Sinir Ağı

ECS : Environmental Control System - Çevresel Kontrol Sistemi

ENN : Edited Nearest Neighbors

GAN : Generative Adversarial Network

GMM : Gaussian Mixture Model

GPGPU : General Purpose Graphics Processing Unit

GPU : Graphics Processing Unit - Grafik İşlem Birimi

GRU : Gated Recurrent Unit

IoT : Internet Of Things - Nesnelerin İnterneti

LSTM : Long Short-Term Memory - Uzun Kısa Süreli Bellek Birim

MQTT : Message Queuing Telemetry Transport

NB : Naive Bayes

PCAP : Packet Capture

PSO : Particle Swarm Optimization

RBM : Restricted Boltzmann Machine

ReLU : Rectified Linear Unit

SELU : Scaled Exponential Linear Unit

SMOTE : Synthetic Minority Oversampling Technique

SMOTEENN : Synthetic Minority Over-sampling Technique + Edited Nearest Neighbors

SSH : Secure Shell

STS : Saldırı Tespit Sistemi

TCP : Transmission Control Protocol

UDP : User Datagram Protocol

XSS : Cross-Site Scripting

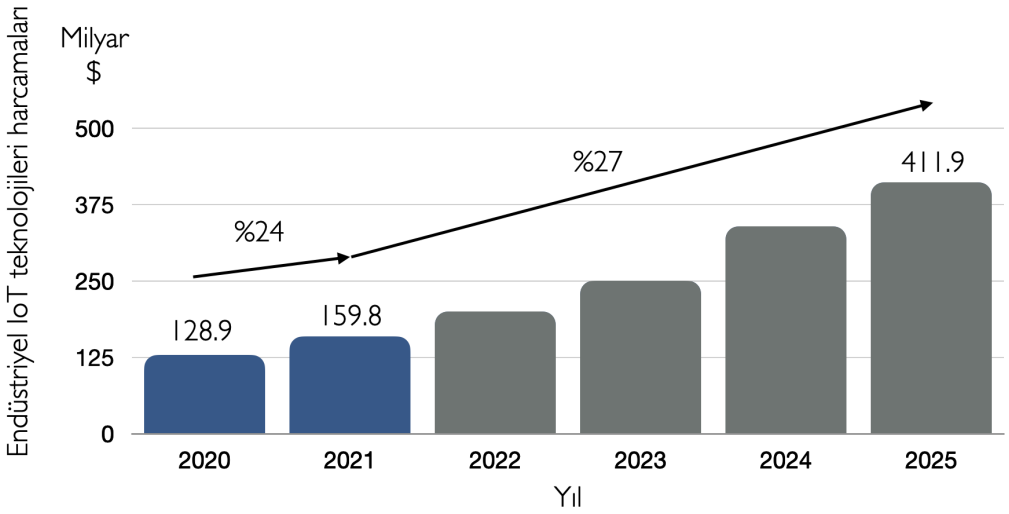
ŞEKİL LİSTESİ

TABLO LİSTESİ

1 GİRİŞ

Son yıllarda teknolojinin daha yaygın kullanımı ve değişen alışkanlıklarla beraber başta bilgisayarlar ve akıllı telefonlar olmak üzere, birçok farklı elektronik cihaz gündelik yaşamın bir parçası haline gelmiştir. Büyük bir kısmı çevresel verileri toplama kabiliyetine sahip sensörler barındıran bu cihazların ev ve işyerlerindeki geleneksel elektronik ve mekanik araçlara entegrasyonu da önceki yıllara oranla önemli ölçüde artış göstermiştir. Yapılan güncel bir anket çalışmasında, 2019 yılında Amerika'da evlerde ortalama 11 adet internete bağlı cihaz olduğu bildirilirken, bu sayının 2021'de ortalama 25'e kadar çıktığı tespit edilmiştir. Dünya ortalamasının da benzer ölçüde bir artış gösterdiği görülmektedir [1]. Tüm bu cihazlar, çeşitli amaçlarla internet alt yapısını kullanarak birbirlerine bağlanmış vaziyettedirler ve bir ağı temsil etmektedirler. Dünya çapında irili ufaklı milyarlarca cihazın internete bağlanarak ve veri transferleri yaparak oluşturduğu bu ağ, Nesnelerin İnterneti (Internet of Things / IoT) olarak tanımlanmaktadır. Artık bir sektör haline gelen bu yapı, endüstriyel seviyede 150 miyar doların üzerinde bir ekosisteme sahiptir ve market hacmi de her geçen gün artmaktadır. Şekil 1.1, günümüze kadarki ve ilerleyen yıllardaki muhtemel endüstriyel IoT harcamalarını göstermektedir. Bireysel kullanımlar da dahil olmak üzere tüm sektörün şimdiden 1,1 trilyon doların üzerinde bir ekosistem halini aldığı da bilinmektedir [2]. Bu artışın ilerleyen yıllarda da sürekliliği öngörülürken, internete bağlı cihazların ürettiği veri miktarında da oldukça büyük bir artış söz konusu olmuştur. Sensörler vasıtasıyla toplanan veriler, kullanım amacına göre değişkenlik göstermekle beraber, cihazlar üzerinde işlenmektedirler ve/veya merkezi platformlara iletilmektedirler.

Günümüzde artık IoT kavramının birçok yerde aktif olarak kullanılmasından ötürü gitgide bu ekosistemin ürettiği ve işlediği veriler hem bireysel kullanımlarda hem de endüstriyel uygulamalarda karar destek mekanizmalarını beslemektedir. Bu veriler ışığında çeşitli aksiyonlar alınmaktadır. Artan bu kullanıma karşın diğer bir yandan IoT sistemlerinin



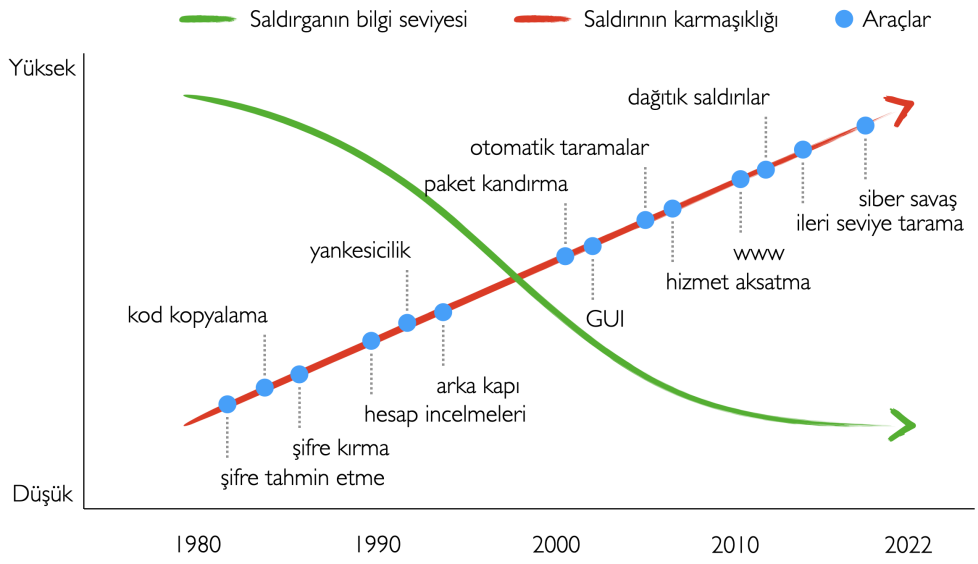
Şekil 1.1: Dünya çapında endüstriyel IoT teknolojilerine yapılan harcamalar [3].

günümüzde karşılaştığı ve hala üzerinde çalışılan bazı problemler vardır. Bu problemler çok farklı çalışma alanlarını içermekle beraber, içinde teknik altyapıyı ilgilendiren üç adet ana başlık olduğu değerlendirilmektedir. Bu başlıklar büyük veri, bağlantı ve güvenlik olarak sıralanabilir [4].

IoT ekosistemindeki cihaz sayısı ve bu cihazlardaki sensörlerin kapasiteleri arttıkça, üretilen veri miktarının yoğunluğu da epey artmaktadır. Buna karşın bu cihazlar tarafından üretilen büyük verilerin, geleneksel veri işleme teknikleriyle işlenmesi oldukça güçleşmektedir ve bu veriler ışığında uygulanacak süreçlerin karar verme adımlarının artık büyük verileri işleme kabiliyeti olan donanımlar ve algoritmalar tarafından yapılması ihtiyacı doğmuştur. Yapılan diğer bir güncel araştırmaya göre 2021 yılında tüm dünyadaki aktif IoT cihazı sayısı yaklaşık 35.8 milyar adetti [5]. Günümüzde ise her bir saniye ortalama 127 adet yeni IoT cihazının aktif olarak kullanıma alındığı ve toplam cihaz sayısının bu şartlar altında 2025 yılında yaklaşık 75 milyar adede çıkabileceği öngörülmektedir [6]. Bu cihazların üreteceği veri miktarının ise en az 79 zettabyte ($79x10^9$ terabyte[1]) olabileceği değerlendirilmektedir [7]. Üretilen büyük verinin işlenmesi konusundaki açığı ise günümüzde çoğunlukla dağıtık hesaplama mimarileri ve grafik işlem üniteleri kullanarak veri işlemeyi daha efektif hale getiren sistemler kapatmaya çalışmaktadır [8, 9, 10]. Üretilen ve işlenen verilerin daha akıcı ve problemsiz şekilde aktarımı ise günümüzde hala

[1] 1 zettabyte = 1 milyar terabyte

üzerinde çalışmalar yürütülen 5G gibi ileri seviye mobil bağlantı altyapıları ile giderilmeye çalışılmaktadır [11, 12, 13]. Güvenlik konusu ise bilgisayar ve internet altyapılarını kullanan tüm sistemler gibi IoT ekosisteminde de nihai bir çözümü olmayan ve sürekli üzerinde çalışılması gerekli araştırma alanlarından biri olarak kalmaya devam etmektedir. Zira kullanımın bu denli yoğun ve etkili olduğu bir sistem, içerisinde irili ufaklı maddi ve manevi değerleri de barındırmaktadır. Farklı üreticiler tarafından ekosisteme yerleştirilen ve dünya çapında güvenlik standartlarını sağlamayan, içi kırılgan olabilecek yapılardan ötürü IoT, kötü niyetli saldırganların da epey ilgi duyduğu bir ekosistem haline gelmiştir. Günümüzde artık karmaşık olarak nitelendirilebilecek saldırıların bile büyük bir deneyim gerektirmediği düşünüldüğünde (Şekil 1.2) güvenlik konusunun önemini hala koruduğu değerlendirilmektedir [14, 15, 16].



Şekil 1.2: Saldırı karmaşıklığı ve buna karşın saldırganların bilgi birikimi [17].

IoT ekosistemindeki güvenlik problemi hem saldırıların karmaşıklığının hem de bu saldırıların neden olduğu yıkımın etki değerinin büyüklüğü nedeniyle insan müdahalesini yetersiz kılmaktadır. Bu nedenle saldırılara karşı programatik ve sürekli manüel müdahale gerektirmeyen bir yaklaşımla çözümler üretme gerekliliği mevcuttur. Bu kapsamda araştırma ve endüstri seviyesinde yapay zeka destekli uygulamaların kullanımı, saldırı tespitlerinde ve saldırıların engellenmesinde insan müdahalesini en aza indirme

konusunda önemli unsurlardan bir tanesidir [18]. Bu alanda geliştirilen sistemlerin önemli bir kısmı yapay zeka konseptinin bir alt kümesi olan makine öğrenmesi yöntemlerini kullanmaktadırlar. Son yıllarda ise yüksek başarımlı sonuçlar elde edebilmek için özellikle makine öğrenmesinin bir dalı olan **Derin Öğrenme** (Deep Learning - DL) yöntemleri ile kapsamlı çalışmalar yapılmaktadır [19]. IoT ekosistemi, yüksek başarım elde edebilmek için veri hakkında örüntüler çıkaran ve büyük veri ile beslenmesi gereken Derin Öğrenme yöntemlerinin uygulanabilmesi için elverişli bir ortam sunmaktadır; ancak Derin Öğrenme, doğası gereği yoğun bir paralel hesaplama gücüne ihtiyaç duymaktadır ve kullandığı yapı taşları merkezi işlem üniteleri (Central Processing Unit - CPU) yerine grafik işlem üniteleri (Graphics Processing Unit - GPU) üzerinde en iyi çalışacak şekilde tasarlanmışlardır.

Bu tez çalışmasında temelde yoğun hesaplama gücü ve paralel işlem kabiliyetine sahip grafik işlem üniteleri üzerinde Derin Öğrenme yöntemleri ile çözülebilecek bir problem ortamı geliştirilmeye çalışılmıştır. Bu bağlamda temel tanımlardan başlayarak, yapılan güncel diğer araştırmalara, önerilere ve çıktılara ilerleyen bölümlerde yer verilmiştir. Çalışmanın problem alanı olarak IoT ekosisteminde yukarıda bahsedilen üç ana problem içindeki güvenlik konusu ele alınmıştır ve detaylandırılan sorunlara yönelik çeşitli çözüm yöntemleri üzerine odaklanılmıştır. İlgili önerilerin deneysel çıktıları literatürdeki diğer çalışmalarla kıyaslanarak alt başlıklar halinde sunulmuştur.

1.1 Yapay Zeka, Makine Öğrenmesi ve Güvenlik

Yapay zeka, hakkında tek bir evrensel tanım yapılamayacak kadar karmaşık olan, kullanıldığı disiplinlerin kendi bakış açılarıyla farklı tanımlarla ifade edilebilen ve kapsayıcılığı geniş olan bir araştırma konusudur [20]. Bilgisayar bilimi perspektifinden ve bu çalışmanın kapsamı açısından yapay zekaya bakıldığında, yapay zeka amaca yönelik verilerle beslenen ve bu veriler ışığında belli matematiksel işlemler yürütüp bunlar neticesinde sayısal çıktılar veren algoritmalar olarak ifade edilebilir [21, 22].

Yapay zeka alanında bilgisayar bilimleri açısından çözüme ulaştırılması hedeflenen

temel problem; görme, duyma ve anlamlandırma gibi insanlar tarafından sezgisel olarak icra edilen ancak matematiksel formülü tam anlamıyla yapılamayan işlevlerin bilgisayarlar tarafından da yapılabilecek şekilde programlanmasıdır [23]. İnsanların günlük yaşamlarında gözlemledikleri, öznel olarak algıladıkları ve bu veriler ışığındaki çıkarsamalar ile kendileri için oluşturdukları bilgi birikiminin (knowledge) bilgisayarlar tarafından nasıl toplanması ve işlenmesi gerektiği ise uzun yıllardır yürütülen aktif bir araştırma konusudur. Bu amaçla geçmişte yapılan çalışmalarda uzmanların denetiminde dünyadaki yaşamın temel kuralları biçimsel diller ile elle kodlanarak kural tabanlı bir yapıda bilgisayara tanıtılmış ve bilgisayarın bu kurallar ışığında, mantıksal çıkarsamalar ile ellerindeki yeni bilgileri anlamlandırmaları beklenmiştir [24]. Bilgi tabanlı (knowledge base) yaklaşım olarak bilinen bu yöntem ile bilgi tabanındaki veriler arttıkça sistemin başarımının düştüğü ve yeni koşullara uyum sağlanabilmesi için tekrar tekrar uzman denetiminde verilerin güncellenmesi gerekliliği ortaya çıkmıştır [24]. Bu nedenle yapay zeka sistemlerinin ham veriler üzerinde kendi bilgi birikimlerini oluşturması gerekliliği ön plana çıkmıştır. Bu da aslında makine öğrenmesi konseptinin yapay zekanın bir alt dalı olarak çıkış noktasını oluşturmaktadır.

Makine öğrenmesi, en basit tabir ile, bilgisayarların gerçek dünyadan elde edilen veriler ışığında ellerindeki problemler için öznel kararlar vermesi olarak tanımlanabilir. Örneğin, en temel makine öğrenmesi algoritmalarından olan lojistik regresyon (logistic regression) ile çeşitli özellikleri verilen evlerin ve fiyatlarının bulunduğu bir verisetini sisteme öğretip, bu doğrultuda elde örneği bulunmayan özelliklere sahip evlerin fiyatları tahmin edilebilmektedir [25] veya bir hamilenin doğumunun sezaryen mı yoksa normal doğum şeklinde mi yapılması gerektiğinin tespiti yapılabilmektedir [26]. Ayrıca yine temel algoritmalardan olan naif bayes (naive bayes) sınıflandırıcısı ile normal e-postalar ve spam e-postalar arasında ayrım yapılabilmektedir [27].

Makine öğrenmesi algoritmalarının kullanımındaki önemli nokta, bu algoritmaları besleyecek verilerin nitelikli olmasından geçmektedir. Nitelikli veri, çözülmesi hedeflenen problemin yapısına ve kullanılacak algoritmaya uygun veri kümesi olarak

tanımlanabilir. Örneğin ev fiyatlarının tahmini için lojistik regresyon fonksiyonuna evin yaşı, kaç metrekare büyüklüğünde olduğu, oda sayısı, kaç katlı olduğu, bahçesinin olup olmadığı ve merkeze yakınlığı gibi çeşitli girdiler nitelikli veri olarak gösterilebilir. Çünkü buradaki değerler birbirleri arasında ilişkilendirilebilir ve evin fiyatına etki edebilirler. Bu değerler ev nesnesinin öznitelikleri (features) olarak adlandırılmaktadırlar. Karşı bir örnek olarak; evin fotoğrafının lojistik regresyon fonksiyonuna verilmesi mantıklı olmayabilir, çünkü fotoğraf içerisinde yer alan piksellerin birbirleri ile olan ilişkileri ev fiyatına etki etmeyebilir. Aynı zamanda bir fotoğraf verisinin işlenmesi de tekil sayısal değerlere göre daha uzun vakit alabilmektedir. Dolayısıyla bu veri, eğer farklı alana odaklanmış modellerin birleşimi meydana gelmezse bu problem için nitelikli değildir yargısına varılabilir. Bu örnekten yola çıkarak, makine öğrenmesi algoritmalarına sağlanan verilerin gösterim şeklinin, algoritmaların çalışma hızına ve başarımına önemli bir etkisinin olduğu söylenebilir.

Algoritmalara sağlanan verilerin öneminin büyük olması, kendi içinde yeni bir problem doğurmaktadır: çözülecek problem ile ilgili hangi verileri çekmeliyiz? Örneğin verilen fotoğraflar içinde siyah renkli bisikletleri tespit etmek istiyoruz. Bu durumda ilk olarak siyah rengin, elimizdeki sayısal piksel değerlerine göre neye benzediğini rahatlıkla program içinde ifade edebiliriz. Ayrıca bir bisikleti tanımlayabilmek için bisikletin bir şasesinin, direksiyonunun ve iki tekerleğinin olduğunu ve bunların birbirlerine göre konumlarının ne olduğunu da biliyoruz; ancak elimizde yalnızca 0-255 aralığında kırmızı, yeşil ve mavi sayısal değeri olan NxN adet pikselden bir bisikleti belli kurallara göre çıkarmamız oldukça güç olabilir. Burada ele alınması gereken birkaç unsur vardır. Bunlardan bazıları; güneş ışığının bisiklete vurma açısı, dolayısıyla piksellerin değerlerinin değişimi, bisikletin fotoğrafının hangi açıdan çekildiğine ve direksiyonun hangi yönde olduğuna göre şekillerin güncel pozisyonlarının yeniden tanımlanması ve birden fazla bisiklet olması durumunda fotoğrafta iç içe geçmiş geometrik şekillerin aslında her birinin ayrı bir bisiklet olduğunun belirlenmesi. Böyle durumlarda bilgi tabanlı yaklaşımların kullanılması problem çözümünü zorlaştırmaktadır. Bu nedenle il-

gili şekillerin temelde neyi temsil ettiğine dair öznitelik çıkarımı yapmaya olanak veren ve bu öznitelikler yardımıyla bu şekilleri (şase, direksiyon, tekerlek) farklı boyutlarda ve açılarda yeniden üretebilecek bir yapı olan temsili öğrenme (representation learning) yöntemi kullanılmaktadır. Bu bağlamda otomatik kodlayıcı (autoencoder) ismi verilen yapılar ile bir girdinin boyutunun indirgenmesi ve boyutu indirgenmiş bu girdinin tekrar eski haline getirilmesini sağlanmaktadır. Burada boyut indirgemedeki kasıt, ilgili girdiyi temsil edebilecek bir özetin çıkarılmasıdır. Bu sayede bu özet üzerinde işlemler yapılarak ilgili girdi ve onun farklı varyasyonları üretilebilecektir. Buna örnek olarak bir tekerleğin farklı ışık açılarına veya görünüm açısına rağmen yine de tekerlek olarak tanınabilmesi verilebilir. Temsili öğrenme yöntemi ile yapay zeka sistemlerinin yeni görevler için daha az müdahale ile daha geniş ölçekli çözümler ürettiği gözlenmiştir.

Temsili öğrenme yöntemi çeşitli girdilerde başarılı sonuçlar verse de, problemler için verilen girdilerin etkilendiği farklı durumlar olabilmektedir. Örneğin bir fotoğrafa vuran güneş ışığının veya tanımlanması beklenen bir nesnenin önünde durup o nesnenin bir kısmını kapatan bir engelin sistem başarımını önemli ölçüde düşürebileceği bilinmektedir. Aynı şekilde ses tonları da bir kişinin uykudan yeni uyanmış olması, gün ortasında olması veya sağlık durumunun farklılık gösterebilmesi gibi nedenlerden dolayı aynı kişi için farklılık gösterebilmektedir. Bu tarz çevresel faktörlerin olumsuz etkilediği durumlarda ise ilgili çevrenin problemin çözümüne yönelik girdi olabilecek özniteliklerinin belirlenmesi ve bu özniteliklerin özetlerinin çıkarılması zorlaşmaktadır. Çünkü çevresel faktörlerden etkilenen yanlış bir öznitelik çıkarımı, problemin çözümünü de olumsuz etkileyecek örnekler doğurabilir. Öznitelik ve özet çıkarımının zor olduğu veya mümkün olmadığı böyle durumlarda da temsili öğrenme yönteminin başarımı düşük seviyelerde kalmaktadır.

Bahsedilen farklı öğrenme tiplerinin dışında yaklaşık son on yıldır araştırmacıların artan bir ilgiyle kullanmaya başladığı ve geleneksel tekniklerin ötesinde bir mimariye sahip olan Derin Öğrenme ise farklı bir yaklaşımla öğrenmeyi hedeflemektedir. Derin öğrenme yaklaşımı, problem içerisindeki girdileri, onları tanımlayabilecek daha basit parçalara

bölerek ve ardından bu basit parçaları birleştirerek temsili öğrenme yaklaşımındaki özellik çıkarımı problemini ortadan kaldırmaktadır. Temelde bir girdiden fazla sayıda basit parçalar elde edip, ardından bu parçaların kombinasyonları ile hem ilgili girdi tekrar üretilebilmekte hem de bu girdinin, parçaların birleşmesinden doğan çeşitli varyasyonları çıkarılabilmektedir. Burada basit parçalardan kasıt, eğer girdimiz fotoğraf ise çizgiler ve köşelerden oluşmaktadır. Çeşitli açılarda bulunan çizgiler birleşerek basit şekilleri oluşturmakta ve bu şekiller de ilerleyen kombinasyonlarda daha karmaşık şekilleri ve en nihayetinde asıl girdiyi oluşturabilmektedirler. Eğer girdimiz bir örüntü (pattern) ise, buradaki basit parçalarımız örüntünün en küçük parçalarının kendi aralarında kombinasyonu ile oluşturulan yeni küçük örüntülerdir ve bunların birleşimiyle oluşturulan daha karmaşık örüntülerdir. Bu sayede belirli bir veri akışının temel hareketleri ve aynı veriyi temsil edebilecek, olması muhtemel diğer hareketler öğrenilebilmektedir. Yine başka bir örnek olarak, girdimizin bir zaman serisi modeli olması durumunda bu girdiden çıkarılabilecek basit parçalar, zamana bağlı olarak girdinin hangi taneciklerinin aktif olduğudur. Dolayısıyla birbirini izleyen farklı zaman aralıklarında hangi taneciklerin hangileri ile kombinasyon oluşturabileceği ve bunlardan doğan asıl girdimiz ve varyasyonları tanınabilmektedir. Buna somut örnek olarak sesten metne çeviri verilebilir. Belli bir lisanda eğitilen ses verilerinden, bu lisandaki cümle yapısı öğrenilebilir, hangi kelimelerden sonra hangi kelimelerin gelebileceği öğrenilebilir hatta kelimelerdeki zaman eklerinin tanınması sağlanıp, daha önce görülmemiş bir ses verisi işlenirken bir sonraki kelimenin ne olacağı tahmin edilebilir. Çok çeşitli alanlarda uygulanabilecek bu yapılar temelde Derin Öğrenme yaklaşımının farklı sahalardaki problemlere uyarlanmasından ibarettir. Derin öğrenmenin bu denli bir öğrenme sürecini takip etmesinden ötürü neredeyse problemden bağımsız bir çözüm havuzu oluşturduğu gözlenmektedir. Bu çalışma kapsamında Derin Öğrenme yaklaşımının çeşitli kaynaklardan akan sensör verileri ile harmanlanması ve bu veriler üzerinde örüntüler tespit edilmesi hedeflenmiştir.

Sensörlerden gelen verilerin işlenmesini ve internete aktarımını sağlayan irili ufaklı cihazların bir ağ ortamında kolektif çalışması özellikle endüstriyel uygulamalarda uzun

süredir yaygın olarak görülmektedir. Son yıllarda ise sensör teknolojilerinin azalan maliyetleri ve genişleyen uygulama alanlarıyla beraber artık son kullanıcıların elektronik cihazlarında da karşılaştığımız ve nesnelerin interneti (IoT) olarak tanımlanan bu yapı birçok açıdan günümüzün önemli teknoloji konseptlerinden biri haline gelmiştir. IoT teknolojisi çok çeşitli uygulama alternatifleri sunarak gündelik yaşamımızın bir parçası olmuştur. Savunma ve güvenlik sanayiinde, modern ulaşım araçlarında, teknolojik donanımlı ev ve iş yerlerinde ve tüketicilerin kullandığı beyaz eşya, küçük ev aletleri ve mobil iletişim cihazları gibi neredeyse günlük yaşamda görülebilecek her alanda sensörlerle donatılmış ve internete bağlı IoT cihazlar karşımıza çıkabilmektedir. IoT cihazlarının çeşitli amaçlarla internete bağlı olması, aslında cihazların kendi görevlerinin yanı sıra, uygun koşullarda her yerden ve her zaman için erişilebilir olabileceği anlamına da gelmektedir. Bu nedenle internet ağına bağlı tüm yapılarda olduğu gibi IoT cihazlarının da güvenlik açıkları nedeniyle kişisel bilgi sızıntıları, finansal kayıplar ve daha birçok farklı potansiyel tehditlerden ötürü kullanıcıların ve kurumların günlük yaşamını etkileyebilecek endişe verici unsurları bulunmaktadır.

Geleneksel anlamdaki siber saldırılar; temel bilgi sızıntıları, finansal çıkarlar ve sistemleri devre dışı bırakma gibi tek seferlik veya kısa süreli amaçlara hizmet ederken, günümüzün siber saldırganları artık daha uzun süreyle sistemlerde hiç fark edilmeden kalıp daha büyük çıkarlar için elektronik ortamlara sızmaktadırlar. Hatta bu tip yeni nesil saldırılardan ötürü devlet ve askeri düzeyde gizli bilgi ve belgelerin bile elde edilmesi kaçınılmazdır. 2019 yılında yapılan bir haberde CNBC, siber saldırıların şirketlere ortalama 200.000$'a mal olduğunu ve tehditlerin ortalama 101 gün boyunca tespit edilemediğini bildirmiştir [28]. Bu süre görece çok uzun olmakla beraber, saldırıların veya saldırganların tespit edilmiş olmasına rağmen sistemlerde gerçekleşen hasarların boyutlarının anlaşılması pek kolay olmamaktadır. WikiLeaks'in 2006 yılında bazı hükümetlerin hassas ve korunan verilerine erişim sağlamış olması ve Stuxnet yazılımının 2010'da İran'ın nükleer araştırmalarını aksatması [29] gibi çok üst düzey olaylar göz önüne alındığında, siber açıkların dünya çapında nelere sebep olabileceği ve bunlarla nasıl

başa çıkılabileceği konusu daha bir önem arz etmektedir. Bu tarz haberlere yansımayan diğer birçok irili ufaklı siber saldırılar, bilgi güvenliği ve siber savunma alanında belirli politikaların oluşturulmasını zorunlu kılmıştır. Aynı zamanda teknolojik imkanların ve buna bağlı olarak siber saldırı çeşitlerinin de gelişmesiyle beraber, artan saldırılara karşı bu politikaların gün geçtikçe sıkılaştırıldığı da gözlenmektedir [30, 31, 32]. Gizlilik dereceli bilgilere veya endüstriyel seviyedeki alanlara yapılabilecek potansiyel saldırıları bir kenara bıraktığımızda, IoT cihazlarının bireysel yaşam alanlarımızla dış dünya arasında bir kapı oluşturduğu gerçeği de göz önündedir. Bu da saldırganların kurumların haricinde artık bireylere veya kitlelere saldırma potansiyelini de oldukça artırmıştır. 2016 yılında Finlandiya'nın Lappeenranta kentindeki iki apartmanın çevresel kontrol sistemleri (ECS) Dağıtık Hizmet Engelleme (DDoS) yöntemiyle bir saldırıya uğramış ve bir kış döneminde iki gün boyunca apartmanlar kelimenin tam anlamıyla soğukta kalmıştır. Bu saldırı sonucunda iki apartmandaki ECS devre dışı kalmıştır [33]. 2017 yılında Kuzey ABD'de bir kumarhanenin sunucularından veri çalmak için Finlandiya merkezli bir saldırı düzenlendiği tespit edilmiştir. Siber saldırganlar, kumarhanede yer alan bir akvaryumun sıcaklığını ve temizlik seviyesini ayarlamak ve gözlemlemek için yerleştirilmiş internete bağlı sensörlerle donatılmış bir işlem birimine sızarak oradan kumarhanenin iç ağına erişmişlerdir [34]. 2015 ve 2016 yıllarında tanınmış otomobil markalarından Jeep'in güvenlik ve güncelleme amaçlı internete bağlı modüllerinden biri üzerinden araca sızmayı deneyen siber saldırganlar aracın sensörlerinden veriler alıp, aynı zamanda aracın frenleri ve direksiyonu gibi bazı elektronik parçalarını uzaktan kontrol edebilmeyi başarmışlardır. Elektronik aksamı kontrol etmek için aracın içinde yer alan CAN bus olarak adlandırılan iç ağı kullanmışlardır [35]. Bunların dışında, ev ortamlarında yer alan güvenlik kameralarının ve bebek kontrol kameralarının da çeşitli zamanlarda saldılara uğrayıp casusluk amaçlı kullanılabildiği bilgisi sıkça rapor edilmiştir [36, 37, 38]. Bahsi geçen bu örneklerin hepsi ve daha fazlası gerçekte yaşanmıştır ve bu haberlerin yaygınlaşma hızının da artmasıyla beraber artık internete bağlı tüm cihazların güvenlik mekanizmalarının daha da güçlü olması gerekliliği desteklenmiştir. Bu süreçler yaşanırken, internete bağlı ci-

hazların ağını korumak amaçlı çeşitli anti-virüs, güvenlik duvarı ve saldırı tespit sistemleri geliştirilmiştir. Bu sistemler ağ altyapısının güvenliğini sağlamak için şirketler ve hükümetler tarafından uzun süre kullanılmıştır.

