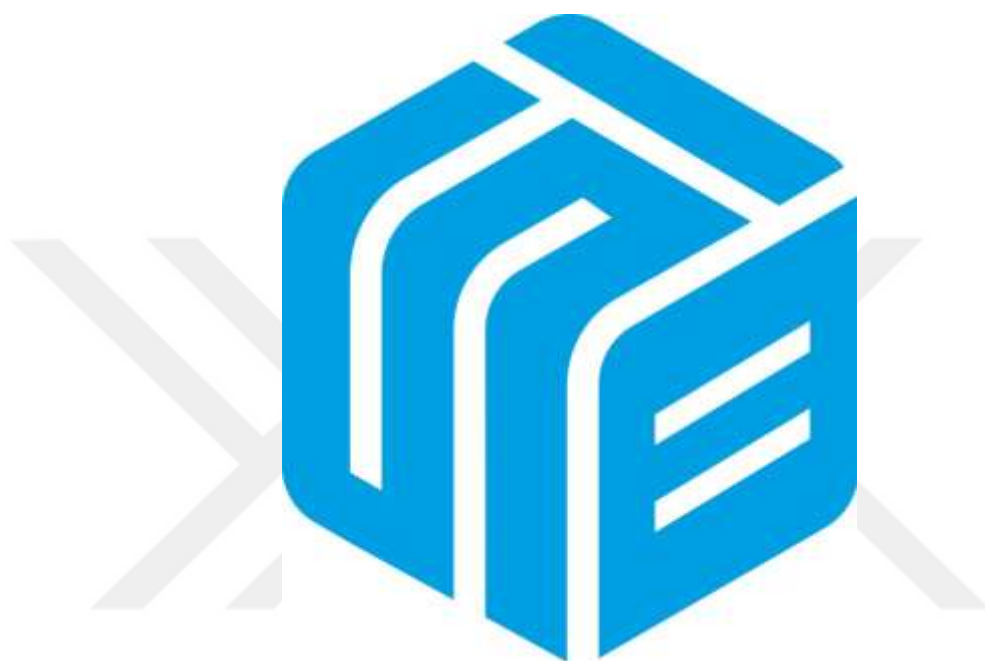




**SÜRÜ ROBOTİK SİSTEMLERİNDE BÜTÜNLEŞİK GÜVENLİK ÇERÇEVESİ:
ELİPTİK EĞRİ KRİPTOGRAFİSİ VE STEGANOGRAFI İLE GÜVENLİ
İLETİŞİM, YOLOV8 TABANLI ARTIRIMLI ÖĞRENME VE
AÇIKLANABİLİR TEHDİT TESPİTİ**

ZEYNEL KUTLU

**DOKTORA TEZİ
SAVUNMA TEKNOLOJİLERİ ANA BİLİM DALI**

**SİVAS BİLİM ve TEKNOLOJİ ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ**

ARALIK 2025

Zeynel KUTLU tarafından hazırlanan “SÜRÜ ROBOTİK SİSTEMLERİNDE BÜTÜNLEŞİK GÜVENLİK ÇERÇEVESİ: ELİPTİK EĞRİ KRİPTOGRAFİSİ VE STEGANOGRAFI İLE GÜVENLİ İLETİŞİM, YOLOV8 TABANLI ARTIRIMLI ÖĞRENME VE AÇIKLANABİLİR TEHDİT TESPİTİ” adlı tez çalışması aşağıdaki jüri tarafından OY BİRLİĞİ ile Sivas Bilim ve Teknoloji Üniversitesi SAVUNMA TEKNOLOJİLERİ Ana Bilim Dalında Doktora Tezi olarak kabul edilmiştir.

Danışman: Doç. Dr. Bülent Gürsel EMİROĞLU

Bilgisayar Mühendisliği, Kırıkkale Üniversitesi

Bu tezin, kapsam ve kalite olarak Doktora Tezi olduğunu onaylıyorum.

Başkan: Prof. Dr. Ramazan KATIRCI

Bilgisayar Mühendisliği, Sivas Bilim ve Teknoloji Üniversitesi

Bu tezin, kapsam ve kalite olarak Doktora Tezi olduğunu onaylıyorum.

Üye: Doç.Dr.Halit BAKIR

Bilgisayar Mühendisliği, Sivas Bilim ve Teknoloji Üniversitesi

Bu tezin, kapsam ve kalite olarak Doktora Tezi olduğunu onaylıyorum..

Üye: Doç.Dr.Hakan KÖR

Bilgisayar Mühendisliği, Hitit Üniversitesi

Bu tezin, kapsam ve kalite olarak Doktora Tezi olduğunu onaylıyorum.

Üye: Dr.Öğretim Üyesi Rıdvan Kaan AĞCA

Bilgisayar ve Öğretim Teknolojileri Öğretmenliği, Kırıkkale Üniversitesi

Bu tezin, kapsam ve kalite olarak Doktora Tezi olduğunu onaylıyorum.

Tez Savunma Tarihi: 30 / 12/ 2025

Jüri tarafından kabul edilen bu çalışmanın Doktora Tezi olması için gerekli şartları yerine getirdiğini onaylıyorum.

.....

