Makine Öğrenmesi Modellerine Yönelik Gelişmiş Çekişmeli Saldırı Mimarisi

Aybüke Kılıç

YÜKSEK LİSANS TEZİ

Bilgisayar Mühendisliği Anabilim Dalı

Ağustos 2025

Advanced Adversarial Attack Architecture for Machine Learning Models

Aybüke Kılıç

MASTER OF SCIENCE THESIS

Department of Computer Engineering

August 2025

Makine Öğrenmesi Modellerine Yönelik Gelişmiş Çekişmeli Saldırı Mimarisi

Aybüke Kılıç

Eskişehir Osmangazi Üniversitesi
Fen Bilimleri Enstitüsü
Lisansüstü Eğitim ve Öğretim Yönetmeliği Uyarınca
Bilgisayar Mühendisliği Anabilim Dalı
Yapay Zeka Bilim Dalında
YÜKSEK LİSANS TEZİ
olarak hazırlanmıştır

Danışman: Dr. Öğr. Üyesi İlker Özçelik

Ağustos 2025

ONAY

Bilgisayar Mühendisliği Anabilim Dalı Yüksek Lisans öğrencisi **Aybüke Kılıç**'ın YÜKSEK LİSANS tezi olarak hazırladığı "**Makine Öğrenmesi Modellerine Yönelik Gelişmiş Çekişmeli Saldırı Mimarisi**" başlıklı bu çalışma, jürimizce lisansüstü yönetmeliğin ilgili maddeleri uyarınca değerlendirilerek oybirliği ile kabul edilmiştir.

Danışman : Dr. Öğr. Üyesi İlker Özçelik

İkinci Danışman :

Yüksek Lisans Tez Savunma Jürisi:

Üye : Dr. Öğr. Üyesi İlker Özçelik

Üye : Prof. Dr. İdiris Dağ

Üye : Dr. Öğr. Üyesi Muhammed Fatih Alaeddinoğlu

Fen Bilimleri Enstitüsü Yönetim Kurulu'nun ..... tarih ve ..... sayılı kararıyla onaylanmıştır.

Prof.Dr. Fatma TÜMSEK
Enstitü Müdürü

ETİK BEYAN

Eskişehir Osmangazi Üniversitesi Fen Bilimleri Enstitüsü tez yazım kılavuzuna göre, Dr. Öğr. Üyesi İlker Özçelik danışmanlığında hazırlamış olduğum "**Makine Öğrenmesi Modellerine Yönelik Gelişmiş Çekişmeli Saldırı Mimarisi**" başlıklı Yüksek Lisans tezimin özgün bir çalışma olduğunu; tez çalışmamın tüm aşamalarında bilimsel etik ilke ve kurallara uygun davrandığımı; tezimde verdiğim bilgileri, verileri akademik ve bilimsel etik ilke ve kurallara uygun olarak elde ettiğimi; tez çalışmamda yararlandığım eserlerin tümüne atıf yaptığımı ve kaynak gösterdiğimi ve bilgi, belge ve sonuçları bilimsel etik ilke ve kurallara göre sunduğumu beyan ederim.
04/08/2025

Aybüke Kılıç

ÖZET

Makine öğrenmesi sistemlerini yanıltmak amacıyla yapılan saldırılar çekişmeli saldırılar olarak tanımlanır. Bu saldırılarda kullanılan küçük pertübasyonlar önemli riskler oluşturur. Bu çalışma, siber güvenlik alanında çekişmeli saldırılar üretmek amacıyla PyTorch tabanlı bir Generative Adversarial Network (GAN) mimarisinin geliştirilmesini sunmaktadır.

Araştırma, dört farklı optimizasyon yaklaşımının sistematik karşılaştırmasını içermektedir: Adam optimizasyonu ile eğitilmiş temel GAN modeli, global arama yetenekleri ile öne çıkan genetik algoritma tabanlı GAN yaklaşımı, Pareto-optimal çözümler üreten çok amaçlı evrimsel algoritma (MOEA) GAN stratejisi ve Adam optimizasyonunun hızlı yakınsama avantajlarını evrimsel algoritmaların global arama yetenekleri ile birleştiren yenilikçi hibrit yaklaşım.

UNSW-NB15 veri seti üzerinde Random Forest sınıflandırıcısına karşı gerçekleştirilen deneysel çalışmalarda, önerilen hibrit yaklaşım %29.82 doğruluk düşüşü ile en yüksek saldırı etkinliğini göstermiştir. Genetik algoritma tabanlı generator optimizasyonu ile MOEA tabanlı discriminator optimizasyonunun kombinasyonu, geleneksel gradyan tabanlı yöntemlere göre %633 oranında daha yüksek saldırı başarısı sağlamıştır. L2 norm kısıtlaması altında gerçekleştirilen saldırılar, gizlilik-etkinlik dengesini başarılı şekilde korumuştur.

Sonuçlar, hibrit optimizasyon stratejilerinin çekişmeli makine öğrenmesinde yeni bir paradigma oluşturabileceğini göstermekte ve hem saldırı hem de savunma perspektifinden siber güvenlik alanında önemli gelişmelere kapı aralamaktadır.

Anahtar Kelimeler: Üretken Çekişmeli Ağlar, Genetik Algoritma, Çok Amaçlı Evrimsel Algoritma, Çekişmeli Örnekler, Makine Öğrenmesi Güvenliği, Optimizasyon, Çekişmeli Saldırılar

SUMMARY

Adversarial attacks are defined as attacks aimed at deceiving machine learning systems. The small perturbations used in these attacks pose significant risks. This study presents the development of a PyTorch-based Generative Adversarial Network (GAN) architecture for generating adversarial attacks in the field of cybersecurity.

The research includes a systematic comparison of four different optimization approaches: a baseline GAN model trained with Adam optimization, a genetic algorithm-based GAN approach distinguished by global search capabilities, a multi-objective evolutionary algorithm (MOEA) GAN strategy producing Pareto-optimal solutions, and an innovative hybrid approach combining the fast convergence advantages of Adam optimization with the global search capabilities of evolutionary algorithms.

In experimental studies conducted against Random Forest classifier on the UNSW-NB15 dataset, the proposed hybrid approach demonstrated the highest attack effectiveness with 29.82% accuracy drop. The combination of genetic algorithm-based generator optimization with MOEA-based discriminator optimization achieved 633% higher attack success compared to traditional gradient-based methods. Attacks performed under L2 norm constraint successfully maintained the stealth-effectiveness balance.

Results indicate that hybrid optimization strategies can establish a new paradigm in adversarial machine learning and open doors to significant developments in cybersecurity from both attack and defense perspectives.
Keywords: Generative Adversarial Networks (GANs), Genetic Algorithm, Multi-Objective Evolutionary Algorithm (MOEA), Adversarial Examples, Machine Learning Security, Optimization, Adversarial Attacks

TEŞEKKÜR

Beni bu çalışmaya sevkeden ve yöneten, çalışma boyunca değerli yardımlarını esirgemeyen Hocam, sayın **Dr. Öğr. Üyesi İlker Özçelik**'e; sevgili annem Aniye, babam Ahmet ve abim İrfan'a sonsuz saygı ve teşekkürlerimi sunarım. Ayrıca kıymetli jüri üyelerim Dr. Öğr. Üyesi Muhammed Fatih ALEADDİNOĞLU ve Prof Dr. İdris DAĞ'ın incelemeleri, ilerlememde destek olan yapıcı dönütleri ve değerli katkıları için teşekkür ederim.

İÇİNDEKİLER

Sayfa

ÖZET . . . . . **vi**

SUMMARY . . . . . **vii**

TEŞEKKÜR . . . . . **viii**

İÇİNDEKİLER . . . . . **ix**

ŞEKİLLER DİZİNİ . . . . . **xii**

ÇİZELGELER DİZİNİ . . . . . **xiii**

SİMGELER VE KISALTMALAR DİZİNİ . . . . . **xiv**

1. GİRİŞ VE AMAÇ **1**

2. LİTERATÜR ARAŞTIRMASI **5**

2.1.. Makine Öğrenmesi ve Çekişmeli Örneklerin Temelleri 5

2.1.1.. Çekişmeli Örneklerin Önemi ve Uygulama Alanları 5

2.2.. Çekişmeli Örnek Oluşturma Yöntemleri 6

2.3.. Optimizasyon Yöntemlerine Genel Bakış 6

2.4.. Üretken Çekişmeli Ağlar (GAN'lar) ve Temel Mimariler 7

2.4.1.. Genetik Algoritmaların GAN Eğitimindeki Rolü 8

2.4.2.. Genetik Algoritmaların GAN Eğitimindeki Rolü 9

2.5.. Çekişmeli Saldırı Performans Metrikleri ve Değerlendirme Kriterleri 10

2.6.. Zorluklar ve Açık Problemler 11

3. TEORİK BİLGİ **12**

3.1.. Makine Öğrenmesi Sınıflandırması 12

3.2.. Makine Öğrenmesi Süreci 13

3.3.. Çekişmeli Saldırılar 14

3.4.. Çekişmeli Örneklerin Tanımı ve Önemi 17

3.5.. Üretici Çekişmeli Ağlar(GAN) 18

3.5.1.. GAN'ların Temel Yapısı ve Çalışma Prensipleri 19

3.5.2.. GAN Mimarileri ve Varyasyonları 20

3.5.3.. GAN'ların Matematiksel Temelleri 21

3.5.4.. GAN Eğitimindeki Zorluklar 22

3.5.5.. GAN'ların Çekişmeli Örnek Üretimindeki Rolü 23
3.6.. Evrimsel Hesaplama 24
3.6.1.. Genetik Algoritmalar 24
3.6.2.. Çok Amaçlı Evrimsel Algoritmalar 26
3.7.. GAN Eğitiminde Optimizasyon Teknikleri 28
3.7.1.. Gradyan Tabanlı Yöntemler 29
3.7.1.1.. Stokastik Gradyan İnişi (SGD) 29
3.7.1.2.. Adam Optimizer 29
3.7.2.. Evrimsel Optimizasyon Yöntemleri 30
3.7.2.1.. Genetik Algoritmalar 31
3.7.2.2.. Çok Amaçlı Optimizasyon 32

4. YÖNTEM 33
4.1.. Veri Seti ve Ön İşleme 33
4.2.. Hedef Model Mimarisi 33
4.3.. Önerilen GAN Mimarisi 34
4.3.1.. Temel GAN Yapısı 34
4.3.2.. Standartlaştırılmış Genetik Operatörler 34
4.3.3.. L2 Norm Kısıtlaması 34
4.3.4.. Karşılaştırılan Yöntemler 34
4.3.4.1.. Adam Tabanlı GAN (Baseline) 34
4.3.4.2.. Genetik Algoritma Tabanlı GAN 34
4.3.4.3.. Çok Amaçlı Evrimsel GAN (MOEA) 34
4.3.4.4.. Önerilen GAN 35
4.4.. Değerlendirme Metrikleri 35
4.4.1.. Model Doğruluk Karşılaştırması 35
4.4.1.1.. Doğruluk (Accuracy) 35
4.4.2.. Saldırı Etkinliği Metrikleri 35
4.4.2.1.. L2 Norm - Trafik Değişim Büyüklüğü 35
4.4.2.2.. Ortalama L2 Norm 36
4.4.3.. Deneysel Kurulum 37

5. BULGULAR VE TARTIŞMA 38
5.1.. GAN Eğitim Dinamikleri ve Yakınsama Analizi 40
5.1.1.. Adam GAN Eğitim Süreci 40
5.1.2.. Genetik Algoritma Tabanlı Yaklaşım 40
5.1.3.. Çok-Amaçlı Evrimsel Algoritma (MOEA) 40
5.1.4.. Hibrit GAN Yaklaşımı 42
5.2.. Güvenlik İmplikasyonları ve Sonuçlar 43

6. SONUÇ VE ÖNERİLER 44
6.1.. Araştırma Sonuçlarının Değerlendirilmesi 44
6.2.. Gelecek Araştırma Yönleri 44
6.3.. Son Değerlendirme 45

KAYNAKLAR DİZİNİ 46

ŞEKİLLER DİZİNİ

Şekil		Sayfa
3.1	Saldırı Sınıflandırması	15
4.1	Yöntemlerin Karşılaştırılması	36
5.1	Model Doğruluk Karşılaştırması (GPU Hızlandırılmış)	38
5.2	Saldırı Etkinliği Karşılaştırması (PyTorch Düşmanca Bozulmaları)	39
5.3	Bozulma Büyüklüğü (L_2 Norm — Düşük Daha Gizli)	40
5.4	PyTorch Adam GAN (Üretici vs Ayırt Edici)	41
5.5	PyTorch Genetik Algoritma (GA Üretici + GA Ayırt Edici)	41
5.6	PyTorch MOEA (3-Amaçlı Üretici + 3-Amaçlı Ayırt Edici)	42
5.7	PyTorch Önerilen Hibrit GAN (GA Üretici + MOEA Ayırt Edici)	42

ÇİZELGELER DİZİNİ

Çizelge **Sayfa**

5.1 Model doğruluğu karşılaştırması ve saldırı etkinliği 39

SİMGELER VE KISALTMALAR DİZİNİ

Simgeler	Açıklamalar
G	Üretici ağ (Generator)
D	Ayırıcı ağ (Discriminator)
θ	Model parametreleri/ağırlıkları
x	Girdi verisi/örneği
x'	Çekişmeli örnek
z	Gizli uzay vektörü/gürültü
η	Çekişmeli pertürbasyon
ε	L2 norm kısıtı
α	Çaprazlama katsayısı
σ	Mutasyon standart sapması
β_1, β_2	Adam optimizatörü momentum katsayıları
η_t	Öğrenme oranı
$f(x)$	Amaç/kayıp fonksiyonu
$\nabla_\theta f(\theta_t)$	Gradyan vektörü
$F(x)$	Çok amaçlı fonksiyon vektörü
m_t, v_t	Adam optimizatöründe moment tahminleri
p_i	Seçilim olasılığı
f_i	Fitness değeri
L_2, L_∞, L_0	Pertürbasyon büyüklük normları

Kısaltmalar	Açıklamalar
AD	Doğruluk Düşüşü (Accuracy Drop)
AdaBoost	Adaptif Artırma (Adaptive Boosting)
Adam	Uyarlanabilir Moment Tahmini
AdaGrad	Uyarlanabilir Gradyan Algoritması
AdvGAN	Çekişmeli GAN
AE	Otomatik Kodlayıcı (Autoencoder)
AI	Yapay Zeka (Artificial Intelligence)
AlexNet	Görüntü Sınıflandırma Ağı (AlexNet)
AME-GAN	Uyarlanabilir Çok Amaçlı Evrimsel GAN
ASR	Saldırı Başarı Oranı
BEGAN	Belirgin Çekişmeli Ağ (Boundary Equilibrium GAN)
BERT	İki Yönlü Kodlayıcı Temsili
BIM	Temel Yinelemeli Yöntem
C&W	Carlini ve Wagner Saldırısı
CDE-GAN	İşbirlikçi Diferansiyel Evrim GAN
CGAN	Koşullu GAN (Conditional GAN)
CNN	Evrişimsel Sinir Ağı
CPU	Merkezi İşlem Birimi
CUDA	Birleşik Bilgi İşlem Aygıt Mimarisi
CycleGAN	Çevrim Tutarlılığı GAN
DCGAN	Derin Evrişimsel GAN
DDoS	Dağıtık Hizmet Dışı Bırakma
DE	Diferansiyel Evrim
DeepFool	Derin Kandırma Saldırısı
DoS	Hizmet Dışı Bırakma
DT	Karar Ağaçları (Decision Trees)
E-GAN	Evrimsel GAN
EC	Evrimsel Hesaplama
FID	Fréchet Başlangıç Mesafesi
FN	Yanlış Negatif
FP	Yanlış Pozitif
FGSM	Hızlı Gradyan İşaret Yöntemi
GA	Genetik Algoritma
GAGAN	Genetik Algoritma GAN
GAN	Üretken Çekişmeli Ağ (Generative Adversarial Network)

Kısaltmalar	Açıklamalar
GBM	Gradient Boosting Makineleri (Gradient Boosting Machines)
GPU	Grafik İşlem Birimi
IDS	Saldırı Tespit Sistemi
IoT	Nesnelerin İnterneti
IS	Başlangıç Skoru (Inception Score)
JSMA	Jacobian Tabanlı Önemlilik Haritası Saldırısı
kNN	k-En Yakın Komşu
L-BFGS	Broyden-Fletcher-Goldfarb-Shanno
LSGAN	Least Squares GAN
LSTM	Uzun Kısa Süreli Bellek
ML	Makine Öğrenmesi (Machine Learning)
MLFFNN	Çok Katmanlı Beslemeli Sinir Ağı
MOEA	Çok Amaçlı Evrimsel Algoritma
MOEA/D	Ayrıştırma Tabanlı Çok Amaçlı Evrimsel Algoritma
MSE	Ortalama Kare Hata (Mean Squared Error)
NSGA-II	Baskın Olmayan Sıralama Genetik Algoritması II
PCA	Temel Bileşen Analizi
PGD	Projeksiyonlu Gradyan İnişi
PSO	Parçacık Sürü Optimizasyonu
ResNet	Kalan Bağlantılı Ağ (Residual Network)
RF	Rastgele Orman (Random Forest)
RMSprop	Kök Ortalama Kare Yayılımı
RNN	Tekrarlayan Sinir Ağı
SGD	Stokastik Gradyan İnişi
SOM	Kendine Organize Olmuş Haritalar (Self-Organizing Maps)
SPEA2	Güç Pareto Evrimsel Algoritması 2
SSIM	Yapısal Benzerlik İndeksi
StackGAN	Yığılmış GAN
SVM	Destek Vektör Makinesi
TN	Gerçek Negatif
TP	Gerçek Pozitif
TTP	Taktik, Teknik ve Prosedürler
UAE	Kısıtsız Çekişmeli Örnekler
UNSW-NB15	University of New South Wales Network Behavior 15
VGG	Görüntü Sınıflandırma Ağı (Visual Geometry Group)
ZOO	Sıfırıncı Derece Optimizasyon

1. GİRİŞ VE AMAÇ

Makine öğrenmesi (ML) sistemleri, eğitim verilerinden öğrenen bir programlama disiplinidir. Sistem tasarımcıları için çekici bir seçenektir. Öğrenme tabanlı yöntemler kullanmayan geleneksel sistemlerin aksine, büyük miktarda veri ile hızlı ve verimli bir şekilde çalışırlar. Makine öğrenmesi teknolojisinin kullanımı şüphesiz hayatımızın birçok yönünü kolaylaştırmıştır. Makine öğrenmesi e-posta filtreleri, öneri sistemleri, reklam yerleştirme, dolandırıcılık tespiti ve web arama gibi çeşitli uygulamalarda kullanılmıştır (Domingos, 2012).