1.2 Tez Çalışmasının Amacı ve Önemi

Bu çalışma, temelde grafik kartları üzerindeki paralel mimaride Derin Öğrenme konsepti ile bir problem çözmeyi hedeflemektedir. Bu kapsamda, son yılların sıcak başlıklarından biri olan ve önemi uzun yıllar daha katlanarak devam edecek olan IoT ve bilgisayar ağları üzerinde gerçekleştirilen saldırı tiplerinin tespit edilip sınıflandırılması üzerine bir uygulama yürütülmüştür. Çalışma kapsamında öncelikle yaygın olarak kullanılan güvenlik mekanizmalarından biri olan Saldırı Tespit Sistemlerinin (STS) de kullandığı olay kayıtları (event logs) dikkate alınmıştır. STS'ler; güvenlik duvarı ve anti-virüs yazılımı gibi önceleme mekanizmalarını aşan saldırıları tespit etmek amaçlı bilgisayar sistemlerinin ağ trafiğinin akışını kontrol eder ve ağın olağan akışının dışında bir hareket olup olmadığını tespit etmeyi amaçlar [39]. Buna rağmen, geleneksel STS mekanizmaları, yapıları gereği veritabanlarında yer alan en güncel kurallar kadar esnek ve kaliteli tespitler yapabilmektedir. STS veritabanlarında veya kurallarında daha önceden hiç yer almamış yeni ve karmaşık saldırı tipleri, bu mekanizmaları da aşıp sisteme girmeyi başarabilmektedirler. Her yeni tip saldırıların ardından STS'lerin kuralları ve veritabanları, üreticileri tarafından güncellenmektedirler. Buna rağmen bu mekanizmaların kurallarını ve davranışlarını da iyi analiz edebilen saldırganlar kendi tekniklerini değiştirip STS'leri aşmanın yolunu bulabilmektedirler. Bu noktada olası güvenlik ihlallerinin üstesinden gelmek için verilerin doğal akışından öğrenebilen bazı akıllı algoritmalara ihtiyaç duyulmaktadır. Buna istinaden, herhangi bir dış müdahale veya el parametre olmadan, verinin olağan akışının yapısını öğrenebilen Derin Öğrenme modelleri STS ve benzeri güvenlik mekanizmalarını güçlendirmek adına bilgisayar ağ trafiğinden çıkarımlar sağlayıcı olarak kullanılmaya başlanmışlardır. Bu mekanizmalar geleceğe yönelik kaliteli tespit mekanizmaları oluşturulmasında büyük rol oynamaya

başlamışlardır [40].

Bu çalışmada, bilgisayar sistemlerinin ağ trafiğini ele alarak ağ saldırılarını yüksek doğruluk ve F_1 puanıyla sınıflandırmak için bir evrişimli özellik çıkarıcı katmanı içeren ve kendi kendini normalleştiren (self normalizing) derin yapay sinir ağı (Deep Neural Network - DNN) [41] modeli tasarlanmıştır ve modelin başarımını ölçebilmek ve karşılaştırılabilir sonuçlara ulaşabilmek için güncel ağ trafiği verisetlerinden biri olan CICIDS2017 verisetine uygulanmıştır. Tercih edilen veriseti ile yakın tarihli literatürde bu veriseti üzerinde benzer çalışmalar yürütmüş diğer araştırmalar ile kıyaslamalar yapılmıştır. Bu veriseti içinde toplam 14 adet saldırı sınıfı bulunmaktadır ve bu sınıflardan bazılarının örnek sayısı diğerlerine göre %99.99'a varan oranlarda daha az olabildiği için verisetindeki sınıf dağılımı oldukça dengesiz bir yapıdadır. Bu nedenle bu çalışma kapsamında nadir görülen sınıf örneklemlerinin çeşitli sentetik veri üretme algoritmaları kullanılarak çoğaltılması yönünde de birtakım önermeler ve denemeler yer almaktadır. Önerilen ve uygulanan veri artırma yöntemlerinin, Derin Öğrenme modelinin hem nadir görülen sınıf türlerindeki düşük performansını iyileştirdiği hem de dolaylı olarak nihai çıktıyı iyileştirdiği deneylerle gözlenmiştir. Bu çalışma kapsamında üç adet veri artırma yöntemi değerlendirilmiştir. Bunlardan biri çalışmanın kendi önerdiği bir rastgele veri üretme fonksiyonu, diğeri SMOTEENN tekniği ve sonuncusu ise tablo verisi üzerinde çalışmaya olanak tanıyan bir çekişmeli üretici ağ (Conditional Tabular Generative Adversarial Network - CTGAN) [42] modelidir. Veri artırma sonuçları da eklendiğinde, deneysel sonuçlara göre bu çalışma kapsamında önerilen her yöntemin kendine has bir çıktıya sahip olduğu ve gözlemlenen literatür dahilinde karşılaştırılan diğer sonuçlara göre en yüksek verimleri gösterdiği değerlendirilmiştir.

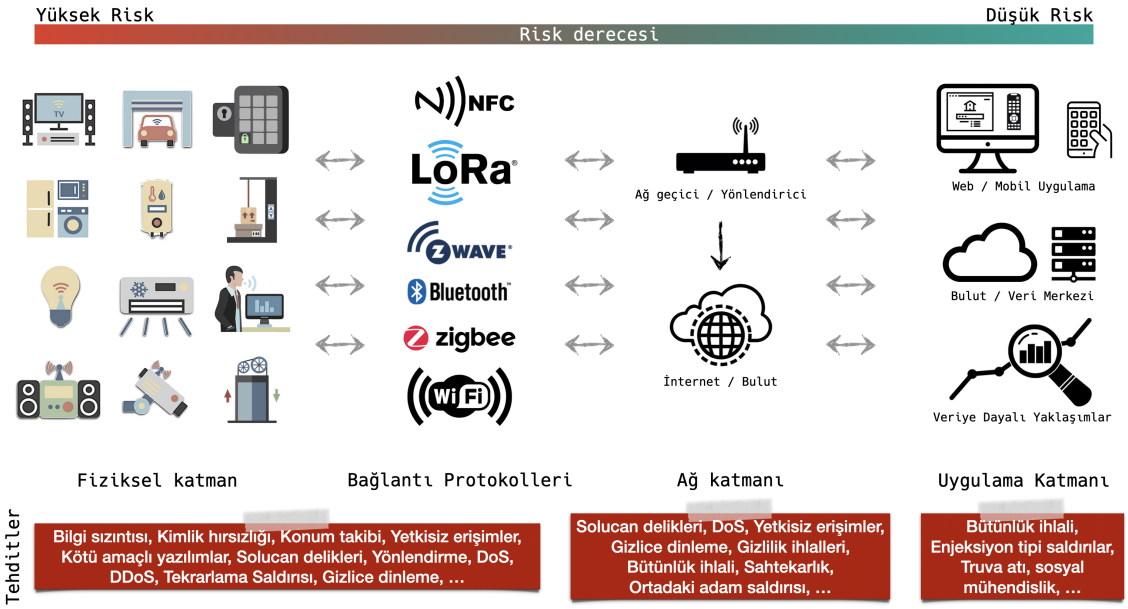
Bu çalışmanın geri kalan kısmı şu şekilde düzenlenmiştir. Başlık 1.3 ile bir IoT sisteminin genel tanımı ve IoT sistemlerindeki güvenlik endişeleri ifade edilmektedir. Başlık 1.4, çalışmanın ana konusu olan saldırıların günümüz teknolojileri ile nasıl tespit edilebildiğini STS'leri tanıtarak açıklamaktadır. Başlık 1.5 ile yakın tarih literatüründeki ilgili çalışmalara değinilmiştir. Başlık 2.1, literatürde geçen ve saldırıların tespiti üzerine

yapılan çalışmaların kullandığı çeşitli verisetleri üzerinde genel bir bakış sunmaktadır. Başlık 1.6 içerisinde çalışma esnasında kullanılan Derin Öğrenme mimarisinin tarihçesi, alt kırılımları ve sonraki bölümde GPU mimarisinin tarihçesi ve çalışma yöntemleri ile ilgili açıklamalar yer almaktadır. Bölüm 3, önerilen Derin Öğrenme modelinin mimarisini açıklamaktadır. Bölüm 4, Derin Öğrenme modelinin ham halinin çıktısını ve ardından modelin başarımını iyileştirmek adına izlenen veri artırma deneylerini sonuçlarıyla beraber açıklamaktadır. Çıktılar diğer makine öğrenmesi algoritmalarından seçilen bazıları ile kıyaslanmış ve başlık 4.3 ile bu çıktılara dair yorumlamalar yapılmıştır. Bölüm 5 bu çalışmanın sonunda elde edilen çıkarımlarla gelecek çalışmalarda ne yapılabileceğini, gelecek çalışmalara nasıl ışık tutabileceğini değerlendirmektedir.

1.3 Nesnelerin İnterneti

Nesnelerin İnterneti (IoT) teriminin aslında standart bir tanımı olmamasına rağmen, Nesnelerin İnterneti dendiğinde genel olarak "şeyler" adı verilen ve benzersiz şekilde kimliklendirmesi (ID) olan fiziksel cihazlardan oluşan bir ağdan söz edilir. Bu cihazlar, üzerlerine bağlı olan sensörler aracılığı ile ortamdan veri okurlar ve/veya ortamı izlerler. Alınan veriler daha sonra çeşitli karar mekanizmalarında kullanılmak üzere işlenir veya işlenmeden tutulabilir. Hatta diğer cihazlarla iletişim kurulur ve kolektif algılamalar ve hareketler elde edilebilir. Bu sayede bu cihazlar tarafından işlenen verilerle bazı görevler cihazların üzerinde veya bulutta çeşitli uygulamalarda kullanılabilir [43].

Kişisel kullanımdan çok büyük ölçekli endüstriyel otomasyon görevlerine kadar birçok ölçeğe uzanan IoT uygulamaları, makineler, sistemler ve ayrıca insanlar arasında güvenilir bir etkileşim oluşturma kabiliyetine sahiptir. IoT uygulamaları, giyilebilir cihazlar, bina ve ev otomasyonu, akıllı şehirler, akıllı üretim, sağlık, otomotiv ve diğer küçük veya büyük ölçekli özel ve endüstriyel ihtiyaçları içerir. Bunlarla sınırlı olmamakla beraber, daha birçok alanda IoT ile faaliyetler geliştirilmektedir [44]. Ulaşım, bilgi işlem hizmetleri, perakende ve toptan satış, bankacılık ve menkul kıymetler, sigorta vb. gibi diğer birçok market, IoT sistemlerini artan bir ivmeyle kullanmaktadır [44]. IoT An-



Şekil 1.3: IoT mimarisi ve katmanlar özelinde muhtemel tehdit çeşitleri.

alytics [45] tarafından yakın zamanda yapılan bir araştırmaya göre, 2020'deki en önemli IoT kullanım örnekleri arasında endüstri, ulaşım, enerji, perakende, akıllı şehirler, sağlık, tedarik zinciri, tarım ve bina uygulamaları yer almıştır.

İlgili araştırma [45], 620 IoT platformundan 1400 aktif IoT projesini araştırmış ve COVID-19 salgını sırasında tedarik zinciri gibi bazı uygulama alanlarının kullanım ve değerinin daha fazla önem kazandığını söylemiştir [2]. Araştırmada yer alan, her bir uygulama alanı için ortak kullanım durumları şu şekilde özetlenebilir:

- Endüstriyel uygulamalar: Uzaktan PLC kontrolü, otomatik kalite kontrol sistemleri, ekipman izleme, üretim alanı izleme, giyilebilir cihazlar vb.
- Ulaşım uygulamaları: araç teşhis / izleme (akü, lastik basıncı ve sürücü izleme vb.).
- Enerji uygulamaları: Şebeke optimizasyonu, uzaktan cihaz izleme, önleyici bakım (predictive maintenance).
- Perakende uygulamaları: Akıllı otomatlar, müşteri katılımı, ürün ve envanter izleme vb.

[2] COVID-19 pandemi sürecinin, yeni normalleşmeden ötürü tüm alanlarda olduğu gibi IoT alanında da karşımıza yeni ihtiyaçlar ve uygulamalar çıkarmaya başladığı gözlenmektedir.

- Akıllı şehir uygulamaları: Çevresel izleme, video ile güvenlik, trafik yönetimi.
- Sağlık uygulamaları: Tıbbi cihaz izleme, destekli yaşam, giyilebilir cihazlar vb.
- Tedarik zinciri uygulamaları: varlık takibi, soğuk zincir izleme, envanter yönetimi vb.
- Tarım uygulamaları: Hassas tarım, alan haritalama, canlı hayvan izleme, kalite kontrol vb.
- Akıllı bina uygulamaları: Akıllı ev sistemleri (aydınlatma, alarmlar, güvenlik mekanizmaları, çeşitli sensörler), bina kullanımı, tüketim izleme vb.

Pandemi sürecinin neden olduğu teknolojik değişiklikler bu çalışmanın kapsamı dışında olsa da, IoT uygulamaları pandemi ile mücadelede önemli bir rol oynamış ve bu tür önemli olayların bir teknoloji dalının kullanımını nasıl etkilediği konusunda araştırmacılara bazı izlenimler vermiştir. Bazı araştırmaların belirttiği gibi, mevcut IoT uygulamaları yeni normalleşme süreçlerinde çeşitli amaçlar için kullanılmaya başlanmıştır. Araç trafiğini yönlendirmek ve büyük kalabalıkları en aza indirmek için geliştirilen akıllı zamanlama işlerinde IoT akıllı şehir uygulamaları kullanılmaya başlanmıştır. Çeşitli özel, kamu ve hastane binalarının kritik noktalarına yerleştirilen termal kameralar ve ateş ölçerler ile binalara ve alanlara giren kişilerin vücut sıcaklıkları ve hastalık durumları tespit edilmeye çalışılmıştır. Belirli uygulamalar içeren akıllı telefonlar, enfekte kişileri işaretleyip, bir enfeksiyon alanı haritası oluşturmak için konumlarını takip etmek amaçlı kullanılmıştır. Bu uygulamalar vasıtasıyla ayrıca, insanlar kritik binalara girmeden önce QR kodlarını okutmuştur. Bu sayede hastalık veya temas durumlarını tespit etmek amaçlı hızlı tanımlamalar sağlanmıştır [46]. Birçok hastanede doğru tedaviler sağlayan hızlı tanı araçları ve herhangi bir acil durum sırasında ilgili sağlık personelini bilgilendirmek için küçük sesli uyarı cihazlarını çeşitli amaçlar için kullanmaya başlanmıştır. [47].

IoT sistem altyapısı genellikle üç ana katmana dayanır. *Fiziksel algı katmanı*, fiziksel bir cihazın çevresinden algılama yaptığı katmandır. *Ağ katmanı*, desteklenen iletişim

ve bağlantı protokollerinin birleşimidir. Bu katman ayrıca verilerin kaynaktan hedefe nasıl iletileceğini de tanımlar. *Uygulama katmanı*, kitleler için mevcut olan içeriğe duyarlı uygulamalar ve hizmetler oluşturur [48]. Bu temel mimari, bir IoT sisteminden beklenenin ne olduğuna bağlı olarak genişletilebilir bir mimaridir. Mimari ayrıca olası tehditlerin nerede bulunduğunu ve sistemin sağlığını nasıl etkilediklerini de tanımlar. IoT cihazlarının çoğu aslında zaten var olan teknolojilerin birleşimi olduğundan, uygulama ve ağ katmanları temel iletişim protokollerini içerir ve IoT ekosisteminden bağımsız olarak kullanılabilirler.

IoT sistemlerine karşı bilinen çeşitli saldırı türleri vardır. Ancak IoT sistemlerindeki heterojenlik nedeniyle saldırıların etkisi tam olarak ölçülemez. Böylece güvenliğin sağlanması ve standardizasyon [49] daha zorlu hale gelmektedir. Belirli katmanlar için olası tehditlerin çoğunu ve risk derecelerini içeren tipik bir IoT mimarisi Şekil 1.3'da gösterilmiştir.

Küçük ya da büyük ölçekli fark etmeksizin birçok alanda günlük hayatın içine dahil olan IoT sistemlerinde, ne amaçla kullanıldığına bağlı olarak çeşitli zorluklarla karşılaşılmaktadır. En temel zorluklar; mimari, ulaşılabilirlik (availability), güvenilirlik (reliability), mobilite, performans, yönetim, ölçeklenebilirlik, birlikte çalışabilirlik (interoperability), güvenlik, gizlilik vb. şeklinde genel olarak tanımlanabilir. Bu çalışmada, IoT dünyasındaki en büyük endişeleri ve temel unsurları içeren her zaman ulaşılabilirlik, gizlilik ve bilgi güvenliği [49, 50, 51] konularına odaklandık.

IoT sistemlerinin ulaşılabilirliği ve güvenliği hakkında artan bir ilgiyle literatürde yayınlar çıkmaktadır. Yakın tarihli bir çalışmada, Tawalbeh ve arkadaşları [50], modern IoT sistemlerinin, tüm sistemi güvenlik ihlallerine karşı daha savunmasız hale getiren geleneksel bilgisayarlardan farklı olduğunu belirtmiştir. Selamat [51], IoT güvenliğinin hala üzerinde aktif çalışılmakta olan açık zorluklarını araştırdı ve uygulama alanıyla ilgili olası tehditleri kategorilere ayırdı. Araştırmaya göre, hizmet reddi saldırıları, yetkisiz erişim, gizlilik ihlali, gizlice dinleme ve kötü niyetli kod enjeksiyonu gibi kötü niyetli saldırılar çoğu durumda karşımıza çıkan olası tehditler arasında yer almaktadır.

Bu tehditler, bilgi güvenliğinin üç temel kavramını doğrudan hedef alır: sistemlerin gizliliği, bütünlüğü ve ulaşılabilirliği. [50, 52] yazarları, büyük ölçekte çalışan birçok IoT sisteminin benzer yazılım ve donanım bileşenlerine dayalı neredeyse aynı bilgi işlem cihazlarından oluştuğunu açıklamaktatır. Bu nedenle, herhangi bir güvenlik ihlali, aynı anda önemli sayıda cihazı etkileyebilir. Benzer şekilde, cihazlar arası otomatik ara bağlantıların seviyesi ve düzensizliği nedeniyle, herhangi bir siber saldırının bir IoT sistemine etkisi de bağlantı sayısı bilinmediğinden tahmin edilemez hale gelebilir. Bu nedenle, bir IoT sisteminde yer alan bir zayıf halkada herhangi bir güvenlik açığı olması durumunda, bilgi güvenliğinin tüm temel kavramları risk altına girer. Benzer şekilde, Hussain ve arkadaşları [53], IoT sistemleri için güvenlik gereksinimlerini ve mevcut çözüm yaklaşımlarını incelemiştir. Yazarlar, mevcut geleneksel birçok çözümün bu sistemler için tüm güvenlik yelpazesini kapsamakta yetersiz kaldığını; ancak bazı yeni tip makine öğrenmesi yaklaşımlarının bir dereceye kadar çözüm sunabileceğini de belirtmişlerdir. Çalışma, araştırdığı tehditleri şu yedi kategoride sınıflandırmış ve kapsamlı bir şekilde detaylandırmıştır: 1) Saldırganların değişken yöntemler kullanarak fiziksel cihaza erişebildiği ve cihaz ayarlarını değiştirebildiği fiziksel saldırılar. 2) Şekil 1.3 ile gösterildiği gibi, cihazların sahip olduğu bağlantı protokollerinin fiziksel katman ve bağlantı katmanındaki güvenlik sorunları. 3) Ortadaki adam saldırısı, sahtekarlık, veri sızıntısı ve yetkisiz erişim dahil olmak üzere ağ katmanı saldırıları. 4) TCP ve UDP protokollerinin dahil olduğu taşıma (transport) ve uygulama katmanı saldırıları. Bu katmanda iyi bilinen saldırılar: kötü amaçlı yazılım saldırıları, DoS, DDoS, kimlik avı, SQL enjeksiyonu, siteler arası komut dosyası çalıştırma (Cross-site scripting - XSS) ve Botnet olarak sıralanabilir. 5) Trafik analizi, yan kanal, tekrar oynatma, ortadaki adam ve protokol saldırılarını içeren çok katmanlı saldırılar. 6) IoT'lerin temel parçalarından biri olan ve çoğu bulut mimarisinde yer alan geleneksel bilindik güvenlik sorunları. 7) Mevcut her bir IoT modülünün kendi güvenlik ve gizlilik ihlallerine sahip olabileceği IoT mimarilerinde güvenlik sorunları. Çalışma daha sonra, önceden tanımlanmış bazı tehditlerin üstesinden gelmek için özel olarak tasarlanmış mevcut makine öğrenimi uygulamalarını

açıklamıştır [53].

1.4 Saldırı Tespit Sistemleri

Saldırı Tespit Sistemleri (STS), temel olarak bir sistemdeki dahili ve harici kullanıcıların ve uygulamaların akışını analiz eden ve bu analizlere göre çeşitli koşullar sağlandığında alarmlar üretebilen bir güvenlik mekanizmasıdır. STS'ler, topladıkları bilgiler üzerinde çeşitli analizler yaparak yetkisiz faaliyetleri ve saldırı durumlarını tespit etmeyi amaçlar [54]. Güvenlik kapılarının temel işleme mekanizması göz önünde bulundurulduğunda ilk adım, kullanıcı adı ve şifre isteme gibi kimlik doğrulama mekanizmaları olarak düşünülebilir. Bu seviyenin ötesindeki kontrol mekanizması özellikle ağ trafiğinin geçtiği güvenlik duvarlarında yoğunlaşmıştır. Güvenlik duvarlarının kullanımı dışarıdan gelen bilinen türdeki saldırıların sayısını azaltsa da, izinsiz girişleri tespit etmek ve bir kaynağın gizliliğini, bütünlüğünü veya ulaşılabilirliğini tehlikeye atabilecek saldırılara karşı açık kapılar bırakmamak için STS(ler) kullanmak bir zorunluluktur gibidir [55].

STS, kullanılan ortamlara ve önleme mekanizmalarının türlerine göre farklı yaklaşımlar kullanılarak uygulanabilir [55, 56, 57]. İlk strateji, ana bilgisayar tabanlı (host-based) olarak adlandırılır ve bilgisayarda kurulu uygulamaların şüpheli hareketleri olup olmadığını tespit etmek için dahili bir izleme mekanizması sunar. İkinci strateji ise STS'nin LAN üzerinde yetkisiz gelen veya giden paket kalıpları olup olmadığını sürekli olarak kontrol ettiği bilgisayar ağına bakmaktır. Ağ tabanlı STS olarak adlandırılan bu tür sistemler, Hizmet Reddi (DoS) gibi bir takım saldırı türlerine karşı etkili olmayabilir. İmza tabanlı STS olarak adlandırılan başka bir STS türü ise, kurallar ve imzalar içeren bir veri tabanından elde ettiği verileri analiz mekanizmasına uygulayarak mükemmel bir şekilde veri tabanında yer alan saldırıları ve şüpheli davranışları tespit eder. Bu tür STS'ler önceden bilinen güvenlik açıklarını tespit edebilmektedir. Kuralların ve imzaların güncellik sorunu yaşadığı anda güvenlik eksikliği oluşması nedeniyle, anomali tabanlı bir STS, altta yatan sistemin oluşturduğu kullanım kalıplarını analiz ederek daha önce bilinmeyen saldırı türlerini bulmak için kullanılan bir başka STS türüdür. Bu STS türü,

önceden tanımlanmış herhangi bir kurala veya imzaya büyük ölçüde bağlı olmadan yeni tür saldırıları tespit etmede daha efektif çalışabilirler. STS'nin altta yatan sistemi, sistemin kendi kararlı veya kararsız davranışlarını analiz ederek öğrendiği göz önüne alındığında, bu STS'lerin sistemi öğrenmesi biraz zor olabileceği ve öğrenme aşamasında yüksek oranda yanlış pozitifler verebileceği değerlendirilmektedir.

Davetsiz misafirler bugün kapsamlı saldırılar geliştirmeye devam ettiğinden, önceden tanımlanmış ve esnek kurallar ve imza tabanlı yaklaşımlar kullanan geleneksel STS'ler yeni saldırıları tespit etmede yetersiz kalmaktadırlar [58]. Saldırganlar bilinen imzaları kullanmak yerine mevcut STS kurallarının üzerine sürekli olarak güncellenmiş veya yeni türde imzalar üretirler ve bu sayede tespit edilmeleri zorlaşır. Bu nedenle son yıllarda makine öğrenmesi ile güçlendirilen STS'ler, geçmiş verileri ile kullanıcı ve uygulama davranışlarını birbiriyle ilişkilendirebilmektedir. Böylece birleştirilmiş bu bilgilerden çıkarımlar yapabilmektedirler. Geleneksel makine öğrenmesi yaklaşımları özellikle neyin anormal davranış olup neyin olmadığının eğitim aşamasında kullanılmaktadır. Bu şekilde, STS'ler normal verilerin nasıl tanımlandığını [59] ayırt edebilirler. Görünmeyen imzaları yakalamak ve önceden etiketlenmiş normal verileri inceleyerek bunları "bilinmeyen saldırı" olarak tanımlamakla birlikte, makine öğrenimi ve Derin Öğrenme teknikleri, imza türlerini doğrudan sınıflandırmak için çok uygundur. İzler bilinen tiplerden farklı olsa bile yeterli veri sağlandığında modeller başarılı olmaktadır.

Bu çalışma kapsamında sadece anomali tespiti uygulamaktan ziyade, STS verilerinden saldırıların yakalanması ve sınıflandırılması üzerine bir yaklaşım amaçlanmaktadır.

1.5 Literatür Araştırması

Son yıllarda donanım temelli iyileştirmeler ve büyük veri teknolojileri ile birlikte Derin Öğrenme temelli yaklaşımların teknoloji ve yapay zeka anlamında birçok alanda çığır açıcı sonuçlar elde edebildiği görülmektedir. IoT sistemlerinin de doğası gereği büyük veri üretebiliyor olması, onu Derin Öğrenmenin popüler uygulama alanlarından biri haline getirmiştir. Bununla birlikte, çok farklı senaryolar olmakla beraber, IoT sistem-

lerinde toplanan verilerin eşit dağılımlı olmaması durumu da beraberinde zor problemleri getirmektedir. Bu, toplanan verilerin farklı kategorilerinin boyutunun birbiriyle eşit veya yaklaşık olmaması demektir. CICIDS2017 verisetinde olduğu gibi, birçok gerçek dünya sorunu dengesiz veriler içerir ve verilerin eşit olarak dağıtılması gereken birçok Derin Öğrenme konseptinin gereksinimleriyle çelişir. CICIDS2017 veriseti 15 saldırı sınıfından oluşan ve saldırı örneklemlerinin dengesiz bir şekilde dağıldığı bir veri kümesi içermektedir. Bu verisetini kullanan çalışmaların önemli bir kısmı da bu dengesiz dağılım nedeniyle örneklem sayısı çok az olan sınıfları yok sayıp sadece belli sınıflar üzerinde odaklanmışlardır. Bu literatür taramasında sadece CICIDS2017 veriseti ile çalışan son araştırmalar incelenmiştir. Literatürde STS'ler ile ilgili bazı çalışmalar bulunmaktadır ve bunlardan sadece birkaçı doğrudan IoT cihazlarına odaklanmıştır. Bu bölümde bu alandaki bazı önemli çalışmalara yer verilmesi amaçlanmıştır.

Literatürdeki çalışmaların bazıları yalnızca belirli bir saldırının tespitine odaklandığı tespit edilmiştir. Roopak ve arkadaşları, IoT ağlarındaki DDoS saldırılarını tespit etmek için bir Derin Öğrenme modeli önermiştir. Amaçları, IoT alanında bir farkındalık yaratmak ve siber güvenlik alanındaki açık araştırma zorluklarına değinmekti. Bu kapsamda DDoS saldırılarını tespit edebilmek için Çok Katmanlı Algılayıcı, Evrişimli Sinir Ağları (CNN), Uzun Kısa Süreli Bellek Birimleri (LSTM) ve hibrit bir CNN+LSTM modeli uyguladılar. Bulgular, hibrit modellerinin diğer modellerin yanı sıra Destek Vektör Makineleri, Naive Bayes (NB) ve Rassal Orman (Random Forest) gibi geleneksel makine öğrenme tekniklerinden daha iyi performans gösterdiğini belirtmiştir. Çalışmalar sonunda 97,15% doğruluk seviyesinde bir başarı elde etmişlerdir [60].