Doç. Dr. Ali Suat YILDIZ
Lisansüstü Eğitim Enstitüsü Müdürü

ETİK BEYAN

Sivas Bilim ve Teknoloji Üniversitesi Lisansüstü Eğitim Enstitüsü Tez Yazım

Kurallarına uygun olarak hazırladığım bu tez çalışmada;

- Tez içinde sunduğum verileri, bilgileri ve dokümanları akademik ve etik kurallar çerçevesinde elde ettiğimi,
- Tüm bilgi, belge, değerlendirme ve sonuçları bilimsel etik ve ahlak kurallarına uygun olarak sunduğumu,
- Tez çalışmada yararlandığım eserlerin tümüne uygun atıfta bulunarak kaynak gösterdiğimi,
- Kullanılan verilerde herhangi bir değişiklik yapmadığımı,
- Bu tezde sunduğum çalışmanın özgün olduğunu,

bildirir, aksi bir durumda aleyhime doğabilecek tüm hak kayıplarını kabullendiğimi beyan ederim.

.....

Zeynel KUTLU

30 / 12 / 2025

SÜRÜ ROBOTİK SİSTEMLERİNDE BÜTÜNLEŞİK GÜVENLİK ÇERÇEVESİ:
ELİPTİK EĞRİ KRİPTOGRAFİSİ VE STEGANOGRAFI İLE GÜVENLİ İLETİŞİM,
YOLOV8 TABANLI ARTIRIMLI ÖĞRENME VE AÇIKLANABİLİR TEHDİT

TESPİTİ

(Doktora Tezi)

Zeynel KUTLU

SİVAS BİLİM VE TEKNOLOJİ ÜNİVERSİTESİ

LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

Aralık 2025

ÖZET

Sürü robotik sistemler, güvenlik, arama-kurtarma ve gözetleme gibi kritik uygulamalarda giderek daha fazla kullanılmaktadır. Ancak bu sistemler, siber saldırılara karşı savunmasızlık ve dinamik ortamlarda tehdit tespitindeki yetersizlikler gibi önemli güvenlik sorunlarıyla karşı karşıyadır. Bu çalışmada, bu problemleri çözmek amacıyla güvenli iletişim ve adaptif tehdit tespitini birleştiren bütünleşik bir güvenlik çerçevesi geliştirilmiştir. Önerilen sistem, hibrit güvenlik protokolü ve artırılmış öğrenme tabanlı tehdit tespit modülü olmak üzere iki temel bileşenden oluşmaktadır. Güvenli iletişim için, Eliptik Eğri Kriptografisi ile dijital imzalama ve steganografi teknikleri birleştirilmiştir. Raspberry Pi 4 üzerinde yapılan testlerde, bu sistem geleneksel RSA yöntemine kıyasla %85 daha az bellek kullanmış, %35,7 daha hızlı çalışmış ve %35,3 daha az enerji tüketmiştir. Sistem, simüle edilen siber saldırılarda %100 başarı oranıyla saldırıları engellemiş ve hiç yanlış alarm üretmemiştir. Tehdit tespiti için YOLOv8-nano yapay zeka modeli kullanılarak silah ve bıçak tespiti yapılmıştır. Model, sürekli öğrenme yaklaşımıyla 5 tur boyunca eğitilmiş ve performansı %71 artarak 0,886 başarı oranına ulaşmıştır. Özellikle azınlık sınıfı olan bıçak tespitinde %196 iyileşme sağlanmıştır. Altıncı turda gerçek dünya verileriyle karşılaşıldığında ortaya çıkan "unutma" problemi, geliştirilen hibrit eğitim yöntemiyle çözülmüş ve performans 0,911'e yükselmiştir. Geleneksel tek seferlik eğitime kıyasla, sürekli öğrenme yaklaşımı %6,3 daha iyi sonuç verirken %41,9 daha az hesaplama gücü kullanmıştır. Sistemin gerçek dünya uygulanabilirliğini göstermek için Intelligent Mission Cruiser (iMC) robot platformu tasarlanıp üretilmiş ve yazılımı geliştirilmiştir. Ground Control Station (Ground CS) kontrol yazılımı gerçekleştirilmiş ve FastDDS tabanlı iletişim altyapısıyla entegre edilmiştir. Geliştirilen yazılım mimarisi, sürü robotiğe uygun modüler yapıda tasarlanmış olup, yeni robot cihazlarının sisteme hızlıca adapte edilebilmesini sağlamaktadır.

Bilim Kodu : 91443,92403, 92418,92424, 92444

Anahtar Kelimeler : Sürü Robotik, Eliptik Eğri Kriptografisi, Steganografi, Tehdit Tespiti,YOLOv8, Katastrofik Unutma, Açıklanabilir Yapay Zeka, Güvenli İletişim, Gömülü Sistemler

Sayfa Adedi : 225

Danışman : Doç. Dr. Bülent Gürsel EMİROĞLU

INTEGRATED SECURITY FRAMEWORK IN SWARM ROBOTIC SYSTEMS:
SECURE COMMUNICATION WITH ELLIPTIC CURVE CRYPTOGRAPHY AND
STEGANOGRAPHY, YOLOV8-BASED INCREMENTAL LEARNING AND
EXPLAINABLE THREAT DETECTION

(Ph.D. Thesis)