Carter'ın "Beyond ChatGPT" raporuna göre, makine öğrenmesi teknolojisi gelecekte çok daha etkili olacaktır. 2025 yılına kadar, büyük kuruluşlardan gelen pazarlama mesajlarının yaklaşık %30'unun makine öğrenmesi veya diğer yapay zeka tabanlı teknikler kullanılarak yapay olarak üretilmesi beklenmektedir. Benzer şekilde, 2030 yılına kadar içeriğinin %90'ı yapay zeka tarafından üretilen bir gişe rekorları kıran filmin vizyona girmesi öngörülmektedir (Wiles, 2023).

İşimizi kolaylaştırmayı amaçlayan bu gelişen teknolojinin, ele alınması gereken bazı güvenlik ve gizlilik riskleri de beraberinde getirdiğini kabul etmek önemlidir. Schneier'in 2023 tarihli makalesinde, yapay zeka kullanan ürünlerin çoğunun potansiyel riskler konusunda farkındalık olmadan piyasaya sürüldüğü belirtilmektedir (Schneier, 2023). Bu risklerin yoğunluğunu ele almak için siber güvenlik uzmanları, makine öğrenmesi mühendisleri ve rakip makine öğrenmesi araştırmacılarının birlikte çalışması önerilmektedir (Madry vd., 2017).

Yapay zeka modelleri kötü niyetli aktörlere maruz kalabilir. Maruz kalan faktörler, modelin sonuçlarının hatalı olacağı durumlar yaratır ve bu kasıtlı eylemler "Düşmanca saldırılar" olarak adlandırılır. Bu saldırılar sırasında, düşmanca örnekler adı verilen küçük sapmalar oluşturulur ve ML sistemine enjekte edilir. Örnekler algılanamaz olacak şekilde tasarlanır ve modelin yanlış sınıflandırmalar yapmasına neden olmayı amaçlar.

Düşmanca saldırılar görüntü tanıma, bilgisayarlı görü, doğal dil işleme ve siber güvenlik dahil olmak üzere birçok alanı hedeflemektedir (Rosenberg vd., 2021). Örneğin, Nesnelerin İnterneti ve otonom arama çalışmalarında, kanal

durum bilgisi örnekleri de düşmanca pertürbasyonlar göstermektedir (Y. Wang vd., 2023).

Makine öğrenmesi modellerinin güvenlik açıklarını test etmek için geliştirilen düşmanca saldırı yöntemleri son yıllarda önemli bir araştırma alanı haline gelmiş ve bu alanda çeşitli optimizasyon stratejilerinin etkinliği sorgulanmaya başlanmıştır. Bu çalışma, düşmanca örnek üretimi için dört farklı optimizasyon yaklaşımının karşılaştırmalı bir analizini sunmaktadır: başlangıç referansı olarak kullanılan klasik Adam optimizasyonu ile eğitilmiş temel GAN modeli, global arama yetenekleri ile öne çıkan genetik algoritma tabanlı GAN yaklaşımı, Pareto-optimal çözümler üreten çok amaçlı evrimsel algoritma (MOEA) GAN stratejisi ve Adam optimizasyonunun hızlı yakınsama avantajlarını evrimsel algoritmaların global arama yetenekleri ile birleştiren yenilikçi bir yaklaşım.

Bu dört metodolojinin performans karşılaştırması, düşmanca makine öğrenmesi alanında optimizasyon tekniklerinin etkinliğini değerlendirmek ve hibrit yaklaşımların potansiyelini ortaya çıkarmak için yürütülmektedir. Bu çalışma, düşmanca makine öğrenmesi alanında evrimsel algoritmaların ve GAN'ların entegrasyonunu inceleyen sistematik bir yapıya sahiptir. Literatür taraması (Bölüm 2), teorik temel (Bölüm 3), dört farklı metodolojinin karşılaştırması (Bölüm 4), deneysel sonuçlar (Bölüm 5) ve kapsamlı değerlendirme (Bölüm 6) aracılığıyla, çok boyutlu performans metrikleri kullanarak düşmanca örnek üretiminde hibrit yaklaşımların etkinliğini analiz etmektedir.

Bu çalışmada, çekişmeli saldırı üretimi için PyTorch tabanlı bir GAN framework'ü geliştirilmiştir. Önerilen yaklaşım, Adam optimizer ile temel GAN eğitimini gerçekleştirirken, generator ağının evrimini Genetik Algoritma (GA) ile, discriminator optimizasyonunu ise Çok Amaçlı Evrimsel Algoritma (MOEA) ile hibrit bir şekilde yürütmektedir. Sistematik bir karşılaştırma sağlamak amacıyla, uniform crossover (0.9 oranı), Gaussian mutation (0.05 oranı), tournament selection ve elitist replacement (%20) içeren standartlaştırılmış genetik operatörler kullanılmıştır. Üretilen adversarial örneklerin bilimsel doğruluğunu sağlamak için L2 norm kısıtları uygulanarak gerçek adversarial pertürbasyon üretimi gerçekleştirilmiştir. MOEA yaklaşımında generator için saldırı etkinliği, çeşitlilik ve gizlilik; discriminator için ise gerçek/sahte tespit doğruluğu ve güven çeşitliliği olmak üzere çok amaçlı optimizasyon kriterleri Pareto front analizi ile eşzamanlı optimize edilmiştir. UNSW-NB15 veri seti üzerinde Random Forest sınıflandırıcısına karşı gerçekleştirilen deneysel çalışmalarda, önerilen yaklaşımın geleneksel gradyan tabanlı yöntemlerin hızlı yakınsama avantajını evrimsel

algoritmaların global arama kabiliyeti ile birleştirerek daha etkili ve çeşitli adversarial örnekler ürettiği gösterilmiştir.

Makine öğrenmesi modellerinin güvenliğini sağlamada Red Team (Kırmızı Takım) ve Blue Team (Mavi Takım) yaklaşımları kritik bir rol oynar. Red Team'ler, kurumsal siber güvenliği güçlendirmek için gerçek tehdit aktörlerinin taktik, teknik ve prosedürlerini (TTP'ler) taklit eden gerçekçi saldırı simülasyonları yürütür. Bu simülasyonlar, sistem açıklarını proaktif bir şekilde ortaya çıkarır ve hem teknik güvenlik önlemlerini hem de yüksek baskı altındaki senaryolarda insan tepkilerini değerlendirir. BT altyapısı, yazılım, personel ve operasyonel süreçler gibi geniş bir yelpazeyi kapsayan bu değerlendirmeler, geleneksel güvenlik denetimlerinde sıklıkla gözden kaçan zayıflıkları ortaya çıkarır.

Blue Team'ler ise, kuruluşun savunma merkezini oluşturur. Red Team operasyonlarından elde edilen içgörüleri analiz ederek, güvenlik protokollerini güçlendirir ve geliştirilmiş koruyucu önlemler uygular. Bu süreçte, tehdit tespiti ile olay müdahale yeteneklerini iyileştirirler. Tespit edilen boşlukları gidererek ve ortaya çıkan tehditlere uyum sağlayarak, Blue Team'ler kuruluşun yalnızca simüle edilmiş saldırılara dayanmasını değil, aynı zamanda gerçek siber tehditlere karşı daha fazla direnç (resilience) inşa etmesini sağlarlar.(Chindrus ve Caruntu, 2023)

Red Team ve Blue Team arasındaki bu sinerjik ilişki, sürekli bir geri bildirim döngüsünü besler. Test etme, öğrenme ve güçlendirmeden oluşan bu devam eden döngü, makine öğrenmesi modellerinin genel güvenlik pozisyonunu (posture) zaman içinde kademeli olarak güçlendirir (Chindrus ve Caruntu, 2023). Bu yaklaşım, hem saldırı hem de savunma stratejilerinin birlikte geliştirilmesine olanak tanıyarak, daha güvenli ve dayanıklı makine öğrenmesi sistemlerinin oluşturulmasını sağlar.

Siber güvenlik açısından incelendiğinde Red Team'ler kurumsal siber güvenliği güçlendirmede hayati işlevler üstlenir. Gerçek tehdit aktörlerinin taktik, teknik ve prosedürlerini (TTP'ler) taklit eden gerçekçi saldırı simülasyonları yürüterek, sistem açıklarını proaktif bir şekilde ortaya çıkarırlar. Değerlendirmeleri, hem teknik güvenlik önlemlerini hem de yüksek baskı altındaki senaryolarda insan tepkilerini değerlendirmek üzere geniş bir yelpazeyi kapsar—BT altyapısı, yazılım, personel ve operasyonel süreçler dahil. Bu kapsamlı yaklaşım, geleneksel güvenlik denetimlerinde sıklıkla gözden kaçan zayıflıkları ortaya çıkarır. (Chindrus ve Caruntu, 2023)

Bu çalışmada, Red Team yaklaşımı çekişmeli saldırıların geliştirilmesi ve test edilmesi sürecinde uygulanmıştır. Önerilen GAN framework'ü, Red Team perspektifinden adversarial örnekler üreterek makine öğrenmesi modellerinin zayıflıklarını ortaya çıkarmayı hedeflemektedir. Blue Team perspektifi ise gelecek çalışmalarda bu saldırılara karşı savunma mekanizmalarının geliştirilmesi ve model dayanıklılığının artırılması için açık bir alan sunmaktadır.

2. LİTERATÜR ARAŞTIRMASI

2.1. Makine Öğrenmesi ve Çekişmeli Örneklerin Temelleri

Makine öğrenmesi, özellikle derin öğrenme modelleri, konuşma tanıma, görsel nesne tanıma ve nesne tespiti gibi çeşitli alanlarda kullanılmaktadır (LeCun, Bengio ve Hinton, 2015). Ancak, bu modellerin güvenliği ve sağlamlığı konusunda önemli endişeler bulunmaktadır. Szegedy vd. tarafından yürütülen çalışma derin öğrenme modellerinin, insan gözüyle algılanamayan küçük ve kasıtlı olarak tasarlanmış pertürbasyonlara karşı son derece hassas olduğunu ortaya koymuştur (I. J. Goodfellow, Shlens ve Szegedy, 2014). Bu pertürbasyonlar "çekişmeli örnekler" olarak adlandırılmaktadır. Bu örnekler modelin eğitim setindeki verilerle benzerlik gösterse de, modelin yanlış sınıflandırma yapmasına neden olacak şekilde tasarlanmıştır. Goodfellow vd. çekişmeli örneklerin varlığının, derin öğrenme modellerinin doğrusal olmayan karakteristiklere sahip olmalarına rağmen, aslında oldukça doğrusal davranış sergilemelerinden kaynaklandığını öne sürmüştür (I. J. Goodfellow, Shlens ve Szegedy, 2014).

2.1.1. Çekişmeli Örneklerin Önemi ve Uygulama Alanları

Çekişmeli örnekler hem akademik hem de pratik açıdan büyük önem taşımaktadır. Akademik perspektiften, bu örnekler makine öğrenmesi modellerinin karar verme mekanizmalarını anlamamıza yardımcı olmakta ve bu modellerin zayıf noktalarını belirlememizi sağlamaktadır(I. J. Goodfellow, Shlens ve Szegedy, 2014). Pratik perspektiften ise, çekişmeli örnekler gerçek dünya uygulamalarında önemli güvenlik riskleri yaratabilmektedir (Kurakin vd., 2018). Örneğin, otonom bir aracın trafik işaretlerini yanlış yorumlaması veya bir yüz tanıma sisteminin yetkisiz bir kişiye erişim vermesi gibi durumlar çekişmeli saldırıların sonucu olarak ortaya çıkabilmektedir. Çekişmeli örneklerin bir diğer önemli özelliği, bir model için oluşturulan çekişmeli örneklerin genellikle diğer modeller için de etkili olmasıdır. Bu özellik, saldırganların hedef modelin mimari detaylarını veya eğitim verilerini bilmeseler bile etkili çekişmeli örnekler oluşturabilecekleri anlamına gelmektedir (Papernot, McDaniel, I. Goodfellow, vd., 2017).

2.2. Çekişmeli Örnek Oluşturma Yöntemleri

Çekişmeli örnek oluşturmak için çeşitli yöntemler geliştirilmiştir. Fast Gradient Sign Method (FGSM) (I. J. Goodfellow, Shlens ve Szegedy, 2014), Projected Gradient Descent (PGD) (Madry vd., 2017) ve Carlini-Wagner (CW) saldırıları (Carlini ve Wagner, 2017) bu yöntemlerden bazılarıdır. Bu yöntemler çoğunlukla gradyan tabanlı yaklaşımlar kullanarak giriş örneklerini modelin kaybını maksimize edecek şekilde pertürbe etmektedir. Bu gradyan tabanlı geleneksel yöntemler etkili olmakla birlikte, bazı sınırlamaları bulunmaktadır. Özellikle kara-kutu saldırıları için gradyan bilgisine erişim olmadığında bu yöntemler yetersiz kalabilmektedir. Bu nedenle, son yıllarda araştırmacılar alternatif optimizasyon yaklaşımlarına yönelmiştir. Üretici Adversarial Ağlar (GAN'lar), adversarial örnek üretimi için güçlü bir alternatif sunarak bu alanda büyük ilgi görmüştür (Xiao vd., 2018). GAN'ların başarısı, geleneksel gradyan tabanlı yöntemlerden farklı olarak, üretici ağın adversarial örnekleri otomatik olarak öğrenebilme kabiliyetinden kaynaklanmaktadır. Bunun yanı sıra, Genetik Algoritmalar (GA) ve Çok Amaçlı Evrimsel Algoritmalar (MOEA) gibi evrimsel hesaplama yöntemleri de adversarial örnek üretiminde umut verici sonuçlar göstermiştir. Bu yöntemler, gradyan bilgisine ihtiyaç duymadan popülasyon tabanlı arama stratejileri kullanarak etkili adversarial örnekler üretebilmektedir (Alzantot vd., 2019). Özellikle MOEA yaklaşımları, adversarial örneklerin etkililik, algılanamazlık ve transferability gibi çoklu kriterlerini eşzamanlı olarak optimize edebilme avantajı sunmaktadır. Bu bağlamda, adversarial örnek üretimi artık sadece gradyan tabanlı optimizasyon problemi olmaktan çıkarak, çok amaçlı optimizasyon ve üretici modelleme alanlarının kesişiminde yer alan interdisipliner bir araştırma alanı haline gelmiştir.

2.3. Optimizasyon Yöntemlerine Genel Bakış

Makine öğrenmesinde optimizasyon, bir modelin performansını maksimize etmek veya hata fonksiyonunu minimize etmek için model parametrelerinin sistematik olarak ayarlanması sürecidir. Geleneksel optimizasyon yöntemleri genellikle gradyan tabanlı yaklaşımları kullanır ve bu yöntemler çoğu denetimli öğrenme probleminde başarılı sonuçlar vermektedir. (Sun vd., 2019) Stokastik Gradyan İnişi (SGD): En temel optimizasyon algoritması olan SGD, kayıp fonksiyonunun gradyanı yönünde parametre güncellemeleri yaparak yerel minimumlara yakınsamaya çalışır(Ruder, 2016). SGD'nin basitliği ve hesaplama verimliliği onu derin öğrenmede yaygın olarak kullanılan bir yöntem haline getirmiştir. Momentum Tabanlı Yöntemler: Momentum, SGD'nin yavaş yakınsama

sorununu çözmek için geliştirilmiş bir tekniktir. Önceki güncellemelerin hızını hesaba katarak salınımları azaltır ve daha hızlı yakınsama sağlar (Dehghani ve Samet, 2020). Uyarlanabilir Öğrenme Oranı Algoritmaları: Adam, RMSprop ve AdaGrad gibi algoritmalar, her parametre için ayrı öğrenme oranları kullanarak optimizasyon sürecini iyileştirirler. Bu yöntemler, farklı parametrelerin farklı hızlarda güncellenmesine olanak tanır (Sun vd., 2019).

Geleneksel gradyan tabanlı yöntemlerin sınırlamalarını aşmak için, doğal evrimden ilham alan evrimsel algoritmalar geliştirilmiştir. Bu algoritmalar, gradyan bilgisine ihtiyaç duymadan çalışırlar ve global arama kabiliyetleri ile öne çıkarlar. Genetik Algoritmalar (GA): Seçilim, çaprazlama ve mutasyon operatörlerini kullanarak bir çözüm popülasyonunu evrimleştiren meta-sezgisel algoritmalar (Holland, 1992). GA'lar, dışbükey olmayan ve çok modlu optimizasyon problemlerinde etkili sonuçlar verebilir. Diferansiyel Evrim (DE): Popülasyon tabanlı başka bir evrimsel algoritma olan DE, vektör farklarını kullanarak yeni çözümler üretir ve sürekli optimizasyon problemlerinde güçlü performans gösterir (Price vd., 2005). Parçacık Sürü Optimizasyonu (PSO): Kuş sürülerinin davranışından ilham alan PSO, parçacıkların sosyal ve bilişsel öğrenme ile en iyi çözümlere doğru hareket etmesini simüle eder (Kennedy ve Eberhart, 1995).