Alsyaibani ve arkadaşları [61], Çift yönlü Bidirectional LSTM alt yapısını kullanarak ve 24 farklı hiper parametre değerini test ederek CICIDS2017 veriseti üzerinde geniş kapsamlı bir çalışma gerçekleştirmiştir. Çift yönlü LSTM, standart LSTM yapısının girdilerin belirli bir zaman aralığındaki gelecek değerlerini işlemesine karşılık olarak, bir girdinin zaman ekseninde önceki değerlerini de hesaba katar ve başarımı artırmayı hedefler. Geliştirdikleri modelleri 1000 iterasyon olacak şekilde eğiten araştırmacılar, tüm sınıflar

dikkate alındığında 98,34% doğruluk değerine ve 98,38% F_1 puanına ulaşabilmişlerdir. Faker ve Dogdu, saldırıları sınıflandırmak için DNN ve topluluk yöntemlerini, Rassal Orman ve Gradient Boosting Tree'yi önermiştir. Ayrıca, K-Means ile özellik sıralaması uygulayarak ve homojenlik metriğini kullanarak verisetini on dört (saldırı sayısı) alt kümeye bölmüşler. Bu şekilde, tüm alt kümeler yalnızca ilgili saldırı ve normal etiketleri içerir ve her alt kümedeki tüm özellikler, en yüksekten en düşük homojenlik puanına göre sıralanır. Puanlama aşamasından sonra en düşük puanlı özellikler silinir. Sağladıkları en yüksek doğruluk, seçtikleri özellikler ve tüm veri kümesi için 99,56% olarak görülmüştür [62].

Hossain ve arkadaşları [63] Brute-Force saldırılarının (BFA), bir saldırganın sistemdeki ağa erişimi olduğundan itibaren artık kaçınılmaz olduğunu belirtmiştir. Çalışmada BFA'yı tespit etmek için çeşitli teknikler kullanan güncel literatür araştırılmış ve deney senaryosu olarak CICIDS2017'nin SSH ve FTP Brute-Force saldırıları araştırılmıştır. Senaryo sadece BFA olduğu için bu çalışmada verisetinin sadece Salı ve Perşembe günü verileri kullanılmıştır. Çalışmada LSTM ve RF, NB, K-Nearest Neighbor gibi diğer geleneksel makine öğrenme teknikleri kullanılmış ve sonuçlar, LSTM'nin Salı veriseti için %99,88 ve Perşembe veriseti için %99,86 doğrulukla diğer sınıflandırıcılardan daha iyi performans gösterdiğini ortaya çıkarmıştır. Çalışmanın sonuçlar bölümünde kesinlik ve duyarlılık değerleri paylaşılmıştır. Bu değerlere göre FTP-Patator saldırısı için F_1 puanı %0,99, SSH-Patator için %0,97, Web Attack Brute-Force için %0,19'dur. Web Attack SQL Injection için F_1 puanı %0 ve Web Attack XSS için %0,03'dür. Bu sonuçlar detaylı incelendiğinde, modelin Salı verileriyle iyi çalıştığı, ancak düşük duyarlılık nedeniyle Perşembe veri kümesiyle iyi bir performans sağlamadığı ve düşük bir F_1 puanına sahip olduğu görülmektedir.

Pelletier ve Abualkibash [64], çalışmalarında CICIDS2017'yi değerlendirdi ve dengesiz sınıfların veri kümesi üzerinde çalışmayı oldukça zorlaştırdığını belirtti. Yine de bir R-dili çatısı altında geliştirdikleri uygulama ile verisetini ön işlemlere tabi tuttular ve veriseti üzerindeki en az önemli olan öznitelikleri veriden çıkardılar. Ardından, model-

lerinin sağlamlığını kontrol etmek için verileri bir derin sinir ağı modeline ve bir rassal ormana uyguladılar. Sonuçlar, sinir ağının %96,53 ve RF ortalamasında %96,24 doğruluk elde ettiğini göstermiştir.

Zhang ve arkadaşları [65], ağa izinsiz giriş tespiti ile uğraşırken dengesiz veriseti sorununa değinmişlerdir ve dengesiz veri dağılımlarının modellerin başarımlarını olumsuz etkilediğini ve saldırı türleri için tespit hızını ve kalitesini sınırladığını belirtmişlerdir. Buna karşılık ilgili çalışmada Derin Öğrenme modellerinde kullanılmak üzere örneklem sayısını düzenlemek için sentetik veri artırma ve azaltma teknikleri önerilmiştir. Bu kapsamda çalışma dahilinde dengesiz veriseti senaryolarında kullanılmak üzere önce Sentetik azınlık örnekleme tekniği (Synthetic Minority Over-Sampling Technique - SMOTE) ile verinin artırılması, ardından Gauss Karışım Modeli (Gaussian Mixture Model - GMM) ile çoğalan örnekler arasından birbirlerine uzayda çok yakın olanların verisetinden çıkarılması sağlanmıştır. Bu sayede verinin daha dengeli ve aynı zamanda nitelikli bir dengeye ulaşmasının hedeflendiği değerlendirilmiştir. Çalışma, SGM-CNN olarak adlandırdıkları sekiz katmanlı CNN tabanlı bir Derin Öğrenme modeli kullanmıştır. Model, iki adet CNN+CNN+MaxPool ve iki adet tam bağlı katman içermektedir. Test veriseti olarak hem CICIDS2017 hem de UNSW-NB15 verisetleri kullanılmış ve başarı oranları birbirleriyle karşılaştırılmıştır. Ön işleme aşaması olarak, kategorik değişkenleri sayısal forma dönüştürmek için One-Hot Encoding isimli bir dönüştürücü kullanılmış ve ardından otomatik öznitelik azaltma ve seçim mekanizması olarak gürültü gidericili bir otomatik kodlayıcı (De-noising Auto Encoder) kullanılmıştır. Bu ön işlemlerden sonra, dengesiz eğitim seti üzerinde SGM adını verdikleri modeli işlemişler ve sonuçları paylaşmışlardır. Çalışmanın sonuçlarına bakıldığında, ağırlıklı ortalama F_1 puanının %99,86 gibi iyi bir sonuç olarak öne çıktığı görülmüştür.

Abdülhammed ve arkadaşları [66], CICIDS2017 veri kümesinde makine öğrenimi görevleri için boyut ve öznitelik küçültme yaklaşımları üzerinde çalışmışlar ve temel veri kümesinden daha ayırt edici ve temsili öznitelikler oluşturmak için hem otomatik kodlayıcı hem de Temel Bileşen Analizi (PCA) kullanmışlardır. Daha sonra elde edilen

düşük boyutlu öznitelikler üzerinde Rassal Orman, Bayesian Network, Doğrusal Ayırma Analizi (Linear Discriminant Analysis - LDA) ve Kuadratik Ayırma Analizi (Quadratic Discriminant Analysis - QDA) gibi çeşitli makine öğrenmesi görevlerini uygulamışlardır. Boyut indirgeme yaklaşımları, Rassal Orman sınıflandırması kullanıldığında %99,6 yüksek sınıflandırma doğruluğunu korurken, CICIDS2017 veri kümesinden en önemli 10 özniteliği ayırt edebilmiştir, ancak çalışmada belirtilen 10 öznitelikli çıktılar için herhangi bir F_1 puanı verilmediği gözlenmiştir. Bunun yerine, öznitelik sayısı 30'a düştüğünde modelin ağırlıklı ortalama F_1 puanının %99,7 olarak gözlemlendiği belirtilmiştir.

Zhang ve arkadaşları [67], dengesiz bir şekilde dağılmış saldırılar içeren verisetleri için daha kaliteli sonuçlar üretmeyi amaçlamışlardır ve buna ilişkin bir Paralel Çapraz Evrişimsel Sinir Ağı (PCCN) modeli önermişlerdir. Bu çalışmada, yazarlar yayınlanmış CSV veri kümesi üzerinden ham veri kümesi (*PCAP* dosyalarını ayrıştırarak) üzerinde çalışmayı seçmiştir. Bu nedenle eğitim ve test setleri için CICIDS2017 araştırmacılarının yayınladıkları açık CSV formatındaki verisetinden daha farklı sayıda eğitim ve test setini elde edip bunun üzerinde çalışma yapmışlardır. İlk modelde her biri dört evrişim katmanı içeren iki paralel CNN katmanını bazı noktalarda birleştirerek çıktılarını paylaştığı üç farklı model tasarlayıp bu modellere verileri aktarmışlardır. Öznitelikleri CNN katmanlarıyla çapraz ilişkilendirdikten sonra, elde edilen öznitelik haritası daha sonra genel bir evrişim katmanına ve ardından bir ortalama havuzlama (average pooling) katmanına gönderilmiştir. Buradaki çıktı ise tam bağlı bir katmana gönderilir ve yapay sinir ağı yapısı tamamlanır. İkinci model, paralel çıktıları birleştirme yerine bu çıktıların toplamını almayı denemiştir ve üçüncü model de sonucun yalnızca son katmanlar düzeyinde birleştirildiği paralel ancak asimetrik evrişimli katmanları kullanmıştır. Deneylerde önerilen PCCN modeli, CNN ve LSTM eşdeğerleri dahil olmak üzere diğer modellerden daha iyi performans göstermiştir. Bu çalışmada modellere iki ayrı veriseti gönderilmiştir. Birinci deneyde saldırı verilerinin içerik bilgileri (payload), ikinci deneyde ise TCP ve UDP başlık bilgileri üzerinden gidilmiştir ve bu iki deney sonucu birbirleri ile kıyaslanmıştır. Yalnızca TCP/UDP başlık bilgisi kullanıldığında %97,31 ağırlıklı orta-

lama F_1 puanı, yalnızca içerik verileri kullanıldığında %98,65 F_1 puanı elde edilmiştir ve hem içerik hem de başlık verileri birlikte kullanıldığında F_1 puanı performansı %99,68 seviyesine kadar çıkmıştır.

Elmasry ve arkadaşları, hem modelleri önceden eğitmek hem de temel modellerin hiper parametrelerini ayarlamak için Parçacık Sürü Optimizasyonu'nu (Particle Swarm Optimization - PSO) kullanarak Derin Öğrenme tekniklerini geliştirmek için temel olarak evrimsel bir algoritma yaklaşımına dayanan iki çalışma üzerine yoğunlaşmışlardır. Yazarlar, 1) Derin Sinir Ağları, 2) Uzun Kısa Süreli Bellek Birimleri, 3) Kapı Özyinelemeli Geçitler (Gated Recurrent Unit - GRU) ve 4) Derin İnanç Ağları olmak üzere dört Derin Öğrenme modelini farklı çalışmalarda incelemişlerdir. Yapılan çalışmalar, PSO kullanarak tespit edilmeye çalışılan saldırıların tespit oranının arttığını ve yanlış alarm oranının belirli bir oranda azaldığını göstermektedir. Çalışma kapsamında geliştirilen modeller ayrı ayrı KDD CUP 99, NSL-KDD, CIDDS ve CICIDS2017 verisetlerinde denenmiştir. Bu yazıda bahsedilen diğer bazı çalışmaların aksine, Elmasry ve arkadaşları eğitim ve test setlerini oluşturmak için CICIDS2017'nin PCAP dosyalarını kullanmışlardır. Bu şekilde, ilk makaleleri [68] için her sınıftan 18750 örnek ve ikinci makaleleri [69] için her sınıftan 2142 örnek seçmişlerdir. BU sayede veriseti tam anlamıyla dengeli bir veriseti haline getirilmiştir. Bu kurulumlarla birlikte ilk makalede yazarlar çok sınıflı sınıflandırıcı %97,37 F_1 puana ulaşmışlardır. İkinci makalede ise PSO ve derin inanç ağlarını kullanan çok sınıflı sınıflandırıcı modeli %95,81 F_1 puanına ve ikili sınıflandırıcı modeli %99,95 F_1 puanına ulaşabilmiştir.

Literatürde saldırı tespiti ile ilgili konuları çeşitli açılardan araştıran başka birçok örnek çalışma vardır. Xavier ve arkadaşları verisetlerinin çok boyutluluklarını azaltarak en sık kullanılan veri kümelerinin baskın özelliklerini gruplandırmayı başarmışlardır. Çalışma kapsamında ISCX 2012 [70], CTU-13 [71], MACCDC [72] ve UGR'16 [73] verisetlerini incelenip üç özellik grubu önerilmiştir: 1) Temel bağlantı özellikleri, 2) içerik özellikleri ve 3) trafik istatistiksel özellikleri. Bu gruplar, farklı saldırı türleri için çeşitli ağırlıkları temsil etmektedir. Çalışma, bu gruplara ileri beslemeli çok katmanlı ve tekrar-

layan sinir ağlarını uygulamış ve sonuçları, çalışmanın yapıldığı tarihteki son literatürle karşılaştırmıştır. Öznitelikler belirli bir miktara indirgenmiş olsa da incelenen yaklaşımlar yine de gözlemlenen literatür [74] arasında en yüksek doğruluğa sahip çıkmıştır.

Saldırı tespit verisetlerinin farklı saldırı türlerini nasıl kapsadığını ve bu saldırılar arasındaki ilişkiyi daha iyi anlamak için Zong ve arkadaşları [75] çok faydalı bir çalışma yapmışlardır. Çalışma, makine öğrenimi modellerinin yanlış alarmlarına ve bunların sonuçlarına siber güvenlik perspektifinden işaret etmektedir. Çalışma daha sonra makine öğrenimi modellerinin verileri nasıl gördüğünü ve buna göre hareket ettiğini daha iyi anlamak için saldırı türlerini üç boyutlu uzayda görselleştirmeye odaklanmıştır. Saldırı türlerini görselleştirmek için yazarlar, NSL-KDD [76] ve UNSW-NB15 [77] veri kümeleri üzerinde çalışmayı seçip, veriler üzerinde bir ön işleme uygulamışlardır. Yazarlar, saldırıları görselleştirmek amacıyla özniteliklerin boyutunu üçe indirmek için PCA dönüşümünü kullanmışlardır. Çalışmada, yazarlar saldırı sınıflarını, verisetleri üzerinde üç boyutlu olarak farklı açılardan da gösterip bunlarla ilgili çeşitli değerlendirmelerde bulunmuşlardır. Görselleştirme aracındaki bazı saldırı türlerinin birbirlerine çok yakın örneklemlerden oluşan sıkı bir grup oluşturduğu (DoS gibi) gözlemlenirken, bazı saldırı türleri yalnızca belirli bir birbirlerine sadece belli açılardan ve perspektiften bakıldığında iyi görünür konumda durmuşlardır. Bu görselleştirme tekniği ile ilgili olarak, yazarlar daha sonra bir dizi makine öğrenimi modeli uygulamışlar ve karar sınırlarını aynı görsele koymuşlardır. Bu çalışma, makine öğrenimi modellerinin özellikle saldırı tespit verilerinde kullanıldıklarında nasıl bir gözlem uzayına hakim olduklarını gözlemlemek açısından yararlı bir iç görü sağladığı için özellikle incelenmesi gereken çalışmalardan biri olarak ön plana çıkmıştır.

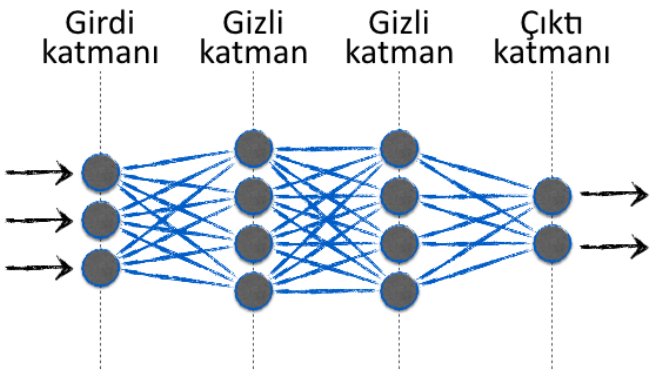
1.6 Derin Öğrenme

Derin öğrenme konsepti temel olarak çok katmanlı yapay sinir ağları modeli üzerine kurulmuştur. En küçük birim olarak nöronlar veya algılayıcılar (perceptron) görülürken, yapay sinir ağı modeli bu nöronların kendi aralarındaki bilgi alışverişi ile öğrenmeyi

gerçekleştirmektedir. Şekil 1.4 ile iki gizli katmanı olan basit bir yapay sinir ağı gösterilmiştir. Aradaki katmanların "gizli" olarak anılmasının nedeni, bu katmandaki nöronlara giden değerlerin işlenen veriden ziyade, modelin kendisi tarafından verilmesidir. Model, ilgili veriyi temsil edebilecek en uygun parçaları ve bu parçaların ilişkilerinden doğan diğer parçaları gizli katmanda belirler ve bu ilişkilere göre çıktı katmanında bir sonuca varılır.

Girdi katmanı haricinde yer alan nöronlar, içlerinde bir hesaplama ünitesi bulundurmaktadırlar. Her bir nöron, bir önceki aşamadan aldığı değerleri belli bir kurala göre, örneğin lojistik sigmoid fonksiyonuna göre işler ve kendi çıktısını bağlı bulunduğu nöronlara girdi olarak gönderir. Bu durumda aslında her bir ünite, önceki nöronlardan sayısal girdiler alan ve bunları bir matematiksel formül ile işleyip yine sayısal çıktılar üreten bir fonksiyondur denilebilir. İlk etapta rastgele olarak belli bir aralıkta belirlenen ağırlık değerleri ile bir nöronun her bir çıktısı diğer nöronlara farklı değerlerde gider. Buna ek olarak her bir nöron için yine rastgele bir "bias" değeri vardır ve bu değer ile de çeşitlilik korunur. Buradaki öğrenme işleminde temel esas; her bi nöronun, ilgili girdi için yeni bir temsil ünitesi olmasıdır. Buna ek olarak, her bir nörondan çıkan değer, ondan sonraki nöronları da etkileyeceği için ağın derinliğinin artması daha fazla nöronun birbirini etkileyeceği anlamına gelir. Bu da ilerleyen katmanlarda artımsal olarak daha karmaşık fonksiyonların işlenmesi ve dolayısıyla daha karmaşık yapıların tanınabilmesi demektir.

Derin öğrenme kavramının son yıllarda araştırmalarda popülerliğinin artmasındaki etken



Şekil 1.4: İki gizli katmanlı basit bir yapay sinir ağı modeli.

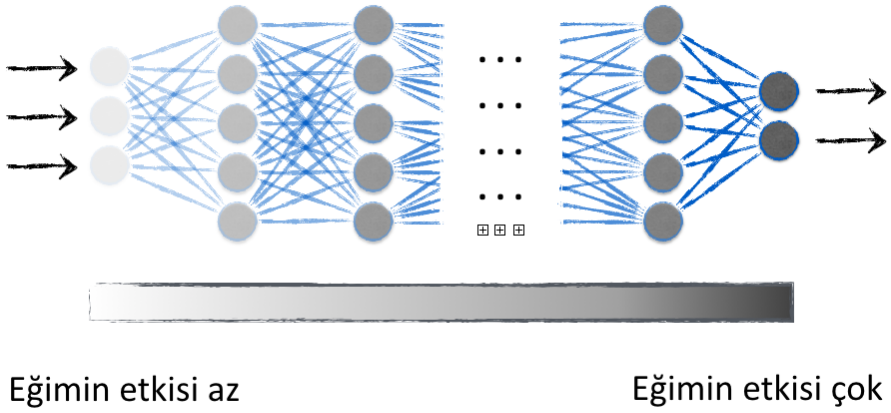
yalnızca sistemin başarılı sonuçlar vermesi değil, aynı zamanda bu sistemlerin altında çalışan donanımsal mimarilerin de ilgili algoritmaların paralel ortamda gerçeklenip kısa sürelerde sonuç alınmasına olanak tanımasıdır.

Artımsal olarak hiyerarşik bir yapıda ve kendi içinde temsili öğrenme yöntemini kullanarak karmaşık problemlerin çözümünde kullanılan Derin Öğrenme yöntemi ile aktif olarak bilgisayarlı görü, ses ve konuşma analizi, doğal dil işleme ve veri işleme uygulamaları üzerinde araştırmalar yapılmaktadır. Çeşitli dillerde yazılmış uygulama çatıları (Google TensorFlow, Caffe, Theano, PyTorch, Keras, PyLearn2 ve diğerleri) ve paralel mimariler ile (Ndivia grafik kartları) hem akademik hem de sektörel uygulamaları hızla artan Derin Öğrenme yöntemi, günümüzde en çok üstünde çalışılan ve gelecek vaadeden makine öğrenmesi yöntemlerinden biri olarak değerlendirilmektedir.

1.6.1 Eğimin Kaybolması

Yapay sinir ağlarında, eğitim için sürekli olarak bir maliyet hesabı yapılır ve üretilmesi beklenen çıktı ile gerçek çıktının karşılaştırılması ile bulunan farka istinaden ağdaki ağırlıklarda ve bias değerlerinde ufak düzenlemeler yapılır. Bu işlemin yeteri kadar tekrarlanması ile normal şartlar altında öğrenme ve eğitim işlemi gerçekleşmiş olur. Burada öğrenme işleminden kasıt, verilen girdinin ağ tarafından üretilen çıktısı ile, çıkması beklenen sonucun aynı veya olabildiğince birbirine yakın olmasıdır. Bu işlem geri yayılım (backpropagation) algoritması vasıtasıyla yapılmaktadır. Yapay sinir ağlarını eğitmek için kullanılan bu yöntem olan backpropagation ile "vanishing gradient" (eğimin kaybolması) problemi ortaya çıkmaktadır. Bu durumda eğitim çok uzun vakit alabilmektedir, hatta öğrenme işlemi bir safhadan sonra durmaktadır.

Öğrenme işlemi, Yapay sinir ağının bir girdi için ürettiği çıktının, aslında olması gerekenden ne kadar farklı bir sonuç ürettiği konsepti üzerinden başlamaktadır. İlgili çıktının olması gereken değere yakınlaşabilmesi için uygulanan backpropagation öğrenme yöntemi ile çıktı üreten nöronlara girdi sağlayan önceki tüm nöronların belli bir oranda değerlerini güncellemeleri gerekmektedir. Bu aşamada ortaya çıkan gradient (eğim) terimi ise, çıktı



Şekil 1.5: Katman sırasına göre eğitim değerinin (gradient) etkisi.

nöronlarının değerlerini belirlenen seviyeye getirebilmeleri için bir önceki katmanda yer alan hangi nöronun kendi çıktısını ne kadar güçlü bir şekilde güncellemesi gerektiğini belirtmektedir. Bir nöron için gradient değeri büyük ise o nöronun ağırlıklarını ve bias değerini o ölçüde daha fazla değiştirmesi gerekecektir. Bu da ilgili nöronun değişime ne kadar eğimli olması gerektiğini göstermektedir. Ağdaki nöronlar, gradient değeri büyük olduğunda hızlı, küçük olduğunda ise yavaş eğitilecektir.

Şekil 1.5 ile görüldüğü üzere eğim değerleri son katmanlarda yüksek iken ilk katmanlara doğru düşmektedir. Bu da problemin yapısıyla ilgili en temel ve küçük yapıları içinde barındırması gereken ilk katmanların daha geç öğrenmesi demektir. Eğer bu katmanlar geç öğrenirse, sonraki katmanlar da bu katmanların birleşiminden meydana geleceğinden ortaya bozuk bir yapı çıkabilmektedir. Örneğin eğer problem yüz tanıma ise, bu durumda sistem düzgün yüz hatlarından ziyade eğri ve bozuk bir yüz modelini kendi içinde standartlaştıracaktır.

1.6.2 Kısıtlı Boltzmann Makinesi

Kısıtlı Boltzmann Makinesi (Restricted Boltzmann Machine - RBM), veri üzerindeki örüntülerin (pattern) ilgili girdilerin yeniden inşa edilmesiyle otomatik olarak bulunabildiği, Geoffrey Hinton tarafından geliştirilmiş bir yapay sinir ağı modelidir [78].

RBM, iki katmanlı sığ ve tam bağlı bir yapay sinir ağıdır. İlk katmanı görülebilir (visible), ikinci katmanı ise gizli (hidden) katman olarak anılmaktadır. Bu yapının ismindeki

kısıtlı (restricted) terimi, görülebilir katmandaki hiçbir nöronun birbiriyle bir bağlantısı olmamasından gelmektedir. RBM, temelde iki yollu bir çeviricinin (2-way translator) matematiksel formülüne karşılık gelmektedir. RBM yapısının ileri beslemesinde (forward pass) ilk katmana gelen girdiler ikinci katmanda sayısal bir karşılık bulacak şekilde kodlanmış olurlar. Bu bir şifreleme işlemi gibidir. Geri beslemede ise (backward pass) bu sayısal değerlerin ilk etapta gönderilen girdileri tekrar üretecek şekilde sonuçlar çıkarması sağlanmaktadır. Bu da deşifre etme işlemi gibidir. İyi eğitilmiş bir RBM, yüksek başarımlarda şifreleme ve deşifreleme işlemi yapabilmektedir.

RBM yapısının önemli özelliklerinden biri, bu yapıların verinin etiketlenmesine ihtiyaç duymamalarıdır. Bu sayede fotoğraf, video, ses ve sensör gibi etiketli olmayan gerçek dünya verileri üzerinde RBM ile önemli çıkarımlar yapılabilmektedir. Bu verilerden herhangi bir manüel etiketleme yöntemine gidilmeden RBM vasıtasıyla girdilerin özellikleri çıkarılabilmekte ilgili girdiler tekrar üretilebilmektedir. Burada RBM yapısındaki önemli husus, bu yapının girdi olarak aktarılan verilerin hangi özellikler vasıtasıyla tekrar üretileceği, dolayısıyla hangi özelliklerinin önemli olduğu konusunda otomatik olarak karar verebilmesidir. Bu durumda RBM, aslında bir özellik çıkarıcı (feature extractor) bir yapı olarak ön plana çıkmaktadır. RBM yapısı, aynı zamanda otomatik kodlayıcı olarak da bilinmektedir.

RBM yapısının eğimin kaybolması problemine nasıl bir çözüm getirdiği, Derin İnanç Ağları konseptinde daha iyi irdelenmektedir.

1.6.3 Derin İnanç Ağları

Geoffrey Hinton tarafından geri yayılma algoritmasına alternatif olarak geliştirilen Derin İnanç Ağları (Deep Belief Networks - DBN), RBM'lerin akıllı bir öğrenme algoritması ile birleşiminden meydana gelmektedir. DBN, ağ bağlantıları ve yapısı itibariyle çok katmanlı bir yapay sinir ağı ile benzerlik göstermektedir. Aralarındaki fark ise öğrenme mekanizmasındadır. DBN, öğrenme mekanizması ile kendisiyle benzer derinlikte olan yapay sinir ağlarına göre yüksek başarımlı sonuçlar üretebilmektedir [79].

DBN, temelde birden fazla RBM'in bir yığın (stack) yapısında arka arkaya dizilmesinden oluşmaktadır ve bir RBM'in gizli katmanı, bir diğerine girdi sağlayan görülebilir katman olarak görev almaktadır.

DBN yapısında öğrenme temelde şu şekilde sağlanmaktadır:

- Öncelikle ilk RBM, ağa sağlanan girdiyi öğrenmek ve olabildiğince başarımlı bir şekilde tekrar üretebilmek için eğitilir.
- İlk RBM eğitimi tamamlandığında, artık sıradaki RBM, ilk RBM'in gizli katmanının değerlerini girdi parametresi olarak görür ve bu aşamada görülebilir katman gibi çalışarak bu ikinci RBM'in eğitilmesi sağlanır. Bu süreç, ağdaki tüm RBM'ler aynı şekilde eğitilene kadar devam eder.

Buradaki önemli husus, bir DBN'in, girdinin tamamını öğrenecek şekilde eğitilmesidir. Karşı bir örnek olarak, Evrişimsel Yapay Sinir Ağlarının her bir katmanda girdi hakkında basit örüntüleri ve özellikleri öğrendiği ve sonraki katmanların da bu özelliklerin kombinasyonlarından oluşan daha karmaşık yapıları tanıdığı bilinmektedir. DBN içerisinde ise ağın tamamı girdinin tamamını bir bütün olarak tanımaktadır. Buna misal olarak bir fotoğraf makinesinin çekilecek görüntüye yavaşça odaklanması verilebilir.

DBN başarımının yüksek olmasının arkasındaki basit mantık, çok katmanlı bir yapay sinir ağının tek katmanlı bir yapay sinir ağına göre daha karmaşık yapıları tanımasındaki gibi, DBN'i oluşturan RBM'lerin bütününün tek bir RBM ünitesinden daha fazla ve karmaşık özellikleri öğrenebilmesidir.

DBN eğitimi tamamlandıktan sonra artık yapı içerisindeki ağırlıklar vasıtasıyla veriseti hakkında elimizde bir model oluşmaktadır. Eğitimin tamamlanması için oluşturulan bu modelin çıktılarının gözetimli öğrenme yöntemi ile iyileştirilmesi gerekmektedir. Bunun için bir miktar etiketli veri kullanılıp DBN'in çıktılarının değerlendirilmesi sağlanmaktadır. DBN, yapısı itibariyle veriden kaliteli özelliklerin çıkarılması, az miktarda etiketli veriye ihtiyaç duyması ve öğrenme süresinin de uygun paralel donanım mimarilerinde kısa sürmesi nedeniyle kolay uygulanabilir bir yapı olarak öne çıkmaktadır.

Bununla birlikte benzer derinlikte birçok katmanlı yapay sinir ağına göre daha kaliteli sonuçlar üretebilmektedir. Bu çıktılar ile DBN yapısı, geri yayılma ile ortaya çıkan eğimin kaybolması problemini de ortadan kaldırmaktadır.

1.6.4 Evrişimsel Yapay Sinir Ağları

Evrişimsel Yapay Sinir Ağları (Convolutional Neural Networks - CNN), Derin Öğrenme konseptinin günümüzde en aktif çalışılan alanlardan biri olmasını sağlayan yapay sinir ağı modellerinden birisidir. CNN modeli, ilk olarak New York Üniversitesinde çalışan Yann Lecun tarafından geliştirilmiştir [80]. CNN'ler son birkaç yıldır bilgisayarlı görü alanındaki uygulamalarda en etkin kullanılan modellerden birisidir. İlk olarak 2012 yılında ImageNet yarışmasında Geoffrey Hinton ve öğrencilerinin açık ara birincilik elde etmesiyle başlayan CNN'lerin popülaritesi bu tarihten itibaren hızla artmıştır. Google, Microsoft Baidu ve Chooch AI gibi büyük teknoloji firmalarının ve açık kaynak komünitesinin çalışmalarıyla ortaya çıkan çeşitli modellerde nesne tanımasında CNN ile üretilmiş algoritmaların bir insan performansına epey yaklaştığı görülmüştür. Bunun ardından birçok teknoloji firmasında ve start-up firmalarında çeşitli alanlarda CNN kullanılmaya başlanmıştır. Bunda destekleyici unsur olarak, modellerin yazılmasını kolaylaştıran açık kaynak kodlu birçok kütüphanenin geliştirilmesi gösterilebilir.