Zeynel KUTLU

SİVAS UNIVERSITY OF SCIENCE AND TECHNOLOGY
INSTITUTE OF GRADUATE STUDIES

December 2025

ABSTRACT

Swarm robotic systems are increasingly utilized in critical applications such as security, search-and-rescue, and surveillance. However, these systems face significant security challenges, including vulnerability to cyber-attacks and inadequate threat detection in dynamic environments. This study develops an integrated security framework combining secure communication and adaptive threat detection to address these problems. The proposed system consists of two fundamental components: a hybrid security protocol and an incremental learning-based threat detection module. For secure communication, Elliptic Curve Cryptography with digital signing and steganography techniques were integrated. Tests on Raspberry Pi 4 demonstrated that this system uses 85% less memory, operates 35.7% faster, and consumes 35.3% less energy compared to traditional RSA methods, achieving 100% blocking success against simulated cyber-attacks with zero false positives. For threat detection, YOLOv8-nano was employed to detect firearms and knives. Through incremental learning over 5 rounds, performance improved by 71%, reaching 0.886 mAP@0.5, with 196% improvement in knife detection F1-score. When confronted with real-world data in round six, catastrophic forgetting was successfully resolved through a hybrid training method, improving performance to 0.911. Compared to one-shot training, incremental learning achieved 6.3% better results while requiring 41.9% less computational resources. To demonstrate real-world applicability, the Intelligent Mission Cruiser (iMC) robotic platform was designed, manufactured, and programmed. Ground Control Station (Ground CS) software was implemented and integrated with FastDDS-based communication infrastructure. The modular software architecture enables rapid integration of new robotic devices, and the system's operational feasibility was successfully validated through iMC platform testing.

Science Code : 91443,92403, 92418,92424, 92444
Key Words : Swarm Robotics, Elliptic Curve Cryptography, Steganography,
Threat Detection, YOLOv8, Catastrophic Forgetting, Explainable
Artificial Intelligence, Secure Communication, Embedded Systems
Page Number : 225
Supervisor : Assoc. Prof. Dr. Bülent Gürsel EMİROĞLU

TEŞEKKÜR

Her şeyden önce, lisansüstü eğitimim boyunca hem yüksek lisans hem de doktora aşamalarında danışmanlığımı üstlenen, yapıcı ve olumlu yaklaşımıyla her daim motivasyonumu yüksek tutan değerli hocam Doç. Dr. Bülent Gürsel EMİROĞLU'na en derin şükranlarımı sunmak isterim.

Tez çalışmam süresince kesintisiz desteklerini esirgemeyen ve nitelikli bir araştırmanın ortaya çıkması için büyük emek harcayan, akademik yolculuğumun ötesinde bir ömür boyu sürdürmeyi arzu ettiğim dostluk bağlarıyla bağlı olduğum değerli hocalarım Doç. Dr. Halit BAKIR ve Dr. Öğr. Üyesi Rıdvan Kaan AĞCA'ya ayrı ayrı içtenlikle teşekkür ederim.

Tez konumun makine öğrenmesi bölümünde bilgi ve tecrübesini cömertçe paylaşan kıymetli kardeşim Öğr. Gör. Emrecan BİNGÖL'e; tez kapsamında geliştirilen iMC robotunun tasarım, montaj ve yazılım süreçlerinde uzmanlığını ve deneyimini esirgemeyen değerli ağabeyim Yüksek Mühendis Tayfun BEKAR'a; doktora sürecimin en zorlu anlarında yapıcı ve motive edici desteğiyle yanımda olan kıymetli kardeşim Gökhan TAŞTAN ve değerli ağabeyim Yakup KORKMAZ'a; çalışmam boyunca ihtiyaç duyduğum her anda yardımını sunan değerli kardeşim Soner KESER'e; ve tezimin tüm aşamalarında desteklerini esirgemeyen, aynı birimde çalışmaktan onur duyduğum hem iş hem gönül arkadaşlarım Doç. Dr. Yusuf AKGÜL, Tuncay ULAŞ, Eren ERTİKE, Bünyamin ARSLAN ve Mehtap OKUTANSOY'a ayrı ayrı teşekkürlerimi sunarım.

Eğitim hayatım boyunca maddi ve manevi tüm desteğini sağlayan, çalışkanlık, beceri ve erdemli insanlık konusunda benim için örnek teşkil eden kıymetli babam Özdemir KUTLU'ya; duaları ve iyi dilekleriyle daima yanımda olan sevgili Annem'e; bu zorlu süreç boyunca gösterdikleri olağanüstü sabır, derin anlayış ve sağladıkları motivasyon ile hayatımın huzur kaynağı olan Hayat Arkadaşım'a, Kızım'a ve Oğlum'a sonsuz minnettarlığımı ve şükranlarımı sunarım. Bu çalışma, değerli ailemin verdiği manevi destek olmaksızın ortaya çıkamazdı.