2.4. Üretken Çekişmeli Ağlar (GAN'lar) ve Temel Mimariler

Son yıllarda, Generative Çekişmeli Networks (GAN'lar) kullanarak çekişmeli örnek oluşturma fikri sıkça kullanılmaktadır (Xiao vd., 2018). GAN'ların gerçekçi ve çeşitli örnekler oluşturabilme kabiliyeti, çekişmeli örnek oluşturmada önemli avantajlar sağlamaktadır. Özellikle GAN'lar kullanılarak, belirli bir hedef modele özgü olmayan genel çekişmeli örnekler oluşturulabilmektedir (Papernot, McDaniel, Jha, vd., 2016). Ian Goodfellow ve ekibi tarafından 2014 yılında kavramsallaştırılan Üretken Çekişmeli Ağlar (GAN'lar), üretken yapay zekanın merkezinde yer alan dönüştürücü bir makine öğrenimi çerçevesini temsil etmektedir. Bu ağlar, verilen bir eğitim veri kümesine istatistiksel olarak benzeyen yeni veri örneklerini özerk bir şekilde öğrenmek ve üretmek için tasarlanmıştır. GAN'ın temel mimarisi, iki farklı, rekabetçi sinir ağından oluşmaktadır: bir Generator(G) ve bir Discriminator (D). Generator Modeli (G), genellikle bir Gauss veya tekdüze dağılım gibi basit bir ön dağılımdan örneklenen rastgele gürültüyü girdi olarak alır . Generatorün amacı, bu gürültüyü, diskriminatörün otantik olarak

sınıflandırması için yeterince gerçekçi sentetik veri örneklerine dönüştürmektir. Diskriminatör Modeli (D), ikili bir sınıflandırıcı olarak işlev görerek iki tür veri değerlendirir: gerçek eğitim veri kümesinden alınan orijinal örnekler ve generator tarafından üretilen sentetik örnekler. Görevi, bu gerçek ve sahte girdiler arasında doğru bir şekilde ayrım yapmak ve girdinin algılanan orijinalliğini gösteren bir olasılık puanı çıkarmaktır (I. J. Goodfellow, Pouget-Abadie, vd., 2014).

GAN'ları etkili bir şekilde eğitmek son derece zordur ve sıklıkla gradyan kaybolması, mod çökmesi ve kararsızlık gibi sorunlarla karşılaşılmaktadır. Gradyan kaybolması, diskriminatörün aşırı güçlü hale gelmesiyle ortaya çıkar ve generator için aşırı küçük gradyanlara yol açar. Mod çökmesi ise generatorün yalnızca sınırlı bir örnek alt kümesi üretmeyi öğrenmesi durumunda ortaya çıkar. Yakınsamama problemi ise GAN eğitim sürecinin istikrarlı bir sonuca ulaşamaması, ağların salınım yapması veya optimizasyonun duraksaması şeklinde kendini gösterir. Bu durumda, model yeterli kalitede ve çeşitlilikte örnekler üretemez; eğitim süreci uzar veya tamamen başarısız olabilir. (S. S. Khanuja ve H. K. Khanuja, 2021)

2.4.1. Genetik Algoritmaların GAN Eğitimindeki Rolü

Genetik algoritmaların GAN eğitimi ve çıktı üretiminde kullanımı son yıllarda dikkat çekici başarılar elde etmiştir. Genetik algoritmalar, başlangıçta rastgele oluşturulan bir çözüm popülasyonunu evrimsel operatörler (seçilim, çaprazlama, mutasyon) aracılığıyla iyileştirerek optimuma yakın çözümler bulmaya çalışır. Bu yaklaşım GAN eğitimine entegre edildiğinde, geleneksel gradyan tabanlı optimizasyonu tamamen değiştirmese de tamamlayıcı bir rol oynayabilir. Örneğin, Konstantopoulou vd. (Konstantopoulou vd., 2024), GAGAN adını verdikleri önerilen modellerinde, klasik GAN eğitim döngüsüne ek olarak discriminator ağ ağırlıklarını GA ile evrimleştirmişlerdir. Bu modelde, her eğitim iterasyonunda önce discriminator ağı backpropagation (Adam optimizasyonu) ile güncellenir, sonrasında genetik algoritma popülasyonundaki ağırlık bireyler seçilim, çaprazlama ve mutasyon adımları ile evrimleştirilerek en iyi ağırlık seti belirlenir.

GAGAN yaklaşının amacı, gradyan tabanlı öğrenmenin yerel arama gücü ile GA'nın global arama kabiliyetini birleştirerek eğitim sürecini stabilize etmektir. Nitekim deneyler, GAGAN yaklaşımının standart DCGAN'a kıyasla daha düşük generator kaybı ve daha yüksek çıktı kalitesi sağladığını göstermiştir; discriminator kaybının eğitim boyunca oldukça stabil kaldığı ve böylece mod

çökmesi ve salınımlı öğrenme gibi sorunların azaldığı bildirilmiştir. Bu sonuçlar, evrimsel algoritmaların derin öğrenme tekniklerini etkili bir şekilde tamamlayabileceğini göstermektedir. Benzer şekilde, diğer çalışmalar GAN mimarisinin tasarımında veya hiperparametre optimizasyonunda GA'ların kullanımıyla GAN eğitimine yeni bir boyut katmıştır (Konstantopoulou vd., 2024).

Evrimsel Hesaplama (EC), seçilim, çaprazlama ve mutasyon gibi doğal evrim ilkelerinden ilham alan meta-sezgisel algoritmalar ailesini kapsamaktadır (Costa vd., 2019). E-GAN, çekişmeli eğitim prosedürünü evrimsel bir optimizasyon problemi olarak yeniden kavramsallaştırarak GAN'lar için yeni bir çerçeve sunmaktadır. Bu paradigmada, diskriminatör dinamik olarak "çevre" işlevi görür, uyarlanabilir kayıp fonksiyonları sağlar ve bir generator popülasyonu bu değişen çevreye yanıt olarak yinelemeli olarak evrilir (C. Wang vd., 2019).

İşbirlikçi birlikte evrim algoritmalarının ilkelerinden ilham alan CDE-GAN, GAN'lardaki mod çökmesi ve eğitim kararsızlığı gibi kalıcı zorlukları doğrudan ele almaktadır (S. Chen vd., 2021). Temel yeniliği, hem generator(ler) hem de diskriminatör(ler) için "ikili evrimi" birleşik bir evrimsel çekişmeli çerçeveye dahil etmesidir.

2.4.2. Genetik Algoritmaların GAN Eğitimindeki Rolü

Çok Amaçlı Evrimsel Algoritmalar (MOEA), bir optimizasyon probleminde birden fazla amaç fonksiyonunu eşzamanlı olarak optimize etmeyi hedefler. Çekişmeli örnek oluşturma gibi senaryolarda tipik olarak saldırı başarısını maksimize etmek ve değişikliklerin algılanamaz veya minimal olmasını sağlamak gibi birden fazla kriter bulunmaktadır. MOEA'lar bu kriterler arasında Pareto optimallik kavramını kullanarak denge arar ve dominan olmayan çözümler kümesi (Pareto cephesi) elde etmeye çalışır. 2023-2025 döneminde çekişmeli saldırılarda ve GAN uygulamalarında MOEA kullanımı üzerine çeşitli çalışmalar yürütülmüştür.

Önemli çalışmalardan biri "Black-Box Sparse Çekişmeli Attack via Multi-Objective Optimization" (Williams ve K. Li, 2023) başlıklı araştırmadır. Bu yöntem, görüntü sınıflandırma modellerine karşı black-box saldırılarda pertürbasyon seyrekliğini ve saldırı başarısını eşzamanlı olarak optimize etmeyi hedeflemiştir. Williams ve Li, saldırıyı iki amaçlı bir optimizasyon problemi olarak yaklaşmışlardır: değiştirilen piksel sayısını ve pertürbasyon büyüklüğünü

minimize etmek, modelin yanlış sınıflandırma oranını (saldırı başarısı) maksimize etmek.

MOEA yaklaşımları sadece görüntü saldırılarında değil, farklı GAN uygulamalarında da kullanılmaktadır. Özellikle ağ güvenliği alanında, birden fazla amacın eşzamanlı optimizasyonu gerektiren durumlar için evrimsel çok amaçlı GAN'lar geliştirilmiştir. Örneğin, Xu vd. (Xu ve Cao, 2025) Research dergisinde yayınlanan çalışmalarında AME-GAN adını verdikleri bir yöntem sunmuşlardır. Adaptive Multiobjective Evolutionary GAN'ın kısaltması olan AME-GAN, Metaverse ortamında ağ saldırı tespit sistemi için tasarlanmıştır [6]. Bu sistemde, saldırı sınıflarından az örnek bulunan dengesiz veri setlerinde azınlık sınıf örnekleri üretmek için GAN'lar kullanılırken, tespit modelinin yapısı evrimsel ağ mimari arama algoritması ile optimize edilmiştir.

2.5. Çekişmeli Saldırı Performans Metrikleri ve Değerlendirme Kriterleri

Çekişmeli saldırıların performansını değerlendirirken birden fazla metrik kullanılmaktadır. Doğruluk Düşüşü, Saldırı Başarı Oranı, Çeşitlilik, Transfer Edilebilirlik, Pertürbasyon Büyüklüğü ve Gerçekçilik temel kriterler arasındadır. Doğruluk Düşüşü ve Saldırı Başarı Oranı: Çekişmeli saldırıların etkinliğini ölçmek için tipik olarak hedef model doğruluğundaki azalma veya saldırının başarı yüzdesi kullanılmaktadır . Pertürbasyon Büyüklüğü ve İzi: Bir çekişmeli saldırının başarılı olması kadar, yarattığı değişikliklerin insan algısı veya savunma sistemleri tarafından tespit edilmemesi de önemlidir. Bu nedenle L_0, L_2, L_∞ normları, değiştirilen piksel sayısı veya yapısal benzerlik indeksi (SSIM) gibi ölçüler kullanılmaktadır. Çeşitlilik: Özellikle üretici modeller söz konusu olduğunda, üretilen örneklerin çeşitliliği anahtar bir performans kriteridir. GAN'lar bazen mod çökmesi yaşayabilir ve monoton örnekler üretebilir (C. Wang vd., 2019). Transfer Edilebilirlik: Çekişmeli örneklerin bir modelden diğerine transfer olabilme kabiliyeti, gerçek dünyada saldırganlar için avantajlı bir durumdur (S. Chen vd., 2021). Gerçekçilik ve Tespit Edilebilirlik: Üretilen sahte veri veya çekişmeli örneklerin insanlar veya otomatik sistemler tarafından gerçek veriden ayırt edilememesi istenir. Bu amaçla çeşitli gerçekçilik ölçüleri kullanılmaktadır (S. Chen vd., 2021).

Bu çalışmada, literatürde tanıtılan yöntemlerden PyTorch tabanlı Üretken Çekişmeli Ağlar (GAN), Genetik Algoritmalar (GA) ve Çok Amaçlı Evrimsel

Algoritmalar (MOEA) yenilikçi bir yaklaşımla birleştirilerek çekişmeli örnek üretimi gerçekleştirilmiştir. Önerilen framework, Adam optimizer ile temel GAN eğitimini yürütürken, generator optimizasyonu için GA ve discriminator optimizasyonu için MOEA kullanarak gradyan tabanlı ve evrimsel optimizasyon yöntemlerinin avantajlarını birleştirmiştir.

2.6. Zorluklar ve Açık Problemler

GA-GAN kombinasyonu ve çekişmeli örnek oluşturma alanı hızlı ilerleme kaydetmesine rağmen çeşitli zorlukları barındırmaktadır. İlk olarak, hesaplama karmaşıklığı önemli bir problemdir. Genetik algoritmalar büyük popülasyonlar ve nesiller üzerinde çalıştıkları için yüksek hesaplama maliyetlerine sahip olabilirler. Bu, özellikle GAN'lar gibi karmaşık modellerle birleştiğinde daha da belirgin hale gelir (Costa vd., 2019). İkinci olarak, çok amaçlı optimizasyonda dengeleme zor olabilir. Birden fazla hedef kriter kullanıldığında, bu kriterlere verilen ağırlıklar veya öncelikler sonucu büyük ölçüde etkiler (C. Wang vd., 2019). Diğer bir konu evrimsel süreçlerin açıklanabilirliği ve kontrol edilebilirliğidir. GA'lar nihayetinde bir "evrim" simülasyonu yaparlar; özellikle çok karmaşık fitness fonksiyonlarında her adımda neden belirli bir çözüme yöneldikleri insanların anlaması için zor olabilir. Son olarak, evrensel bir değerlendirme protokolü oluşturma ihtiyacı da bir problemdir. Çekişmeli saldırılar için farklı makaleler farklı veri setleri, farklı ağ mimarileri ve farklı metrikler kullanmaktadır (I. J. Goodfellow, Shlens ve Szegedy, 2014).

3. TEORİK BİLGİ

Yapay zeka, bilgisayarların akıllı davranışlar sergilemesini sağlamayı hedefleyen bir disiplindir ve tasarlanan sistemler tipik olarak insan beyninin işleyişini taklit eder. Yapay zeka disiplini içinde birçok alt alan bulunmaktadır ve bunlardan biri makine öğrenmesidir. Makine öğrenmesi, belirli özellikler öğrenebilen veya bir problemi çözebilen algoritmalar kullanır. Sistemler verileri analiz eder ve modeller oluşturur. Makine öğrenmesi iki ana türe kategorize edilebilir: denetimli öğrenme ve denetimsiz öğrenme. (I. Goodfellow, Bengio, vd., 2016)

3.1. Makine Öğrenmesi Sınıflandırması

Denetimli Öğrenme etiketli bir veri seti üzerinde eğitim yapılmasını içerir ve dört ana kategori altında incelenebilir: sınıflandırma algoritmaları, derin öğrenme, derin transfer öğrenme ve regresyon yöntemleri.

Sınıflandırma algoritmaları etiketli çıktılara dayalı çıkarımlar yapar. Bu algoritmaların örnekleri arasında Çok Katmanlı İleri Beslemeli Sinir Ağları (MLFFNN) (Z. Wang vd., 1994), k-En Yakın Komşu Algoritması (kNN) (Guo vd., 2003), Destek Vektör Makineleri (SVM) (Hearst vd., 1998) ve Karar Ağaçları (DT) (Quinlan, 1986) bulunmaktadır.

Derin öğrenme genellikle görüntü işlemede kullanılır ve tüm öğrenme aşamaları otomatiktir. Derin öğrenme algoritmalarının örnekleri arasında Evrişimli Sinir Ağları (CNN) (LeCun, Bottou, vd., 2002), Tekrarlayan Sinir Ağları (RNN) (Elman, 1990), Uzun Kısa Süreli Bellek (LSTM) (Hochreiter ve Schmidhuber, 1997) ve Otokodlayıcılar (AE) (Rumelhart vd., 1986) bulunmaktadır.

Derin Transfer Öğrenme(Donahue vd., 2014), önceden eğitilmiş bir modelin ağırlık değerlerini yeni bir göreve uyarlama sürecidir. Bu yaklaşım, sınırlı veri setlerinde etkili sonuçlar elde etmeyi sağlar. Transfer öğrenme teknikleri arasında özellik çıkarımı (feature extraction), ince ayar (fine-tuning), ve alan adaptasyonu (domain adaptation) bulunmaktadır. Yaygın olarak kullanılan önceden eğitilmiş modeller arasında ResNet (He vd., 2016), VGG (Simonyan ve Zisserman, 2014), AlexNet (Krizhevsky vd., 2012) ve BERT (Devlin vd., 2019) yer almaktadır.

Regresyon algoritmaları, hedef değişkenin sürekli sayısal değerler aldığı durumlarda kullanılır (James vd., 2013). Regresyon algoritmalarının örnekleri arasında Doğrusal Regresyon, Polinom Regresyon, Ridge Regresyon, Lasso Regresyon ve ElasticNet Regresyon bulunmaktadır (Hastie vd., 2009).

Denetimli Öğrenme yöntemleri arasında ayrıca ensemble (topluluk) algoritmaları da yaygın olarak kullanılmaktadır. Bu yöntemler, birden fazla temel modelin (örneğin karar ağaçları) birleştirilmesiyle daha güçlü ve kararlı tahmin modelleri oluşturmayı hedefler (Zhou, 2012). Ensemble yöntemlerinin en bilinen örnekleri arasında Rastgele Orman (Random Forest) (çok sayıda karar ağacının bagging yöntemiyle birleştirilmesi) (Breiman, 2001), Gradient Boosting Makineleri (GBM) (hataları artımlı olarak düzelten boosting tabanlı bir yöntem) (Friedman, 2001) ve AdaBoost (zayıf sınıflandırıcıların ağırlıklı oylama ile birleştirilmesi) (Freund ve Schapire, 1997) sayılabilir. Bu yöntemler, özellikle aşırı öğrenme (overfitting) problemini azaltma ve model genelleme performansını artırma konusunda oldukça etkilidir.

Denetimsiz Öğrenme algoritmaları, etiketlenmemiş verilerden öğrenir ve bu yöntem doğrudan veriden yapısal bilgi çıkarır (Murphy, 2012). Kümeleme, boyut azaltma ve örüntü tanıma denetimsiz öğrenmenin ana kategorileridir. Kümeleme algoritmaları, verileri benzerlikler temelinde önceden belirlenmiş sayıda kümeye dağıtır (Jain vd., 1999). Kümeleme algoritmalarının örnekleri arasında K-Ortalamalar (K-Means), Hiyerarşik Kümeleme ve Kendini Organize Eden Haritalar (SOM) bulunmaktadır (Kohonen, 2012). Boyut azaltma algoritmaları, veri setindeki özellik sayısını azaltmayı hedefler ve Temel Bileşenler Analizi (PCA) bu yöntemin yaygın bir örneğidir (Jolliffe, 2011). Örüntü tanıma algoritmaları, verilerdeki gizli yapıları ve düzenlilikleri keşfetmeyi amaçlar. Bu algoritmalar, veri içindeki tekrarlayan yapıları, ilişkileri ve anomalileri tespit eder. Örüntü tanıma yöntemleri arasında Association Rules (Birliktelik Kuralları), Sequence Mining (Dizi Madenciliği) ve Anomaly Detection (Anomali Tespiti) algoritmaları bulunmaktadır (Bishop ve Nasrabadi, 2006).