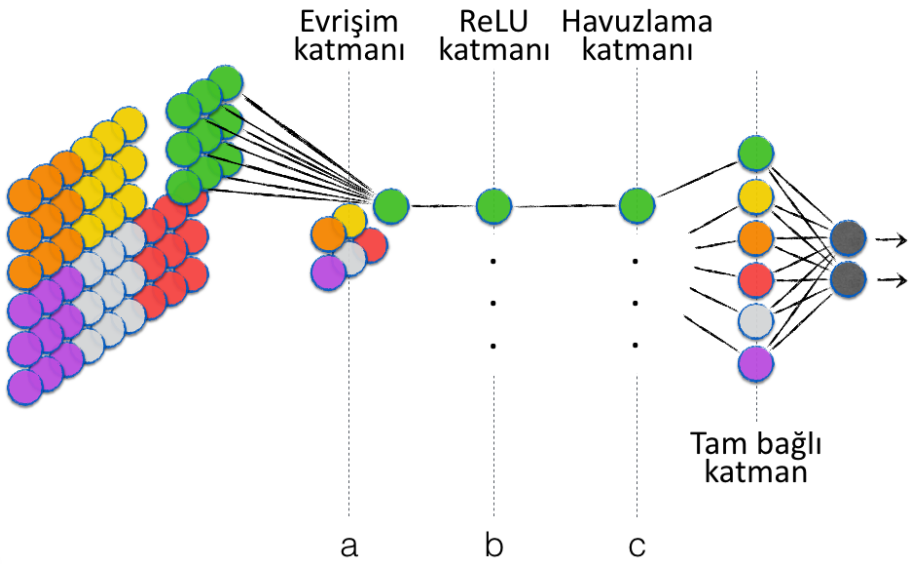
CNN, birkaç adet temel ve tekrarlayan katmandan oluşmaktadır. Bunlardan ilki evrişim katmanıdır. Bu katman içinde girdi olarak verilen resimler üzerinde piksel grupları (örneğin 3x3 piksellik bir kare alanı) sırayla belli filtreler ile dolaşılır. Bu sayede bu gruplar içinde aynı özelliklere sahip olanlar filtrenin uygulanmasından sonra belirlenmiş olur. Filtre sonunda ilgili girdinin belli özelliklerini ön plana çıkarmış bir yapı ortaya çıkmaktadır. Benzer işlem farklı filtreler ile sürdürülerek aynı girdi için birçok farklı belirleyici özellik çıkarılmış olur. Bir evrişim işlemini, resim üzerinde dolaşan NxN'lik küçük dürbünler olarak nitelemek yerinde olacaktır. Farklı her bir filtre birbirinden bağımsız olarak çalışıp yeni bir çıktı üreteceğinden, her bir filtrenin paralel olarak çalışması da mümkündür. Bu da özellik çıkarımının hızını artıran bir unsurdur. Şekil 1.6(a) üzerinde

örnek bir 9x9 piksel boyutlarında resim ve 3x3'lük grupların bağlı bulunduğu bir nöron görülmektedir. Ayırt edilmesi açısından renklendirilen kısımlar bir resmin ardışık 3x3'lük piksel gruplarıdır. Her bir grup evrişim katmanındaki ilgili nörona ulaşmaktadır. Evrişim katmanındaki her bir nöron aynı filtrenin, resmin bağlı bulundukları bölgelerine uygulanan bir kopyası olarak düşünülmektedir. Bu durumda bu katmandaki her nöronun aynı ağırlık ve bias değerlerine sahip olması gerekmektedir. Filtre tüm resme uygulandığında, bu katmandaki her bir nöron ilgili resim içindeki kendilerine bağlı bölgelerde aynı belirgin özellikleri belirlemiş olacaklardır.

Evrişim katmanıyla birlikte filtrelenen resim, Rectified Linear Unit (ReLU) ve havuzlama katmanlarından geçirilerek işlemler devam eder. Bu katmanlar evrişim katmanının girdi için bulduğu özellikler üzerinde işlemler yapmaktadırlar. Şekil 1.6(b) ile gösterilen ReLu katmanı, aktivasyon nöronlarının bulunduğu katmandır. CNN modeli geri yayılma ile eğitildiğinden, eğimin kaybolması problemi burada da ortaya çıkabilmektedir. ReLU ise öğrenme esnasında eğim değerini neredeyse sabit tuttuğundan, bu aktivasyon fonksiyonu ile önem derecesi yüksek olan ilk katmanların bozulmasının önüne geçmektedir ve CNN'in daha kaliteli eğitilmesine olanak vermektedir. Şekil 1.6(c) ile gösterilen havuzlama katmanı ise veride boyut indirgemek için kullanılmaktadır. Havuzlama katmanı ile CNN'in yalnızca en belirgin ve girdi hakkında nitelikli özet bilgileri verebilecek özellikleri bulduğu garanti edilmektedir. Bu sayede havuzlama katmanının kullanılmamasına kıyasla öğrenme işleminde hem bellek hem de işlemci gücünden tasarruf edilebilmektedir [81].

CNN'deki katmanlar, art arda sıralanıp birbirlerinin çıktılarını da işleyebilmektedirler. Bu sayede daha karmaşık yapıları da öğrenebilmektedirler; ancak bu katmanların girdi hakkında buldukları özellikler ve örüntülerin anlamı, bu katmanların ardına eklenmiş tam bağlı yapay sinir ağları ile ortaya çıkmaktadır. Bu sayede verilen girdiler sınıflandırılabilmektedir.

CNN'ler gözetimli bir yapıda olduklarından ve derin katmanlar içerdiklerinden eğitim için büyük miktarda etiketli veriye ihtiyaç duymaktadırlar. Bu da gerçek dünya problem-



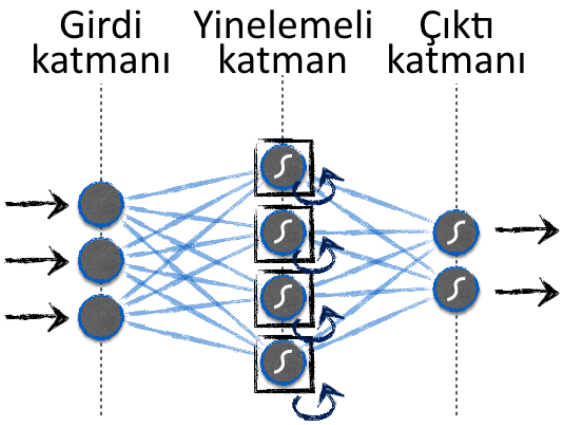
Şekil 1.6: Bir Evrişimsel Yapay Sinir Ağının genel görünümü.

lerinin çözümü için yeterli miktarda veri toplama gereksinimi oluşturmaktadır.

1.6.5 Geri Beslemeli Yapay Sinir Ağları

Geri Beslemeli Yapay Sinir Ağları (Recurrent Neural Networks - RNN), zamanla davranışı değişen veriler için, içinde basit bir geri bildirim yapısı bulunan bir tahmin mekanizması gibi çalışmaktadır. Sepp Hochreiter [82], Jurgen Schmidhuber vd. [83] ve Geoffrey Hinton'un [84] çalışmalarıyla günümüzde popülerliği artan ve kendilerine yer edinen RNN'ler; konuşma tanıma, örüntü tanıma, gerçek zamanlı dil çevirisi, borsa tahmini ve hatta sürücüsüz araçlar gibi çeşitli uygulama alanlarında akan verileri anlamlandırma işlemlerinde kullanılabilmektedirler.

RNN yapısı, katmanların çıktılarını yalnızca bir sonraki katmana ilettiği ileri beslemeli yapay sinir ağlarına kıyasla (örneğin Evrişimsel Yapay Sinir Ağları) bir farklılık içermektedir. Bu ağ yapısında, bir katmanın çıktısı hem bir sonraki katmana girdi olarak gönderilmekte hem de aynı katmana geri bildirimde bulunmaktadır. Şekil 1.7 üzerinde basit bir RNN'in yapısı tasvir edilmiştir. Ayrıca RNN'ler değişken boyutlu bir veri dizisini girdi olarak alıp, yine değişken boyutlu bir veri dizisini çıktı olarak üretebilmektedirler. Bu sayede ileri beslemeli yapay sinir ağlarının sabit çıktılı sınıflandırma veya tahmin etme gibi özelliklerine kıyasla RNN'ler ile daha farklı alanlarda uygulamalar



Şekil 1.7: Bir RNN'in temel görünümü.

geliştirilebilmektedir. N-e-N yapısında girdi ve çıktı düzeni mümkündür. Örneğin tekil girdi ve çoğul çıktıya örnek olarak resim alt yazısı üretme gösterilebilir. Bu durumda girdi olarak bir resim ve çıktı olarak metin dizisi üretilebilir (örneğin verilen bir manzara resminin içeriğindeki nesneleri isimleri ve konumlarıyla tespit edebilir). Çoğul girdi ve tekil çıktıya örnek olarak doküman sınıflandırma gösterilebilir (örneğin girdi olarak verilen haber metinlerinin ortak kategorisini tespit etmek). Çoğul girdi ve çoğul çıktıya örnek olarak ise video görüntülerinin sınıflandırılması verilebilir (örneğin saniyede N tane görüntü içeren bir videonun akışına göre o video içeriğinde ne anlatmak istediğinin belirlenmesi). RNN'lere eklenecek bir zaman etiketiyle de bu yapıların istatistiksel veri tahminleri yapması sağlanabilir. Buna örnek olarak borsa verilerinin geçmiş değerlere, sayısal indikatörlere ve ilgili hisselerle alakalı çıkan haberlerin olumlu veya olumsuz olmasına göre bir sonraki aşamada hangi yöne gideceğinin tahmin edilmesi verilebilir.

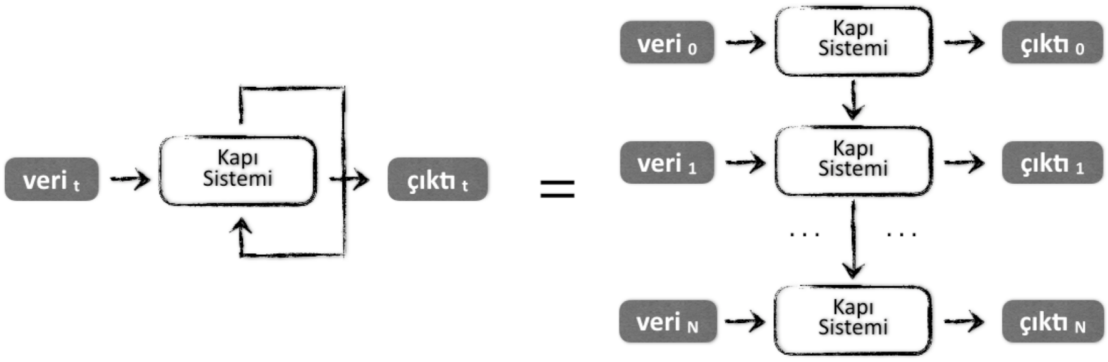
RNN, temelde tek bir katmandan oluşan bir ağı temsil etmektedir. RNN'lerin yanyana dizilip birbirlerini beslemesiyle daha karmaşık yapıların tanınması ve daha kaliteli çıktıların üretilmesi sağlanabilmektedir. RNN'ler, geri yayılım yöntemi ile eğitilmektedirler. RNN'lerdeki her bir zaman biriminin bir katmana karşılık gelmesinden ötürü, çok sayıda zaman birimine karşılık gelecek bir RNN'in eğitimi de dolayısıyla sanki aynı oranda artan bir ileri beslemeli yapay sinir ağını eğitmek kadar vakit alacaktır. Bu da derin yapının ilk katmanlarının eğimlerinin çok küçük oranlarda değişmesi ve ağın öğrenme işleminin çok yavaş olmasına neden olmaktadır. Burada ortaya çıkan

bir diğer sorun ise verilerin zamana bağlı gelişine karşılık ağın yavaş öğrenmesi nedeniyle bu verilerin önemini yitirmesi durumudur. Bu olay temelde ağın verilerdeki kısa süreli değişimleri öğrenip uzun süreli daha genel değişimleri görememesine neden olmaktadır. Bu problemlere çözüm olarak kapı sistemleri (gating systems) ortaya çıkmıştır. Uzun Kısa Süreli Bellek Birimleri (Long Short-Term Memory - LSTM) [82, 85] ve Geçitli Tekrarlayan Birimler (Gated Recurrent Unit - GRU) [86] bu çözümlerden en çok kullanılanlar arasındadır. Bu sistemler basit manasıyla ilgili verinin gerektiğinde unutulması, gerektiğinde ise ilerleyen zaman birimlerinde hatırlanması üzerine kuruludur. RNN'ler yapı itibariyle derin ağları temsil ettiğinden ve eğitim işlemi uzun sürdüğünden, bu ağların eğitimi için GPU'ların tercih edilmesi bu süreci epey hızlandıracak bir katkı sağlamaktadır. Yapılan deneysel çalışmalarda, GPU ve CPU ile eğitilen aynı ağların eğitim sürelerinde muazzam bir zaman farkının olabileceği tabir edilmektedir. Kısacası, uygun şartlar sağlandığında bir ağın eğitimi belli kapasitede bir CPU ile aylarca sürecekken, aynı ağın GPU ile bir günde veya daha kısa sürede eğitiminin tamamlanabilmesi mümkün görülebilir.

1.6.6 Uzun Kısa Süreli Bellek Birimleri

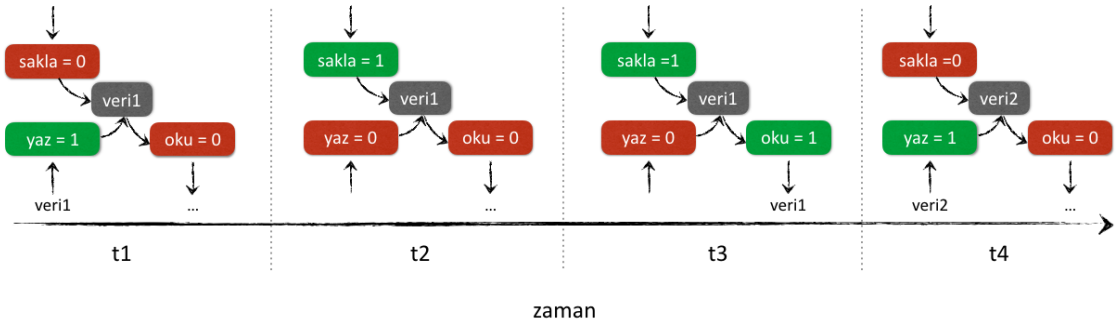
Uzun Kısa Süreli Bellek Birimleri (Long Short-Term Memory - LSTM) yapısı, lojistik ve lineer ünitelerin birbirleri ile etkileşime geçmesiyle oluşan bir hafıza hücresi olarak RNN'lerin uzun zaman etiketlerini hatırlamasını sağlayabilmektedir [82]. Bu yapı, her girdide eski girdinin üstüne yazan klasik yineleme birimlerine kıyasla, içerisinde geliştirilen kapı sistemiyle mevcut hafıza alanını gerektiğinde koruyabilmektedir. Eğer bir LSTM birimi önemli bir özniteliği tespit ederse, bu durumda ilgili özniteliği uzun bir süre saklayıp muhtemel uzun süreli örüntüleri tespit edebilmektedir [86]. Bir LSTM hücresinin zaman içerisindeki akışı özet olarak Şekil 1.8 ile gösterilmiştir.

Bir LSTM içinde temelde üç adet hafıza kapısı (gate) bulunmaktadır. Bunlar sırasıyla yaz (write), sakla (keep) ve oku (read) kapılarıdır. Yaz kapısı ile ilgili hafıza hücresine önceki hücrelerden veri yazılabilmektedir. Sakla kapısı ile hücredeki mevcut veri koru-



Şekil 1.8: LSTM hücresinin temel çalışma prensibi.

nabilmekte veya silinebilmektedir. Oku kapısı ile de hücre içindeki veri okunup sonraki hücrelere aktarılabilmektedir. Şekil 1.9 ile bir LSTM hücresinin zaman içerisinde işleyişi gösterilmiştir. t1 zamanında sakla kapısı 0, yaz kapısı 1 iken yeni bir veri yazılmaktadır. t2 zamanında sakla kapısı 1, yaz kapısı 0 olduğunda yeni bir veri yazılmamakta, hücre içindeki mevcut veri korunmaktadır. Herhangi bir okuma yazma işlemi yapılmadığı için bu anda ilgili hücrenin ağın geri kalanından izole edildiği görülmektedir. t3 anında içerdeki veri korunup oku kapısı 1 olduğu için ilgili veri çıktı olarak döndürülmektedir ve t4 zamanında ise sakla kapısı 0 olduğu için içerdeki veri silinmektedir. Yaz kapısı da 1 olduğundan dolayı hücreye yeni veri yazılmaktadır.



Şekil 1.9: LSTM hafıza hücresinin zaman içerisinde işleyişi.

1.7 CUDA Mimarisi

Bu çalışma kapsamında model geliştirme, eğitme ve test süreçlerinde faydalanılan Derin Öğrenme kütüphaneleri ve literatürde kullanılan diğer birçok makine öğrenmesi

kütüphanesi, grafik kartları üzerinde hesaplama süreçlerini eniyilemişlerdir. Bu sayede grafik kartlarının paralel işlem gücünden faydalanarak geleneksel merkezi işlem birimlerine kıyasla onlarca kat hız kazanımı elde edebilmişlerdir. Bu kütüphaneler, grafik kartlarındaki eniyileme işlemleri için CUDA mimarisini baz almaktadırlar. Bu nedenle bu çalışma kapsamında bu mimarinin işleyişinin açıklanmasının faydalı olduğu değerlendirilmiştir.

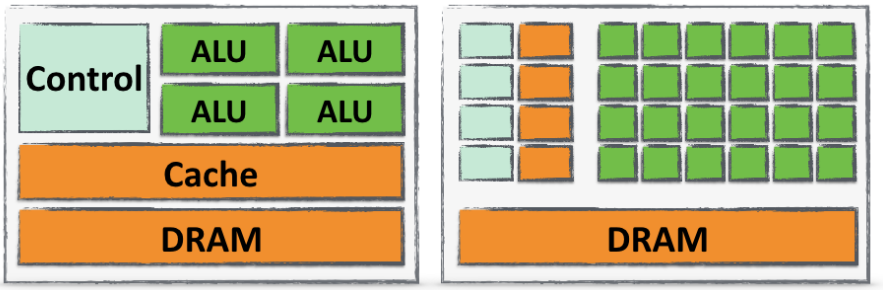
CUDA, NVIDIA firmasının 2006 yılından itibaren Compute Unified Device Architecture adının kısaltılmış hali olarak yayınlanan, NVIDIA grafik kartlarının 200 serisinden sonraki modellerinde desteklenen ve bu grafik işlem ünitelerinin genel amaçlı hesaplama işlemleri için kullanılabilmesine olanak tanıyan bir yazılım mimarisidir. Temel olarak bu mimariye sahip grafik kartları, GPGPU (General Purpose Graphics Processing Units) olarak nitelendirilmektedir [87].

Grafik İşlem Üniteleri, üç boyutlu grafiksel işlemler için gerekli hesaplamaları efektif bir biçimde işlemeleri için tasarlanmışlardır. Genelde her bir piksel için basit ve aynı işlemleri yaparlar; ancak piksel sayısının çokluğu ve bu piksellerin aynı zamanda gerçek-zamanlı olarak çalıştığı ve saniyede 60-100 defa yenilendiği de göz önünde bulundurulduğunda, grafik işlem ünitelerin paralel yapıya ne kadar elverişli olduğu açıkça görülmektedir. Bu bağlamda GPU'larda büyük ölçekli verilerin aynı işlemlerden geçerek paralel bir yapıda hesaplanması mümkündür. Bu nedenle GPU'lar için, yoğun hesaplama odaklı yüksek paralel yapılar için tasarlanmışlardır denilebilir [88].

Genel amaçlı Grafik İşlem Ünitelerini CPU'lardan ayıran önemli faktörlerden biri, transistörlerin bellek yönetimi ve veri akış kontrolünden ziyade çoğunlukla veri işleme için ayrılmış olmasıdır. CPU'larda ayrık işlemlerin farklı modelde yapılarla heterojen bir biçimde çalışmasına olanak tanıyan bu modüller GPU'larda daha basit ve aynı/benzer işlemlerin yürütülmesinde görev alarak, yapıyı mümkün olduğunca homojen tutup görevleri de GPU çekirdeklerine dağıtıp azami başarımı sağlamaktadırlar [88].

Geleneksel CPU donanım mimarisi ve GPU mimarisi Şekil 1.10 ile tasvir edilmiştir. CPU'da az sayıda Aritmetik Lojik Üniteleri (ALU) ve geniş kontrol ünitesinin be-

raberinde, paylaşımlı önbellek büyük yer kaplamaktadır. GPU'da ise çok sayıda küçük kontrol üniteleri ve paylaşımlı önbellek, daha çok sayıda çekirdeğe bloklar halinde atanıp işleyişi yönetmektedirler.

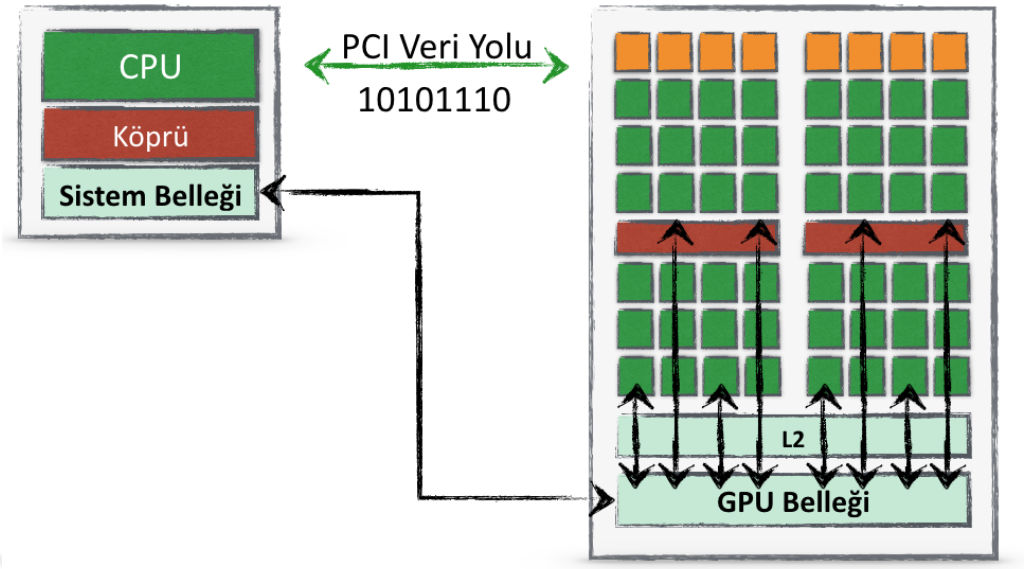


Şekil 1.10: CPU ve GPU'nun kontrol üniteleri, çekirdekleri ve bellek yapılarının kıyaslanması.

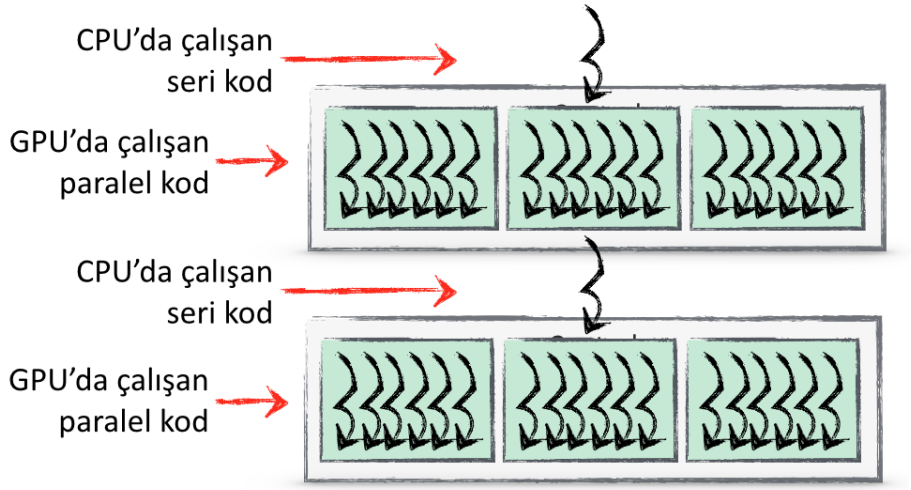
1.7.1 CUDA İş Akışı

CUDA mimarisinde gerçeklenen paralel programlar, belli bir yolu takip ederek GPU üzerinde çalışmaya başlar. Bu konuda ilk adım, üzerinde çalışılacak verinin öncelikle ana bellekten grafik kartı üzerindeki hızlı erişimli belleğe kopyalanmasıdır. Kopyalanan veri CUDA iş parçacıklarına belirlenen oranda dağıtılarak ilgili kod parçacıkları paralel bir biçimde koşturulur ve işlem sonrasında güncellenen veya yeni oluşturulan veri tekrar ana belleğe kopyalanarak paralel işlem tamamlanır. Bu işlem Şekil 1.11 ile gösterilmiştir. CPU ve GPU bellekleri arasında çift yönlü bir iletişim vardır. Veri bir defa GPU belleğine girdikten sonra GPU üzerinde işlenen verinin kontrolü CPU'dan çıkar, geri-bildirim gelinceye kadar verinin durumuyla ilgili bilgi CPU'da bulunmaz [88].

CUDA programlarının işleyişi ile ilgili bir diğer husus ise, programlarda yoğun paralellik gerektirmeyen bölümlerin düşük frekanslı GPU çekirdekleri yerine daha yüksek frekanslı olan CPU üzerinde çalıştırılmasıdır. Bu da gerektiğinde CPU ile asenkron çalışabilecek CUDA program parçacıklarının ve standart CPU kod parçacıklarının bir arada çalışmasına olanak vermektedir. İlgili anlatım Şekil 1.12 ile tasvir edilmiştir.



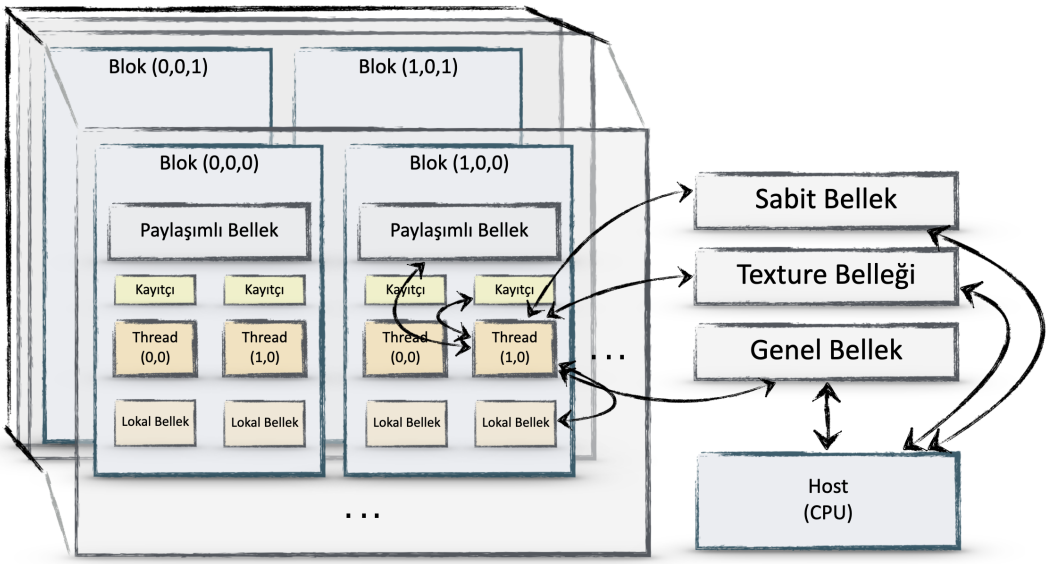
Şekil 1.11: CUDA ile gerçeklenen programlamların genel CPU-GPU veri akışları ve işlenmesi.



Şekil 1.12: CUDA ile gerçeklenen programlamların genel çalışma akışı.

1.7.2 CUDA Yazılım Mimarisi

CUDA, NVIDIA grafik kartları üzerinde çalışacak şekilde tasarlanıp, yazılım modelinde de ilgili grafik kartın donanımıyla uyum sağlayabilmektedir. Temel yazılım mimarisi Şekil 1.13 ile gösterildiği gibidir. GPU Grid; Genel, Sabit ve Doku Bellek Alanlarıyla birlikte çok sayıda İş Parçacığı bloğu içermektedir. Bloklar ilgili bellek alanlarıyla çift yönlü iletişim içindedirler. Bloklar arası senkronizasyon Genel Bellek Alanı üzerinden yapılmaktadır. İlgili bellek alanları aynı zamanda sistem ana belleği ile de çift yönlü iletişim içindedirler. Her bir iş parçacığı bloğu, içinde çok sayıda iş



Şekil 1.13: CUDA blok, iş parçacığı, bellek ve veri iletişimi için genel görünüm.

parçacığı barındırabilmektedir. Her bir blok içinde iş parçacıklarının birbirleri arasındaki haberleşmelerini ve senkronizasyonlarını sağlayabilmek adına paylaşımlı bellek alanları mevcuttur. Bloklar içindeki iş parçacıkları diğer bloklar içindeki iş parçacıkları ile bu paylaşımlı bellek üzerinden iletişim kuramazlar. İlgili bellekler yalnızca üzerlerindeki bloğun içinde yer alan iş parçacıkları tarafından okunabilir ve yazılabilirler. Her bir iş parçacığının, değişkenlerini saklamak amacıyla sahip olduğu bir kayıtçı bellek ve yerel bellek mevcuttur. Bu bellekler sayesinde fonksiyonları yürüten iş parçacıkları paylaşılan belleğe veya genel belleğe erişime gerek kalmadan hızlı bir biçimde görevlerini icra edebilirler [89].