Zeynel KUTLU

Sivas, 2025

	Sayfa
ÖZET	i
ABSTRACT.....	ii
TEŞEKKÜR.....	iii
İÇİNDEKİLER	iv
ÇİZELGELERİN LİSTESİ.....	vi
ŞEKİLLERİN LİSTESİ.....	vii
RESİMLERİN LİSTESİ.....	viii
SİMGELER VE KISALTMALAR	ix
1. GİRİŞ.....	1
2. KAYNAK ARAŞTIRMASI.....	9
2.1.Sürü Otonom Sistemler ve Güvenlik	9
2.2. Kriptografik Yöntemler ve Kaynak Kısıtlı Sistemler	12
2.3. Steganografi Teknikleri ve Hibrit Yaklaşımlar.....	17
2.4. Dağıtık Hizmet Reddi (DDoS) Saldırıları ve Tespit Mekanizmaları.....	21
2.5. Derin Öğrenme Tabanlı Nesne Tespiti	24
2.6. Artırımlı Öğrenme ve Adaptif Sistemler.....	27
2.7. Açıklanabilir Yapay Zekâ (XAI)	31
2.8. Literatür Sentezi ve Değerlendirme	34
3. MATERYAL ve YÖNTEM	39
3.1.Genel Sistem Mimarisi.....	39
3.1.1. Sistem Bileşenleri.....	39
3.1.2. Sistem Tasarım Felsefesi	41
3.2. Hibrit Güvenlik Protokolü.....	45
3.2.1. Eliptik Eğri Kriptografisi.....	46
3.2.2.Steganografik Gizleme	51
3.2.3.İki Aşamalı Hibrit Protokol	63
3.2.4.DDoS Koruması	67
3.2.5. Hibrit Güvenlik Protokolü Test Ortamı.....	71
3.3. Tehdit Tespit ve Artırımlı Öğrenme	76
3.3.1. YOLOv8 Model Mimarisi	76
3.3.2.Verit Seti.....	79
3.3.3. İlk Model Eğitimi (One-Shot Model).....	81

3.3.4. Artırımlı Öğrenme Stratejisi.....	82
3.3.5. Açıklanabilir Yapay Zekâ (Grad-CAM)	85
3.3.6. Performans Değerlendirme Metrikleri	87
3.4. Gerçek Dünya Uyarlaması: iMC ve Ground Cs Sistemi	90
3.4.1. iMC Donanım Mimarisi ve Sensör Entegrasyonu	92
3.4.2. iMC Yazılım Mimarisi	100
3.4.3. GroundCS Mimarisi	115
4. BULGULAR ve TARTIŞMA	125
4.1. Hibrit Güvenlik Protokolü Performans Analizi	125
4.1.1. ECC ve RSA Karşılaştırmalı Analizi	126
4.1.2. Steganografik Gömme Kalitesi ve Görsel Bütünlük	129
4.1.3. İki Aşamalı Hibrit Protokol İşlem Süreleri	132
4.1.4. DDoS Koruması Etkinliği ve Sistem Dayanıklılığı.....	144
4.2. Artırımlı Öğrenme ve Açıklanabilir Yapay Zeka Performans Analizi	148
4.2.1. Başlangıç Model Performansı (1.Tur).....	148
4.2.2. Artırımlı Öğrenme Süreci ve Performans Gelişimi (Tur 2-5).....	150
4.2.3. Gerçek Dünya Verileri ile Artırımlı Öğrenme ve Katastrofik Unutmanın Çözümü (6.tur)	157
4.2.4. Tam Veri Seti (One-Shot) ile Eğitilmiş Model Karşılaştırmalı Analizi.....	162
4.2.5. Grad-CAM Tabanlı Açıklanabilirlik Analizi	172
SONUÇ ve ÖNERİLER	179
KAYNAKLAR	185
ÖZGEÇMİŞ	207

ÇİZELGELERİN LİSTESİ

Çizelge	Sayfa
Çizelge 2.1. RSA ve ECC Kriptografik Algoritmalarının Karşılaştırmalı Analizi	14
Çizelge 2.2.YOLO Ailesi Versiyonlarının Karşılaştırmalı Performans Analizi	26
Çizelge 2.3.Artırımlı Öğrenme Yaklaşımlarının Karşılaştırmalı Analizi.....	28
Çizelge 2.4 Literatür Boşlukları ve Bu Tezin Katkıları Matrisi	38
Çizelge 3.1. GroundCS Donanım Özellikleri	115
Çizelge 3.2. GroundCS Ağ Yapısı.....	116
Çizelge 4.1. ECC ve RSA Kriptografik Sistemlerin Kaynak Kullanımı Karşılaştırması	126
Çizelge 4.2. ECC ve RSA İşlem Süresi ve Ek Yük Karşılaştırması.....	127
Çizelge 4.3. Steganografik Gömme Kalite Metrikleri	129
Çizelge 4.4. LSB Steganografi Performans Karşılaştırması	132
Çizelge 4.6. Operasyonel Gereksinimler	137
Çizelge 4.7. Güvenlik Protokolleri İşlem Süresi Karşılaştırması	139
Çizelge 4.8. Hibrit Protokol Enerji Tüketimi	143
Çizelge 4.9. DDoS Saldırı Senaryoları Test Sonuçları	146
Çizelge 4.10.DDoS Koruması Performans Karşılaştırması.....	147
Çizelge 4.11.Başlangıç Model Performans Metrikleri (1.Tur).....	149
Çizelge 4.12. Veri Akış İstatistikleri	151
Çizelge 4.13. Artırımlı Öğrenme Turlarında Genel Performans Metrikleri	153
Çizelge 4.14. Sınıf Bazlı Performans Metrikleri (Peak F1-Score Noktasında)	155
Çizelge 4.15. Hibrit Eğitim Model Karşılaştırması	160
Çizelge 4.16. Artırımlı Öğrenme vs. Tek Seferlik Eğitim Performans Karşılaştırması	163
Çizelge 4.17. Eğitim Süresi ve Hesaplama Maliyeti Karşılaştırması	167
Çizelge 4.18. Artırımlı Öğrenme ile Tek Seferlik Eğitim Kapsamlı Karşılaştırma	171