3.2. Makine Öğrenmesi Süreci

Makine öğrenmesi algoritmalarının ana çerçevesi sırayla şu adımlardan oluşur: veri toplama, veri ön işleme, özellik çıkarma, eğitim ve makine öğrenmesi sınıflandırıcısının test edilmesi. (I. Goodfellow, Bengio, vd., 2016)

1. Veri Toplama: Makine öğrenmesi çalışmalarında ilk adım işlenecek verilerin toplanmasıdır. Bir problemi çözmede kullanılacak verinin ele alınan probleme ve uygulanacak uygulamaya uygun olması beklenir. Üzerinde çalışılacak veri iyileştirilebilir bir öğrenme sürecine sahip olmalı ve daha iyi yöntemler bulmak ve modeli geliştirmek için yüksek çeşitliliğe sahip olmalıdır.

2. Veri Ön İşleme: Toplanan verilerin işlemeye uygun hale getirilmesi için uygulanan süreçtir. Gerçekleştirilecek işlemler verileri temizlemek ve analiz için hazırlamaktır. Bu aşamada genellikle aykırı değerler tanımlanır ve çeşitli yöntemler kullanılarak normalleştirilir, ilgisiz özellikler kaldırılır. Enterpolasyon veya diğer teknikler aracılığıyla eksik değerler bulunur ve kararı olumsuz etkilemesi engellenir. Bu süreçten çıkan veriler daha güvenilir ve analitik uygulamalarda kullanım için daha uygun olacaktır.

3. Özellik Çıkarma: Bu aşamada problemi çözme üzerinde en yüksek etkiye sahip özellikler belirlenir. Tanımlanan bu özelliklerin makine öğrenmesinin anlayacağı en uygun formatta organize edilmesi ve sunulması gerekir.

4. Eğitim: Bu aşamada çözmeyi planladığımız problem için bir model oluşturulur. Bu model oluşturulurken önemli olarak tanımlanan özellikler ve varsa verilerin etiketleri kullanılır. Model, veri setinde eğitim için ayrılan veriler üzerinde çalışarak özellik kombinasyonlarına göre en uygun çözümün ne olması gerektiğini öğrenir.

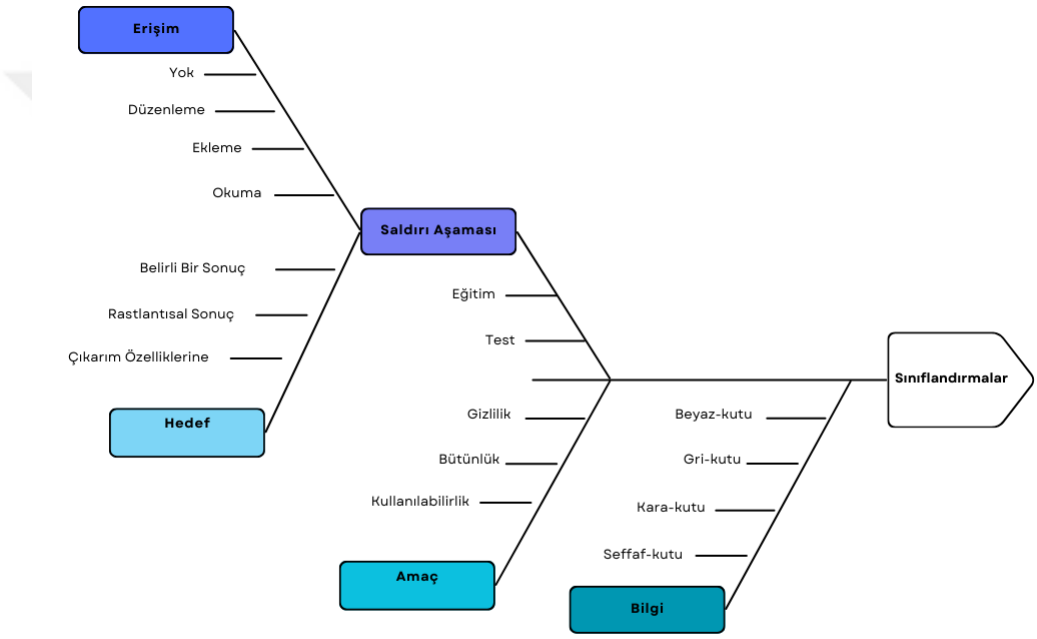
5. Test: Test aşamasında model, eğitim aşamasında kullanılmayan veri setinin bir bölümü ile test edilir. Amaç modelin performansını yeni ve daha önce görülmemiş veriler üzerinde objektif bir değerlendirmeye tabi tutmaktır. Modelin performansı genellikle doğruluk, kesinlik, hatırlama ve F1 Skoru gibi modelin tahmin başarısını değerlendiren metriklerle belirlenir.

3.3. Çekişmeli Saldırılar

Makine öğrenmesi modellerinin güvenliği ve çekişmeli saldırılara karşı güvenlik açıkları son derece önemli bir araştırma konusu haline gelmiştir. Özellikle görüntü tanıma, ses işleme ve metin analizinde derin öğrenme modellerinin insan gözüyle algılanamayan küçük pertürbasyonlarla kandırılabileceği kanıtlanmıştır(I. J. Goodfellow, Shlens ve Szegedy, 2014).Bu tür çekişmeli örnekler modelin çıktısını manipüle ederek hatalı sınıflandırmalara yol açabilir. Üstelik

otonom araçlardan yüz tanıma sistemlerine kadar birçok güvenlik açısından kritik uygulamada çekişmeli örneklerin varlığı ciddi riskler doğurmaktadır. Bu nedenle hem siber güvenlik uzmanları hem de makine öğrenmesi araştırmacılarının bu modellerin saldırılara karşı direncini artırmak için işbirliği yapması gerekmektedir.

Çekişmeli saldırılar detaylı olarak incelendiğinde birçok farklı özelliğe göre sınıflandırılabilir. Bu çalışmada bunları üç başlık altında analiz etmeyi seçtik. Şekil 3.1'te gösterildiği gibi amaç, bilgi seviyesi ve saldırı aşamasına göre incelenmektedirler.



Şekil 3.1 Saldırı Sınıflandırması

Çekişmeli saldırılarda, saldırganın sistem hakkında sahip olabileceği bilgi seviyesi dört kategori altında incelenir: kara kutu, gri kutu, beyaz kutu ve şeffaf kutu.

Kara-kutu: Saldırgan sistemin iç işleyişi hakkında hiçbir bilgiye sahip değildir. Sadece sistemin girdilerini ve çıktılarını gözlemleyebilir. Saldırgan sistemi dışarıdan inceleyerek olası güvenlik açıklarını belirlemeye çalışır.

Gri-kutu: Saldırgan bazı ara süreçler, girdiler ve çıktılar hakkında bilgiye sahiptir. Saldırgan sistemin iç işleyişi hakkında kısmi bilgiye sahiptir. Bu saldırganın hedefli saldırılar geliştirmesini sağlar.

Beyaz-kutu: Saldırgan sistemin tüm iç işleyişine tam erişime sahiptir. Saldırgan sistemin zayıflıklarını detaylı olarak inceleyebilir ve en etkili saldırıları tasarlayabilir.

Şeffaf-kutu: Beyaz-kutu bilgisine ek olarak saldırgan sistemin tasarımına, kodlarına, güvenlik önlemlerine ve diğer iç detaylara da erişebilir. Bu bilgi seviyesi saldırgana sistemin en derin güvenlik açıklarını belirleme ve en yıkıcı saldırıları gerçekleştirme fırsatı sağlar.

Bir siber saldırının amacı üç kategoride incelenir. Gizlilik sadece yetkili kişilerin verilere erişebilmesini sağlamayı ve saldırganların verilere erişimini engellemeyi hedefler. Kullanılabilirlik saldırılara rağmen bir sistem veya hizmetin sürekliliğinin sağlanmasını hedefler. Bütünlük verilerin bozulmamış ve zarar görmemiş kalmasını ve sistemin bazı veya tüm örnekler için doğru sonuçlar üretmesini sağlamayı hedefler.

Çekişmeli saldırının gerçekleştirildiği makine öğrenmesi aşaması, saldırıda kullanılacak yöntemler üzerinde en önemli etkiye sahiptir. Öğrenme süreci saldırılar açısından iki ana başlıkta incelenir: eğitim ve test. Bu çalışmada esas olarak eğitim süreci boyunca yapılan saldırılar üzerinde durulmuştur. Eğitim süreci iki farklı perspektiften incelenebilir: saldırganın veri setine erişim izninin seviyesi ve saldırganın hedeflediği model bileşeni.

Çekişmeli saldırılar saldırganın hangi hizmeti hedeflediğine göre incelendiğinde, saldırgan sadece belirli bir bölgede kalan sonuçları manipüle etmek isteyebilir, ayrım yapmadan rastgele saldırabilir veya sonuç yerine çıkarılan özelliklere saldırmak isteyebilir.

Saldırganların veri setine erişimi dört kategori altında incelenebilir: okuma, ekleme, değiştirme ve erişim yok. Değiştirme izinlerinde saldırgan veri setinde değişiklikler yapabilir; okuma seçeneğinde saldırgan sadece verileri görüntülemeye erişime sahiptir. Ekleme seçeneğinde veri setine eklemeler yapılabilir; erişimin olmadığı durumlarda ise veri setini bilmeden sadece model üzerinde saldırılar yapılır.

3.4. Çekişmeli Örneklerin Tanımı ve Önemi

Goodfellow vd. çekişmeli örnekleri matematiksel olarak formülize etmiş ve bu örneklerin hedef modelleri yanıltma mekanizmasını açıklamıştır. Literatürde çekişmeli örnekler, orijinal giriş verilerine imperceptible (algılanamaz) pertürbasyonlar eklenerek oluşturulan ve makine öğrenmesi modellerinin yanlış sınıflandırma yapmasına neden olan örnekler olarak tanımlanmaktadır (I. J. Goodfellow, Shlens ve Szegedy, 2014).

Bir sınıflandırıcının girişi x'e çekişmeli bir pertürbasyon η eklenirse, $x' = x+\eta$ gibi bir girişin orijinalden farklı tepki vermesi mantıklı değildir. $\|\eta\|_\infty < \epsilon$ koşulu sağlanırsa, girdilere farklı atamalar yapılmasını bekleriz. w'nin bir ağırlık vektörü olması ile x''in iç çarpımı aşağıdaki gibidir. Çekişmeli pertürbasyon η, aktivasyonda $w^\top \eta$'ya kadar bir artışa neden olur.

$$w^\top x' = w^\top x + w^\top \eta \tag{3.1}$$

(I. J. Goodfellow, Shlens ve Szegedy, 2014)

Ayrıca, çekişmeli örnekler sadece bir tehdit olarak değil, aynı zamanda modelleri daha sağlam hale getirme fırsatı olarak da görülebilir. Goodfellow vd. (I. J. Goodfellow, Shlens ve Szegedy, 2014) tarafından önerilen "çekişmeli eğitim" yaklaşımında, çekişmeli örneklerin eğitim sürecine dahil edilmesi modellerin bu tür örneklere karşı sağlamlığını artırabilir.

Saldırı Türleri

Literatürde sıklıkla kullanılan ve incelenen çeşitli çekişmeli saldırı yöntemleri bulunmaktadır. Bu saldırılar, beyaz-kutu, gri-kutu ve kara-kutu senaryolarına göre farklı stratejiler izlemekte ve aşağıda özetlenmektedir:

1. L-BFGS (Broyden-Fletcher-Goldfarb-Shanno): Beyaz-kutu saldırısıdır. Görüntülerde minimum sayıda değişiklik yaparak çekişmeli örnekler oluşturmayı hedefler. Ancak yöntem zaman alıcıdır ve büyük veri setlerinde etkisizdir. (I. J. Goodfellow, Shlens ve Szegedy, 2014)

2. FGSM (Fast Gradient Sign Method): Beyaz-kutu saldırısıdır. Gradyan işaretlerini kullanarak basit, hızlı ve etkili şekilde çekişmeli örnekler oluşturmayı hedefler. Pertürbasyonlar optimal değildir, bu nedenle başarı oranı düşüktür. (I. J. Goodfellow, Shlens ve Szegedy, 2014)

3. JSMA (Jacobian-based Saliency Map Attack): Beyaz-kutu saldırısıdır. Modelin karar verme sürecinde önemli olan özellikler seçilir ve önem sırasına göre azalan saliency'ler ile saldırı gerçekleştirilir. (Papernot, McDaniel, Jha, vd., 2016) FGSM'den daha fazla hesaplama gerektirir. (Macas vd., 2024)

4. DeepFool: Beyaz-kutu saldırısıdır. Modelin karar sınırları belirlenir ve girdi değerleri bu sınırları en küçük değerlerle geçmek için iteratif olarak manipüle edilir. Küçük değişiklikler yapar, ancak çekişmeli örnekler genellikle optimal seçime yakın değildir. (Moosavi-Dezfooli vd., 2016)

5. PGD (Projected Gradient Descent): Beyaz-kutu saldırısıdır. Projeksiyon bir vektörü tanımlı bir bölge içine dahil etmeye çalışır. Projeksiyon kısıtları içinde çekişmeli örnekler oluşturur. (Madry vd., 2017)

6. BIM (Basic Iterative Method): Beyaz-kutu saldırısıdır. FGSM'yi iteratif bir yaklaşımla uygular. FGSM'den daha iyi sonuçlar üretebilmesine rağmen, bazı uygulamalarda performans düşüklüğü yaşanabilir. (Kurakin vd., 2018)

7. C&W (Carlini and Wagner) Saldırısı: Beyaz-kutu algoritmasıdır. (Qiuyu vd., 2021) L-BFGS gibi çalışır ancak kutu kısıtları olmadan ve farklı amaç fonksiyonları ile daha verimli saldırı örnekleri oluşturur. C&W (Carlini & Wagner, 2017) FGSM, JSMA ve DeepFool'dan daha fazla hesaplama gerektirir. (Macas vd., 2024)

8. ZOO (Zeroth Order Optimization): Gri-kutu algoritmasıdır. Modelin çıktılarını analiz ederek çekişmeli örnekler oluşturur. Girişe ihtiyaç duymadan çalışabilmesi olumlu olmasına rağmen, bir modelin birçok kez çalıştırılması gerektiği için bazı senaryolarda performans düşüklüğü yaşanabilir. (P.-Y. Chen vd., 2017)

9. GAN (Generative Çekişmeli Networks): Kara-kutu algoritmasıdır. İki sinir ağının rekabeti yoluyla çekişmeli saldırılar oluşturulur. Bir sinir ağının üretici, diğerinin ayırıcı olarak hareket ettiği sinir ağları kullanılır. Eğitimde kullanılacak yeni örnekler üretebilir ancak çok yoğun hesaplamalar gerektirir. GAN (I. Goodfellow, Pouget-Abadie, vd., 2020)

3.5. Üretici Çekişmeli Ağlar(GAN)

Üretici Çekişmeli Ağlar (GAN'lar) Ian Goodfellow vd. tarafından 2014'te tanıtılan yenilikçi bir derin öğrenme yaklaşımıdır (I. J. Goodfellow, Pouget-Abadie, vd., 2014). GAN'lar veri üretimi için kullanılan güçlü bir model sınıfıdır ve

görüntü, ses ve metin gibi çeşitli alanlarda başarıyla uygulanmıştır. Çekişmeli saldırı alanında da GAN'lar, geleneksel gradyan tabanlı yöntemlerin ötesinde, daha karmaşık ve tespit edilmesi zor adversarial örnekler üretmek için kullanılmaktadır.

GAN'ların temel fikri iki sinir ağının (üretici ve ayırıcı) karşılıklı rekabet içinde eğitilmesidir. Üretici ağ rastgele gürültüden başlayarak gerçek veri dağılımına benzer örnekler üretmeye çalışırken, ayırıcı ağ gerçek örnekler ile üretici tarafından oluşturulan sahte örnekleri ayırt etmeye çalışır. Bu rekabet süreci oyun teorisi perspektifinden bir minimax oyunu olarak modellenebilir ve Nash dengesi ulaşıldığında üretici gerçek veri dağılımını mükemmel şekilde taklit edebilme kabiliyetine sahip olur.

3.5.1. GAN'ların Temel Yapısı ve Çalışma Prensipleri

GAN'lar iki temel bileşenden oluşur: üretici ve ayırıcı. Bu iki bileşen birbirleriyle rekabet eden ve eşzamanlı olarak eğitilen iki farklı sinir ağıdır.

Üretici ağ (G) latent uzaydan (tipik olarak normal dağılımlı rastgele gürültü) gerçek veri uzayına bir haritalama oluşturmaya çalışır. Üretici bu haritalama yoluyla gerçek veri dağılımına mümkün olduğunca benzer örnekler üretmeyi hedefler. Matematiksel olarak üretici $G : Z \rightarrow X$ fonksiyonunu öğrenmeye çalışır, burada Z latent uzay ve X gerçek veri uzayıdır.

Ayırıcı ağ (D) girdi olarak aldığı bir örneğin gerçek veri dağılımından mı yoksa üretici tarafından mı üretildiğini tahmin etmeye çalışır. Ayırıcı bir sınıflandırma görevi gerçekleştirir ve çıktısı girdinin gerçek olma olasılığını temsil eder. Matematiksel olarak ayırıcı $D : X \rightarrow [0, 1]$ fonksiyonunu öğrenir, burada $D(x) = 1$ girdinin gerçek olduğunu, $D(x) = 0$ ise girdinin sahte olduğunu gösterir.