CUDA'da bloklar ve blokların içlerindeki iş parçacıkları kendi aralarında üç boyutlu kübik bir yapıda dağılım gösterebilirler. Çözülmesi hedeflenen probleme göre belirlenen bu dağılım, aynı zamanda GPU çekirdeklerinin en etkin biçimde kullanılması açısından önem arz ederken, dolayısıyla çözülmesi hedeflenen problemin de yüksek başarımla işlem görmesine etki eder.

2 MATERYAL VE YÖNTEM

2.1 Ağ Trafik Verisetleri

Ağ trafiğinde saldırı tespit problemine bir yaklaşım önermek için, normal ve anormal veri kayıtlarının güncel örneklerini içeren kapsamlı bir veri üzerinde çalışmak esastır. Benzer şekilde, önerilen modelin son yıllardaki literatüre karşı sağlamlığını ölçebilmek için seçilen veriseti diğer çalışmalar tarafından da inceleniyor ve çeşitli senaryolarla ve modellerle denenmiş veya deneniyor olmalıdır. Bu amaçla, literatürde karşılaştırılabilir deney sonuçları sağlamak adına bir dizi genel veriseti mevcuttur. DARPA 98 [92], Birleşik Devletler Hava Kuvvetleri'nin yerel bilgisayar ağının simüle edildiği, MIT Lincoln laboratuvarı tarafından oluşturulan bir verisetidir. Veriseti, yaygın olarak kullanılan protokolleri ve DoS, U2R, R2L ve Probe dahil olmak üzere bir dizi saldırı örneğini içermektedir. Yedi haftalık eğitim ve iki haftalık test verilerinden oluşmaktadır; ancak yaşı ve yapısı gereği artık gerçek dünyadaki saldırı ve trafik akış senaryolarını pek yansıtmamaktadır. Bunun yerine, yakın zamana kadarki birçok araştırmada da temel referans noktası olarak kullanılmış olan KDD CUP 99 [90], Darpa 98'e göre daha popülerdir. Yakın zamana kadar test verisetinin yapısı gereği, bilinen ve bilinmeyen saldırıların temsil edildiği birçok çalışmada temel olarak kullanılmıştır. Bu veriseti, makine öğrenimi görevleri için daha uygun görünmektedir; ancak veriseti içerisinde algoritmaların performansını etkileyen çok fazla tekrar eden örneklemin olduğu gözlemlenmiştir. Bu nedenle NSL-KDD [76] isimli yeni bir veriseti üretilmiştir ve KDD CUP 99'un eksikliklerinin giderilmesi hedeflenmiştir. Bu kapsamda Kanada Siber Güvenlik Enstitüsü tarafından oluşturulmuş bu veriseti makine öğrenmesi çalışmaları için uzun yıllar fiili test ortamı haline gelmiştir. NSL-KDD hala birçok çalışma tarafından temel referans test ortamı olarak kullanılmaktadır. NSL-KDD'nin oluşumundan birkaç yıl sonra, daha güncel, etiketli ve geniş kapsamlı gerçek ağ trafiği saldırı verileri sağlamak için ISCX 2012 [70] isimli yeni bir veriseti oluşturulmuştur. ISCX 2012 güncel veriler içermesine

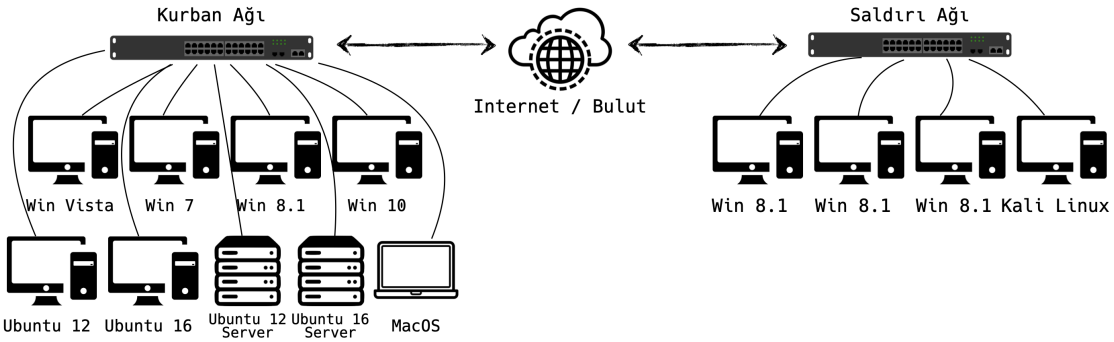
Tablo 2.1: Literatürde sıkça kullanılan STS verisetleri.

Veriseti	Yıl	Saldırı Türleri Ailesi	Öznitelik Sayısı
DARPA 98 [90, 91]	1998	DoS, probing, R2L, U2R	41
KDD CUP 99 [92, 93]	1999	DoS, probing, R2L, U2R	41
NSL-KDD [76]	2009	DoS, probing, R2L, U2R	41
ISCX 2012 [70]	2012	DoS, DDoS, Brute-Force, Infiltration	20
UNSW-NB15 [77]	2015	Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode, Worms	49
CICIDS2017 [94]	2017	DoS, DDoS, PortScan, Bot, Brute Force, Web Attack, Infiltration, Heartbleed	80
CICIDS2018 [94]	2018	DoS, DDoS, PortScan, Bot, Brute Force, Web Attack, Infiltration, Heartbleed	80
IoTID20 [95]	2020	DoS, Brute Force, Flooding (UDP, ACK and HTTP), ARP Spoofing, PortScan	80
MQTT-IoT-IDS2020 [96]	2020	Scan (Aggressive, UDP) Brute Force (SSH, MQTT)	44

rağmen, SSL ve TLS trafiğini içermemesinden ötürü çeşitli çalışmalarca tercih dışı kaldığı değerlendirilmektedir. Çünkü günümüz internet trafiğinin önemli bir kısmı bu katmanları kullanmaktadır. Tüm bu anlatılan verisetleri çok sayıda çalışma tarafından test, değerlendirme ve karşılaştırma ortamı olarak kullanılmış olsa da artık günümüz gereksinimlerini, saldırılarını ve teknolojilerini pek desteklemedikleri ve artık neredeyse güncel tüm makine öğrenmesi algoritmaları tarafından %100'e yakın bir oranda çözülebildikleri için artık daha yeni verisetlerine ihtiyaç doğmuştur. Bu kapsamda 2015 yılında hem normal hem de dönemin güncel 10 adet saldırı ailesini içeren (Exploit, worms, backdoor, DoS vb. gibi) güncel bir veriseti çalışması yapılmıştır [77].

Gelişen teknoloji, yenilenen saldırı ve tehditler ve buna bağlı olarak değişen ihtiyaçlar nedeniyle Kanada Siber Güvenlik Enstitüsü, saldırı tespit sistemlerinin eğitilmesini sağlamak ve başarımını test edebilmek amacıyla CICIDS2017 isimli yeni bir veriseti hazırlamıştır [94]. Veriseti beş günlük gerçek bir bilgisayar ağı trafiğini içermektedir. Birinci gün yalnızca normal bir akışın temsil edildiği verisetinde kalan dört gün hem nor-

mal hem de çeşitli güncel kategorilerde saldırılar içeren bir ağ trafiğinden oluşmaktadır. Enstitünün yayınladığı etiketli CSV verilerinde iki milyondan fazla örneklem satırı bulunmaktadır. Bu veriseti, gerçek dünya senaryolarını daha fazla yansıtan daha fazla sayıda örneğe sahip olmasının yanı sıra, daha dengesiz verilere sahip olmasıyla da UNSW-NB15'ten farklıdır. Verisetine dahil edilen tüm saldırılar Tablo 2.2 ile gösterilmektedir. Bu verisetindeki ağ trafiği kayıtları, saldırganlar ve kurbanlar arasında Mac OS, Windows ve Linux makinelerinin kullanıldığı gerçek bir bilgisayar ağından elde edilmiştir. Simülasyon iki adet ağdan oluşmaktadır: 1) Saldırı ağı, izinsiz giriş yapanların ve saldırganların trafiğini hedeflerine nasıl yönlendirdiğini simüle etmek için kullanılmıştır ve 2) bir saldırı gerçekleşene kadar iyi huylu/normal davranışı temsil eden kurban ağı. CICIDS2017 verisetinin temel mimarisini Şekil 2.1 ile gösterildiği gibidir. Veriseti iki formatta araştırmacılara sunulmuştur. Birincisi PCAP dosyalarından oluşan ham trafik verisi ve ikincisi de toplamda 80 adet sayısal ve kategorik özniteliklerden oluşan CSV dosyalarıdır. Ayrıntılı bir şekilde incelendiğinde, CICIDS2017 verisetinin incelenen literatürdeki diğer verisetlerine kıyasla en yüksek sınıf dengesizliğine sahip olduğu görülmektedir. 2018 yılında İletişim Güvenliği Kurumu (The Communications Security Establishment) da bu projeye dahil olmuş ve CICIDS2017'nin yeni bir varyasyonu yayınlamışlardır. Bu veri kümesine CSE-CIC-IDS2018 [94] adı verilmiştir. Yapısal olarak CICIDS2017'ye benzer, ancak 10 günlük ağ trafiğinden kayıtların toplandığı 16 milyondan fazla satıra sahiptir. Bu veri kümesi de aynı zamanda yüksek sınıf dengesizlik oranına sahiptir, ancak önceki CICIDS2017 ile karşılaştırıldığında sınıf bazında dengesizlik oranı daha düşüktür.



Şekil 2.1: CICIDS2017 verisetinin oluşturulduğu test/simülasyon ortamı [94]

Tablo 2.2: CICIDS2017 normal ve saldırı trafiğinin günler içerisindeki dağılım senaryosu.

Kayıt günü	**Süre**	**Saldırı**
Pazartesi	Tüm gün	Normal trafik
Salı	Tüm gün	FTP-Patator, SSH-Patator
Çarşamba	Tüm gün	DoS (Hulk, GoldenEye, Slowloris, Slowhttptest) Heartbleed
Perşembe	Sabah Öğleden sonra	Brute Force, XSS, SQL Injection Infiltration
Cuma	Sabah Öğleden sonra	Bot DDoS, PortScan

IoT dünyasının kısa sürede hızlı bir şekilde genişlemesi ve bu alandaki çalışmaların çeşitliliği nedeniyle sadece IoT özelinde farklı veriseti çalışmaları da yürütülmeye başlanmıştır. Bu amaçla araştırmacılar 2019 yılında örnek bir IoT veri akışı kümesi sağlayabilmek için internete bağlı bir gece mumu (gece lambası) ve bir Wi-Fi kamera kullanarak, bu iki cihazın kendileri arasındaki haberleşmelerini kayıt altına alarak bir çalışma gerçekleştirmişlerdir [97]. Bu veri kümesi, DoS, Scan (Port ve OS), Flooding (UDP, ACK ve HTTP) ve Telnet Brute Force dahil olmak üzere çeşitli ağ saldırıları türlerini içermektedir. Veriseti, dizüstü bilgisayarlar ve akıllı telefonlar dahil olmak üzere ağa

Tablo 2.3: CICIDS2017 verisetinin sınıf dağılımı ve saldırı tiplerinin detayları.

Etiket	Dağılım (%)	Açıklama
Benign	80,3189	Normal trafik örnekleri.
DoS Hulk	8,1376	DoS tipindeki saldırıya zemin hazırlamak için Hulk isimli bir araç kullanıp trafik üretir.
PortScan	5,6156	Kurban makineden veri sağlayabilmek için hedefteki bilgisayarın farklı portlarını tarar ver bu portlara veri göndermeye çalışır.
DDoS	4,5272	DoS saldırısı için çok sayıda saldırı makinesini aynı anda kullanır.
DoS GoldenEye	0,3639	DoS saldırısını gerçekleştirmek için GoldenEye isimli bir araç kullanır.
FTP-Patator	0,2805	Kurban makineden kullanıcı giriş bilgilerini çalmak için FTP-Patator kullanarak Brute Force (Kaba Kuvvet) saldırısı düzenler.
SSH-Patator	0,2085	Kurban makineden kullanıcı giriş bilgilerini çalmak için SSH-Patator kullanarak Brute Force (Kaba Kuvvet) saldırısı düzenler.
DoS slowloris	0,2049	Slow Loris isimli bir araç vasıtasıyla DoS saldırısı düzenler.
DoS Slowhttptst	0,1944	Bir sunucuyu yavaşlatmak için sunucunun kaldırabileceğinden çok daha fazla HTTP bağlantısı kurup talep göndermeye çalışır.
Bot	0,0691	Truva atları kullanarak bir sunucunun içine girmeye ve bu sunucunun bağlı bulunduğu tüm ağa yayılmaya çalışır.
WA Brute Force	0,0532	Kullanıcı giriş bilgilerini çalmak için deneme yanılma tekniği ile çok sayıda talep gönderir.
WA XSS	0,0230	Bir web uygulamasının form alanlarına zararlı kod parçacığı göndermeye çalışır.
Infiltration	0,0012	Sızma (infiltration) araçları kullanarak bir sunucuya tam erişim alacak şekilde sızmaya çalışır.
WA SQL Inj.	0,0007	Uygulamaları kırmak için SQL komutları gönderir ve çalıştırır.
Heartbleed	0,0003	OpenSSL'deki bir açığı kullanarak web uygulamalarından değerli bilgiler sızdırmaya çalışır.

bağlı farklı istemci cihazları ile 42 ham ağ paket dosyasından (PCAP) oluşmaktadır. Kısa bir süre sonra, başka bir çalışma, PCAP dosyalarından birçok özelliği IoTID20 [95] adlı daha okunabilir CSV dosyalarına çıkararak bu verisetinin bir türevini geliştirmiştir. Bu çalışma, IoT ağlarında izinsiz giriş tespiti araştırması için karşılaştırılabilir bir test ortamı sağlamayı amaçlamıştır. Nispeten yeni bir veriseti olduğu için literatürde bu verisetini kullanan, makine öğrenmesi çıktılarıyla saldırı tespiti açısından kıyaslama yapılabilecek çok fazla çalışma bulunmamaktadır.

IoT ağları, çok çeşitli cihazlarla uyumlu olabilmek için sıklıkla hafif ve spesifik yapılar kullandığından bu yapılara özgü güvenlik açıklarını da barındırmaktadırlar.

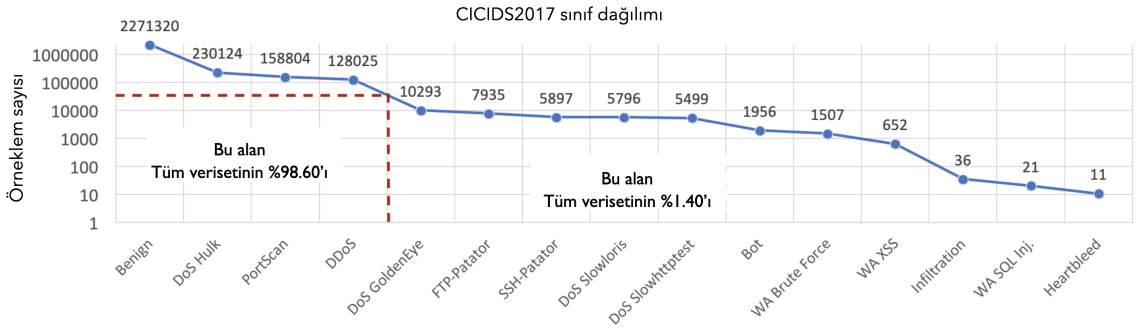
Bu yapılardan biri, IoT cihazlarının (diğer benzer protokollerle birlikte) iletişim amacıyla kullandığı, abone ve yayıncı mantığına dayanan bir Mesaj gönderme (Message Queuing Telemetry Transport - MQTT) protokolüdür. Yakın tarihli bir araştırma, MQTT tabanlı çok sayıda gerçek dünya senaryosu olmasına rağmen hali hazırda bunu baz alan bir IDS verisetinin eksikliğini göstermiştir. Bu amaçla araştırmacılar, MQTT-IoT-IDS2020 [98, 96] adlı yeni bir veriseti geliştirmişlerdir. Bu veriseti, Scan (Agresif, UDP) ve Brute Force (SSH, MQTT) dahil olmak üzere dört IoT saldırısı içermektedir. Veriseti 44 öznitelik içermektedir ve hem PCAP dosyaları ile birlikte hem de işlenmiş CSV dosyası ile yayınlanmaktadır. Bu veriseti, özellikle MQTT tabanlı saldırılara yöneldiği ve diğer birçok saldırı türünü içermediği için çalışmamızın kapsamı dışındadır; ancak IoT güvenliği konusunda çalışmak isteyen gelecek araştırmacılar için önemli bir referans noktası olduğu değerlendirilmektedir. Bu veriseti ayrıca nispeten yenidir ve literatürde bu verisetini kullanan ve kıyaslamalar yapan bir çalışma bu yayının hazırlandığı tarih neticesinde henüz bulunmamıştır.

Başlık 1.3 ile daha önce bahsedilen gerçek dünya IoT tehdit senaryolarının uyumluluğunu ve çeşitliliğini sağlamak için bu çalışma temel veriseti olarak CICIDS2017 seçilmiştir. Ayrıca, halefi CSE-CIC-IDS2018'e kıyasla daha yüksek sınıf dengesizliği ve daha düşük hesaplama ve işlem süresi gereksinimi nedeniyle bu çalışmanın amacına daha iyi hizmet edeceği değerlendirilmiştir.

Çalışma kapsamında önerilecek yaklaşımları denemek için seçilen CICIDS2017 veriseti, aşağıdaki temel adımlarla bir ön işleme tabi tutulmuştur:

- Normal ve saldırı verilerini içeren beş günlük ayrık veriler, öncelikle yekpare bir yığın oluşturacak şekilde bir araya getirilmiştir. Bu esnada verisetinde yer alan ve birbirinin aynısı olan satırlar ham veriden silinmiştir.

- Makine öğrenmesi ve türevi yaklaşımların ana unsuru veriden örüntüler yakalamaktır ve bunu verideki irili ufaklı değişimleri gözlemleyerek sağlayabilirler. Bu verisetindeki bazı özniteliklerin [1] sadece sıfır (0) değeri içerdiği tespit edilmiştir. Bu özniteliklerin çalışmaya olumlu hiçbir katkısı olmayacağı, aksine gereksiz işlem gücü tüketeceği değerlendirildiği için bu öznitelikler de ham veriden silinmiştir. Bunun dışında *flow ID* ve *timestamp* isimli öznitelikler de sayısal değerlere çevrilemeyeceği için ham veriden silinmişlerdir. Son olarak da örneklerin içinde çeşitli kolonlarda eğer *NaN, -inf, +inf* gibi sonlu bir şekilde sayısallaştırılamayacak ve algoritmaları hataya zorlayacak tüm satırlar ham veriden silinmişlerdir.



Şekil 2.2: CICIDS2017 verisetinin dengesiz dağılımını gösteren sınıf sayıları.

- Dört adet öznitelik (*source port ve source IP, destination port ve destination IP*) ikili kodlama (binary encoding) tekniği ile sayısal veriye çevrilmişlerdir.

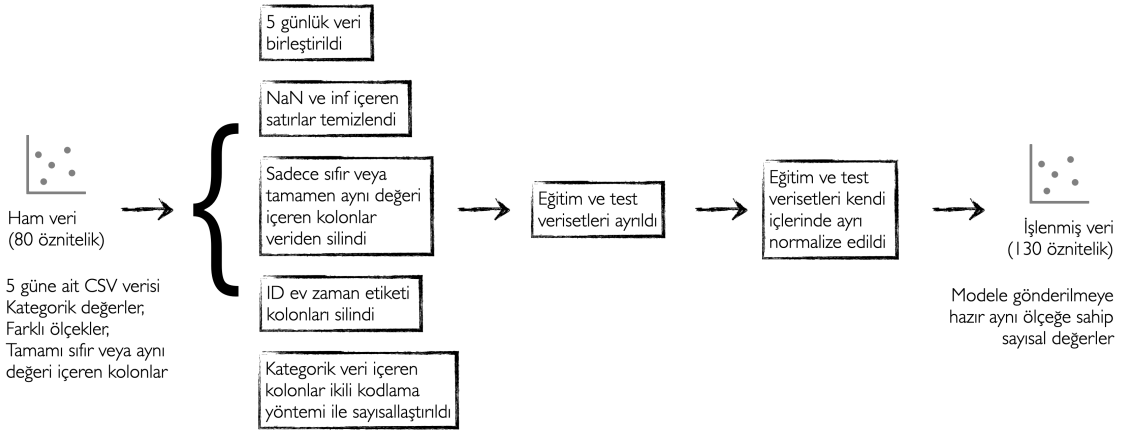
- Temizleme ve sayısallaştırma işlerinin ardından elde kalan veri yığını artık tamamen sayısal değerlerden oluşmaktadır. Makine öğrenmesi algoritmalarındaki ak-

[1] bwd psh flags, bwd urg flags, fwd avg bytes/bulk, fwd avg packets/bulk, fwd avg bulk rate, bwd avg bytes/bulk, bwd avg packets/bulk, bwd avg bulk rate

tivasyon fonksiyonlarına da uyumluluğu esas almak için bu sayısal veriler 0 ile 1 arasında *(Python sklearn kütüphanesi ile)* ölçeklenmiştir ve ardından kolonların ortalama değerleri çıkarılıp birim varyansına ölçeklenmiştir, yani normalize edilmiştir.

- Tüm bu işlemlerin ardından elde edilen veri artık 130 adet normalize edilmiş özniteliğe sahip, toplamda 2.827.876 adet örneklem içeren bir veri haline gelmiştir. Bu ön işleme ile elde edilen verinin detaylı sayısal sınıf istatistikleri Tablo 2.3 ile gösterilmiştir ve bu sınıfların veriseti içerisindeki dağılımı ise Şekil 2.2 ile tasvir edilmiştir.

İlgili ön işlem adımları Şekil 2.3 ile kısaca tasvir edilmiştir.



Şekil 2.3: CICIDS2017 verisetinin bu çalışma kapsamındaki ön işlem adımları akışı.

2.2 Rastgele Gürültü ile Veri Artırma

Bu ilk veri artırma tekniğinde, mevcut veriye belirlenen sayı kadar basit bir şekilde rastgele gürültü ekleyen ve bu gürültülü verileri mevcut verilerin üzerine ekleyerek yeni örneklemler oluşturan bir yaklaşım güdülmüştür. Veri üretici fonksiyon, rastgele seçilen örneklere normal dağılım olacak şekilde ve sınıftaki verilerin özniteliklerinin sayısal ortalamaları alınarak ve bu ortalamaları $0,001$ ile çarpıp yeni gürültülü veri elde etmektedir. Her bir iterasyonda ise fonksiyon, bir önceki katsayıyı yarıya bölüp tekrar tüm veriye gürültü eklemektedir. Bu sayede yeni oluşan sentetik daha önceki sentetik veriler kul-

lanılarak da üretilmesini ve buna rağmen düzlemde özniteliklerin çok kaybedilmeden belli grupların içerisine yakınsamasını sağlamıştır. Rastgele veri üreten fonksiyon Algoritma 1 ile gösterilmiştir.

Algoritma 1 Rastgele Veri Üreteci

Input: class, num_samples_to_generate
Output: noisy_samples
Initialization :
$noise = class.samples.mean() * 0.001$
LOOP Process
for $i = 1$ to $num\_samples\_to\_generate$ **do**
  $noisy\_sample = sample + noise * random.normal(size = sample.shape)$
  $noise = noise * 0.5$
end for
return $noisy\_sample$

2.3 SMOTEENN ile Veri Artırma

SMOTEENN, *SMOTE* (Sentetik azınlık örnekleme tekniği) ve *ENN* (Düzenlenmiş en yakın komşu) tekniklerinin birleştirilerek örneklem sayısı az olan sınıflara sentetik veri üretme amacıyla kullanılan bir tekniktir. *SMOTE* ile belirlenen sınıfların örneklemleri lineer deformasyona (linear interpolation) uğratılarak bu örneklemler üzerinden yeni sentetik veriler üretilir. Ardından *ENN* ile üretilen yeni örneklemlerin oluşan merkez kümeye ve birbirlerine en uzak olan olanları kümeden çıkarılır [2] [99].

2.4 CTGAN ile Veri Artırma

Modele uygulanan veri artırma tekniklerinin üçüncüsü olarak iki eşik değeri üzerinden CTGAN yapay öğrenme modeli kullanılmıştır. Bu senaryoda CTGAN algoritması 100 iterasyon boyunca, her iterasyonda 10 batch seçili olmak kaydıyla eğitim setindeki (belirlenen eşik değerlerine göre veri artırma tekniği uygulanacak) sınıflar üzerinden eğitilmiştir. SMOTEENN ve rastgele veri üreteci fonksiyonuna kıyasla bu senaryoda kullanılan model, verinin karakteristiğini öğrenip onun üzerinden sentetik veri üreteceği için bu eğitim safhasının tamamlanması gerekmiştir. CTGAN, temelde bir çekişmeli üretici ağ

[2] Bu çalışmada ENN algoritmasının varsayılan değeri 3 olarak değiştirilmeden kullanılmıştır.

yapısı olan GAN'ın tablo verilerine uyarlanmış halidir. GAN, normalde girdi olarak verilen resimlerin karakteristiğini bir üretici bir de bunu sorgulayan iki ağ tipi ile karşılıklı puanlama yoluyla öğrenmeyi dener. Üretici ağ, öğrenilen resimlerin bir benzerini sentetik ortamda üretmeye çalışırken, sorgulayıcı ağ ise üretilen sentetik resimlerin gerçek mi yoksa sahte mi olduğunu kendi puanı ile belirler. Buna istinaden de üretici ağ sentetik verilerinin kalitesini artırmaya çalışır. Aynı yapının tablo ve sayısal verilere uyarlanlanmış hali ise CTGAN'dır.

3 ÖNERİLEN MODEL

Bu tez çalışması kapsamında, temelde bir IoT ağındaki akan trafik verilerinin Derin Öğrenme ve veri üretme teknikleri kullanılarak analiz edilmesi ve bu ağa yapılan saldırıların türlerinin yüksek başarımla sınıflandırılması amaçlanmaktadır. Bu amaçla CICIDS2017 veriseti baz alınmış ve verisetinde yer alan normal ve 14 adet saldırı tipini içeren farklı ağ paketlerinin TCP başlık bilgileri kullanılmıştır. Bu başlık bilgilerinin saldırı verisi içerip içermediği, eğer saldırı var ise hangi tip bir saldırı olduğu Derin Öğrenme tekniği ile kontrol edilmiştir. Çalışma kapsamında önerilen Derin Öğrenme modeli, otomatik kaliteli öznitelik üretimi ve çıkarımı için iki katmandan oluşan bir ayrılabilir evrişimsel sinir ağı (Separable CNN) modeli içermektedir. CNN yığınının içinde her biri 3x1 boyutunda olan filtreleme kerneli ve Rectified Linear Unit (ReLU) aktivasyon fonksiyonu bulunmaktadır. ReLU ardından her bir örneklem, boyut indirgemesi için MaxPool katmanına gönderilmiştir. Bu katman ile örneklemlerin her dört özelliğinden yalnızca en yüksek değere sahip olan verisi bir sonraki katmana gönderilmiştir. Bu sayede verinin baskın karakterleri korunurken aynı zamanda veri boyutu da düşürülmüştür. Bu iki katmanlı ayrılabilir evrişim modeli, girdilerin her bir özelliği için derinlik bazında bir evrişim sağlamaktadır ve orijinal olarak ilk Xception modelinde kullanılmış ve iyi bir izlenim vermiştir [100]. Bu modelin çıktısı daha sonra dört katmanlı kendi kendini normalize edebilen (self-normalizing) yapıdaki bir tam bağlı yapay sinir ağına gönderilmiştir. Bu modelin çıktısı ise 15 adet çıktı nöronu içeren bir softmax fonksiyonudur.

Önerilen modeldeki CNN katmanlarından ilki 128 ve ikincisi 256 adet 3x1 boyutunda kernele sahip olan filtreler içermektedir. Modelin bu kısmı, girdi verilerinden çıkan bir öznitelik haritası oluşturmaktadır ve verilerin kendi içindeki özniteliklerin özellik korelasyonlarının çıkarılmasını sağlamaktadır. Modelin ikinci kısmı ise toplamda dört katman içerir ve sırasıyla 128, 64, 32 ve 15 adet kendini normalize edebilen tam bağlı

nöron grubundan oluşan bir sınıflandırıcıdır. Son sınıflandırıcı katmanı dışındaki ilk üç tam bağlı katman, Üstel Doğrusal Ünite (Scaled Exponential Linear Unit - SELU) aktivasyon fonksiyonunu ve akabinde batch normalizasyonu [101] tekniğini kullanmaktadır ve nöronlar arasındaki bağlantılarda %10 oranında bir alfa veri düşürme tekniğini (alpha dropout) uygulamaktadır. Modelin bu bölümünde SELU fonksiyonu, nöron aktivasyonlarını sıfır ortalama ve birim varyansına otomatik olarak getirmeye çalışır. Böylece her iterasyon sonunda kendi kendini normalleştirebilen bir özellik oluşturur. Alpha dropout ise ağın tamamının her gelen veriyi öğrenmesi yerine, gelen girdileri ağın farklı bölümlerinin öğrenmesini sağlamak için bazı bağlantıları belirli bir algoritmaya göre keser ve böylece normalleştirme özelliğini korumak için SELU aktivasyonunu doygunluk değerine ayarlar. Aynı zamanda ağın genelleştirilmesine katkıda bulunur ve ezberleme oranını azaltır. Önerilen modelde aynı zamanda sisteme ilk giren ham verilerin hiç işlenmemiş halleri de son katmandan hemen önceki işlenmiş verilere dahil edilir ve bu sayede hem öznitelikleri çıkarılmış ve işlenmiş veri hem de bu verinin orijinal özelliklerinin korunduğu ham hali en son sınıflandırıcıya gönderilir. Bu teknik çeşitli araştırmalarda ve güncel bazı modellerde uygulanmış olup öğrenme mekanizmasının bir plato şeklinde kalmasının önüne geçmiş olur [102]. Modeldeki en son katman, verisetinde de yer alan 15 adet farklı sınıfı temsil eder ve modelin çıktısının olasılık dağılımını sağlayan softmax aktivasyon fonksiyonunu kullanır. Aynı zamanda Adam isimli, Derin Öğrenme teknikleri içinde çok yaygın olan optimizasyon fonksiyonunun daha yeni bir versiyonu olan AdamW optimizasyon fonksiyonu da bu modelde kullanılmış ve test edilmiştir. AdamW, Adam fonksiyonunun bilinen bazı eksiklerini kapatmayı hedefleyen ve hem öğrenme oranının hem de bu oranla ilişkili başka katsayıların daha iyi kullanılmasını sağlayıp modellerin önceki versiyona göre daha hızlı öğrenmesini hedeflemektedir [103]. Bu optimizasyon fonksiyonu sayesinde modeller daha erken iterasyonlarda daha iyi öğrenme becerileri geliştirmeyi başarmışlardır.