ŞEKİLLERİN LİSTESİ

Şekil	Sayfa
Şekil 3.1.Önerilen Entegre ECC-Steganografi Çerçevesi Mimarisi.....	45
Şekil 3.2. İMC tarafında işlemlerin akışı.....	73
Şekil 3.3. Sunucu Tarafı İşlem Akışı.....	75
Şekil 3.4. Tehdit Tespit ve Artırımlı Öğrenme Sistemi.....	76
Şekil 3.5. iMC Teknik 3B çizimi.....	93
Şekil 3.6.iMC Donanım mimarisi.....	99
Şekil 3.6.iMC Katmanlı yazılım mimarisi.....	100
Şekil 3.7.iMC Ana Kontrol Döngüsü	114
Şekil 4.1. Başlangıç Model Performans Eğrileri	148
Şekil 4.2. 2.Tur Model Performans Eğrileri	153
Şekil 4.3. 3.Tur Model Performans Eğrileri	154
Şekil 4.4. 4.Tur Model Performans Eğrileri	154
Şekil 4.5. 5.Tur Model Performans Eğrileri	155
Şekil 4.6. 6.Tur Model Performans Eğrileri	158
Şekil 4.7. 6.Tur Hibrit Eğitilmiş Model Performans Eğrileri	160
Şekil 4.6. 1.Tur (Baseline) Isı Haritası Dağılımı	172
Şekil 4.7. 2.Tur Isı Haritası Dağılımı.....	173
Şekil 4.8. 3.Tur Isı Haritası Dağılımı.....	174
Şekil 4.9. 4.Tur Isı Haritası Dağılımı.....	174
Şekil 4.10. 5.Tur Isı Haritası Dağılımı.....	175
Şekil 4.12. 6.Tur Hibrit Isı Haritası Dağılımı	176
Şekil 4.13. One Shot Isı Haritası Dağılımı	177

RESİMLERİN LİSTESİ

Resim	Sayfa
Resim 3.1. iMC sağdan görünüş	94
Resim 3.2. iMC soldan görünüş.....	94
Resim 3.3. iMC arkadan görünüş	95
Resim 3.4. iMC önden görünüş	95
Resim 3.5. GroundCS Aktif iMC görüntüleme ve seçme ekranı	117
Resim 3.6. GroundCS Kontrol Ekranı	118
Resim 3.7. GroundCS Tehdit Yönetim Web Modülü	123
Resim 4.1. Orijinal Görsel ile Stego Görsel	131



SİMGELER VE KISALTMALAR

Bu çalışmada kullanılmış simgeler ve kısaltmalar, açıklamaları ile birlikte aşağıda sunulmuştur.

Simgeler

°C	Santigrat derece (sıcaklık birimi)
bit	Binary digit (İkili basamak, bilgi birimi)
byte	Sekiz bitten oluşan veri birimi
dB	Desibel (ses şiddeti/sinyal gücü birimi)
GHz	Gigahertz (frekans birimi)
MHz	Megahertz (frekans birimi)
mJ	Milijul (enerji birimi)
ms	Milisaniye (zaman birimi)
µs	Mikrosaniye (zaman birimi)

Açıklamalar

Kısaltma

3DES	Triple DES (Üçlü DES)
AES	Advanced Encryption Standard (Gelişmiş Şifreleme Standardı)
AI	Artificial Intelligence (Yapay Zekâ)
AMR	Autonomous Mobile Robot (Otonom Mobil Robot)
BRE	Building Research Establishment (Bina Araştırma Kurumu)
CFD	Computational Fluid Dynamics (Hesaplamalı Akışkanlar Dinamiği)
ChaCha20	ChaCha20 Akış Şifreleme Algoritması
CLEFIA	CLEFIA Hafif Blok Şifreleme Algoritması
CNN	Convolutional Neural Network (Evrişimli Sinir Ağı)
DCT	Discrete Cosine Transform (Ayrık Kosinüs Dönüşümü)
DDoS	Distributed Denial of Service (Dağıtık Hizmet Reddi Saldırısı)
DES	Data Encryption Standard (Veri Şifreleme Standardı)
DNS	Domain Name System (Alan Adı Sistemi)
DoS	Denial of Service (Hizmet Reddi Saldırısı)
DWT	Discrete Wavelet Transform (Ayrık Dalgacık Dönüşümü)
ECC	Elliptic Curve Cryptography (Eliptik Eğri Kriptografisi)