GAN'ların eğitim süreci oyun teorisi çerçevesinde anlaşılabilir. Üretici ve ayırıcı sıfır toplamlı bir oyunda iki oyuncu gibi davranır. Ayırıcı doğru sınıflandırmalar yaparak kazanmaya çalışırken, üretici sahte örneklerin gerçek olduğuna ayırıcıyı ikna ederek kazanmaya çalışır. Bu rekabet üreticinin giderek daha gerçekçi örnekler üretmesine ve ayırıcının daha iyi bir sınıflandırıcı olmasına yol açar.

GAN eğitimi başarılı olduğunda, üretici gerçek veri dağılımına çok benzer bir dağılım üretmeyi öğrenir ve ayırıcı gerçek ve sahte örnekleri ayırt etmekte zorlanır

(tüm x için $D(x) \approx 0.5$). Bu durumda üretici tarafından üretilen örnekler gerçek örneklerden ayırt edilemez hale geldiği için ayırıcı rastgele tahmin etmekten daha iyi performans gösteremez.

3.5.2. GAN Mimarileri ve Varyasyonları

GAN'ların ilk ortaya çıkışından bu yana çeşitli GAN mimarileri ve varyasyonları önerilmiştir. Bu varyasyonlar GAN'ların eğitim kararlılığını artırmak, üretilen örneklerin kalitesini iyileştirmek veya belirli görevlere daha iyi uyum sağlamak için geliştirilmiştir.

Deep Convolutional GAN (DCGAN) (Salimans vd., 2016) GAN'ların eğitim kararlılığını artırmak ve daha yüksek kaliteli görüntüler üretmek için evrişimli sinir ağları (CNN) kullanan bir GAN varyasyonudur. DCGAN batch normalization, LeakyReLU aktivasyon fonksiyonu ve transposed convolution gibi çeşitli mimari iyileştirmeler içerir.

Conditional GAN (CGAN) (Mirza ve Osindero, 2014) üretim sürecini kontrol etmek için ek bilgiler (örn. sınıf etiketleri) kullanan bir GAN varyasyonudur. CGAN'da hem üretici hem de ayırıcı ek bilgileri girdi olarak alır ve üretici bu bilgileri kullanarak belirli bir sınıfa ait örnekler üretmeye çalışır.

Wasserstein GAN (WGAN) (Arjovsky vd., 2017) GAN eğitiminin kararlılığını artırmak için Wasserstein mesafesini (Earth Mover's Distance olarak da bilinir) kullanan bir GAN varyasyonudur. WGAN, GAN eğitiminde sıkça karşılaşılan gradyan kaybolması ve mod çökmesi gibi sorunları azaltabilir.

CycleGAN (Zhu vd., 2017) eşleştirilmiş eğitim verisi olmadan iki domain arasında çeviri için kullanılan bir GAN varyasyonudır. CycleGAN cycle consistency loss kullanarak çevirinin tutarlılığını sağlar. Örneğin CycleGAN at görüntülerini zebra görüntülerine çevirebilir ve tekrar at görüntülerine geri çevirebilir.

StackGAN (H. Zhang vd., 2017) metin açıklamalarından yüksek çözünürlüklü görüntüler üretmek için kullanılan bir GAN varyasyonudur. StackGAN iki aşamalı bir süreç kullanır: ilk aşama düşük çözünürlüklü bir taslak oluşturur, ikinci aşama bu taslağı kullanarak daha yüksek çözünürlüklü görüntü üretir.

Bu varyasyonlar GAN'ların farklı görevler ve veri türlerine uygulanabilirliğini genişletmiş ve GAN'ların performansını iyileştirmiştir. Ancak tüm bu varyasyonlarda da GAN eğitimi genellikle kararsız ve zor olmaya devam etmektedir.

3.5.3. GAN'ların Matematiksel Temelleri

Üretici Çekişmeli Ağlar (GAN'lar) iki sinir ağının, üretici ve ayırıcının, karşılıklı rekabet içinde eğitildiği bir üretici model sınıfıdır. GAN'ların matematiksel formülasyonu bir minimax oyunu olarak ifade edilebilir.

Üretici G latent uzaydan (Z) gerçek veri uzayına (X) bir haritalama oluşturmaya çalışır: G: Z $\rightarrow$ X. Ayırıcı D bir örneğin gerçek veri dağılımından mı yoksa üretici tarafından mı üretildiğini tahmin etmeye çalışır: D: X $\rightarrow$ [0,1].

Matematiksel olarak GAN eğitimi aşağıdaki minimax problemi olarak formüle edilebilir:

$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim p_{\text{data}}(x)}[\log D(x)] + \mathbb{E}_{z\sim p_z(z)}[\log(1 - D(G(z)))] \tag{3.2}$$

(I. J. Goodfellow, Pouget-Abadie, vd., 2014)

Burada G üretici ağı, D ayırıcı ağı, p_{data} gerçek veri dağılımını ve p_z latent uzay dağılımını temsil eder. Bu formülasyonda D gerçek örnekler için 1, sahte örnekler için 0 değeri üretmeye çalışırken, G, D'nin G(z) için 1 değeri üretmesine neden olacak örnekler oluşturmaya çalışır.

Teorik olarak bu oyunun Nash dengesi, üreticinin gerçek veri dağılımını mükemmel şekilde taklit ettiği ve ayırıcının gerçek ve sahte örnekleri ayırt edemediği (tüm x için D(x) = 0.5) bir durumdur. Ancak pratikte bu dengeye ulaşmak tipik olarak zordur ve GAN eğitimi kararsız olabilir. GAN'ların eğitim dinamiği üretici ve ayırıcı arasındaki güç dengesine bağlıdır. Eğer ayırıcı çok güçlüyse, üretici için faydalı gradyanlar sağlamayabilir; eğer üretici çok güçlüyse, ayırıcıyı kolayca kandırabilir ve yeterince öğrenemez. Bu nedenle GAN eğitiminde üretici ve ayırıcı arasında denge korunması önemlidir. Goodfellow vd. (I. J. Goodfellow, Shlens ve Szegedy, 2014) orijinal GAN formülasyonunda üretici

için alternatif bir kayıp fonksiyonu önermiştir:

$$L_G = -\mathbb{E}_{z\sim p_z(z)}[\log D(G(z))] \tag{3.3}$$

Bu, üreticinin $\log(1 - D(G(z)))$'yi minimize etmek yerine $\log D(G(z))$'yi maksimize etmeye çalıştığı anlamına gelir. Bu alternatif formülasyon özellikle eğitimin başında daha iyi gradyanlar sağlayabilir.

3.5.4. GAN Eğitimindeki Zorluklar

GAN eğitimi üretici ve ayırıcı arasındaki karmaşık dinamikler nedeniyle genellikle zor ve kararsızdır. GAN eğitiminde karşılaşılan ana zorluklar şunlardır:

Mod Çökmesi (Mode Collapse): Üretici veri dağılımının sadece küçük bir bölümünü taklit etmeyi öğrenir ve çeşitliliği kaybeder. Bu durumda üretici ayırıcıyı kandırmak için sadece birkaç "güvenli" örnek üretir ve veri dağılımının diğer bölgelerini keşfetmez. mod çökmesi GAN'ların çeşitli ve gerçekçi örnekler üretme kabiliyetini sınırlar.

Gradyan Kaybolması: Ayırıcı çok iyi hale geldiğinde, üretici için faydalı gradyanlar sağlamayabilir. Ayırıcı gerçek ve sahte örnekleri mükemmel şekilde ayırt edebildiğinde, üretici için gradyanlar sıfıra yaklaşır ve üretici öğrenemez. Bu problem özellikle eğitimin başında yaygındır, çünkü üretici henüz gerçekçi örnekler üretememiştir.

Kararsız Eğitim: GAN eğitimi üretici ve ayırıcı arasındaki güç dengesine bağlıdır ve bu denge kolayca bozulabilir. Eğer ayırıcı çok hızlı öğrenirse, üretici için faydalı gradyanlar sağlamayabilir; eğer üretici çok hızlı öğrenirse, ayırıcı geride kalabilir ve yeterince ayırt edici özellikler öğrenemeyebilir.

Hiperparametre Hassasiyeti: GAN eğitimi öğrenme oranı, batch boyutu ve ağ mimarisi gibi hiperparametrelere karşı oldukça hassastır. Uygun hiperparametrelerin seçilmesi başarılı GAN eğitimi için kritiktir, ancak bu tipik olarak deneme yanılma gerektirir.

Yakınsama Belirsizliği: GAN eğitiminin ne zaman yakınsadığını belirlemek zor olabilir. Üretici ve ayırıcı kayıpları eğitim ilerledikçe kararlı bir değere yakınsamayabilir, bu da eğitimi ne zaman durduracağına karar vermeyi zorlaştırır.

Bu zorlukları aşmak için çeşitli teknikler önerilmiştir:

DCGAN, WGAN, BEGAN(Berthelot vd., 2017) gibi gelişmiş GAN mimarileri eğitim kararlılığını artırmak için çeşitli mimari değişiklikler içerir. WGAN, LSGAN, BEGAN gibi modeller alternatif kayıp fonksiyonları kullanarak gradyan kaybolması ve mod çökmesi problemlerini azaltmaya çalışır.

Düzenlileştirme Teknikleri: Spectral normalization(Miyato vd., 2018), gradient penalty(Gulrajani vd., 2017), feature matching(Gulrajani vd., 2017) gibi teknikler GAN eğitimini düzenlileştirerek daha kararlı hale getirebilir.

Çift Zaman Ölçekli Güncelleme(Two-Timescale Update)(Heusel vd., 2017): Üretici ve ayırıcıyı farklı oranlarda güncelleme (örneğin ayırıcıyı daha sık güncelleme) güç dengesini korumaya yardımcı olabilir.

Mini Yığın Ayrımı(Minibatch Discrimination)(Gulrajani vd., 2017): Ayırıcının bir mini-batch içindeki örnekler arasındaki ilişkileri de değerlendirmesini sağlayarak mod çökmesi problemini azaltmaya çalışır.

Bu tekniklere rağmen GAN eğitimi hâlâ zor bir problem olarak kalmaktadır ve tamamen çözülmemiştir. Bu nedenle GAN eğitimi için alternatif optimizasyon yaklaşımları (örneğin evrimsel algoritmalar) araştırılmaktadır.(C. Wang vd., 2019)

3.5.5. GAN'ların Çekişmeli Örnek Üretimindeki Rolü

GAN'lar çekişmeli örnek üretimi için oldukça uygun bir çerçeve sağlar. Geleneksel çekişmeli örnek üretim yöntemleri (örn. FGSM, PGD) tipik olarak tek bir örneği pertürbe etmek için gradyan tabanlı yaklaşımlar kullanırken, GAN'lar daha genel bir yaklaşım sunar ve aynı anda birden fazla örnek üretebilir.

Xiao vd. (Xiao vd., 2018) çekişmeli örnek üretimi için GAN'ları kullanan AdvGAN adlı bir çerçeve geliştirmiştir. AdvGAN bir üretici ağ kullanarak orijinal örneklere pertürbasyonlar ekleyerek çekişmeli örnekler oluşturur. Üretici üç temel amacı optimize etmeye çalışır: (Domingos, 2012) üretilen çekişmeli örnekler hedef modeli kandırır, (Wiles, 2023) çekişmeli örnekler orijinal örneklere görsel olarak benzer, ve (Rosenberg vd., 2021) çekişmeli örnekler gerçekçi görünür.

GAN tabanlı çekişmeli örnek üretimi geleneksel yöntemlere kıyasla önemli avantajlara sahiptir. İlk olarak, GAN'lar sadece mevcut örneklere pertürbasyon eklemekle kalmaz, aynı zamanda yeni çekişmeli örnekler de üretebilir. İkinci olarak, GAN'lar üretilen çekişmeli örneklerin gerçekçiliğini artırabilir çünkü ayırıcı

ağ gerçek ve sahte örnekleri ayırt etmeye çalışarak üreticiyi daha gerçekçi örnekler oluşturmaya teşvik eder. Üçüncü olarak, GAN'lar transfer edilebilirliği iyileştirebilir, yani bir model için üretilen çekişmeli örnekler diğer modeller için de etkili olabilir.

Song vd. (Song vd., 2018) GAN'lar kullanarak "kısıtsız" çekişmeli örnekleri başarıyla üreten Unrestricted Çekişmeli Examples (UAE) adlı bir yaklaşım önermiştir. Geleneksel çekişmeli örnekler tipik olarak orijinal örneklere küçük pertürbasyonlar ekleyerek oluşturulurken, UAE yaklaşımı herhangi bir Lp-norm kısıtı olmadan tamamen farklı görünebilen ancak anlamsal olarak özdeş çekişmeli örnekler üretmeyi hedefler.

Liu vd. (Qiu vd., 2020) GAN'lar kullanarak "anlamsal" çekişmeli örnekler üretmeyi önermiştir. Bu yaklaşımda GAN'lar anlamsal olarak anlamlı değişiklikler yaparak (örn. bir yüzün ifadesini değiştirme) çekişmeli örnekler oluşturur. Bu tür anlamsal çekişmeli örnekler geleneksel yöntemlere kıyasla daha az tespit edilebilir ve daha doğal görünebilir.

Özetle, GAN'lar çekişmeli örnek üretimi için güçlü ve esnek bir çerçeve sağlar. GAN'ların gerçekçi ve çeşitli örnekler üretebilme kabiliyeti çekişmeli saldırıların etkinliğini artırabilir ve daha sağlam makine öğrenmesi modellerinin geliştirilmesine katkıda bulunabilir.(Xiao vd., 2018)

3.6. Evrimsel Hesaplama

3.6.1. Genetik Algoritmalar

Genetik Algoritmalar (GA) doğal seçilim ve genetik prensiplerden ilham alan bir optimizasyon tekniğidir. GA'lar çözüm adaylarından oluşan bir popülasyon kullanır ve bu popülasyonu seçilim, çaprazlama ve mutasyon operatörleri aracılığıyla evrimleştirir. Bu çalışmada GA'lar, çekişmeli saldırı üretiminde geleneksel gradyan tabanlı optimizasyonun kısıtlarını bertaraf etmek amacıyla kullanılmıştır. Özellikle GAN mimarisinde generator ağının parametrelerini optimize etmek için GA yaklaşımı benimsenmiş, böylece yerel optimal noktalarında sıkışma problemi çözülmeye çalışılmıştır. GA'nın popülasyon tabanlı doğası, adversarial örneklerin çeşitliliğini artırırken, evrimsel operatörleri sayesinde farklı pertürbasyon stratejilerinin keşfedilmesine olanak sağlamıştır. Bu bölümde GA'ların temel prensiplerine, çalışmada kullanılan spesifik operatörlere,

parametre seçim kriterlerine ve GAN mimarisiyle entegrasyonuna ilişkin detaylı açıklamalar sunulacaktır. GA'ların temel adımları şunlardır:

Başlangıç Popülasyonu: Çözüm adaylarından (bireyler) oluşan rastgele bir başlangıç popülasyonu oluşturulur. Her birey probleme özgü bir şekilde kodlanmış bir "kromozom" ile temsil edilir. Kromozom tipik olarak bit dizisi, gerçel sayı dizisi veya başka bir veri yapısı olabilir.

Fitness Değerlendirmesi: Her birey için problemin amaçlarına ne kadar uyduğunu ölçen bir fitness değeri hesaplanır. Fitness fonksiyonu problemin doğasına bağlıdır ve GA'nın başarısı için kritiktir.

Seçilim: Daha yüksek fitness değerlerine sahip bireyler sonraki nesil için ebeveyn olarak seçilme olasılığı daha yüksektir. Yaygın seçilim yöntemleri arasında rulet çarkı seçilimi, turnuva seçilimi ve sıralama seçilimi bulunur.

Çaprazlama: Seçilen ebeveynler genetik materyallerini birleştirerek çocuk üretir. Çaprazlama ebeveynlerin kromozomlarını belirli noktalardan keserek kesilen parçaları değiştirme ile gerçekleştirilir. Yaygın çaprazlama operatörleri arasında tek nokta çaprazlama, iki nokta çaprazlama ve tekdüze çaprazlama bulunur.

Mutasyon: Çaprazlamadan kaynaklanan çocukların kromozomlarında küçük rastgele değişiklikler yapılır. Mutasyon popülasyonun genetik çeşitliliğini korumaya ve yeni çözüm alanlarını keşfetmeye yardımcı olur. Mutasyon oranı tipik olarak düşüktür, çünkü yüksek mutasyon oranları iyi çözümlerin bozulmasına neden olabilir.

Değiştirme: Yeni nesil ebeveyn popülasyonunun yerini alır. Bazı GA varyasyonlarında elitizm olarak bilinen bir strateji kullanılır; bu stratejide en iyi bireyler doğrudan sonraki nesle aktarılır, bu da iyi çözümlerin korunmasını sağlar.

Durma Kriteri: Belirli bir nesil sayısına ulaşıldığında, yeterince iyi bir çözüm bulunduğunda veya başka bir durma kriteri sağlandığında algoritma sonlanır.

GA'lar özellikle gradyan bilgisinin mevcut olmadığı karmaşık, çok modlu optimizasyon problemleri için uygundur. Ana avantajları incelendiğinde, kayıp fonksiyonunun gradyanını gerektirmediği için türevi alınamayan fonksiyonlar veya gradyan hesaplamanın zor olduğu problemler için ideal bir optimizasyon

yaklaşımı sunmaktadır. Bu gradyan bağımsızlığı özelliği, GA'ların çözüm uzayında paralel arama gerçekleştirmesine olanak tanırken, aynı zamanda yerel optimalerden kaçınarak global arama kabiliyeti kazanmasını sağlamaktadır. Özellikle birden fazla optimuma sahip çok modlu problemlerde, tek bir çözüm yerine birden fazla optimal veya optimal yakını çözümü eşzamanlı olarak bulabilme avantajına sahiptir. Ayrıca GA'ların esneklik özelliği sayesinde çeşitli problem türlerine ve farklı kodlama şemalarına kolaylıkla uyarlanabilmesi, bu algoritmaları geniş bir uygulama yelpazesinde kullanılabilir kılmaktadır.