Bu çalışmada önerilen ve yukarıda anlatılan modelin detaylı mimarisi Şekil 3.1 ile tasvir edilmiş ve modelde kullanılan hiper parametrelerin detayları Tablo 3.1 ile ayrıntılı şekilde

Tablo 3.1: Önerilen modelin hiper parametreleri.

Parametreler	Değerler
Girdi/özellik sayısı	130
Evrişim filtre sayısı	128x3 - 256x3
CNN katmanı aktivasyon fonksiyonu	ReLU
Tam bağlı nöron katmanları	128 - 64 - 32 - 15
Tam bağlı katmanlar aktivasyon fonksiyonu	SELU
Bağlantı düşürme metodu	alpha dropout
Düşürme (dropout) oranı	10%
Normalleştirme metodu	Batch normalization
Çıktı sayısı	15
Çıktı katmanı aktivasyon fonksiyonu	softmax
Optimizasyon fonksiyonu	AdamW

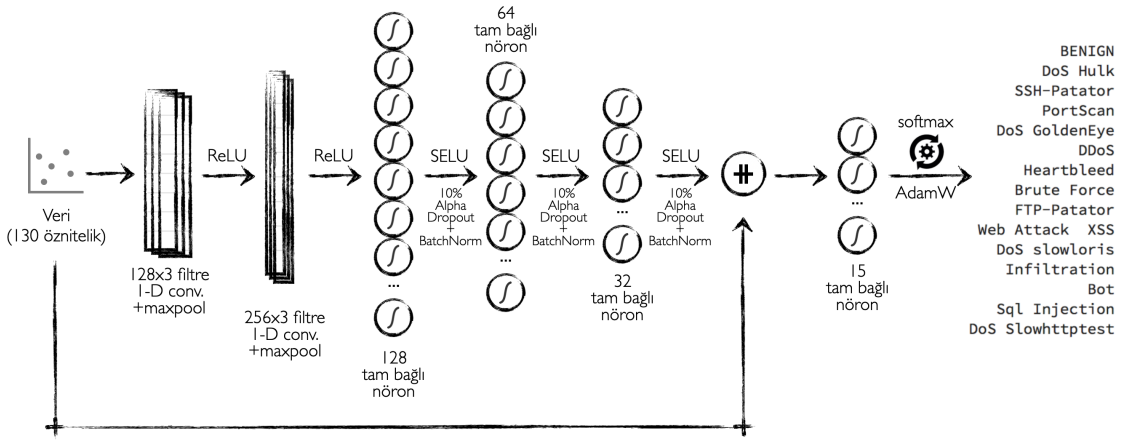
gösterilmiştir.

Önerilen modelin testleri yapılırken, her defasında farklı rastgele süreçlerden kaçınmak ve makul bir strateji kurabilmek adına, sınıflar içindeki örneklem sayısına bağlı özelleştirilmiş bir bias (sapma) değeri modeldeki son katmana eklenmiştir. Bu bias değeri statik bir değer olup temel olarak eğitim setindeki en az örnekleme sahip sınıfın sayısının, eğitim setindeki tüm örneklere oranının logaritmasına karşılık gelmektedir. Bu değerin varlığı ve içeriği, aynı zamanda çıktı katmanındaki 15 nöronun daha düşük örneklem içeren sınıflardaki öğrenme sürecinde daha büyük bir öğrenme katsayısı olmasını sağlar. Bu özel bias [1] değeri Formül 3.1 ile gösterildiği gibidir.

$$ozel\_bias = \log(\frac{en\_az\_orneklem\_sayisi}{toplam\_orneklem\_sayisi}) \tag{3.1}$$

Özel bias formülünü hesapladıktan sonra, buna bağlı olarak son katmanda tüm sınıflar için bir sınıf ağırlık değeri hesaplanmaktadır. Bu ağırlık değerleri de temel olarak sınıflardaki örneklem sayısı ile ters orantılı olmaktadır. Bunu da özel bias değeri sağlamaktadır. Sınıf ağırlıklarının oluşturulmasının, özellikle geri yayılma (back propagation) işlemi esnasında sistemin başarımını artması yönünde olumlu katlıları olduğu değerlendirilmektedir.

[1] Özel bias yaklaşımı, Derin Öğrenme topluluğunun deneyimlerinden faydalanılarak Tensorflow dokümantasyona [104] en iyi pratikler olarak eklenmiştir. Bu çalışmada da buradan faydalanılmıştır.



Şekil 3.1: Önerilen modelin detaylı mimarisi. Model, tek boyutlu evrişim katmanları ve tam bağlı katmanlar kullanır. Çıktı katmanında özel bias ilklendiricisi ve diğer katmanlarda Lecun Normal kernel ilklendiricisi yer almaktadır.

4 BULGULAR VE TARTIŞMA

4.1 Deneysel Çalışmalar

Bu bölüm, önceki bölümlerde anlatıldığı şekilde işlenmiş olan CICIDS2017 veriseti üzerinde gerçekleştirilen sınıflandırma deneylerini göstermektedir. Bu amaçla önerilen modele gönderilen çeşitli verileride açıklamaktadır ve her tekniğin sonucunu karşılaştırmaktadır. Deneyler toplamda iki adet ana senaryo içermektedir. İlk senaryo önişlemden geçirilmiş CICIDS2017 verisetinin eğitim setini başka herhangi bir işlemden geçirmeden direkt olarak önerilen modele göndermektir. İkinci senaryo ise kendi içinde birtakım alt kırılımlar içermektedir. Bunlar, önişlemden geçirilmiş veriye çeşitli sentetik veri artırma tekniklerinden uygulamak ve bu yeni oluşan veriler ile eğitim işlemini gerçekleştirmektedir ve bu sayede de verilerin düzensiz dağılımı ile öne çıkan bazı handikapları ortadan kaldırmaya çalışmaktır. Bu iki senaryo da kendi karakteristiğine sahiptir ve çözülmesi hedeflenen problem üzerinde kendilerine has çıktılar sunmaktadırlar. Bahsedilen ikinci ana senaryoda üç adet sentetik veri artırma tekniği kullanılmıştır ve her üç teknik için iki adet alt kırılım mevcuttur.

Bu çalışmadaki kodlamalar Python dili kullanılarak TensorFlow 2.2 kütüphanesi üzerinden yazılmıştır ve Ubuntu 20.04 işletim sisteminde çalıştırılmıştır. Eğitim ve test aşaması, her birinin GPU'lardan birini aynı anda kullandığı toplam üç adet nVidia GPU ile hesaplanmıştır. Tablo 4.1, deneylerin yapıldığı donanımı kısaca özetlemektedir.

Tablo 4.1: Deneylerin yapıldığı donanım.

Kategori	**CPU**	**GPU (x4)**
Üretici	Intel	NVIDIA
Model	Xeon(R) Gold 6138	Tesla V100
Saat Frekansı	2000 MHz	1246 MHz
Çekirdek sayısı	80	5120
DRAM Bellek	100 GB DDR4	16 GB DDR5

Önerilen model ve senaryolardan daha iyi iç görüler elde etmek ve yaklaşımın

sağlamlığını incelemek için tüm deneyler, k'nin beş olduğu k-fold çapraz doğrulama tekniği ile yürütülmüştür. beş değerindeki k-fold ile verisetinden birbirinden farklı beş adet eğitim ve test kümeleri çıkarılmıştır. Her bir çıktıdaki eğitim kümesi ve test kümesi oranı %80 eğitim ve %20 test olacak şekilde ayarlanmıştır. Burada k değerinin beş olmasının ardındaki temel motivasyon, en nadir görülen sınıfın eğer tüm verilerinin %20'si test olarak ayarlanırsa test kümesinin yalnızca iki adet örneklem içermesidir. Bu durumda en az örneklem sayısına sahip olan Heartbleed, eğitim kümesinde dokuz örneklem ve test kümesinde iki örneklem almıştır. k'nın daha yüksek sayıda olması bu test kümesi sayısını 1'e indirmektedir. k'nın daha düşük olması ise çapraz doğrulamanın kalitesini düşürmektedir. Bu nedenle çalışma kapsamında verisetinin karakteristik yapısı düşünüldüğünde k-fold tekniğindeki k değerinin beş olması uygun görülmüştür. Artık modele uygulayacağımız beş farklı eğitim ve test kümemiz bulunmaktadır. Deneyler kapsamında eğitilen Derin Öğrenme modeline gönderilen training ve test setlerindeki her bir dağılım (fold), 100 adet iterasyonla eğitilmiştir ve 10240 batch sayısına sahip olmuştur. Her bir testin çıktısı ise daha sonra aritmetik ortalama ile birleştirilip deneylerin genel çıktı kalitesi belirlenmiştir. Bu nedenle aşağıda yer alan tüm çıktılar en iyi eğitim verisinin çıktısından ziyade, beş adet eğitim-test sürecinin ortalamasını [105] içermektedir.

Bir veri kümesinin işlenmesi ve ardından çıktıları hakkında net iç görüler elde edebilmek için şimdiye kadar çok sayıda değerlendirme metriği tanımlanmıştır. Problemin yapısına ve ihtyaçlara bağlı olarak, bazen sadece doğruluk (accuracy) metriği bile yeterli olmaktadır. Ancak özellikle dengesiz sayıda sınıflar içeren problem uzayında daha donanımlı metriklerle çalışmak değerlendirme sağlığı için oldukça önemlidir. Kesinlik ve duyarlılık metrikleri, dengesiz sınıflar içeren problemlerde önemli ölçütlerden ikisidir. Kesinlik, aslında pozitif sınıfa ait olan pozitif gözlemlerin oranını ifade eder (Formül 4.1 içinde tanımlanmıştır). Duyarlılık olarak da adlandırılan geri çağırma, veri kümesindeki (Formül 4.2 içinde tanımlanan) tüm pozitif örneklerden yapılan pozitif gözlemleri nitelendirir. Bu nedenle, bu iki ölçütü en üst düzeye çıkarmak, bir sorun için en iyi çözümü bulmada daha yüksek verimlilik sağlar. Bundan ötürü, bu tarz problemlerde her iki değeri

de aynı anda üst düzeye çıkarmak adına geliştirilen F_1 puanı (F-ölçütü ya da F_1-skoru olarak da adlandırılır) literatürde çok sık kullanılmaktadır. F_1 puanı, kesinlik ve duyarlılığın harmonik ortalamasıdır (Formül 4.3). Bu çalışmada, F_1 puanına güvenmek, bu dengesiz dağılımda etkinlik arayışını tatmin etmektedir. Bu amaçla, tüm sonuçlar F_1 puanı ve ayrıca ilgili kesinlik ve duyarlılık değerleri ile hesaplanmıştır [1].

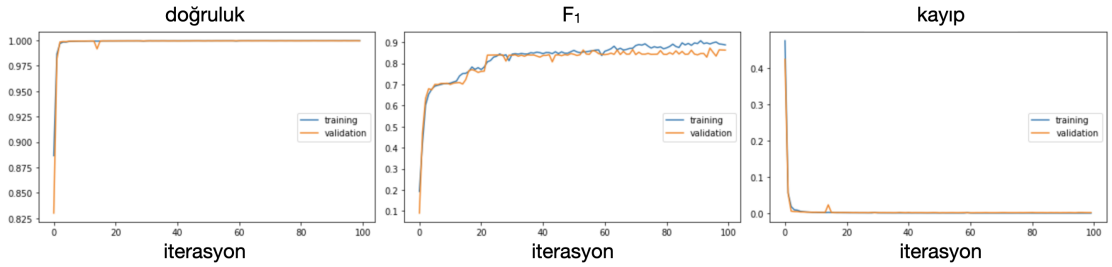
$$kesinlik = \frac{TP}{TP + FP} \tag{4.1}$$

$$duyarlilik = \frac{TP}{TP + FN} \tag{4.2}$$

$$F_1\ puani = 2\frac{kesinlik * duyarlilik}{kesinlik + duyarlilik} = \frac{TP}{TP + \frac{1}{2}(FP + FN)} \tag{4.3}$$

Deneylerin ilk senaryosu, 130 özniteliğe sahip işlenmiş CICIDS2017 verilerinin %80'inin k-fold (k=5) ile direkt olarak Şekil 3.1 ile gösterilmiş olan önerilen modele uyarlamasıdır. Her deneyde Derin Öğrenme modeli toplamda 100 iterasyon sonucunda durdurularak elde edilen sonuçlar kaydedilmiştir ve tüm dağılımlar bittiğinde bunların çıktılarının aritmetik ortalamaları alınmıştır. Bu alınan ortalama da deneyin ilgili senaryosunun doğruluk, kesinlik, duyarlılık ve F_1 puanına karşılık gelmiştir. Bu ilk senaryodaki işlenmiş verinin önerilen modele gönderilmesiyle birlikte, 15 sınıftan oluşan verisetinin 11 tanesinin F_1 puanı %98,4'ün üzerinde bir değer almıştır. Kalan sınıflar ise %22,1 ile %93,5 arasında bir dağılım göstermiştir. Bu senaryoda web attack ailesinde yer alan sınıfların en az başarıya sahip olduğu gözlenmiştir. Bu atak ailesinden *WA XSS* %16,84 duyarlılık ve %22,15 F_1 puanına sahipti. *WA SQL Injection*, %32,99 duyarlılık ve %41,33 F_1 puanına sahipti. Bu atak ailesinde modelin düşük başarım elde etmesinin sebebi, veri içerisindeki özniteliklerin verinin karakteristiğini olabildiğince yansıtmamasından kaynaklanmaktadır. Bunun anlamı, verisetinde yalnızca TCP/UDP paket bilgilerinin yer alması, buna rağmen web attack ailesindeki saldırı tiplerinin parmak izlerinin çoğunlukla paketlerin

[1] TP: True Positive, FP: False Positive, FN: False Negative



Şekil 4.1: Modelin 100 iterasyon boyunca eğitilmesi süreci. Grafikler sırasıyla doğruluk, F_1 ve modelin kayıp (loss) değerlerini göstermektedir.

içeriğindeki yükte (payload) yer almasından kaynaklanmaktadır. Bununla birlikte ayrıca bu atak ailesinin örneklem sayısı görece diğer bazı sınıflara göre az sayıda kalmıştır. Bu da zaten az olan saldırı karakteristiği verisinin model tarafından yeterince öğrenilememesine sebep olmuş ve düşük başarımda etkenler arasında olduğu değerlendirilmiştir. Benzer şekilde, *WA Brute Force* en başarısız üçüncü sınıf olarak karşımıza çıkmıştır. Brute-Force tipi saldırılar, doğası gereği uzun bir zaman dilimine yayılabilecek türden saldırı tipleridir. Buna istinaden, bu tarz saldırılar üstünde zamana bağlı değişkenleri ve verileri kontrol edebilen bir modelin (örneğin LSTM gibi, tek tek örneklemleri inceleyen bir modelden ziyade daha iyi başarım sağlayacağı değerlendirilmektedir. Buna rağmen, verileri tek tek inceleyerek öğrenen önerilen model, CNN'deki özellik çıkarımının ve hiper parametrelerin de yardımıyla *WA Brute Force* saldırısında yine de önemli bir başarı elde etmiştir. Örnek sayısı Tablo 2.3 ve Şekil 2.2'de görülebileceği gibi, verideki yüksek dengesizlik sonuçları da etkilemiş ve sınıflandırma için daha fazla örneğin hazırda bulunması gerekliliğini ön plana çıkarmıştır. Bunların dışında, verisetindeki en az örnekleme sahip sınıf olmasına rağmen *Heartbleed* sınıfı, daha fazla örneklem içeren saldırı tiplerine göre çok daha kolay öğrenilmiş ve bu saldırıyı bulma başarımı %100 olarak görülmüştür. Bunun arkasındaki ana sebep, *Heartbleed*'in TCP/UDP başlıklarında çok belirgin parmak izlerine sahip olması ve özelliklerinin problem uzayındaki diğer sınıflarla örtüşmemesidir. Bu ilk deneyde, F_1 puanının tüm dağılımlardaki toplam ağırlıklı ortalaması %99,9 ve makro ortalaması [2] %88,7 olarak hesaplanmıştır. Bu deneydeki makro ortalama du-

[2]Makro ortalama, örnek sayısından bağımsız olarak tüm puanların aritmetik ortalamasıdır.

Tablo 4.2: raw senaryosu için önerilen modelin detaylı sınıflandırma sonuçları.

Etiket	Kesinlik	Duyarlılık	F_1 puanı
Benign	0,9899	0,9889	0,9894
DoS Hulk	0,9993	0,9998	0,9995
PortScan	0,9998	0,9994	0,9996
DDoS	0,9998	0,9998	0,9998
DoS GoldenEye	0,9930	0,9928	0,9929
FTP-Patator	0,9986	0,9998	0,9992
SSH-Patator	0,9991	0,9986	0,9988
DoS slowloris	0,9852	0,9917	0,9884
DoS Slowhttptst	0,9899	0,9889	0,9894
Bot	0,9237	0,9488	0,9356
WA Brute Force	0,7074	0,8951	0,7887
WA XSS	0,4160	0,1684	0,2215
Infiltration	1,0000	0,9714	0,9846
WA SQL Inj.	0,5800	0,3299	0,4133
Heartbleed	1,0000	1,0000	1,000
Doğruluk	0,9995		
Makro Ortalama	0,9061	0,8856	0,8874
Ağırlıklı Ortalama	0,9994	0,9995	0,9994
süre (dk)	53,9100		

yarlılık değeri ise %88,5 seviyesinde kalmıştır. Kesinlik, duyarlılık ve F_1 puanı dahil olmak üzere, bu deneydeki her sınıfın ayrıntılı ortalama sonuçları raw isimli senaryo olarak kaydedilmiştir ve Tablo 4.2 ile gösterildiği gibidir. Deneyler esnasında modelin ortalama 27-28. iterasyonda [3] en iyi sonuca yakınsadığı ve kalan 72-73 iterasyonda ise yakın seviyelerde kaldığı gözlenmiştir. Bununla birlikte modelin yaklaşık 60. iterasyonlarda artık öğrenmeyi geçip ezberlemeye başladığı da tespit edilmiştir. Daha net bir karşılaştırılabilirlik elde etmek için ise bu çalışmada tüm deneylerde 100 iterasyonda sabit kalmaya karar verilmiştir. 100 iterasyon boyunca elde edilen eğitim çıktıları Şekil 4.1 ile gösterildiği gibi kaydedilmiştir.

İlk deneyin sonuçlarını alındıktan sonra, bu yaklaşım öncelikle Tablo 4.3 ile gösterildiği gibi son dönemde yayınlanmış olan literatür çalışmalarıyla karşılaştırılmıştır. Çalışmalardan bazıları [63, 60] modellerinde ve deneylerinde sadece belirledikleri hedef sınıfları içermiş, bu nedenle yalnızca temel sınıfların karşılık gelen alt sonuçları

[3] Yakınsama iterasyon sayıları her dağılımda farklılık göstermiştir.

bu çalışmadaki sonuçlarla karşılaştırılmıştır. Bazı çalışmalar ise bu çalışmadaki gibi problem setindeki tüm sınıfların çözümünü hedef alan tam bir model üzerinde yoğunlaşmışlardır. Bahsedilen tüm çalışmalarla kıyaslama yapıldığında, bu çalışmanın raw isimli senaryosundaki modelin literatürde incelenen tüm çalışmalardan daha iyi performans gösterdiği değerlendirilmiştir.

Tablo 4.3: Litreratürde CICIDS2017 verisetini kullanan diğer bazı çalışmalar ile bu çalışmadaki raw senaryosunun sonuçlarının kıyaslaması.

Referans Model	**Ölçüm Tipi**	**Referans (%)**	**Bu çalışmanın Sonuçları (%)**
CNN+LSTM, DDoS [60]	Doğruluk	97,15	**99,98**
Bidirectional LSTM, tüm sınıflar [61]	Doğruluk	98,34	**99,95**
Bidirectional LSTM, tüm sınıflar [61]	F_1 puanı	98,38	**99,94**
DNN, tüm sınıflar [62]	Doğruluk	99,50	**99,95**
LSTM, SSH-Patator [63]	F_1 puanı	97,00	**99,88**
LSTM, FTP-Patator [63]	F_1 puanı	99,00	**99,92**
LSTM, WA BF [63]	F_1 puanı	19,00	**78,87**
LSTM, WA XSS [63]	F_1 puanı	3,00	**22,15**
LSTM, WA SQL Inj. [63]	F_1 puanı	0,00	**41,33**
PCCN, tüm sınıflar [67]	F_1 puanı	97,31	**99,94**
PCA+RF, tüm sınıflar [66]	F_1 puanı	99,70	**99,94**
SGM-CNN, tüm sınıflar [65]	F_1 puanı	99,86	**99,94**

Bu çalışmada önerilen model, verisetindeki sınıfların çoğu için önemli bir performansa sağlasa bile, *WA XSS*, *WA SQL Inj.* ve *WA Brute Force* sınıflarında diğer sınıflara kıyasla daha düşük bir başarım göstermiştir. Bu sınıflarda sırasıyla *WA XSS* %22,15, *WA SQL Inj.* %41,33 ve *WA Brute Force* %78,87 F_1 puanları elde edilmiştir. Bu sınıfların aynı zamanda en az beş örnekleme sahip sınıflar olduğu da gözlenmiştir (*Infiltration* ve *Heartbleed* ile birlikte). Bu veriler ışığında bu çalışmada, örneklem sayısı az olup da başarımı görece düşük olan sınıflara veri artırma teknikleri uygulayarak bu sınıfların başarımını yükseltme adına çeşitli teknikler önerilmiştir. Bu tekniklerle aşağıdaki deneyler düzenlenmiştir.

Örneklem sayısı az olan sınıfların başarımını artırmak için üç deney ve her deney için iki alt kırılım düzenlenmiştir. Deneylerin tamamında çeşitli veri artırma teknikleri kullanılmıştır. Her deneyin iki alt kırılımında uygulanan tekniğe iki farklı eşik değeri

verilmiş ve hangi sınıfların ne kadar veri artışına maruz bırakılacağını belirlenmiştir. Çıkan sonuçlar ise birbirleri ile kıyaslanmıştır. Bu deneyler 1) rastgele örneklem üretme (random sample generation), 2) SMOTEENN (Synthetic Minority Over-sampling Technique + Edited Nearest Neighbors) ve 3) CTGAN (Conditional Tabular Generative Adversarial Network) tekniklerini kullanmıştır.

Bu senaryolarda az sayıda örneklem içerdiği veya mevcut örneklemleri yetersiz parmak izi verdiği için F_1 puanı düşen sınıflara sentetik olarak üretilen yeni veriler eklenmesi hedeflenmiştir. Bu sentetik verilerin mevcut veriler göz önünde bulundurularak yakın olması planlanmıştır. Öncelikle verisetindeki hangi sınıflara veri artırma tekniği uygulanacağı konusu değerlendirilmiştir. Bu amaçla verisetindeki açık ara en çok örneklem içeren *Benign* sınıfı hariç tutulup, kalan diğer saldırı sınıfları üzerinden iki farklı eşik değeri hesaplanmış ve verisi artırılacak sınıflar bu eşik değerlerine göre seçilmiştir. Hangi sınıfların verilerinin artırılacağı eşik değeri belirlendikten sonra, belirlenen değerin altında örnekleme sahip sınıfların örneklemleri bu eşik değerine varıncaya kadar artırılmıştır. Bu eşik değerlerinin problemin sınıf ve örneklem sayılarından bağımsız olması için dinamik hesaplanmasına karar verilmiştir.

Veri artırma teknikleri için kullanılacak eşik değerlerinden ilkinde, sınıflar *Benign* sınıfı hariç tutularak örneklem sayılarına göre büyükten küçüğe sıralanmış ve eşik değeri bu sıralamada ortada kalan sınıfın örneklem sayısına eşitlenmiştir. Bu sayı ilgili veriseti için 4637 olarak belirlenmiştir. İkinci eşik değerinin hesaplaması için *Benign* sınıfı hariç tutularak kalan 14 sınıftaki örneklem sayıları toplanıp sınıf sayısı olan 14'e bölünmüş ve eşik değeri hesaplanmıştır. Bu sayı ilgili veriseti için 31803 olarak belirlenmiştir. Bu değerler hesaplanırken *Benign* sınıfının hesaplama dışı bırakılmasının sebebi, bu sınıfın verisetinin çok büyük bir kısmını oluşturması ve bu nedenle eşik değeri hesaplamasında performansı olumsuz yönde etkilememesidir. İlgili değerler Tablo 4.4 ile sınıf bazında detaylıca gösterilmiştir.

Eşik değerleri hesaplandıktan sonra, aşağıda anlatılan üç adet veri artırma tekniği iki eşik değeri için de işlenmiştir.

Tablo 4.4: Eğitim seti için sentetik veri artırma teknikleri uygulanacak sınıflar, eşik değerleri ve sınıflara ne kadar yeni sentetik örneklem ekleneceği.

Etkilenen Sınıflar	**Mevcut örneklem sayısı**	$esik_1$ (4637)	$esik_2$ (31803)
FTP-Patator	6348	-	25455
DoS GoldenEye	8234	-	23569
SSH-Patator	4717	-	27086
DoS slowloris	4637	-	27166
DoS Slowhttptest	4400	237	27403
Bot	1565	3072	30238
WA Brute Force	1205	3432	30598
WA XSS	522	4115	31281
Infiltration	29	4608	31774
WA SQL Inj.	16	4621	31787
Heartbleed	9	4628	31794

4.1.1 Rastgele Gürültü ile Veri Artırma

$esik_1$ varyasyonu (senaryosu) için yukarıdaki hesaplamada belirtildiği üzere 4637 değeri belirlenmiştir. Bu değer, hesaplama gereği sıralamada ortadaki sınıfın örneklem sayısına eşittir. Bu da eşik değerinin *DoS Slowloris* sınıfına karşılık gelmektedir. Bu aşamadan sonra, örneklem sayısı 4637'den daha az olan sınıfların rastgele gürültü ile veri artırılması sürecine başlanmıştır ve bu sınıfların örneklem sayısı *DoS Slowloris* sınıfının örneklem sayısına eşitlenene kadar veri artırma tekniği iterasyonlar boyunca veriye uygulanmıştır. $esik_1$ senaryosunda rastgele gürültü ile veri artırma tekniği kullanılarak tüm sınıfların test setinde %91,56 makro ortalamalı bir F_1 puanı elde edilmiştir. Ortalamaya bakıldığında, raw modeline kıyasla yaklaşık %3,15'lik bir F_1 puanı artışının meydana geldiği görülmektedir. Verisi artırılan sınıflarda ise *WA XSS* %92,3 artış göstermiş, *WA SQL Inj.* %80,1 artış göstermiş ve *WA Brute Force* sınıfı %11,1 oranında düşüş göstermiştir. Sentetik veri artırma tekniği uygulanan sınıflar dışında, sistemin genel performansında bir değişiklik olduğu için haliyle veri artırma tekniği uygulanmamış sınıfların da başarım puanlarında toplam puanı %100'de dengeleyecek şekilde farklı irili ufaklı artış ve azalışların kaydedildiği de görülmüştür. Bunlar içinden *Infiltration* sınıfının F_1 puanının %1,52 azalması örnek olarak gösterilebilir. Bu farklılıkların sebebinin ise

veriye gürültü eklemenin ve bu gürültünün kalitesinin, veriye otomatikman eklenen bias değerini değiştirmesi olduğu değerlendirilmektedir. Bu deneyin çıktıları, rnd_{t1} başlığı altında Tablo 4.5 ve Tablo 4.6 üzerinde detaylandırılmıştır.

$esik_1$ değeri için deneyler yapıldıktan sonra, $esik_2$ için de benzer hiperparametrelerle deneylere devam edilmiştir. Bu deneyde eşik değeri belirlenen sınıfların örneklem sayıları ortalaması olan 31803'e eşitlenerek bu sayıdan daha az örneklem içeren sınıfların örneklem sayıları ilgili veri artırma tekniği kullanılarak bu sayıya eşitlenene kadar sentetik olarak çoğaltılmıştır. Bu aşamada $esik_2$ senaryosundan elde edilen çıktılar $esik_1$ ile kıyaslanmıştır.

$esik_2$ varyasyonundan elde edilen sonuçlara göre, *WA Brute Force* saldırısının F_1 puanı %1,57 oranında artmış, ancak *WA XSS* saldırısının puanı %8,47 oranında düşmüştür. Ayrıca *WA SQL Inj.* saldırı tipinde de %10,61 oranında düşüş gözlenmiştir. Tüm sınıflar için toplam makro ortalama F_1 puanı %90,6 olarak hesaplanmıştır. Bu sonuçlar herhangi bir veri artırma tekniği içermeden eğitim setinin direkt olarak modele verildiği raw senaryosuna kıyasla daha iyi bir ortalamaya sahip olmasına rağmen, $esik_1$ varyasyonu ile kıyaslandığında daha düşük kalmıştır. Buradaki önemli unsurlardan bir tanesi, her ne kadar katsayısı her iterasyonda yarıya indirilse de, rastgele veri üreteci fonksiyonunun bu eşik değerinde öncekine görece çok daha fazla iterasyon yapması gerektiği ve bu nedenle de yeni üretilen rastgele sentetik verilerin öncekilere göre problem uzayında daha çok iç içe girmesinden ötürü olduğu değerlendirilmektedir. Haliyle daha önce bahsedilen bias değeri bu senaryoda çok daha fazla etkisini göstermiş ve yeni üretilen sınıf örneklemlerinin diğer sınıflara karışma ihtimali oldukça artmıştır. Bu sonuç elbette eşik değerlerinin etkisinin belli bir seviyeye ve belirli sınıflara etkisinin olumlu olduğu halde belli koşullarda ise olumsuz etki yapabileceğini göstermektedir. Bu eşik değerinin kullanıldığı senaryodaki detaylı sonuçlar, rnd_{t2} adı altında Tablo 4.5 ve Tablo 4.6 ile gösterilmiştir.