Açıklama

GAN	Generative Adversarial Network (Üretici Çekişmeli Ağ)
GPS	Global Positioning System (Küresel Konumlama Sistemi)
Ground CS	Ground Control Station (Yer Kontrol İstasyonu)
GUID	Globally Unique Identifier (Küresel Benzersiz Tanımlayıcı)
HEVC	High Efficiency Video Coding (Yüksek Verimli Video Kodlama)
HTTP	Hypertext Transfer Protocol (Hiper Metin Aktarım Protokolü)
HTTPS	Hypertext Transfer Protocol Secure (Güvenli HTTP)
ICMP	Internet Control Message Protocol (İnternet Kontrol Mesaj Protokolü)
IDS	Intrusion Detection System (Saldırı Tespit Sistemi)
iMC	Intelligent Mission Cruiser (Akıllı Görev Aracı)
IoT	Internet of Things (Nesnelerin İnterneti)
IP	Internet Protocol (İnternet Protokolü)
JPEG	Joint Photographic Experts Group
JSteg	JSteg Steganografi Aracı
LiDAR	Light Detection and Ranging (Işık Algılama ve Menzil Belirleme)
LSB	Least Significant Bit (En Az Anlamlı Bit)
MAC	Message Authentication Code (Mesaj Doğrulama Kodu)
NIST	National Institute of Standards and Technology (Ulusal Standartlar ve Teknoloji Enstitüsü)
NSA	National Security Agency (Ulusal Güvenlik Ajansı)
OTA	Over-The-Air (Havadan Güncelleme)
OutGuess	OutGuess Steganografi Aracı
PHOTON	PHOTON Hafif Hash Fonksiyonu
PKI	Public Key Infrastructure (Açık Anahtar Altyapısı)
Poly1305	Poly1305 Mesaj Doğrulama Kodu
PRESENT	PRESENT Hafif Blok Şifreleme Algoritması
PSNR	Peak Signal-to-Noise Ratio (Tepe Sinyal-Gürültü Oranı)
RSA	Rivest-Shamir-Adleman (Asimetrik Şifreleme Algoritması)
SAS	Swarm Autonomous Systems (Sürü Otonom Sistemler)
SHA-256	Secure Hash Algorithm 256-bit (Güvenli Hash Algoritması)
SHA-3	Secure Hash Algorithm 3 (Güvenli Hash Algoritması 3)

SIMON	SIMON Hafif Blok Şifreleme Algoritması
SPECK	SPECK Hafif Blok Şifreleme Algoritması
SPONGENT	SPONGENT Hafif Hash Fonksiyonu
SSIM	Structural Similarity Index Measure (Yapısal Benzerlik İndeksi)
SVD	Singular Value Decomposition (Tekil Değer Ayrışımı)
TCP	Transmission Control Protocol (İletim Kontrol Protokolü)
TinyOS	TinyOS İşletim Sistemi (Gömülü Sistemler için)
TLS	Transport Layer Security (Taşıma Katmanı Güvenliği)
TRL	Technology Readiness Level (Teknoloji Hazırlık Seviyesi)
UAV	Unmanned Aerial Vehicle (İnsansız Hava Aracı)
UDP	User Datagram Protocol (Kullanıcı Veri Birimi Protokolü)
UGV	Unmanned Ground Vehicle (İnsansız Kara Aracı)
Wi-Fi	Wireless Fidelity (Kablosuz Ağ)
YOLO	You Only Look Once (Gerçek Zamanlı Nesne Tespit Algoritması)
YOLOv8	You Only Look Once version 8 (YOLO Sürüm 8)

1. GİRİŞ

Sürü otonom sistemler (Swarm Autonomous Systems - SAS), doğadaki sosyal böceklerin (arılar, karıncalar) kolektif davranışlarından esinlenerek geliştirilmiş, merkezi bir kontrol mekanizması olmaksızın çok sayıda otonom ajanın koordineli şekilde çalıştığı dağıtık sistemlerdir (Brambilla ve ark., 2013). Bu sistemlerin temel özellikleri arasında özerklik (her ajan bağımsız karar verebilme), yerellik (ajanlar sadece komşularıyla iletişim kurar), dağıtık kontrol (merkezi otorite yoktur), ölçeklenebilirlik (ajan sayısı dinamik olarak değişebilir) ve dayanıklılık (bir ajanın arızalanması sistemi çökertmez) yer almaktadır (Sahin, 2005).

Sürü otonom sistemler, son yıllarda geniş bir uygulama yelpazesinde kullanılmaya başlanmıştır:

Askeri Uygulamalar: Düşman hatlarının keşfi, gözetim ve istihbarat toplama, mayın tespiti, hedef takibi ve koordineli saldırı senaryolarında insansız hava araçları (UAV) ve insansız kara araçları (UGV) sürüleri kullanılmaktadır (Chung ve ark., 2018). Özellikle düşman hava savunma sistemlerinin yoğun olduğu bölgelerde, tek bir pahalı platform yerine çok sayıda düşük maliyetli otonom ajanın kullanılması, kayıp riskini dağıtmakta ve operasyonel esneklik sağlamaktadır (Scharre, 2014).

Arama-Kurtarma Operasyonları: Deprem ve sel gibi doğal afetlerde, engebeli arazide veya yıkık binalarda mahsur kalan kişilerin tespiti için otonom robotlar kullanılmaktadır (Murphy ve ark., 2008). Sürü sistemler, geniş alanları paralel olarak tarayarak arama süresini önemli ölçüde kısaltmaktadır (Queralta ve ark., 2020).