Ancak GA'ların bazı dezavantajları da vardır. Her nesilde birden fazla bireyi değerlendirmeleri gerektiği için hesaplama açısından pahalı olabilirler ve bu durum özellikle büyük problemlerde önemli bir maliyet yaratabilir. Bunun yanı sıra, GA'ların performansı popülasyon boyutu, çaprazlama ve mutasyon oranları gibi hiperparametrelere oldukça hassastır ve bu parametrelerin optimal değerlerinin belirlenmesi zorlu bir süreç olabilmektedir. Özellikle çözüm uzayının büyük olduğu durumlarda, GA'lar gradyan tabanlı yöntemlere kıyasla daha yavaş yakınsama sergileyebilir ve bu da zaman açısından dezavantaj yaratabilir. Son olarak, GA'lar yakınsama veya optima bulma konusunda gradyan tabanlı yöntemler gibi güçlü teorik garantilere sahip olmadığından, performanslarının önceden tahmin edilmesi ve garanti altına alınması mümkün değildir.

GA'lar üretici ağın parametrelerini optimize etmek için GAN eğitiminde kullanılabilir. Bu yaklaşımda her birey üretici ağın bir parametre setini temsil eder ve bireylerin fitness değeri ayırıcı ağın çıktısına dayalı olarak hesaplanır. GA'ların GAN eğitiminde gradyan tabanlı yöntemlere kıyasla avantajları ve dezavantajları vardır ve bu araştırma topluluğunda aktif bir tartışma konusu olmaya devam etmektedir.

3.6.2. Çok Amaçlı Evrimsel Algoritmalar

Çok Amaçlı Evrimsel Algoritmalar (MOEA) potansiyel olarak çelişen birden fazla amacı eşzamanlı olarak optimize etmeye çalışan evrimsel hesaplama teknikleridir. Tek amaçlı optimizasyondan farklı olarak, çok amaçlı optimizasyon tek bir optimal çözüm yerine bir dizi Pareto-optimal çözüm (Pareto cephesi olarak bilinir) üretir. Bu çalışmada MOEA yaklaşımı, çekişmeli saldırı üretiminde çoklu performans kriterlerini eşzamanlı olarak optimize etmek amacıyla discriminator ağının eğitiminde kullanılmıştır. Saldırı etkinliği, pertürbasyon çeşitliliği ve gizlilik korunumu gibi çelişebilen amaçlar arasında dengeli çözümler elde etmek için MOEA'nın Pareto dominance prensibi benimsenmiştir. Bu yaklaşım, tek bir metriği

maksimize etmek yerine, çok boyutlu performans uzayında optimal trade-off noktalarının belirlenmesini mümkün kılmıştır.

Bu bölümde MOEA'ların temel prensiplerine, Pareto optimal çözüm kavramına, çalışmada kullanılan spesifik algoritmalara ve discriminator optimizasyonundaki rolüne ilişkin detaylı açıklamalar sunulacaktır.

MOEA'ların temel prensibi Pareto dominansı kavramına dayanır. Bir çözüm başka bir çözümü domine eder eğer en az bir amaçta daha iyi ve tüm diğer amaçlarda en az eşit kadar iyiyse. Pareto-optimal bir çözüm hiçbir başka çözüm tarafından domine edilmeyen bir çözümdür. Başka bir deyişle, Pareto-optimal bir çözüm başka bir amacı iyileştirmek için bir amacı feda etmelidir.

MOEA'ların temel adımları Genetik Algoritmalara benzer olmakla birlikte, birden fazla amacın değerlendirilmesi ve Pareto dominansına dayalı seçilim mekanizmalarını içermektedir. Süreç, çözüm adaylarından oluşan rastgele bir başlangıç popülasyonunun oluşturulmasıyla başlar ve her birey birden fazla amaç fonksiyonu açısından değerlendirilerek bir amaç vektörü elde edilir. Pareto tabanlı seçilim aşamasında bireyler, Pareto dominansı ve çeşitlilik gibi kriterlere göre seçilirken, non-dominated sorting, crowding distance ve strength Pareto gibi yaygın seçilim mekanizmaları kullanılmaktadır. Seçilen bireyler çaprazlama ve mutasyon operatörleri aracılığıyla yeni bireyler üretir ve bu süreçte Pareto-optimal çözümler tipik olarak bir arşivde saklanarak sonraki nesillere aktarılır. Algoritma, belirli bir nesil sayısına ulaşıldığında veya Pareto cephesi yeterince kararlı hale geldiğinde sonlanır ve böylece çok amaçlı optimizasyon probleminin çözümü tamamlanmış olur.

MOEA'ların en yaygın örnekleri arasında NSGA-II (Non-dominated Sorting Genetic Algorithm II), SPEA2 (Strength Pareto Evolutionary Algorithm 2), MOEA/D (Multi-Objective Evolutionary Algorithm based on Decomposition) ve NSGA-III bulunur.

Deb vd. (Deb vd., 2002) tarafından önerilen NSGA-II, MOEA alanında en yaygın kullanılan algoritmalardan biridir. NSGA-II hızlı non-dominated sorting algoritması, crowding distance hesaplama ve elitizm stratejilerini içerir. NSGA-II'nin ana avantajları hızlı hesaplama süresi, iyi çeşitlendirilmiş Pareto cephesi ve elitizm yoluyla iyi çözümlerin korunmasıdır.

MOEA'ların ana avantajları çok boyutlu optimizasyon problemlerinde kendini göstermektedir. Bu algoritmalar birden fazla amacı eşzamanlı olarak optimize ederek, tek amaçlı optimizasyondaki kaçınılmaz amaçlar arası feda etme problemini çözmekte ve karar vericilere kapsamlı bir perspektif sunmaktadır. Tek bir çözüm yerine bir dizi Pareto-optimal çözüm üretmeleri sayesinde, karar vericilere farklı feda etmeler arasında seçim yapma esnekliği sağlarlar ve böylece problem çözümünde daha geniş bir seçenek yelpazesi sunarlar. Ayrıca, amaç fonksiyonlarının gradyanlarını gerektirmemeleri nedeniyle türevi alınamayan fonksiyonlar veya gradyan hesaplamanın zor olduğu karmaşık problemler için ideal bir yaklaşım oluştururlar. Son olarak, çeşitli mekanizmalar kullanarak Pareto cephesi boyunca çözümlerin çeşitliliğini korumaları, algoritmanın kapsamlı ve dengeli sonuçlar üretmesini sağlamaktadır.

MOEA'ların ana dezavantajları ise çeşitli pratik zorluklardan kaynaklanmaktadır. Her nesilde birden fazla bireyi ve birden fazla amaç fonksiyonunu değerlendirmeleri gerektiği için hesaplama açısından oldukça pahalı olabilirler ve bu durum özellikle büyük ölçekli problemlerde önemli bir kısıt yaratabilir. Amaç sayısı arttıkça Pareto dominansının ayırt etme gücü azalması (many-objective optimization problemi olarak bilinir) algoritma performansını olumsuz etkileyebilir ve çözüm kalitesinde düşüşe neden olabilir. Bunlara ek olarak, MOEA'ların performansı popülasyon boyutu, çaprazlama ve mutasyon oranları gibi hiperparametrelere oldukça hassastır ve bu parametrelerin optimal değerlerinin belirlenmesi karmaşık ve zaman alıcı bir süreç olabilmektedir.

MOEA'lar üretici ağı birden fazla amaç açısından optimize etmek için GAN eğitiminde kullanılabilir. Örneğin üretici hem ayırıcıyı kandırmak hem de gerçekçi ve çeşitli örnekler üretmek için optimize edilebilir. Bu yaklaşım geleneksel GAN eğitimine kıyasla daha kararlı bir eğitim süreci ve daha yüksek kaliteli örnekler sağlayabilir.

3.7. GAN Eğitiminde Optimizasyon Teknikleri

Generatif Çekişmeli Ağların (GAN) eğitimi, üretici ve ayırıcı ağlar arasındaki dinamik etkileşim nedeniyle karmaşık bir süreçtir I. J. Goodfellow, Pouget-Abadie, vd., 2014. Bu bölümde, GAN eğitiminde kullanılan temel optimizasyon teknikleri ve karşılaştırmalı analizleri sunulmaktadır.

3.7.1. Gradyan Tabanlı Yöntemler

GAN eğitiminde en yaygın kullanılan optimizasyon teknikleri:

3.7.1.1. Stokastik Gradyan İnişi (SGD)

$$\theta_{t+1} = \theta_t - \eta_t \nabla_\theta f(\theta_t) \tag{3.4}$$

(LeCun, Bottou, vd., 2002)

Bu denklemde:

- θ_t: t zamanındaki model parametreleri (ağırlıklar ve bias değerleri)
- θ_{t+1}: Güncellenmiş model parametreleri
- η_t: Öğrenme oranı (learning rate) - parametre güncellemelerinin büyüklüğünü kontrol eder
- $\nabla_\theta f(\theta_t)$: Kayıp fonksiyonunun parametrelere göre gradyanı - en dik düşüş yönünü gösterir
- $f(\theta_t)$: Kayıp fonksiyonu (üretici için: $-\log D(G(z))$, ayırıcı için: $-\log D(x) - \log(1 - D(G(z)))$)

SGD'nin GAN eğitimindeki temel avantajı hesaplasal verimliliğidir LeCun, Bottou, vd., 2002. Ancak, η_t öğrenme oranının uygun şekilde ayarlanması gerekmektedir. Çok yüksek değerler sapmaya, çok düşük değerler ise yavaş yakınsamaya neden olabilir.

3.7.1.2. Adam Optimizer

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t \tag{3.5}$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2 \tag{3.6}$$

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t} \tag{3.7}$$

$$\hat{v}_t = \frac{v_t}{1 - \beta_2^t} \tag{3.8}$$

$$\theta_{t+1} = \theta_t - \eta \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon} \tag{3.9}$$

(Kingma ve Ba, 2014)

Adam optimizatöründe kullanılan parametreler:

- $g_t = \nabla_\theta f(\theta_t)$: t zamanındaki gradyan vektörü
- m_t: Gradyanların exponential moving average'ı (momentum), gradyanların yönünü stabilize eder
- v_t: Gradyanların karelerinin exponential moving average'ı, adaptif öğrenme oranı sağlar
- β_1: Birinci moment decay oranı momentum hesaplamalarını kontrol eder
- β_2: İkinci moment decay oranı, varyans hesaplamalarını kontrol eder
- $\hat{m}_t, \hat{v}_t$: Bias-corrected moment estimatları - başlangıç bias'ını düzeltir
- η: Öğrenme oranı, SGD'ye göre daha az ayarlama gerektirir
- ϵ: Sayısal kararlılık için küçük sabit - sıfıra bölmeyi önler

Adam'ın GAN eğitimindeki performansı, özellikle öğrenme oranı ayar gereksinimini azaltması ve adaptif gradyan ölçeklendirmesi nedeniyle dikkat çekicidir Kingma ve Ba, 2014. Sparse gradyanlarda ve non-stationary ortamlarda üstün performans gösterir.

3.7.2. Evrimsel Optimizasyon Yöntemleri

Evrimsel algoritmalarla çekişmeli saldırı üretme, yani bir modelin güvenlik açıklarını bulan örnekleri evrimsel yöntemlerle optimize etme önemli bir araştırma alanı olarak ortaya çıkmıştır. Evrimsel yöntemler özellikle kara-kutu saldırılar (modelin iç yapısını bilmeden sadece çıkış skorlarına dayalı yapılan saldırılar) için uygun kabul edilmektedir. Gradyan tabanlı klasik saldırılar hedef modelin türev bilgisini gerektirirken, evrimsel algoritmalar sadece modele sorgu erişimi (sadece çıktıları) ihtiyaç duyar ve türev bilgisi kullanmadan arama yapabilir. Bu nedenle evrimsel yaklaşımlar saldırı yüzeyini geniş çapta arayarak ve popülasyon tabanlı global optimizasyon gerçekleştirerek güçlü çekişmeli örnekler bulabilir. Sonuç olarak GAN'lar gibi üretici modellerin kabiliyetlerini genetik algoritmaların arama kabiliyetleri ile birleştiren yaklaşımlar hem sentetik veri üretimi hem de çekişmeli saldırı tasarımında yeni ufuklar açmaktadır.

3.7.2.1. Genetik Algoritmalar

Genetik algoritmalar, GAN eğitiminde aşağıdaki operatörlerle uygulanır:

Seçilim (Selection):

$$p_i = \frac{f_i}{\sum_{j=1}^{N} f_j} \tag{3.10}$$

(Goldberg, 1989)

Bu denklemde:

- p_i: i'inci bireyin seçilme olasılığı
- f_i: i'inci bireyin fitness değeri (GAN için: generator loss veya discriminator accuracy)
- N: Popülasyon büyüklüğü
- Bu yöntem fitness-proportionate selection (roulette wheel) olarak bilinir.

Çaprazlama (Crossover):

$$x_{\text{child}} = \alpha x_{\text{parent1}} + (1 - \alpha) x_{\text{parent2}} \tag{3.11}$$

(Goldberg, 1989)

Bu işlemde:

- $x_{\text{parent1}}, x_{\text{parent2}}$: Seçilen ebeveyn bireylerinin gen vektörleri (GAN parametreleri)
- $\alpha \in [0, 1]$: Çaprazlama katsayısı - ebeveynlerden gelen katkının oranını belirler
- x_{child}: Oluşturulan yavru bireyin gen vektörü
- Bu linear combination crossover olarak adlandırılır

Mutasyon (Mutation):

$$x' = x + \mathcal{N}(0, \sigma^2) \tag{3.12}$$

(Goldberg, 1989)

Mutasyon parametreleri:

- x: Mutasyona uğrayacak birey (GAN ağırlık vektörü)
- x': Mutasyon sonrası yeni birey
- $\mathcal{N}(0, \sigma^2)$: Ortalaması 0, varyansı σ^2 olan normal dağılım
- σ: Mutasyon şiddeti - çeşitliliği kontrol eder
- Gaussian mutation yöntemi kullanılmıştır

3.7.2.2. Çok Amaçlı Optimizasyon

Pareto optimal çözümler:

$$\text{Minimize } F(x) = [f_1(x), f_2(x), ..., f_k(x)]^T \tag{3.13}$$

Çok amaçlı optimizasyon formülasyonunda:

- $F(x)$: Çok boyutlu amaç fonksiyonu vektörü
- $f_i(x)$: i'inci amaç fonksiyonu (örneğin: f_1 = generator loss, f_2 = discriminator loss, f_3 = mode collapse metriği)
- k: Toplam amaç fonksiyonu sayısı
- x: Karar değişkenleri vektörü (GAN parametreleri)
- Pareto frontu: Hiçbir amacı diğerini kötüleştirmeden iyileştirilemeyen çözümler kümesi

NSGA-II (Non-dominated Sorting Genetic Algorithm II) algoritması Deb vd., 2002 GAN eğitiminde başarıyla uygulanmıştır. Bu algoritma:

- Fast non-dominated sorting ile Pareto katmanlarını belirler
- Crowding distance ile çözüm çeşitliliğini korur
- Elite preservation ile en iyi çözümleri saklar
- GAN'larda generator ve discriminator arasındaki dengeyi optimize etmek için idealdir

4. YÖNTEM

Bu çalışmada, siber güvenlik alanında çekişmeli saldırılar üretmek amacıyla PyTorch tabanlı bir Generative Adversarial Network (GAN) mimarisi geliştirilmiştir. Önerilen yaklaşım, geleneksel Adam optimizasyonu, genetik algoritma (GA), çok amaçlı evrimsel algoritma (MOEA) ve önerilen metodolojilerin sistematik karşılaştırmasını içermektedir.

4.1. Veri Seti ve Ön İşleme

Deneysel çalışmalarda kullanılan UNSW-NB15 veri seti, 175.341 eğitim ve 82.332 test örneği olmak üzere toplam 257.673 ağ trafiği kaydından oluşmaktadır. Veri seti; Normal, Generic, Exploits, Fuzzers, DoS, Reconnaissance, Analysis, Backdoor, Shellcode ve Worms olmak üzere 10 farklı saldırı kategorisini kapsamaktadır. (Moustafa ve Slay, 2015) Ön işleme aşamasında kategorik değişkenler (proto, service, state) için One-Hot Encoding, sayısal değişkenler için ise StandardScaler normalizasyonu uygulanmıştır. İşlemler sonucunda orijinal 49 özellikten 195 boyutlu özellik uzayı elde edilmiş olup (kategorik değişkenlerin One-Hot Encoding ile genişlemesi sonucu), tüm veriler PyTorch tensor formatına dönüştürülerek CUDA desteği ile GPU üzerinde işlenmiştir. CPU üzerinde gerçekleştirilen hesaplamaların 24 saati aşan süre gerektirmesi nedeniyle, hesaplama verimliliğini artırmak amacıyla GPU tabanlı paralel işleme mimarisine geçiş yapılmıştır. Geliştirilen kod yerel ortamda başarıyla çalıştırılmış ve test edilmiştir.

4.2. Hedef Model Mimarisi

Çalışmada referans model olarak 100 karar ağacından oluşan Random Forest sınıflandırıcısı kullanılmıştır. Bu model, saldırı kategorilerini sınıflandırmak üzere eğitilmiş olup, adversarial saldırıların etkinliğinin değerlendirilmesinde temel karşılaştırma metriği olarak hizmet etmiştir.