İkinci eşik değerinin sonuçlarının öncekinden daha düşük olmasına rağmen raw modelden daha üstün çıkmasından ötürü, veri artırma tekniklerine bir başka senaryo üzerinden

devam edilmiştir. Sonraki senaryolar ile sırasıyla SMOTEENN ve CTGAN teknikleri olacaktır.

4.1.2 SMOTEENN ile Veri Artırma

Bu yöntem, $esik_1$ varyasyonu ile beş adet dağılımın (fold) çalıştırılmasının ardından, ortalamalar alındığında %91,40 F_1 puanı vermiştir. Bu sonucun raw modele kıyasla daha iyi olduğu, ancak rnd_{t1} senaryosuna kıyasla biraz geride kaldığı gözlenmiştir. Önceden başarımı düşük olan sınıflara bakıldığında ise, *WA Brute Force* ve *Infiltration* sınıflarının başarımının rnd_{t1}'e göre arttığı; ancak *WA XSS* ve *WA SQL Inj.* sınıflarında başarımın azaldığı görülmüştür. Önceki senaryolarda olduğu gibi bu sınıflarda yaşanan puan değişimlerine istinaden diğer sınıflarda da ufak farklılaşmalar gözlenmiştir. Bu deneyin detaylı sonuçları smt_{t1} adı altında Tablo 4.5 ve Tablo 4.6 ile gösterilmiştir.

SMOTEENN veri artırma tekniğinin 2. eşik değeri deneyi, %90,80'lik bir F_1 puanı üretmiştir. Bir önceki deneyde olduğu gibi bu çıktı da raw modeline kıyasla iyi ancak rastgele veri üretme içeren rnd_{t1} deneyine kıyasla aşağıda kalmıştır. Yine de rnd_{t2}'ye kıyasla %0,22 daha iyi makro ortalama tutturabilmiştir. Bundaki etki ise *WA SQL Inj.* ve *Infiltration* sınıflarının F_1 puanlarının iyileşmesinden kaynaklanmaktadır. Tüm sınıfların çıktıları bakıldığında, aslında rnd_{t2} deneyine kıyasla ufak gürültülü sonuç farklılıkları haricinde muazzam bir fark olmadığı gözlenmektedir. Bu deneyin detaylı çıktıları smt_{t2} adı altında Tablo 4.5 ve Tablo 4.6 ile gösterilmiştir.

4.1.3 CTGAN ile Veri Artırma

$esik_1$ değeri için F_1 puanının makro ortalaması %90,41 olarak ölçülmüştür. Bu değer raw modelinden daha iyi bir çıktı vermiştir ancak önceki veri artırma teknikleri ile kıyaslandığında daha az bir başarım elde edildiği görülmüştür. $esik_2$ değeri için de F_1 puanının makro ortalaması %90,48 olarak ölçülmüştür [4]. Bu deneydeki makro or-

[4]Tüm deneylere ait ilgili kodlar ve dokümantasyona bu adresten açık kaynak olarak erişilebilir: *https://github.com/ucekmez/phd*

Tablo 4.5: Önerilen raw model ve diğer veri artırma senaryolarının duyarlılık değerlerinin birbirleriyle kıyaslanması. Kalın değerler ilgili sınıf için elde edilen en iyi çıktıları ifade etmektedir.

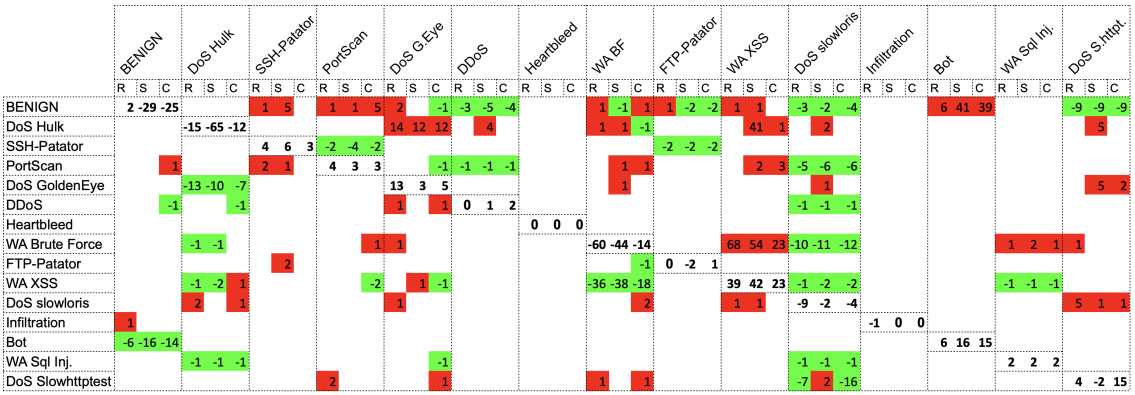
Sınıf	raw	rnd_{t1}	rnd_{t2}	smt_{t1}	smt_{t2}	$ctgan_{t1}$	$ctgan_{t2}$
Benign	**0,9999**	0,9998	0,9998	0,9998	0,9998	0,9998	0,9998
DoS Hulk	**0,9998**	0,9997	0,9992	0,9993	0,9995	0,9997	0,9993
PortScan	**0,9994**	0,9994	0,9992	0,9993	0,9992	0,9991	0,9994
DDoS	0,9998	0,9998	0,9996	**0,9998**	0,9995	0,9997	**0,9998**
DoS GoldenEye	0,9928	0,9916	**0,9948**	0,9880	0,9926	0,9940	0,9938
FTP-Patator	**0,9998**	**0,9998**	0,9997	0,9996	0,9996	**0,9998**	**0,9998**
SSH-Patator	0,9986	0,9991	0,9993	**0,9998**	0,9984	0,9996	0,9996
DoS slowloris	**0,9917**	0,9860	0,9875	0,9898	0,9887	0,9775	0,9889
DoS Slowhttptst	0,9889	0,9894	0,9896	0,9830	0,9856	**0,9901**	0,9849
Bot	0,9488	0,9657	0,9683	0,9673	0,9690	**0,9693**	0,9555
WA Brute Force	**0,8951**	0,6708	0,6987	0,7311	0,6947	0,8348	0,8560
WA XSS	0,1684	**0,4843**	0,4111	0,4246	0,4125	0,2484	0,2134
Infiltration	0,9714	0,9428	0,9428	0,9714	**1,0000**	**1,0000**	**1,0000**
WA SQL Inj.	0,3299	0,8099	0,6599	0,7600	0,7600	**0,9099**	0,7700
Heartbleed	**1,0000**	**1,0000**	**1,0000**	**1,0000**	**1,0000**	**1,0000**	**1,0000**
Doğruluk	**0,9995**	0,9994	0,9993	0,9993	0,9993	0,9994	0,9994
Makro Ort.	0,8856	0,9225	0,9100	0,9208	0,9199	**0,9281**	0,9173
Ağırlıklı Ort.	**0,9995**	0,9994	0,9993	0,9993	0,9993	0,9994	0,9994
süre (dk.)	**53,91**	58,24	66,28	65,01	754,57	549,19	2156,99

talamaların ikisi de önceki veri artırma tekniklerine kıyasla düşük kalsa da, *Infiltration* sınıfının CTGAN kullanılarak hazırlanan deneyde her iki eşik değerinde de tüm dağılımlarda %100 F_1 puanı elde edildiği gözlenmiştir.

Yürütülen deneyler neticesinde CTGAN tekniğinin bazı sınıflar için diğer tekniklerden daha düşük performans göstermesinin nedenlerine bakıldığında, ana nedenlerde biri olarak sınıflardaki örneklemlerin dağılım kalitesi ön plana çıkmaktadır. GAN yapısının örneklemleri iyi anlamasındaki en temel unsurun hem örneklem sayısı hem de ilgili sınıfların en vurucu özniteliklerini iyi öğrenmesi gibi kriterler olduğu bilinmektedir. GAN, esasen gücünü örneklemlerin kalitesinden alan bir üretici modeldir. Bu nedenle bu çalışma kapsamında yürütülen CTGAN deneyinde örneklemlerin sayısının ve problem uzayındaki dağılımının etkilerinin CTGAN deneyinin sonuçlarına etki ettiği değerlendirilmektedir. Karşıt bir sonuç olarak ise, Infiltration sınıfının örneklem kalitesinin iyi olmasından ötürü bu sınıfla ilgili iyi veriler üretildiği ve bunun da model

Tablo 4.6: Önerilen raw model ve diğer veri artırma senaryolarının F_1 değerlerinin birbirleriyle kıyaslanması. Kalın değerler ilgili sınıf için elde edilen en iyi çıktıları ifade etmektedir.

Sınıf	raw	rnd_{t1}	rnd_{t2}	smt_{t1}	smt_{t2}	$ctgan_{t1}$	$ctgan_{t2}$
Benign	**0,9999**	0,9999	0,9999	0,9999	0,9999	0,9999	0,9999
DoS Hulk	0,9995	**0,9995**	0,9993	0,9994	0,9994	0,9995	0,9993
PortScan	0,9996	0,9996	0,9994	0,9996	0,9994	0,9995	**0,9996**
DDoS	0,9998	**0,9998**	0,9997	0,9995	0,9996	**0,9998**	0,9995
DoS GoldenEye	0,9929	0,9918	0,9900	0,9893	0,9901	**0,9932**	0,9924
FTP-Patator	0,9992	0,9994	**0,9995**	0,9994	0,9988	0,9993	0,9994
SSH-Patator	0,9988	**0,9989**	0,9985	0,9975	0,9968	0,9945	**0,9989**
DoS slowloris	**0,9884**	0,9883	0,9866	0,9877	0,9857	0,9861	0,9884
DoS Slowhttptst	**0,9894**	0,9883	0,9868	0,9859	0,9865	0,9853	0,9883
Bot	**0,9356**	0,9293	0,9201	0,9196	0,9164	0,9233	0,9232
WA Brute Force	**0,7887**	0,7003	0,7115	0,7319	0,7071	0,7631	0,7713
WA XSS	0,2215	**0,4257**	0,3896	0,3971	0,3830	0,2881	0,2342
Infiltration	0,9846	0,9692	0,9560	0,9846	0,9632	**1,0000**	**1,0000**
WA SQL Inj.	0,4133	**0,7444**	0,6657	0,7326	0,6960	0,6306	0,6777
Heartbleed	**1,0000**	**1,0000**	**1,0000**	**1,0000**	**1,0000**	**1,0000**	**1,0000**
Doğruluk	**0,9995**	0,9994	0,9993	0,9993	0,9993	0,9994	0,9994
Macro Ort.	0,8874	**0,9156**	0,9068	0,9149	0,9081	0,9041	0,9048
Weighted Ort.	**0,9994**	0,9994	0,9993	0,9994	0,9993	0,9994	0,9994
süre (dk.)	**53,91**	58,24	66,28	65,01	754,57	549,19	2156,99



Şekil 4.2: Sentetik veri artırma tekniklerinin çıktılarının $t1$ eşik değeri için raw_{t1} senaryosu çıktıları ile sınıf bazındaki farklılıkları

başarımını %100'e çıkardığı gözlenmiştir. CTGAN deneyinde sonuçların kalitesine etki edebilecek bir diğer unsur ise modelin eğitilmesi esnasında belirlenen hiperparametrelerdir. Bunlar özellikle batch sayısı ve eğitim esnasındaki iterasyon sayısıdır. Bu tekniğin hiperparametrelerinin optimize edilmesi konusu bu çalışmanın bir parçası olmadığı için bu konuda sabit değerler kullanılmıştır. Uygun koşullar ve doğru parametre optimizasyonları ile sonuçların kalitesinin yükseltilebileceği değerlendirilmektedir. Problem bazında manuel veya dinamik olarak müdahale edilmesi gereken bu detaylı gereksinimlere rağmen CTGAN tekniğinin statik bir yapısı olan diğer iki yönteme kıyasla makine öğrenmesi yaklaşımını ele almasından ve sonuçların yine diğer iki yöntem ile yarışabilir konumda olmasından ötürü, CTGAN'ın gerçekçi tablo verileri üretme konusunda üzerinde çalışılmaya değer olduğu gözlenmiştir. CTGAN deneyinin çıktıları 1. eşik değeri için $ctgan_{t1}$ ve 2. eşik değeri için $ctgan_{t2}$ başlıkları altında kaydedilmiş ve sırasıyla tüm deneylerin sonuçları gibi Tablo 4.5 ve Tablo 4.6 ile detaylandırılmıştır.

Şekil 4.2, sentetik veri üretme senaryolarının $t1$ eşik değeri için raw_{t1} modelinin çıktıları ile sınıf bazındaki farklılıklarını göstermektedir. Yeşil arkaplanlı negatif değerler, ilgili sentetik veri senaryosundan elde edilen çıktının sınıflar bazında yapılan hatalı tespitleri ne kadar azaltığını gösterirken, kırmızı arkaplanlı pozitif değerler ise raw_{t1} senaryosunun çıktısına göre ekstra hatalı tespitler olduğunu göstermektedir. Arkaplanı beyaz değerler ise ilgili sınıf için toplam değişimi ifade etmektedir. Bu tablodan yapılabilecek bir çıkarım, aynı eşik değerindeki tüm sentetik veri senaryolarının

etki oranı farklılık göstermesine karşın çoğunlukla aynı sınıflarda olumlu veya olumsuz etki gösterdiğidir. Tekil sınıf bazında bakılacak olursa, sentetik veri senaryolarında $BENIGN$, $DoSHulk$, $WebAttackBruteForce$ ve $DoSSlowloris$ saldırı tiplerinin tespit başarımının raw modeline göre düştüğü, diğer sınıfların başarımının ise artmaya meyilli olduğu gözlemlenmiştir. $BENIGN$ sınfının sentetik veri senaryolarında en çok Bot sınıfı ile karıştırıldığı gözlenmiştir. $DoSHulk$ sınıfının tüm sentetik veri senaryolarında $DosGoldenEye$ ve özellikle SMOTEENN senaryosunda $WebAttackXSS$ ile karıştırıldığı gözlenmiştir. $WebAttackBruteForce$ sınıfının ise tüm sentetik veri senaryolarında en çok $WebAttackXSS$ sınıfı ile karıştırıldığı gözlenmiştir. Buna karşın Bot sınıfı tespitlerinin raw modelinin çıktılarına göre tüm sentetik veri senaryolarında $BENIGN$ sınıfından kendine doğru iyileştirme olduğu da gözlenmiştir. Tablo detaylı incelendiğinde problem uzayında öznitelikleri daha çok dağınık halde olan saldırı tiplerinin duyarlılık kaybettiği gözlenirken, daha dar bir alana yayılmış saldırı tiplerinin ise duyarlılık kazandığı değerlendirilmiştir.

4.2 Diğer Makine Öğrenmesi Teknikleri ile Kıyaslama

Çalışma kapsamındaki çıktılar ayrıca referans olması adına başka makine öğrenmesi algoritmaları ile de kıyaslanmıştır. raw senaryosunda kullanılan ham veriseti; Rassal Orman, Karar Ağacı, Naive Bayes ve en yakın K-Komşu (K-NN) gibi artık algoritmaları standartlaşmış makine öğrenmesi teknikleri üzerinde test edilmiş ve hem çıktı kalitesi hem de eğitim süreleri kaydedilmiştir. Algoritmalar, python sklearn kütüphanesinde yer alan varsayılan parametreler kullanılarak eğitilmiştir. İlgili kıyaslama Tablo 4.7 üzerinde detaylandırılmıştır. Naive Bayes algoritmasının tüm yöntemler içinde yaklaşık 0,5 dakikalık eğitim süresi ile en hızlı öğrenmeyi sağladığı gözlenirken, aynı zamanda kullanılan yöntemler içinde en düşük başarıma sahip olduğu tespit edilmiştir. Karar ağacı algoritması yaklaşık 1 dakikalık zaman diliminde eğitilmiştir ve Naive Bayes'e göre oldukça iyi bir çıktı üretmiştir. K-NN ise diğer makine öğrenmesi algoritmaları içinde yaklaşık 515 dakika ile en uzun süren algoritma olmuştur. Başarım değerleri ise karar

Tablo 4.7: Önerilen senaryolardaki algoritmalar ve geleneksel makine öğrenmesi tekniklerinin çıktılarının karşılaştırılması. **: MLP algoritması sekiz CPU çekirdeği üzerinde paralel çalıştırılmıştır.*

Algoritma	F_1	Duyarlılık	Süre (dk)	Donanım
raw	0,9994	0,9995	53,91	GPU
rnd_{t1}	0,9994	0,9994	58,24	GPU
rnd_{t2}	0,9993	0,9993	66,28	GPU
smt_{t1}	0,9994	0,9993	65,01	GPU
smt_{t2}	0,9993	0,9993	754,57	GPU
$ctgan_{t1}$	0,9994	0,9994	549,19	GPU
$ctgan_{t2}$	0,9994	0,9994	2156,99	GPU
Rassal Orman	0,9532	0,9287	5,41	CPU
Karar Ağacı	0,9620	0,9543	1,04	CPU
Naive Bayes	0,5410	0,8220	0,5	CPU
K-NN $(K = 3)$	0,6623	0,8490	515,75	CPU
MLP	0,9210	0,9311	51,2	CPU*

ağacı ve Naive Bayes algoritmalarının arasında yer almıştır. Geleneksel çok katmanlı yapay sinir ağı modeli (MLP) de kıyaslama yapılan diğer modellere eklenmiştir. Bu ağ, aynı önerilen model gibi dört katmandan oluşmaktadır ve katmanları sırasıyla 128, 64, 32 ve 15 adet nöron içermektedir. 100 iterasyon çalışan bu model yapısı gereği üstünde çalıştığı donanımdaki CPU çekirdeklerinin tamamını (8 adet) kullanmış ve toplam 51,2 dakika süresince çalışmıştır. Bu çalışma kapsamında önerilen yöntem GPU donanımını kullanırken, diğer makine öğrenmesi yöntemleri, MLP hariç, CPU donanımı üzerinde tek çekirdeği kullanacak şekilde çalıştırılmıştır. Bu nedenle süre kıyaslamaları önerilen yöntem ile değil, sadece kendi aralarında değerlendirilebilmektedir. Diğer bir yandan bu algoritmaların hazırlanma süreleri ve uygulanabilirliği kullanılan kütüphanenin kolaylığı ile doğru orantılı olmuştur ve standart parametreler değiştirilmemiştir. Önerilen yöntemin arkasında ise başlı başına özellikle tasarlanmış bir mimari yer almaktadır. Çıktılar kıyaslanırken buna dikkat edilmesi de elzemdir.

Bu çıktılar, çok boyutlu ve çok sınıflı verilerde önerilen Derin Öğrenme yönteminin geleneksel diğer makine öğrenmesi yöntemlerine kıyasla da daha yüksek başarımlar elde ettiğini göstermektedir.

4.3 Çıktıların Yorumlanması

Bu çalışma kapsamında yapılan deneyler temelde veri artırma tekniklerinin örneklem sayısı az olan sınıflarda nasıl bir etki gösterdiğini test etmek ve bu tekniklerin, kullanılan modellerde başarımı ne kadar artırdığını gözlemlemek amacıyla gerçekleştirilmiştir. Bu amaçla farklı eşik değerleri ile deneyler çeşitlendirilmiş olup, problem setindeki birçok sınıfın bu veri artırma tekniklerinden olumlu yönde etkilendiği, en çok da örneklem sayısı az olan sınıfların bu çalışmalarda yüksek başarımla tespit edildiği görülmüştür. Çalışmalar kapsamında örneklem sayısı zaten yüksek olan bazı sınıfların ise model üzerinde tespit performansının bir miktar olumsuz etkilendiği de gözlenmiştir. Bunun nedeninin ise örneklem sayısı az olan sınıfların veri artırma tekniği uygulandıktan sonra oluşan yeni sentetik örneklemlerin bir kısmının problem uzayında diğer sınıflarla üst üste gelmesinden ötürü olduğu düşünülmektedir. Yine de bu gözlem çalışma kapsamı dışında olduğu için derinlemesine araştırılmamıştır.

Veri artırma tekniklerinin, gözlemlenen sonuçlar itibariyle farklı eşik değerlerinde farklı çıktılar verdiği ve sınıflara uygulansın veya uygulanmasın, problem setindeki birçok sınıfın güncel tespitinde çeşitli etkileri olduğu görülmüştür. Bu sonuçlar, veri artırma tekniklerinin dengesiz sınıf dağılımları olan verisetlerinde makro sonuçları epey iyileştirdiğini ve makine öğrenmesi çalışmalarının kalitesini artırdığını ortaya çıkarmıştır; ancak veri artırma konusu, doğası gereği kendine has başka hiperparametreleri karşımıza çıkarmıştır. Bunlardan birisi verisetinde hangi sınıfların veri artırma tekniğine gireceği ve diğeri ise tespit edilen sınıfların verilerinin neye göre ve ne kadar artırılacağıdır. Bu çalışmada bu hiperparametrelerin gerçekten var olup olmadığının belirlenmesi için veri artırma tekniği uygulanacak sınıfların seçilmesi ve bunların ne kadar artırılacağı konusu için iki adet eşik değeri ve sınıf bazında veri artırma sayısı dinamik olarak problemdeki sınıf dağılımlarına göre belirlenmeye çalışılmıştır. İki farklı eşik değeri seçilmiştir. $esik_1$ yedi adet sınıfı, $esik_2$ ise 11 adet sınıfı etkilemiştir. Çoğu durumda $esik_1$ değerinin diğerinden daha iyi performans gösterdiği gözlenmiştir. Bunun nedeninin ise $esik_2$

değerinin daha fazla sınıfı etkilemesi, bunlarda veri artırımı yapılması ve bu üretilen fazla verinin dağılımının yaygın olmasından ötürü üretilen sentetik verilerin diğer sınıflarla çakışması olduğu değerlendirilmektedir.

Tüm veri artırma deneyleri ve eşik varyasyonları dikkate alındığında, *WA Brute Force* sınıfının tüm deneylerde raw modelinin herhangi bir veri artırma tekniği kullanılmayan ham haline kıyasla daha az doğru tespit edilebildiği gözlenmiştir. Buna karşın *WA XSS* ve *WA SQL Inj.* gibi sınıfların ise veri artırma tekniğine maruz bırakıldığında F_1 puanı bazında sınıf başarımlarının %80,11 ile %92,1 oranlarında arttığı gözlenmiştir.

Tüm sınıflar bazında ağırlıklı F_1 puanına bakıldığında, raw modelin başarımının veri artırma tekniklerine kıyasla önde olduğu, ancak makro ortalamalar göz önünde bulundurulduğunda ise F_1 puanları olarak tüm veri artırma tekniklerinin model başarımını iyileştirdiği ortaya çıkmıştır. Bunlar içerisinde ise en iyi sonucun rnd_1 tekniği ile %91,56 F_1 puanı ve %92,25 duyarlılık puanı üzerinden elde edildiği gözlenmiştir.

Deneylerde dikkatimizi çeken bir diğer konu ise rastgele veri üreteci fonksiyonunun çoğu sınıfta SMOTEENN'e göre daha iyi sonuçlar vermesidir. Rastgele veri üreteci, temel olarak örneklemlerin birbirlerine olan mesafelerine bakmaksızın eğitim setindeki her örneğin öznitelik değerlerine belirli oranda gürültü ekleyerek yeni sentetik örneklemler oluşturan ve her iterasyonda önceki iterasyonda üretilen sentetik verileri de kendine girdi olarak kullanan bir sentetik veri üreteci olarak geliştirilmiştir. Örneklemlerin lineer deformasyona uğratılarak sentetikleştirildiği SMOTEENN'in aksine, rastgele veri üreteci fonksiyonu sentetik verileri başka iki gerçek verinin ortasında yer alacak şekilde üretmeyi garanti edemez, değerlerini rastgele seçer ve uzayda belirlenen gürültü kadar herhangi bir noktaya yeni sentetik örneklemi yerleştirebilir. Problem uzayında sentetik örneklemlerin çok dağılmaması ve yüksek bias değeri oluşturmaması için de her yeni iterasyonda gürültü miktarını azaltır ve belli bir küme grubunun en fazla dış sınırları kadar bir sentetik örneklem grubu oluşturmaya devam eder. Yapılan testlerde SMOTEENN'in kalitesinin de yine mevcut gerçek örneklemlerin dağılımı kadar kaliteli sonuçlar verdiği gözlenirken, rastgele veri üreteci fonksiyonunun örneklerin kalitesinden bağımsız olarak rastgele

çıktılar verdiği ve bu nedenle kalitesiz dağılıma sahip örneklemlerde ve örneklem sayısı çok az olan sınıflarda SMOTEENN'e göre daha iyi sonuçlar ürettiği gözlemlenmiştir.

Veri artırma tekniklerinin model başarımını artırmasının yanısıra, aynı zamanda bu teknikler uygulandığında modellerin çalışma sürelerine ne kadar etki ettikleri konusu da bir fizibilite faktörü olarak düşünülmelidir. Tablo 4.6 ile gösterildiği üzere, raw modelini eğitmek için gereken süre yaklaşık 54 dakikadır. Bu modele rnd_{t1} senaryosu eklendiğinde toplam süreye etkisi fazladan dört dakika, smt_{t1} senaryosu ile fazladan 12 dakika ve $ctgan_{t1}$ senaryosu ile de raw modelin eğitim süresine fazladan 495 dakika eklenmiştir. Bu zamanlar dikkate alındığında, CTGAN senaryolarının diğer deneylere kıyasla çok fazla vakit aldığı ve buna karşın diğer senaryolardan daha iyi çıktılar veremediği gözlenmiştir. Bundaki en önemli etken, CTGAN tekniğinin de veri üretmeden önce ilgili örneklemler ile bir eğitim süreci olmasıdır. Tüm çıktıları değerlendirdiğimizde yine de üretken bir çekişmeli ağın sayısal veriler üzerinde örneklemler üretmesinin gürültü tabanlı ve lineer deformasyon modellerine kıyasla nasıl bir konumda bulunduğunu göstermek adına kıymetli bir örnek olduğu değerlendirilmektedir.

5 SONUÇ

IoT teknolojisi ve uygulamaları, içerdiği tüm alt dalların karmaşıklığı ve potansiyel güvenlik zafiyetleri nedeniyle siber güvenlik konseptinin de ilgisini epey çeken bir araştırma alanıdır. Bu teknolojiler birçok gerçek dünya işini basitleştirse de, beraberinde birçok güvenlik açığını da ortaya çıkarmaktadır. Bu çalışma kapsamında üzerinden geçilen ve detaylandırılan tüm güvenlik açıklarının ve potansiyel tehlikelerin, çeşitli araştırmacılar tarafından makine öğrenmesi ve Derin Öğrenme yöntemleri ile üstesinden gelinmeye çalışıldığı ve bu çalışmaların belli seviyelere kadar başarılı olduğu gözlemlenmektedir. Bu nedenle de bahsi geçen yapay zeka yaklaşımlarının neredeyse birçok IoT ağını korumada ilk seçenek olarak tercih edilmeye başlandığı bilinmektedir; ancak IoT kullanım alanları ve karmaşıklığı arttıkça, gelenekselleşen yaklaşımların güncelliği tehlikeye girmekte ve daha yeni algoritmaların geliştirilmesi gerekliliği doğmaktadır. Özellikle karmaşıklığı yüksek ve dengesiz dağılıma sahip IoT verilerinde problemin yüksek başarımlarla çözülmesi ihtimali de azalmaktadır. Bu da sistemleri tehlikeye atmaktadır. Bu tarz yaklaşımlarda kaliteli çıktılar elde edebilmek için yapay zeka modellerinin de doğru verilerle eğitilmesi ve özellikle online/aktif olarak kendilerini güncellemeleri elzemdir.

Her problem için gerçek bir ortamdan veri alınamadığı durumlarda araştırmacılar, ilgili problemlere yönelik daha önceden oluşturulan ve olabildiğince geniş bir spektruma veya spesifik bir alana hitap edecek doğru verisetlerini bulmak durumundadırlar. Bu çalışmada da hem IoT alanında yer alan saldırı tiplerine sahip olan, hem karmaşıklığı ve dengesizliği yüksek olan hem de literatürdeki farklı çalışmaların da kullandığı ve bu sayede kıyaslanabilirliği yüksek olan CICIDS2017 veriseti üzerinde çalışılmıştır. Bu veriseti ayrıca probleme uygun bir şekilde önişlemlere ve veri artırma tekniklerine tabi tutulup farklı senaryolar ve deneyler yapılmıştır. Bu kapsamda CNN destekli, kendi kendini normalleştirebilen bir sinir ağı sınıflandırıcı olarak önerilmiş ve ilgili veriseti kul-

lanılarak gerçek bir ortam simüle edilmiştir. Bu sayede verisetinde belirlenmiş saldırı tiplerinin tespiti yapılmıştır. Bu çalışmanın deneyleri sonucunda, çıktıların %99,94 ağırlıklı F_1 puanı ile incelenen literatürdeki tüm diğer çalışmalardan daha iyi olduğu belirlenmiştir. Yine de verisetindeki az örnekleme sahip bazı sınıfların F_1 puanlarının örneklem sayısı daha çok olan diğer sınıflara göre az olduğu gözlenmiştir. Düşük örnekleme sahip sınıflara sentetik veriler eklenmesinin ise hem dengesiz veriseti problemini çözdüğü hem de iyileştirilmesi hedeflenen sınıfların tespit başarımını artırdığı gözlenmiştir. Bu amaçla gerçekleştirilen üç adet farklı veri artırma tekniğinin problem üzerindeki başarımları incelenmiştir. Veri artırma sürecinde hangi sınıfların üzerinde yapılacağı ise dinamik olarak belirlenmiştir ve bu anlamda hazırlanan iki farklı basit eşik değeri kıyaslanmıştır. Deneyler, veri artırma tekniklerinin hedeflenen sınıfların çoğu için daha iyi bir bölgeye yakınsama sağladığını ve ayrıca problem uzayının örnek dağılımını genişlettiklerini ve bunun sonucunda yanlış sınıflarla örtüşen bazı örneklemler oluşturduğunu da göstermiştir. Buna rağmen, sonuç itibariyle tüm sınıflar göz önünde bulundurulduğunda dengesiz bir verisetine uygulanan belli veri artırma tekniklerinin, bu tekniklerin uyarlanmadığı senaryolara göre çok daha iyi başarım gösterdiği tespit edilmiştir. Yine de bu yaklaşımın CSE-CIC-IDS2018 veya gelecekte üretilecek olan yeni dengesiz verisetlerinde de doğrulanması gerekliliği hala geçerlidir.