Hassas Tarım: Tarım alanlarının izlenmesi, bitki sağlığının analizi, zararlı tespiti ve hedeflenmiş ilaçlama için otonom drone sürüleri kullanılmaktadır (Albani ve ark., 2017). Bu yaklaşım, kimyasal kullanımını azaltarak çevresel sürdürülebilirliğe katkı sağlamaktadır.

Endüstriyel Otomasyon: Depo yönetimi, lojistik ve üretim hatlarında otonom mobil robotlar (AMR), malzeme taşıma ve envanter yönetimi görevlerini koordineli şekilde gerçekleştirmektedir (Wurman ve ark., 2008).

Çevre İzleme: Orman yangınlarının tespiti, hava kalitesi ölçümü, deniz kirliliği izleme ve vahşi yaşam gözlemi için otonom sensör ağları kullanılmaktadır (Dunbabin & Marques, 2012).

Bu geniş uygulama yelpazesi, sürü otonom sistemlerin esneklik, maliyet etkinliği ve hata toleransı gibi avantajlarını ortaya koymaktadır (Hamann, 2018). Ancak bu sistemlerin yaygınlaşması, güvenlik açıklarını da beraberinde getirmiştir (Sedjelmaci ve ark., 2017).

Sürü otonom sistemlerde güvenlik, literatürde iki temel boyutta ele alınmaktadır: İletişim Güvenliği ve Siber Saldırı Direnci (Sedjelmaci ve ark., 2017; Ferrag ve ark., 2020). Bu iki boyut, sistemin hem veri iletiminin güvenliğini hem de kötü niyetli saldırılara karşı direncini sağlamak üzere birbirini tamamlayıcı nitelikte tasarlanmaktadır.

İletişim Güvenliği, sürü içindeki düğümler arasında ve düğümlerle yer kontrol istasyonu arasında gerçekleşen veri iletiminin gizliliğini, bütünlüğünü ve kimlik doğrulamasını sağlamayı amaçlamaktadır (Sedjelmaci ve ark., 2017). Bu boyut, üç temel güvenlik ilkesi üzerine inşa edilmiştir:

- Kriptografi, iletişim kanallarında aktarılan verilerin yetkisiz erişime karşı korunmasını sağlar. Kaynak kısıtlı gömülü sistemlerde çalışan sürü sistemlerde, geleneksel RSA gibi asimetrik şifreleme yöntemleri, yüksek hesaplama yükü ve enerji tüketimi nedeniyle verimsiz olabilir (Koblitz ve ark., 2000). 128-bit güvenlik seviyesi için 3072-bit RSA yaklaşık 384 byte anahtar boyutu gerektirirken, 256-bit Eliptik Eğri Kriptografisi (ECC) sadece 32 byte ile aynı güvenlik seviyesini sağlamaktadır (Hankerson ve ark., 2004). Bu fark, özellikle düşük bant genişliğine sahip kablosuz ağlarda ve akıllı kartlar gibi hesaplama kaynakları kısıtlı cihazlarda kritik önem taşımaktadır.
- Kimlik doğrulama mekanizmaları, sürü içindeki her düğümün meşru bir sistem elemanı olduğunu doğrulayarak, sahte düğümlerin sisteme dahil olmasını engellemektedir (Sedjelmaci ve ark., 2017). Dijital imza algoritmaları, düğümlerin kimliklerini doğrulamak için kullanılmaktadır (Johnson ve ark.,

2001). Bu sayede, yalnızca geçerli özel anahtara sahip düğümler sistemle iletişim kurabilmekte ve veri gönderebilmektedir.

- Veri bütünlüğü, iletim sırasında verilerin değiştirilmediğini veya bozulmadığını garanti altına alır. Hash fonksiyonları ve dijital imzalar, veri bütünlüğünü sağlamak için kullanılan temel araçlardır (Sedjelmaci ve ark., 2017). Özellikle kritik görev verilerinin manipüle edilmesi, sürü operasyonlarının başarısızlığına yol açabileceğinden, veri bütünlüğü mekanizmalarının entegrasyonu zorunluluk arz etmektedir.

Siber Saldırı Direnci, sistemin kötü niyetli saldırıları tespit etme, önleme ve bu saldırılara rağmen operasyonel sürekliliğini koruma yeteneğini ifade etmektedir (Sedjelmaci ve ark., 2017; Ferrag ve ark., 2020). Bu boyut, özellikle Dağıtık Hizmet Reddi saldırıları, Küresel Konumlama Sistemi (GPS) sahtekarlığı ve sızma girişimleri gibi tehditlere karşı savunma mekanizmalarını içermektedir.

Saldırı tespit sistemi, sistem trafiğini sürekli olarak izleyerek anormal davranışları tespit etmektedir (Sedjelmaci ve ark., 2017; Ferrag ve ark., 2020). Geleneksel imza tabanlı tespit yöntemlerinin yanı sıra, son yıllarda derin öğrenme tabanlı Saldırı Tespit Sistemi (IDS) yaklaşımları, yüksek doğruluk oranları ile öne çıkmaktadır (Ferrag ve ark., 2020). Özellikle sürü sistemlerinde, düğümler arasındaki iletişim paternlerindeki anormalliklerin tespiti, erken uyarı sistemlerinin etkinliğini artırmaktadır (Sedjelmaci ve ark., 2017).