4.3. Önerilen GAN Mimarisi

4.3.1. Temel GAN Yapısı

Generator Ağı latent uzaydan (100 boyut) adversarial pertürbasyon üretmek üzere tasarlanmıştır. Ağ mimarisi: Giriş katmanı: 100 boyutlu latent vektör Gizli katmanlar: 128 $\rightarrow$ 256 $\rightarrow$ 512 nöron (LeakyReLU aktivasyon, %30 dropout) Çıkış katmanı: 195 boyutlu pertürbasyon vektörü (lineer aktivasyon)

Discriminator Ağı ise gerçek ve üretilmiş örnekleri ayırt etmekle görevlidir: Giriş katmanı: 195 boyutlu özellik vektörü Gizli katmanlar: 512 $\rightarrow$ 256 $\rightarrow$ 128 nöron (LeakyReLU aktivasyon, %30 dropout) Çıkış katmanı: 1 nöron (Sigmoid aktivasyon)

4.3.2. Standartlaştırılmış Genetik Operatörler

Tüm evrimsel yaklaşımlarda tutarlılık sağlamak amacıyla aşağıdaki operatörler kullanılmıştır: Çaprazlama: Uniform crossover (%90 oran) Mutasyon: Gaussian mutasyon (%5 oran, $\sigma = 0.2$) Seçim: Turnuva seçimi (boyut=3) Değiştirme: Elitist replacement (%20 elitizm)

4.3.3. L2 Norm Kısıtlaması

Pertürbasyonların gerçekçi sınırlar içinde kalmasını sağlamak amacıyla $\epsilon = 0.1$ L2 norm kısıtlaması uygulanmıştır.

4.3.4. Karşılaştırılan Yöntemler

4.3.4.1. Adam Tabanlı GAN (Baseline)

Geleneksel GAN eğitimi için Adam optimizasyonu (lr=0.0002, β_1=0.5) kullanılmış olup, 500 epoch boyunca eğitim gerçekleştirilmiştir.

4.3.4.2. Genetik Algoritma Tabanlı GAN

30 bireylik populasyon 50 epoch evrimsel optimizasyon Fitness fonksiyonu: Hedef modelin doğruluk kaybı

4.3.4.3. Çok Amaçlı Evrimsel GAN (MOEA)

NSGA-II algoritması kullanılarak generator için üç amaç optimize edilmiştir:

1. Saldırı etkinliği ($accuracy_drop$)

2. Pertürbasyon çeşitliliği
3. Gizlilik skoru

4.3.4.4. Önerilen GAN

Yöntem, 100 epoch Adam ön-eğitimi ve 50 epoch hibrit optimizasyon (Generator GA + Discriminator MOEA) içermektedir.

4.4. Değerlendirme Metrikleri

Bu bölümde, PyTorch GAN tabanlı ağ saldırısı karşılaştırması çalışmasında kullanılan değerlendirme metriklerinin detaylı açıklamaları ve matematiksel formülleri sunulmaktadır.

4.4.1. Model Doğruluk Karşılaştırması

4.4.1.1. Doğruluk (Accuracy)

Ağ saldırısı tespit modelinin genel performansını ölçen temel bir metriktir:

$$\text{Doğruluk} = \frac{TP + TN}{TP + TN + FP + FN} \tag{4.1}$$

Burada TP (gerçek pozitif - saldırı doğru tespit), TN (gerçek negatif - normal trafik doğru tespit), FP (yanlış pozitif - normal trafiği saldırı olarak tespit) ve FN (yanlış negatif - saldırıyı normal olarak tespit) değerlerini temsil etmektedir (Sokolova ve Lapalme, 2009).

4.4.2. Saldırı Etkinliği Metrikleri

4.4.2.1. L2 Norm - Trafik Değişim Büyüklüğü

L2 norm, orijinal ağ trafiği ile GAN tarafından üretilen adversarial trafik arasındaki Öklid mesafesini ölçer:

$$L_2 = \sqrt{\sum_{i=1}^{n} (f_i - f_i')^2} \tag{4.2}$$

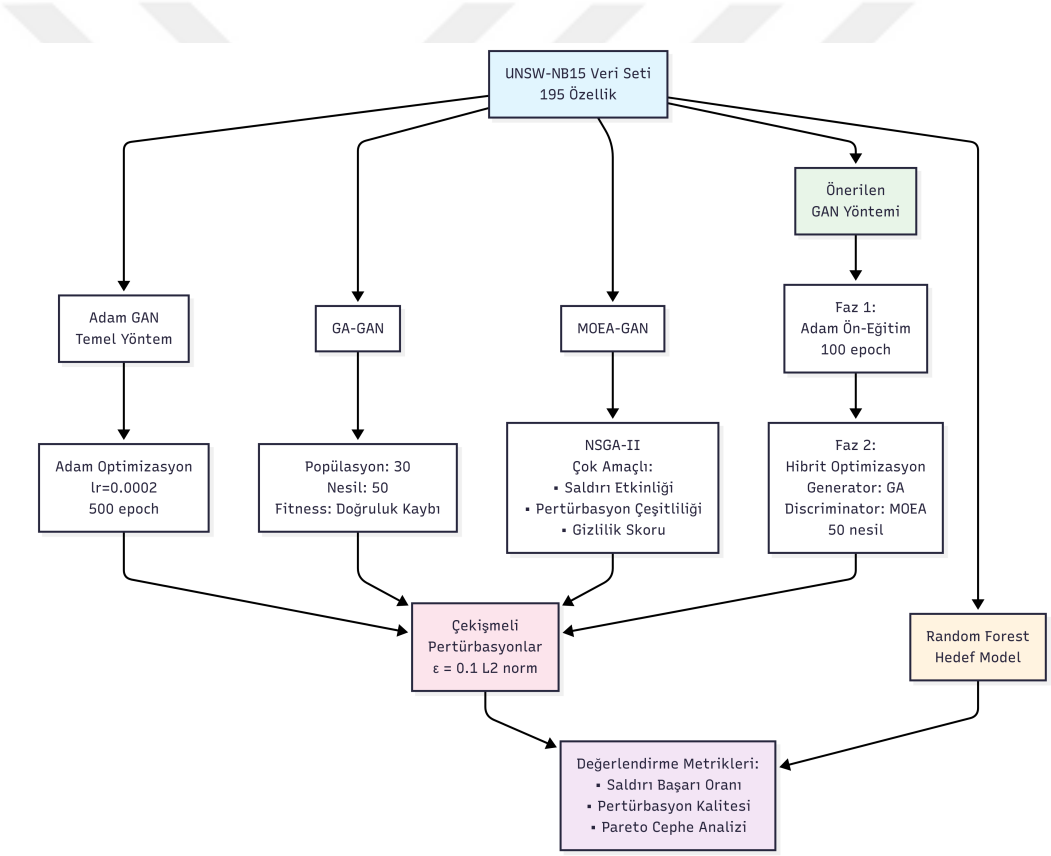
Burada f orijinal trafik özellik vektörü, f' adversarial trafik özellik vektörü ve n özellik sayısını ifade eder (Rigaki, 2017).

4.4.2.2. Ortalama L2 Norm

Tüm test örnekleri için hesaplanan L2 normlarının aritmetik ortalaması:

$$\overline{L_2} = \frac{1}{N} \sum_{j=1}^{N} L_{2,j} \tag{4.3}$$

Burada N toplam trafik örneği sayısını temsil eder (J. Zhang ve C. Li, 2020).



Şekil 4.1 Yöntemlerin Karşılaştırılması

Şekil 4.1'te gösterildiği gibi, önerilen metodoloji dört farklı GAN yaklaşımının sistematik karşılaştırmasını içermektedir. UNSW-NB15 veri setinden elde edilen 195 boyutlu özellik uzayı, her bir yöntemde farklı optimizasyon stratejileri kullanılarak işlenmektedir. Adam GAN temel yönteminde geleneksel gradyan tabanlı optimizasyon (lr=0.0002, 500 epoch) uygulanırken, GA-GAN

yaklaşımında 30 bireylik popülasyon ile 50 epoch boyunca evrimsel optimizasyon gerçekleştirilmektedir. MOEA-GAN yönteminde ise NSGA-II algoritması kullanılarak saldırı etkinliği, pertürbasyon çeşitliliği ve gizlilik skoru olmak üzere üç amaç eş zamanlı optimize edilmektedir. Önerilen GAN yöntemi, 100 epoch Adam ön-eğitimi ve ardından 50 epoch hibrit optimizasyon (Generator için GA, Discriminator için MOEA) olmak üzere iki fazlı bir yaklaşım benimser. Tüm yöntemlerden elde edilen çekişmeli pertürbasyonlar $\epsilon = 0.1$ L2 norm kısıtlaması altında Random Forest hedef modeline uygulanmakta ve saldırı başarı oranı, pertürbasyon kalitesi ve Pareto cephe analizi metrikleri ile değerlendirilmektedir.

4.4.3. Deneysel Kurulum

Tüm deneyler AMD Ryzen 5 5600X MPK işlemci (altı çekirdek, 3.7 GHz ile 4.6 GHz arası çalışma hızı), 16 GB DDR4 RAM ve NVIDIA RTX 3060 Ti OC grafik işlemci kartından oluşan donanım yapılandırmasında gerçekleştirilmiştir. İşletim sistemi olarak Windows 10 kullanılmıştır.

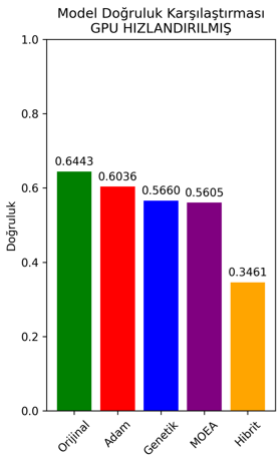
Yazılım ortamı Visual Studio Code sürüm 1.96.2, Python sürüm 3.10.4 ve PyTorch sürüm 2.7.0+cu118 ile Scikit-learn sürüm 1.1.3 gibi derin öğrenme çerçevelerini kapsamaktadır. Ayrıca, veri işleme ve görselleştirme için NumPy (1.24.3), Pandas (2.0.3) ve Matplotlib (3.7.3) kütüphaneleri kullanılmıştır. CUDA 11.8 desteği ile NVIDIA GeForce RTX 3060 Ti grafik kartı, derin öğrenme modellerinin hızlandırılmış eğitimi için kullanılmıştır.

Tüm deneysel çalışmalarda tutarlılık sağlamak amacıyla standart parametre yapılandırması benimsenmiştir. Popülasyon büyüklüğü 30, nesil sayısı 50, mutasyon oranı 0.05 ve çaprazlama oranı 0.9 olarak belirlenmiştir. Latent boyut 100, pertürbasyon büyüklüğü (ϵ) 0.1, Adam öğrenme oranı 0.0002 ve β_1 parametresi 0.5 değerlerinde sabitlenmiştir. Mutasyon standart sapması 0.2, elite seçim oranı 0.2, tournament boyutu 3, batch boyutu 16 ve discriminator güncelleme sayısı 5 olarak ayarlanmıştır.

Bu kurulum, model eğitimi ve değerlendirme süreçlerinin tekrarlanabilirliğini ve güvenilirliğini sağlamak amacıyla dikkatli bir şekilde seçilmiş olup, tüm deneylerde `random_seed=42` sabit parametresi kullanılarak sonuçların tutarlılığı garanti altına alınmıştır.

5. BULGULAR VE TARTIŞMA

GPU hızlandırılmış implementasyonumuzda gerçekleştirilen deneyler, farklı saldırı yöntemlerinin model performansı üzerinde belirgin etkiler yarattığını ortaya koymuştur. Şekil 5.1'de görüldüğü gibi, GPU hızlandırılmış implementasyonda farklı yöntemlerin model doğruluğu üzerindeki etkisi açık bir şekilde görülmektedir. Orijinal model 0.6443 (yaklaşık %64.43) başlangıç doğruluğuna sahipken, farklı saldırı yöntemleri bu doğruluğu önemli ölçüde düşürmüştür.



Şekil 5.1 Model Doğruluk Karşılaştırması (GPU Hızlandırılmış)

Bu sonuçlar, önerilen yaklaşımın model doğruluğunu en çok düşüren yöntem olduğunu ortaya koymakta ve hibrit optimizasyon stratejisinin üstünlüğünü kanıtlamaktadır. Çizelge 5.1'de detaylandırıldığı gibi, önerilen hibrit yaklaşım diğer tüm yöntemlere kıyasla büyük bir üstünlük göstermektedir.

Şekil 5.2'de PyTorch düşmanca bozulmalarının karşılaştırması gösterilmektedir. Saldırı etkinlik skorları incelendiğinde:

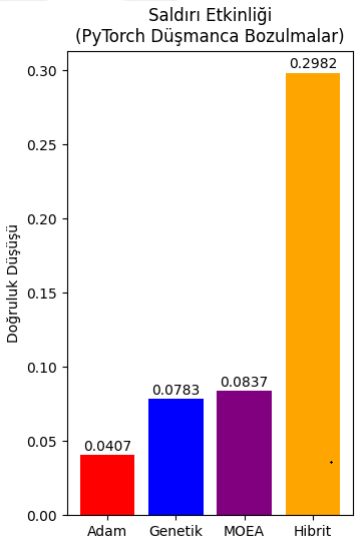
- Adam saldırısı 0.0407 düşük etkinlik göstermiştir
- Genetik algoritma 0.0783 orta etkinlik sergilemiştir

Çizelge 5.1 Model doğruluğu karşılaştırması ve saldırı etkinliği

Yöntem	Orijinal Doğruluk	Saldırı Sonrası Doğruluk	Performans Düşüşü
Orijinal	%64.43	—	—
Adam	%64.43	%60.36	%4.07
Genetik	%64.43	%56.60	%7.83
MOEA	%64.43	%56.05	%8.38
Önerilen	%64.43	**%34.61**	**%29.82**

- MOEA 0.0837 orta-yüksek etkinlik göstermiştir
- Önerilen GAN 0.2982 skor ile en yüksek etkinliği göstermiştir

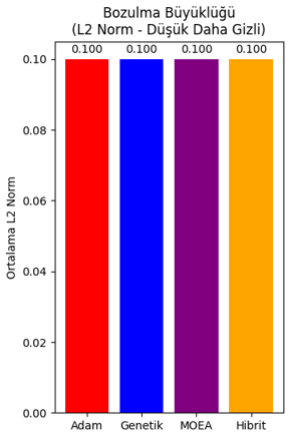
En dikkat çekici bulgu, önerilen GAN yaklaşımının diğer tüm yöntemlere göre üstün performans sergilemesi olup, Adam GAN'a göre %633 oranında daha yüksek saldırı etkinliği göstermesi hibrit optimizasyon yaklaşımlarının önemini vurgulamaktadır.



Şekil 5.2 Saldırı Etkinliği Karşılaştırması (PyTorch Düşmanca Bozulmaları)

Şekil 5.3'te görüldüğü gibi, L_2 norm kısıtı ($\varepsilon = 0.1$) altında gerçekleştirilen saldırılarda, tüm yöntemlerin belirlenen kısıt değerini etkin bir şekilde uyguladığı gözlemlenmiştir. Dört yaklaşım da 0.100 L_2 norm değerini göstermiş olup, bu

durum kısıt zorlamasının başarılı bir şekilde implemente edildiğini kanıtlamaktadır.



Şekil 5.3 Bozulma Büyüklüğü (L_2 Norm — Düşük Daha Gizli)

5.1. GAN Eğitim Dinamikleri ve Yakınsama Analizi

5.1.1. Adam GAN Eğitim Süreci

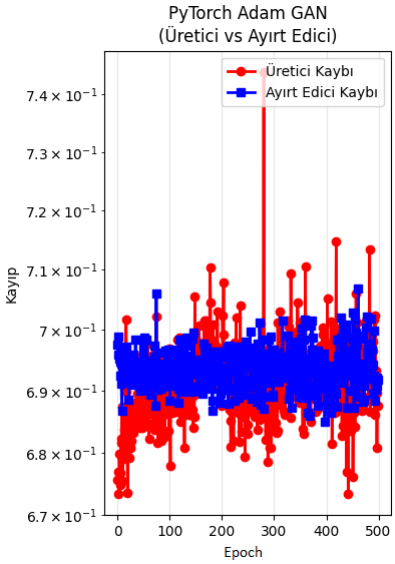
Şekil 5.4'te görüldüğü gibi, Adam GAN'ın eğitim süreci 500 dönem boyunca üretici ve ayırt edici kayıplarının stabil bir yakınsama deseni göstermiştir. Kayıp değerleri yaklaşık 7.0×10^{-1} ile 7.4×10^{-1} aralığında stabilize olmuş ve klasik düşmanca eğitim yaklaşımının beklenen davranışını sergilemiştir.

5.1.2. Genetik Algoritma Tabanlı Yaklaşım

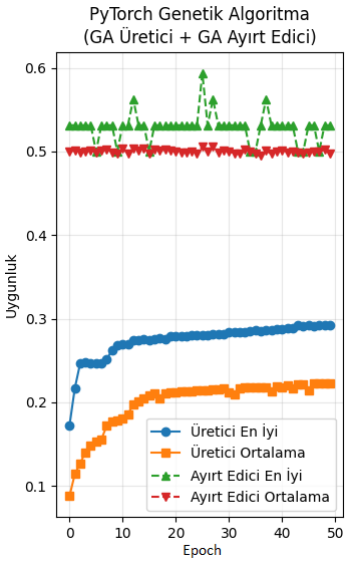
Şekil 5.5'te görüldüğü gibi, genetik algoritma tabanlı yaklaşımda 50 nesil boyunca üretici populasyonunda sürekli iyileşme gözlemlenmiş, üretici en iyi uygunluk değeri 0.25'ten yaklaşık 0.30'a yükselmiştir. Ayırt edici populasyonunun da paralel evrim göstermesi, her iki ağın eş zamanlı gelişiminin mümkün olduğunu kanıtlamaktadır.