Bu çalışma, yukarıda anlatılanlara ek olarak aynı zamanda bir dizi hiperparametre oluşturma tekniği olarak da değerlendirilmektedir. Bu çalışmada belirli bir problemi farklı yapay zeka modelleri geliştirerek ve bunları birbirleri ile kıyaslayarak çözmek yerine, veriyi farklı şekillere dönüştürerek aynı modelin beslenmesi sağlanmış ve problem üzerinde başarımı artırmaya yönelik veri merkezli bir yaklaşım ele alınmıştır. Bunun paralelinde de hem veri artırma teknikleri olarak, hem verisetinde hangi sınıfların verilerinin artırılması gerekliliğinin kararı olarak hem de farklı eşik değerleri belirleme açısından, veri merkezli yaklaşımlarda nasıl bir yeni tip hiperparametre havuzu oluştuğu bu çalışmada ortaya çıkmıştır.

Literatürde, geliştirilen Derin Öğrenme modellerindeki CNN katmanlarının sayısını, her

bir katmandaki nöron/filtre sayısını, öğrenme katsayısını ve eğer kullanılıyorsa veri düşürme tekniğinin oranını optimize etmek için çok çeşitli yöntemler ve yaklaşımlar bulunmaktadır. Bu amaçla AutoML (model başarımını artırmak için gerekli hiperparametreleri otomatik olarak bulmaya çalışan) uygulamaları veya optimizasyon algoritmaları (Genetik Algoritma, Karınca Kolonisi Optimizasyonu, Parçacık Sürü Optimizasyonu vb. gibi) kullanılabilmektedir. Bu çalışmada tanımlanan hiperparametre optimizasyonlarıyla birlikte çok daha iyileşme ihtimalinin olduğu da göz önünde bulundurulmalıdır; ancak bu optimizasyonlar bu tez çalışmasının kapsamı dışındadır ve yalnızca veri merkezli bir iyileştirme üzerine odaklanılmıştır. Bu kapsamda ilgili hiperparametreler üzerinde çalışılması için literatüre açık bir araştırma bırakılmıştır. Hem model hem de veri merkezli hibrit bir yaklaşımın dengesiz veri problemi olan karmaşık ve çok sınıflı verisetlerinde daha iyi başarımlar elde edebileceği ihtimalinin yüksek olduğu değerlendirilmektedir.

Tüm bu çıktılara ek olarak, bu çalışmanın yapıldığı dönemde başlayan ve hala süregelmekte olan pandemi ve buna istinaden ortaya çıkan yeni normalleşme alışkanlıklarıyla beraber IoT uygulamalarının sağlık alanında ve toplumsal süreçlerde çok daha geniş kapsamda kullanılmaya başlanacağı öngörülmektedir. Yeni temassız sistemler, sağlık verilerinin toplanması ve kontrolleri, sosyal mesafe korunumu ve diğer ilgili yeni ortaya çıkan ihtiyaçlarda 5G, blok-zincir mimarisi, robotik altyapılar, ve IoT sistemlerinin rolü çok daha önemli bir konuma gelmiştir. Bu nedenle IoT altyapılarını yakın gelecekte büyük potansiyel güvenlik zafiyetlerinin beklediği değerlendirilmektedir [106]. Bu alanda yapılacak gelecek çalışmaların ise önemini artırarak koruduğu bir gerçektir.

KAYNAKLAR

[1] Deloitte,, "How the pandemic stress-tested the increasingly crowded digital home," https://www2.deloitte.com/content/dam/insights/articles/6978_TMT-Connectivity-and-mobile-trends/DI_TMT-Connectivity-and-mobile-trends.pdf, 2021, accessed: 2022-09-01.

[2] "Prognosis of worldwide spending on the internet of things (iot) from 2018 to 2023," https://www.statista.com/statistics/668996/worldwide-expenditures-for-the-internet-of-things/, accessed: 2022-09-01.

[3] "Global iot spending to grow 24% in 2021, led by investments in iot software and iot security," https://iot-analytics.com/2021-global-iot-spending-grow-24-percent, accessed: 2022-09-01.

[4] Balaji, S., Nathani, K., and Santhakumar, R., "Iot technology, applications and challenges: a contemporary survey," *Wireless personal communications*, vol. 108, no. 1, pp. 363–388, 2019.

[5] Steward, J., "The ultimate list of internet of things statistics for 2022," https://findstack.com/internet-of-things-statistics, 2020, accessed: 2022-09-01.

[6] Andre, L., "Iot device statistics you must read: 2022 data on market size, adoption & usage," https://financesonline.com/iot-device-statistics, 2022, accessed: 2022-09-01.

[7] G., N., "How many iot devices are there in 2022?" https://techjury.net/blog/how-many-iot-devices-are-there, 2022, accessed: 2022-09-01.

[8] Lv, Z. and Singh, A. K., "Big data analysis of internet of things system," *ACM Transactions on Internet Technology*, vol. 21, no. 2, pp. 1–15, 2021.

[9] Chen, Y., "Iot, cloud, big data and ai in interdisciplinary domains," p. 102070, 2020.

[10] Cuzzocrea, A. and Mumolo, E., "A novel gpu-aware histogram-based algorithm for supporting moving object segmentation in big-data-based iot application scenarios," *Information Sciences*, vol. 496, pp. 592–612, 2019.

[11] Akpakwu, G. A., Silva, B. J., Hancke, G. P., and Abu-Mahfouz, A. M., "A survey on 5g networks for the internet of things: Communication technologies and challenges," *IEEE access*, vol. 6, pp. 3619–3647, 2017.

[12] Varsier, N., Dufrène, L.-A., Dumay, M., Lampin, Q., and Schwoerer, J., "A 5g new radio for balanced and mixed iot use cases: Challenges and key enablers in fr1 band," *IEEE Communications Magazine*, vol. 59, no. 4, pp. 82–87, 2021.

[13] Li, S., Da Xu, L., and Zhao, S., "5g internet of things: A survey," *Journal of Industrial Information Integration*, vol. 10, pp. 1–9, 2018.

[14] Kumar, G., Lydia, M., and Levron, Y., "Security challenges in 5g and iot networks: A review," *Secure Communication for 5G and IoT Networks*, pp. 1–13, 2022.

[15] Lata, N. and Kumar, R., "Security in internet of things (iot): Challenges and models," *Mathematical Statistician and Engineering Applications*, vol. 71, no. 2, pp. 75–81, 2022.

[16] Ahanger, T. A., Aljumah, A., and Atiquzzaman, M., "State-of-the-art survey of artificial intelligent techniques for iot security," *Computer Networks*, p. 108771, 2022.

[17] Sharifi, A., Zad, F. F., Noorollahi, A., and Sharifi, J., "An overview of intrusion detection and prevention systems (idps) and security issues," *IOSR J. Comput. Eng.(1OSR-JCE)*, vol. 16, no. 1, pp. 47–52, 2014.

[18] Kilincer, I. F., Ertam, F., and Sengur, A., "Machine learning methods for cyber security intrusion detection: Datasets and comparative study," *Computer Networks*, vol. 188, p. 107840, 2021.

[19] Ahmad, Z., Shahid Khan, A., Wai Shiang, C., Abdullah, J., and Ahmad, F., "Network intrusion detection system: A systematic study of machine learning and deep learning approaches," *Transactions on Emerging Telecommunications Technologies*, vol. 32, no. 1, p. e4150, 2021.

[20] Abbass, H., "What is artificial intelligence?" *IEEE Transactions on Artificial Intelligence*, vol. 2, no. 2, pp. 94–95, 2021.

[21] Fetzer, J. H., "What is artificial intelligence?" in *Artificial intelligence: its scope and limits*. Springer, 1990, pp. 3–27.

[22] McCarthy, J., "What is artificial intelligence," *URL: http://www-formal. stanford. edu/jmc/whatisai. html*, 2004.

[23] Bengio, I. G. Y. and Courville, A., *Deep Learning*. MIT Press, 2016. [Online]. Available: http://www.deeplearningbook.org

[24] Lenat, D. B. and Guha, R. V., *Building large knowledge-based systems; representation and inference in the Cyc project*. Addison-Wesley Longman Publishing Co., Inc., 1989.

[25] Allenby, G. M. and Lenk, P. J., "Modeling household purchase behavior with logistic normal regression," *Journal of the American Statistical Association*, vol. 89, no. 428, pp. 1218–1231, 1994.

[26] Mor-Yosef, S., Samueloff, A., Modan, B., Navot, D., and Schenker, J. G., "Ranking the risk factors for cesarean: logistic regression analysis of a nationwide study." *Obstetrics & Gynecology*, vol. 75, no. 6, pp. 944–947, 1990.

[27] Androutsopoulos, I., Koutsias, J., Chandrinos, K. V., Paliouras, G., and Spyropoulos, C. D., "An evaluation of naive bayesian anti-spam filtering," *arXiv preprint cs/0006013*, 2000.

[28] Steinberg, S., "Cyberattacks now cost companies $200,000 on average, putting many out of business," https://www.cnbc.com/2019/10/13/cyberattacks-cost-small-companies-200k-putting-many-out-of-business.html, 2019, accessed: 2022-09-01.

[29] Jenkins, S., "Learning to love siem," *Network Security*, vol. 2011, no. 4, pp. 18–19, 2011.

[30] Farwell, J. P. and Rohozinski, R., "Stuxnet and the future of cyber war," *Survival*, vol. 53, no. 1, pp. 23–40, 2011.

[31] Bronk, C. and Tikk-Ringas, E., "The cyber attack on saudi aramco," *Survival*, vol. 55, no. 2, pp. 81–96, 2013.

[32] Walters, R., "Cyber attacks on us companies in 2014," *The Heritage Foundation*, vol. 4289, pp. 1–5, 2014.

[33] at Forbes, L. M., "Hackers use ddos attack to cut heat to apartments," https://www.forbes.com/sites/leemathews/2016/11/07/ddos-attack-leaves-finnish-apartments-without-heat, 2016, accessed: 2022-09-01.

[34] Mathews, L., "Criminals hacked a fish tank to steal data from a casino," https://www.forbes.com/sites/leemathews/2017/07/27/criminals-hacked-a-fish-tank-to-steal-data-from-a-casino, 2017, accessed: 2022-09-01.

[35] Greenberg, A., "The jeep hackers are back to prove car hacking can get much worse," https://www.wired.com/2016/08/jeep-hackers-return-high-speed-steering-acceleration-hacks/, 2016, accessed: 2022-09-01.

[36] Chiu, A., "She installed a ring camera in her children's room for 'peace of mind.' a hacker accessed it and harassed her 8-year-old daughter," https://www.washingtonpost.com/nation/2019/12/12/she-installed-ring-camera-her-childrens-room-peace-mind-hacker-accessed-it-harassed-her-year-old-daughter/, 2019, accessed: 2022-09-01.

[37] Jain, S., "Woman says hacker spied on her through webcam," https://www.ndtv.com/offbeat/woman-says-hacker-spied-on-her-through-webcam-video-will-give-you-chills-1761567, 2017, accessed: 2022-09-01.

[38] Matyszczyk, C., "Hacker shouts at baby through baby monitor," https://www.cnet.com/news/hacker-shouts-at-baby-through-baby-monitor/, 2014, accessed: 2022-09-01.

[39] Hodo, E., Bellekens, X., Hamilton, A., Tachtatzis, C., and Atkinson, R., "Shallow and deep networks intrusion detection system: A taxonomy and survey," *arXiv preprint arXiv:1701.02145*, 2017.

[40] Cekmez, U., Erdem, Z., Yavuz, A. G., Sahingoz, O. K., and Buldu, A., "Network anomaly detection with deep learning," in *2018 26th Signal Processing and Communications Applications Conference (SIU)*. IEEE, 2018, pp. 1–4.

[41] Klambauer, G., Unterthiner, T., Mayr, A., and Hochreiter, S., "Self-normalizing neural networks," in *Advances in Neural Information Processing Systems*, 2017, pp. 971–980.

[42] Xu, L., Skoularidou, M., Cuesta-Infante, A., and Veeramachaneni, K., "Modeling tabular data using conditional gan," in *Advances in Neural Information Processing Systems*, 2019, pp. 7335–7345.

[43] Ray, P. P., "A survey on internet of things architectures," *Journal of King Saud University-Computer and Information Sciences*, vol. 30, no. 3, pp. 291–319, 2018.

[44] Shah, S. H. and Yaqoob, I., "A survey: Internet of things (iot) technologies, applications and challenges," in *2016 IEEE Smart Energy Grid Engineering (SEGE)*. IEEE, 2016, pp. 381–385.

[45] Scully, P., "Top 10 iot applications in 2020," https://iot-analytics.com/top-10-iot-applications-in-2020/, 2019, accessed: 2022-09-01.

[46] Ndiaye, M., Oyewobi, S. S., Abu-Mahfouz, A. M., Hancke, G. P., Kurien, A. M., and Djouani, K., "Iot in the wake of covid-19: A survey on contributions, challenges and evolution," *IEEE Access*, vol. 8, pp. 186 821–186 839, 2020.

[47] Singh, R. P., Javaid, M., Haleem, A., and Suman, R., "Internet of things (iot) applications to fight against covid-19 pandemic," *Diabetes & Metabolic Syndrome: Clinical Research & Reviews*, 2020.

[48] Yan, Z., Zhang, P., and Vasilakos, A. V., "A survey on trust management for internet of things," *Journal of network and computer applications*, vol. 42, pp. 120–134, 2014.

[49] Cvitić, I., Vujić, M. *et al.*, "Classification of security risks in the iot environment." *Annals of DAAAM & Proceedings*, vol. 26, no. 1, 2015.

[50] Tawalbeh, L., Muheidat, F., Tawalbeh, M., Quwaider, M. *et al.*, "Iot privacy and security: Challenges and solutions," *Applied Sciences*, vol. 10, no. 12, p. 4102, 2020.

[51] Selamat, A. and Iqal, Z., "Open challenges in internet of things security," in *Journal of Physics: Conference Series*, vol. 1447, no. 1. IOP Publishing, 2020, p. 012054.

[52] Alaba, F. A., Othman, M., Hashem, I. A. T., and Alotaibi, F., "Internet of things security: A survey," *Journal of Network and Computer Applications*, vol. 88, pp. 10–28, 2017.

[53] Hussain, F., Hussain, R., Hassan, S. A., and Hossain, E., "Machine learning in iot security: current solutions and future challenges," *IEEE Communications Surveys & Tutorials*, 2020.

[54] Chae, H.-S. and Choi, S., "Feature selection for efficient intrusion detection using attribute ratio," *International Journal of Computers and Communications*, vol. 8, 2014.

[55] Yeo, L. H., Che, X., and Lakkaraju, S., "Modern intrusion detection systems," *CoRR*, vol. abs/1708.07174, 2017. [Online]. Available: http://arxiv.org/abs/1708.07174

[56] Aburomman, A. A. and Reaz, M. B. I., "A survey of intrusion detection systems based on ensemble and hybrid classifiers," *Computers & Security*, vol. 65, pp. 135 – 152, 2017.

[57] Patel, A., Qassim, Q., and Wills, C., "A survey of intrusion detection and prevention systems," *Information Management & Computer Security*, vol. 18/4, pp. 277–290, 2010.

[58] Sahingoz, O. K. and Erdogan, N., "Rubdes: A rule based distributed event system," in *Computer and Information Sciences - ISCIS 2003*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 284–291.

[59] Karatas, G. and Sahingoz, O. K., "Neural network based intrusion detection systems with different training functions," in *Proceedings of the 6th International Symposium on Digital Forensic and Security (ISDFS), Antalya, Turkey*, 2018.

[60] Roopak, M., Tian, G. Y., and Chambers, J., "Deep learning models for cyber security in iot networks," in *2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC)*. IEEE, 2019, pp. 0452–0457.

[61] Alsyaibani, O. M. A., Utami, E., and Hartanto, A. D., "An intrusion detection system model based on bidirectional lstm," in *2021 3rd International Conference on Cybernetics and Intelligent System (ICORIS)*. IEEE, 2021, pp. 1–6.

[62] Faker, O. and Dogdu, E., "Intrusion detection using big data and deep learning techniques," in *Proceedings of the 2019 ACM Southeast Conference*, 2019, pp. 86–93.

[63] Hossain, M. D., Ochiai, H., Doudou, F., and Kadobayashi, Y., "Ssh and ftp brute-force attacks detection in computer networks: Lstm and machine learning approaches," in *2020 5th International Conference on Computer and Communication Systems (ICCCS)*. IEEE, 2020, pp. 491–497.

[64] Pelletier, Z. and Abualkibash, M., "Evaluating the cic ids-2017 dataset using machine learning methods and creating multiple predictive models in the statistical computing language r," *Science*, vol. 5, no. 2, pp. 187–191, 2020.

[65] Zhang, H., Huang, L., Wu, C. Q., and Li, Z., "An effective convolutional neural network based on smote and gaussian mixture model for intrusion detection in imbalanced dataset," *Computer Networks*, p. 107315, 2020.

[66] Abdulhammed, R., Musafer, H., Alessa, A., Faezipour, M., and Abuzneid, A., "Features dimensionality reduction approaches for machine learning based network intrusion detection," *Electronics*, vol. 8, no. 3, p. 322, 2019.

[67] Zhang, Y., Chen, X., Guo, D., Song, M., Teng, Y., and Wang, X., "Pccn: Parallel cross convolutional neural network for abnormal network traffic flows detection in multi-class imbalanced network traffic flows," *IEEE Access*, vol. 7, pp. 119 904–119 916, 2019.

[68] Elmasry, W., Akbulut, A., and Zaim, A. H., "Empirical study on multiclass classification-based network intrusion detection," *Computational Intelligence*, vol. 35, no. 4, pp. 919–954, 2019.

[69] Elmasry, W., Akbulut, A., and Zaim, A. H., "Evolving deep learning architectures for network intrusion detection using a double pso metaheuristic," *Computer Networks*, vol. 168, p. 107042, 2020.

[70] Shiravi, A., Shiravi, H., Tavallaee, M., and Ghorbani, A. A., "Toward developing a systematic approach to generate benchmark datasets for intrusion detection," *computers & security*, vol. 31, no. 3, pp. 357–374, 2012.

[71] Terzi, D. S., Terzi, R., and Sagiroglu, S., "Big data analytics for network anomaly detection from netflow data," in *2017 International Conference on Computer Science and Engineering (UBMK)*. IEEE, 2017, pp. 592–597.

[72] Krovich, D., Cottrill, A., and Mancini, D. J., "A cloud based entitlement granting engine," in *National Cyber Summit*. Springer, 2019, pp. 220–231.

[73] Maciá-Fernández, G., Camacho, J., Magán-Carrión, R., García-Teodoro, P., and Therón, R., "Ugr'16: A new dataset for the evaluation of cyclostationarity-based network idss," *Computers & Security*, vol. 73, pp. 411–424, 2018.

[74] Larriva-Novo, X. A., Vega-Barbas, M., Villagrá, V. A., and Rodrigo, M. S., "Evaluation of cybersecurity data set characteristics for their applicability to neural networks algorithms detecting cybersecurity anomalies," *IEEE Access*, vol. 8, pp. 9005–9014, 2020.

[75] Zong, W., Chow, Y.-W., and Susilo, W., "Interactive three-dimensional visualization of network intrusion detection data for machine learning," *Future Generation Computer Systems*, vol. 102, pp. 292–306, 2020.

[76] Dhanabal, L. and Shantharajah, S., "A study on nsl-kdd dataset for intrusion detection system based on classification algorithms," *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 4, no. 6, pp. 446–452, 2015.

[77] Moustafa, N. and Slay, J., "Unsw-nb15: a comprehensive data set for network intrusion detection systems (unsw-nb15 network data set)," in *2015 military communications and information systems conference (MilCIS)*. IEEE, 2015, pp. 1–6.

[78] Salakhutdinov, R., Mnih, A., and Hinton, G., "Restricted boltzmann machines for collaborative filtering," in *Proceedings of the 24th international conference on Machine learning*. ACM, 2007, pp. 791–798.

[79] Mohamed, A.-r., Dahl, G. E., and Hinton, G., "Acoustic modeling using deep belief networks," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 1, pp. 14–22, 2012.

[80] LeCun, Y., Bengio, Y. *et al.*, "Convolutional networks for images, speech, and time series," *The handbook of brain theory and neural networks*, vol. 3361, no. 10, p. 1995, 1995.

[81] Sermanet, P., Chintala, S., and LeCun, Y., "Convolutional neural networks applied to house numbers digit classification," in *Pattern Recognition (ICPR), 2012 21st International Conference on*. IEEE, 2012, pp. 3288–3291.

[82] Hochreiter, S. and Schmidhuber, J., "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[83] Srivastava, R. K., Greff, K., and Schmidhuber, J., "Training very deep networks," in *Advances in neural information processing systems*, 2015, pp. 2377–2385.

[84] Graves, A., Mohamed, A.-r., and Hinton, G., "Speech recognition with deep recurrent neural networks," in *2013 IEEE international conference on acoustics, speech and signal processing*. IEEE, 2013, pp. 6645–6649.

[85] Greff, K., Srivastava, R. K., Koutník, J., Steunebrink, B. R., and Schmidhuber, J., "Lstm: A search space odyssey," *arXiv preprint arXiv:1503.04069*, 2015.

[86] Chung, J., Gulcehre, C., Cho, K., and Bengio, Y., "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv preprint arXiv:1412.3555*, 2014.

[87] "CUDA compute unified device architecture," https://developer.nvidia.com, accessed: 2022-09-01.

[88] Sanders, J. and Kandrot, E., "Cuda by example," *An Introduction to General-Purpose GPU Programming/J. Sanders, E. Kandrot-Addison-Wesley Professional*, 2010.

[89] Kirk, D., "Nvidia cuda software and gpu parallel computing architecture," *NVIDIA*, vol. 7, pp. 103–104, 2007.

[90] Tavallaee, M., Bagheri, E., Lu, W., and Ghorbani, A. A., "A detailed analysis of the kdd cup 99 data set," in *Computational Intelligence for Security and Defense Applications, 2009. CISDA 2009. IEEE Symposium on*. IEEE, 2009, pp. 1–6.

[91] "Darpa 98 data set," https://www.ll.mit.edu/r-d/datasets/1998-darpa-intrusion-detection-evaluation-dataset, 1998, accessed: 2022-09-01.

[92] McHugh, J., "Testing intrusion detection systems: a critique of the 1998 and 1999 darpa intrusion detection system evaluations as performed by lincoln laboratory," *ACM Transactions on Information and System Security (TISSEC)*, vol. 3, no. 4, pp. 262–294, 2000.

[93] "Kdd cup 99," http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html, 1999, accessed: 2022-09-01.

[94] Sharafaldin, I., Lashkari, A. H., and Ghorbani, A. A., "Toward generating a new intrusion detection dataset and intrusion traffic characterization." in *ICISSP*, 2018, pp. 108–116.

[95] Ullah, I. and Mahmoud, Q. H., "A scheme for generating a dataset for anomalous activity detection in iot networks," in *Canadian Conference on Artificial Intelligence*. Springer, 2020, pp. 508–520.

[96] Hindy, H., Tachtatzis, C., Atkinson, R., Bayne, E., and Bellekens, X., "Mqtt-iot-ids2020: Mqtt internet of things intrusion detection dataset," 2020. [Online]. Available: https://dx.doi.org/10.21227/bhxy-ep04

[97] Kang, H., Ahn, D. H., Lee, G. M., Yoo, J. D., Park, K. H., and Kim, H. K., "Iot network intrusion dataset," 2019. [Online]. Available: https://dx.doi.org/10.21227/q70p-q449

[98] Hindy, H., Bayne, E., Bures, M., Atkinson, R., Tachtatzis, C., and Bellekens, X., "Machine learning based iot intrusion detection system: An mqtt case study (mqtt-iot-ids2020 dataset)," 2020.

[99] Hossen, M. A., Siddika, F., Chanda, T. K., and Bhuiyan, T., "A comparison of some soft computing methods on imbalanced data," in *International Conference on Cyber Security and Computer Science*, 2018.

[100] Chollet, F., "Xception: Deep learning with depthwise separable convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1251–1258.

[101] Ioffe, S. and Szegedy, C., "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *CoRR*, vol. abs/1502.03167, 2015. [Online]. Available: http://arxiv.org/abs/1502.03167

[102] Kim, B.-C., Sung, Y. S., and Suk, H.-I., "Deep feature learning for pulmonary nodule classification in a lung ct," in *2016 4th International Winter Conference on Brain-Computer Interface (BCI)*. IEEE, 2016, pp. 1–3.

[103] Loshchilov, I. and Hutter, F., "Fixing weight decay regularization in adam," *arXiv preprint arXiv:1711.05101*, 2017.

[104] "Tensorflow documentation for imbalanced data," tensorflow.org/tutorials/structured_data/imbalanced_data, 2020, accessed: 2022-09-01.

[105] Gencoglu, O., van Gils, M., Guldogan, E., Morikawa, C., Süzen, M., Gruber, M., Leinonen, J., and Huttunen, H., "Hark side of deep learning–from grad student descent to automated machine learning," *arXiv preprint arXiv:1904.07633*, 2019.

[106] Chamola, V., Hassija, V., Gupta, V., and Guizani, M., "A comprehensive review of the covid-19 pandemic and the role of iot, drones, ai, blockchain, and 5g in managing its impact," *IEEE Access*, vol. 8, pp. 90 225–90 265, 2020.

ÖZGEÇMİŞ

Adı Soyadı : Uğur ÇEKMEZ

Yabancı Dili : İngilizce

Öğrenim Durumu

Derece	**Bölüm / Program**	**Üniversite / Lise**	**Yılı**
Lise	Fen Bilimleri	İstanbul Vatan Anadolu Lisesi	2004-2008
Lisans	Bilgisayar Bilimleri	İstanbul Bilgi Üniversitesi	2008-2012
Y. Lisans	Bilgisayar Mühendisliği	Hava Harp Okulu Komutanlığı	2012-2014
Doktora	Bilgisayar Mühendisliği	Marmara Üniversitesi	2017-2022

İş Deneyimi

Görevi	**Firma / Kurum**	**Yılı**
Araştırma Görevlisi	Yıldız Teknik Üniversitesi Bilgisayar Mühendisliği Bölümü	2013-2017
Kurucu Ortak	Fili Labs ARGE Ltd.	2014-2019
Uzman Araştırmacı	TÜBİTAK Bulut Bilişim ve Büyük Veri Ar-Ge Laboratuvarı	2017-2018
Uzman Ar-Ge Mühendisi	SIEMENS Türkiye	2018-2020
Uzman Yapay Zeka Mühendisi	Türkiye Radyo Televizyon Kurumu (TRT) - Data&Insight Lab	2020-2021
Lead MLOps Engineer	Chooch Intelligence Technologies	2021-Halen

Bilimsel Eserler

Uluslararası hakemli dergilerde yayınlanan makaleler:

1. Sahingoz, O. K., Cekmez, U., Buldu, A. (2021). Internet of Things (IoTs) Security: Intrusion Detection using Deep Learning. *Journal Of Web Engineering*, https://doi.org/10.13052/jwe1540-9589.2062.

Bilimsel toplantılarda sunulan ve bildiri kitabında basılan bildiriler:

1. Cekmez U., Ozsiginan M., Sahingoz O. K., Adapting the GA Approach to Solve Traveling Salesman Problems on CUDA Architecture, *14th IEEE International Symposium on Computational Intelligence and Informatics*, Kasım 2013, Budapeşte, Macaristan.

2. Cekmez U., Ozsiginan M., Sahingoz O. K., Parallel Ant Colony Optimisation for UAV Route Planning on CUDA Architecture, *22th IEEE Conference on Signal Processing and Communications Applications*, Nisan 2014, Trabzon, Türkiye.

3. Cekmez U., Ozsiginan M., Sahingoz O. K., A UAV Path Planning with Parallel ACO Algorithm on CUDA Platform, *IEEE ICUAS '14, The 2014 International Conference on Unmanned Aircraft Systems*, Mayıs 2014, Orlando, Florida, ABD.

4. Cekmez U., Ozsiginan M., Aydin M., Sahingoz O. K., UAV Path Planning with Parallel Genetic Algorithms on CUDA architecture, *World Congress on Engineering*, Temmuz 2014, Londra, İngiltere.

5. Cekmez, U., Sahingoz, O. K., Parallel solution of large scale Traveling Salesman Problems by using clustering and Evolutionary Algorithms. *2016 24th Signal Processing and Communication Application Conference*, Mayıs 2016, Zonguldak, Türkiye.

6. Ozsigınan, M., Cekmez, U., Sahingoz, O. K. (2016, May). Controlling moving targets by using unmanned aerial vehicles. *2016 24th Signal Processing and Communication Application Conference*, Mayıs 2016, Zonguldak, Türkiye.

7. Cekmez, U., Ozsiginan, M., Sahingoz, O. K., Multi colony ant optimization for UAV path planning with obstacle avoidance. *IEEE ICUAS '16, The 2016 International Conference on Unmanned Aircraft Systems*, Haziran 2016, Arlington, ABD.

8. Cekmez, U., Ozsiginan, M., Sahingoz, O. K., Multi-UAV path planning with parallel genetic algorithms on CUDA architecture. *The 2016 on Genetic and Evolutionary Computation Conference Companion*, Temmuz 2016, Denver, Colorado, ABD.

9. Sezer, A., Cekmez, U., Cells classification with deep learning. *2017 25th Signal Processing and Communications Applications Conference*, Mayıs 2017, Antalya, Türkiye.

10. Cekmez, U., Ozsiginan, M., Sahingoz, O. K., Multi-UAV path planning with multi colony ant optimization. *In International Conference on Intelligent Systems Design and Applications*, Springer, Aralık 2017, Delhi, Hindistan.

11. Cekmez, U., Erdem, Z., Yavuz, A. G., Sahingoz, O. K., Buldu, A., Network anomaly detection with deep learning. *2018 26th Signal Processing and Communications Applications Conference*, Mayıs 2018, İzmir, Türkiye.

12. Demiral, Y., Carkaci, N., Cekmez, U., DevOps Architecture in the Cloud, *2019 27th Signal Processing and Communications Applications Conference*, Nisan 2019, Sivas, Türkiye.