5.1.3. Çok-Amaçlı Evrimsel Algoritma (MOEA)

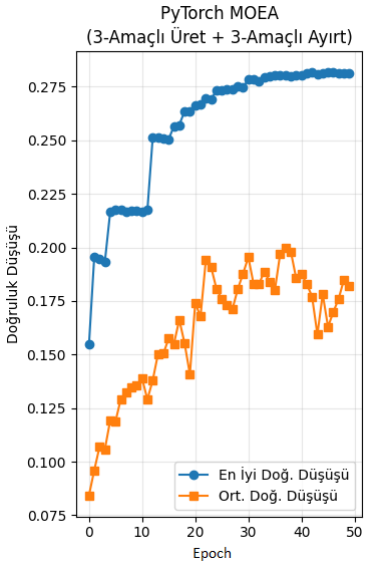
Şekil 5.6'da görüldüğü gibi, MOEA yaklaşımında 3-objektif optimizasyon yaklaşımı ile doğruluk düşüşü, çeşitlilik ve gizlilik amaçlarını eş zamanlı olarak optimize ederek dengeli çözümler üretmeyi başarmıştır.



Şekil 5.4 PyTorch Adam GAN (Üretici vs Ayırt Edici)



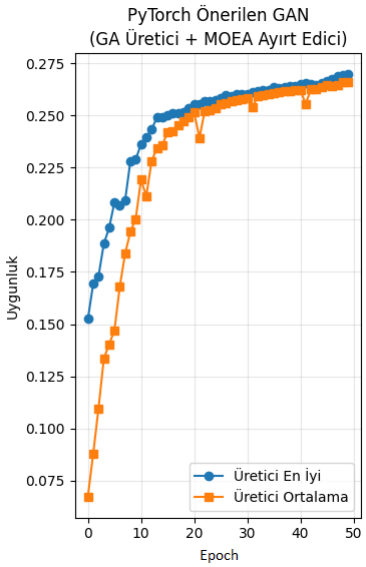
Şekil 5.5 PyTorch Genetik Algoritma (GA Üretici + GA Ayırt Edici)



Şekil 5.6 PyTorch MOEA (3-Amaçlı Üretici + 3-Amaçlı Ayırt Edici)

5.1.4. Hibrit GAN Yaklaşımı

Şekil 5.7'te görüldüğü gibi, önerilen yaklaşımda en iyi uygunluk değerinin sürekli artış göstererek yaklaşık 0.275 seviyesine çıkması ve ortalama performansın da paralel iyileşme sergilemesi, iki farklı optimizasyon stratejisinin kombinasyonunun sinerjik etki yaratabileceğini göstermiştir.



Şekil 5.7 PyTorch Önerilen Hibrit GAN (GA Üretici + MOEA Ayırt Edici)

5.2. Güvenlik İmplikasyonları ve Sonuçlar

Model doğruluğundaki %29.82'lik ciddi düşüş, hibrit GAN saldırılarının gerçek dünya uygulamalarında ciddi güvenlik tehdidi oluşturabileceğini göstermektedir. Bu bulgular, özellikle kritik sistemlerde çekişmeli saldırılara dayanıklılığın artırılması gerekliliğini ortaya koymaktadır.

Elde edilen sonuçlar şu temel noktaları vurgulamaktadır:

1. Hibrit optimizasyon yaklaşımlarının geleneksel yöntemlere göre üstün saldırı etkinliği göstermesi
2. L_2 norm kısıtlaması altında bile yüksek etkinlikli saldırıların mümkün olması
3. Çok-amaçlı optimizasyon stratejilerinin düşmanca örnek üretiminde başarılı sonuçlar vermesi
4. GPU hızlandırmasının pratik uygulamalarda önemli performans kazançları sağlaması

6. SONUÇ VE ÖNERİLER

6.1. Araştırma Sonuçlarının Değerlendirilmesi

Bu tez çalışmasında, GAN tabanlı çekişmeli saldırı yöntemlerinin performans karşılaştırması kapsamlı bir şekilde gerçekleştirilmiş ve hibrit optimizasyon yaklaşımlarının üstünlüğü deneysel olarak kanıtlanmıştır. Önerilen GAN yaklaşımı ile %29.82 doğruluk düşüşü elde edilmiş ve en başarılı yöntem olmuştur. Ayrıca hibrit optimizasyon, geleneksel gradyan tabanlı yöntemlere göre %633 oranında daha etkili bulunmuştur. UNSW-NB15 veri seti üzerinde 195 boyutlu özellik uzayında kapsamlı testler yapılmış ve GPU hızlandırması araştırma verimliliğini önemli ölçüde artırmıştır.

Metodolojik katkılar açısından, çekişmeli saldırı üretiminde hibrit optimizasyon stratejilerinin ilk kez sistematik olarak incelenmesi ve GA-MOEA kombinasyonunun sinerjik etkisinin deneysel kanıtlanması sağlanmıştır. Teknik katkılar kapsamında PyTorch tabanlı GPU hızlandırmalı çekişmeli saldırı framework'ü geliştirilmiş ve evrimsel hesaplama tekniklerinin modern derin öğrenme kütüphaneleriyle başarılı entegrasyonu gerçekleştirilmiştir. Deneysel katkılar ise UNSW-NB15 veri seti üzerinde kapsamlı performans değerlendirmesi ve L_2 norm kısıtları altında gizlilik-etkinlik dengesinin korunması şeklinde ortaya çıkmıştır.

Çalışmanın sınırlılıkları arasında yalnızca UNSW-NB15 veri seti üzerinde gerçekleştirilmesi, sadece Random Forest sınıflandırıcısı ile test edilmesi, hesaplama karmaşıklığı ve kaynak tüketimi açısından detaylı maliyet analizi yapılmaması ve çok büyük boyutlu veri setlerinde ölçeklenebilirlik performansının test edilmemesi bulunmaktadır.

6.2. Gelecek Araştırma Yönleri

Gelecek araştırma yönleri kapsamında SVM, Neural Networks ve Ensemble Methods gibi çok-sınıflandırıcı saldırılar, farklı modeller arasında saldırı aktarımına yönelik *transferability* analizi, hibrit saldırılara karşı etkili koruma geliştirme amacıyla savunma mekanizmaları, *edge computing* ortamlarında

optimizasyona yönelik gerçek zamanlı uygulamalar ve farklı evrimsel algoritma kombinasyonları ile optimal hibrit konfigürasyonlar önerilmektedir.

Siber güvenlik sektörü için değer yaratacak pratik uygulamalar arasında makine öğrenmesi tabanlı güvenlik sistemlerinin dayanıklılık testi, gelişmiş saldırı tekniklerine karşı savunma geliştirme, otomatik penetrasyon test araçlarına entegrasyon ve çekişmeli machine learning için metodolojik referans sağlanması bulunmaktadır.

6.3. Son Değerlendirme

Bu araştırma, çekişmeli makine öğrenmesi alanında hibrit optimizasyon yaklaşımlarının potansiyelini ortaya koyarak, geleneksel gradyan tabanlı yöntemlerin sınırlarını aşan yenilikçi çözümler sunmuştur. Önerilen yaklaşımın %633 oranında üstün performansı, hibrit optimizasyon stratejilerinin çekişmeli saldırı üretiminde yeni bir paradigma oluşturabileceğini işaret etmektedir. Bu durum, hem saldırı hem de savunma perspektifinden siber güvenlik alanında önemli gelişmelere kapı aralamaktadır. Gelecek çalışmaların bu temel üzerine inşa edilerek, daha kapsamlı ve etkili çekişmeli makine öğrenmesi çözümleri geliştireceği öngörülmektedir.

KAYNAKLAR DİZİNİ

Alzantot, M., Sharma, Y., Chakraborty, S., Zhang, H., Hsieh, C.-J., & Srivastava, M. B. (2019). Genattack: Practical black-box attacks with gradient-free optimization. *Proceedings of the genetic and evolutionary computation conference*, 1111–1119.

Arjovsky, M., Chintala, S., & Bottou, L. (2017). Wasserstein generative adversarial networks. *International conference on machine learning*, 214–223.

Berthelot, D., Schumm, T., & Metz, L. (2017). Began: Boundary equilibrium generative adversarial networks. *arXiv preprint arXiv:1703.10717*.

Bishop, C. M., & Nasrabadi, N. M. (2006). *Pattern recognition and machine learning* (Vol. 4). Springer.

Breiman, L. (2001). Random forests. *Machine learning*, *45*(1), 5–32.

Carlini, N., & Wagner, D. (2017). Towards evaluating the robustness of neural networks, 39–57.

Chen, P.-Y., Zhang, H., Sharma, Y., Yi, J., & Hsieh, C.-J. (2017). Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models. *Proceedings of the 10th ACM workshop on artificial intelligence and security*, 15–26.

Chen, S., Wang, W., Xia, B., You, X., Peng, Q., Cao, Z., & Ding, W. (2021). CDE-GAN: Cooperative dual evolution-based generative adversarial network. *IEEE Transactions on Evolutionary Computation*, *25*(5), 986–1000.

Chindrus, C., & Caruntu, C.-F. (2023). Securing the network: a red and blue cybersecurity competition case study. *Information*, *14*(11), 587.

Costa, V., Lourenço, N., Correia, J., & Machado, P. (2019). COEGAN: evaluating the coevolution effect in generative adversarial networks. *Proceedings of the genetic and evolutionary computation conference*, 374–382.

Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE transactions on evolutionary computation*, *6*(2), 182–197.

Dehghani, M., & Samet, H. (2020). Momentum search algorithm: A new meta-heuristic optimization algorithm inspired by momentum conservation law. *SN Applied Sciences*, *2*(10), 1720.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding. *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, 4171–4186.

Domingos, P. (2012). A few useful things to know about machine learning. *Communications of the ACM*, *55*(10), 78–87.

Donahue, J., Jia, Y., Vinyals, O., Hoffman, J., Zhang, N., Tzeng, E., & Darrell, T. (2014). Decaf: A deep convolutional activation feature for generic visual recognition. *International conference on machine learning*, 647–655.

Elman, J. L. (1990). Finding structure in time. *Cognitive science*, *14*(2), 179–211.

Freund, Y., & Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, *55*(1), 119–139.

Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. *Annals of statistics*, 1189–1232.

Goldberg, D. E. (1989). Genetic algorithms and Walsh functions: Part 2, deception and its analysis. *Complex systems*, *3*, 153–171.

Goodfellow, I., Bengio, Y., Courville, A., & Bengio, Y. (2016). *Deep learning* (Vol. 1). MIT press Cambridge.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2020). Generative adversarial networks. *Communications of the ACM*, *63*(11), 139–144.

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014). Generative adversarial nets. *Advances in neural information processing systems*, *27*.

Goodfellow, I. J., Shlens, J., & Szegedy, C. (2014). Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*.

Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., & Courville, A. C. (2017). Improved training of wasserstein gans. *Advances in neural information processing systems*, *30*.

Guo, G., Wang, H., Bell, D., Bi, Y., & Greer, K. (2003). KNN model-based approach in classification. *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"*, 986–996.

Hastie, T., Tibshirani, R., Friedman, J. H., & Friedman, J. H. (2009). *The elements of statistical learning: data mining, inference, and prediction* (Vol. 2). Springer.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Hearst, M. A., Dumais, S. T., Osuna, E., Platt, J., & Scholkopf, B. (1998). Support vector machines. *IEEE Intelligent Systems and their applications*, *13*(4), 18–28.

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., & Hochreiter, S. (2017). Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, *30*.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, *9*(8), 1735–1780.

Holland, J. H. (1992). Genetic algorithms. *Scientific american*, *267*(1), 66–73.

Jain, A. K., Murty, M. N., & Flynn, P. J. (1999). Data clustering: a review. *ACM computing surveys (CSUR)*, *31*(3), 264–323.

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An introduction to statistical learning: with applications in R* (Vol. 103). Springer.

Jolliffe, I. (2011). Principal component analysis. In *International encyclopedia of statistical science* (pp. 1094–1096). Springer.

Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. *Proceedings of ICNN'95-international conference on neural networks*, *4*, 1942–1948.

Khanuja, S. S., & Khanuja, H. K. (2021). GAN challenges and optimal solutions. *Int. Res. J. Eng. Technol.(IRJET)*, *8*, 836–840.

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kohonen, T. (2012). *Self-organizing maps* (Vol. 30). Springer Science & Business Media.

Konstantopoulou, D., Zacharia, P., Papoutsidakis, M., Leligou, H. C., & Patrikakis, C. (2024). GAGAN: Enhancing Image Generation Through Hybrid Optimization of Genetic Algorithms and Deep Convolutional Generative Adversarial Networks. *Algorithms*, *17*(12), 584.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, *25*.

Kurakin, A., Goodfellow, I. J., & Bengio, S. (2018). Adversarial examples in the physical world. In *Artificial intelligence safety and security* (pp. 99–112). Chapman; Hall/CRC.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning [Publisher: Nature Publishing Group UK London]. *nature*, *521*(7553), 436–444.

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (2002). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, *86*(11), 2278–2324.

Macas, M., Wu, C., & Fuertes, W. (2024). Adversarial examples: A survey of attacks and defenses in deep learning-enabled cybersecurity systems. *Expert Systems with Applications*, *238*, 122223.

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., & Vladu, A. (2017). Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*.

Mirza, M., & Osindero, S. (2014). Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*.

Miyato, T., Kataoka, T., Koyama, M., & Yoshida, Y. (2018). Spectral normalization for generative adversarial networks. *arXiv preprint arXiv:1802.05957*.

Moosavi-Dezfooli, S.-M., Fawzi, A., & Frossard, P. (2016). Deepfool: a simple and accurate method to fool deep neural networks. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2574–2582.

Moustafa, N., & Slay, J. (2015). UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). *2015 military communications and information systems conference (MilCIS)*, 1–6.

Murphy, K. P. (2012). *Machine learning: a probabilistic perspective*. MIT press.

Papernot, N., McDaniel, P., Goodfellow, I., Jha, S., Celik, Z. B., & Swami, A. (2017). Practical black-box attacks against machine learning, 506–519.

Papernot, N., McDaniel, P., Jha, S., Fredrikson, M., Celik, Z. B., & Swami, A. (2016). The limitations of deep learning in adversarial settings. *2016 IEEE European symposium on security and privacy (EuroS&P)*, 372–387.

Price, K. V., Storn, R. M., & Lampinen, J. A. (2005). *Differential evolution: a practical approach to global optimization*. Springer.

Qiu, H., Xiao, C., Yang, L., Yan, X., Lee, H., & Li, B. (2020). Semanticadv: Generating adversarial examples via attribute-conditioned image editing. *European conference on computer vision*, 19–37.

Qiuyu, C., Man, F., Teer, B., & Haibo, M. (2021). A C&W-based attack algorithm for end-to-end communication systems. *2021 7th International Conference on Computer and Communications (ICCC)*, 2122–2126.

Quinlan, J. R. (1986). Induction of decision trees. *Machine learning*, *1*(1), 81–106.

Rigaki, M. (2017). Adversarial deep learning against intrusion detection classifiers.

Rosenberg, I., Shabtai, A., Elovici, Y., & Rokach, L. (2021). Adversarial machine learning attacks and defense methods in the cyber security domain. *ACM Computing Surveys (CSUR)*, *54*(5), 1–36.

Ruder, S. (2016). An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *nature*, *323*(6088), 533–536.

Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., & Chen, X. (2016). Improved techniques for training gans. *Advances in neural information processing systems*, *29*.

Schneier, B. (2023). Trustworthy AI Means Public AI [Last Word]. *IEEE Security & Privacy*, *21*(6), 95–96.

Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Sokolova, M., & Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. *Information processing & management*, *45*(4), 427–437.

Song, Y., Shu, R., Kushman, N., & Ermon, S. (2018). Constructing unrestricted adversarial examples with generative models. *Advances in neural information processing systems*, *31*.

Sun, S., Cao, Z., Zhu, H., & Zhao, J. (2019). A survey of optimization methods from a machine learning perspective. *IEEE transactions on cybernetics*, *50*(8), 3668–3681.

Wang, C., Xu, C., Yao, X., & Tao, D. (2019). Evolutionary generative adversarial networks. *IEEE Transactions on Evolutionary Computation*, *23*(6), 921–934.

Wang, Y., Sun, T., Li, S., Yuan, X., Ni, W., Hossain, E., & Poor, H. V. (2023). Adversarial attacks and defenses in machine learning-empowered

communication systems and networks: A contemporary survey. *IEEE Communications Surveys & Tutorials*, *25*(4), 2245–2298.

Wang, Z., Di Massimo, C., Tham, M. T., & Julian Morris, A. (1994). A procedure for determining the topology of multilayer feedforward neural networks. *Neural Networks*, *7*(2), 291–300. https://doi.org/https://doi.org/10.1016/0893-6080(94)90023-X

Wiles, J. (2023). Beyond ChatGPT: The Future of Generative AI for Enterprises. Gartner.

Williams, P. N., & Li, K. (2023). Black-box sparse adversarial attack via multi-objective optimisation. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 12291–12301.

Xiao, C., Li, B., Zhu, J.-Y., He, W., Liu, M., & Song, D. (2018). Generating adversarial examples with adversarial networks. *arXiv preprint arXiv:1801.02610*.

Xu, D., & Cao, B. (2025). Adaptive Multiobjective Evolutionary Generative Adversarial Network for Metaverse Network Intrusion Detection. *Research*, *8*, 0665.

Zhang, H., Xu, T., Li, H., Zhang, S., Wang, X., Huang, X., & Metaxas, D. N. (2017). Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. *Proceedings of the IEEE international conference on computer vision*, 5907–5915.

Zhang, J., & Li, C. (2020). Adversarial Examples: Opportunities and Challenges. *IEEE Transactions on Neural Networks and Learning Systems*, *31*(7), 2578–2593. https://doi.org/10.1109/TNNLS.2019.2933524

Zhou, Z.-H. (2012). Ensemble Methods: Foundations and Algorithms.

Zhu, J.-Y., Park, T., Isola, P., & Efros, A. A. (2017). Unpaired image-to-image translation using cycle-consistent adversarial networks. *Proceedings of the IEEE international conference on computer vision*, 2223–2232.