Dağıtık Hizmet Reddi Saldırısı (DDoS) koruması, hedef sisteme aşırı sayıda istek göndererek sistem kaynaklarının tükenmesine ve operasyonların kesintiye uğramasına neden olan saldırılara karşı savunma sağlamaktadır (Mirkovic & Reiher, 2004). Özellikle merkezi yer kontrol istasyonu sunucusunun, çoklu düğümlerden gelen yoğun veri trafiğini yönetmesi gerektiği göz önüne alındığında, Dağıtık Hizmet Reddi Saldırısı (DDoS) saldırılarına karşı etkili savunma mekanizmalarının entegrasyonu kritik önem taşımaktadır (Mirkovic & Reiher, 2004; Zargar ve ark., 2013). Uygulama katmanı filtreleme mekanizmaları, ağ katmanı saldırılarının yanı sıra, HTTP/HTTPS protokolü üzerinden gerçekleştirilen sofistike saldırılara karşı da etkili savunma sağlamaktadır (Zargar ve ark., 2013; Ranjan ve ark., 2009). Bu mekanizmalar, gelen her isteğin içeriğini

analiz ederek, yalnızca geçerli veri yapısına sahip isteklerin sunucu tarafından işlenmesine izin vermektedir (Ranjan ve ark., 2009). Aşamalı IP engelleme stratejisi, belirli bir IP adresinden gelen tekrarlı geçersiz isteklerin tespit edilmesi durumunda, o IP adresini otomatik olarak engelleyerek saldırı etkisini minimize etmektedir (Zargar ve ark., 2013; Peng ve ark., 2007).

Anomali tespiti, sistem davranışlarındaki beklenmedik sapmaları tanımlayarak, bilinmeyen veya sıfır gün saldırılarına karşı proaktif bir savunma sağlamaktadır (Ferrag ve ark., 2020; Peng ve ark., 2007). Makine öğrenmesi ve derin öğrenme tabanlı anomali tespit algoritmaları, normal sistem davranışını öğrenerek, bu davranıştan sapan aktiviteleri otomatik olarak işaretlemektedir (Ferrag ve ark., 2020). Bu yaklaşım, özellikle imza tabanlı sistemlerin tespit edemediği yeni saldırı türlerine karşı etkili bir savunma mekanizması sunmaktadır.

Araştırmanın Temel Amacı ve Katkıları

Bu çalışmanın temel amacı, sürü otonom sistemlerde güvenli ve sürdürülebilir yapay zekâ gelişimi sağlayan, uçtan uca entegre edilmiş bir güvenlik çerçevesi geliştirmektir. Sürü robotik sistemlerde tehdit tespit modellerinin güncellenmesi sürecinde, veri zehirlenmesi (data poisoning) saldırıları ciddi bir güvenlik riski oluşturmaktadır (Biggio & Roli, 2018; Gu ve ark., 2017). Mevcut çalışmalar, ya kriptografik güvenlik protokollerine (Sedjelmaci ve ark., 2017) ya da makine öğrenmesi güvenliğine (Lyu ve ark., 2020) odaklanmakta, ancak bu iki alanın entegrasyonu ve gerçek dünya uygulanabilirliği yeterince ele alınmamaktadır. Bu araştırma, güvenlik protokolü, donanım tasarımı ve yazılım mimarisini bütünlük bir sistem olarak sunarak, laboratuvar ortamından gerçek dünya uygulamasına geçişi mümkün kılan özgün bir yaklaşım sunmaktadır.

Temel Problem: Artırımlı Öğrenme Sürecinde Veri Güvenliği

Sürü otonom sistemlerde, kenar cihazlar (edge devices) operasyonel ortamda karşılaştıkları yeni tehditleri tespit eder ve bu verileri kullanarak modellerini artırımlı olarak geliştirir. Ancak bu süreçte kritik bir güvenlik açığı ortaya çıkmaktadır: Manipüle edilmiş verilerin eğitim setine dahil olması, modelin yanlış gelişmesine ve operasyonel başarısızlığa yol açabilir.

Bir saldırgan, aşağıdaki yöntemlerle sisteme müdahale edebilir:

- Yetkisiz erişim sağlayarak sahte tehdit verileri gönderebilir,

- Sosyal mühendislik ile meşru bir operatörü kandırarak yanlış etiketli veriler ekletebilir,
- Ortadaki adam (Man-in-the-middle) saldırısı ile iletişim kanalına müdahale ederek veri akışını manipüle edebilir.

Bu tür saldırılar, modelin arka kapı saldırısı (backdoor attack) veya veri zehirlleme (adversarial poisoning) yoluyla kasıtlı olarak bozulmasına neden olur. Özellikle artırılmış öğrenme sürecinde, bozuk veriler modelin yeni sınıfları öğrenirken eski bilgisini de etkileyebilir ve katasforik unutma problemini şiddetlendirebilir (Kirkpatrick ve ark., 2017). Aynı zamanda tehdit tespiti için çalışan modellerde tehdit olmayan bir içeriğin tehdit gibi modele öğretilmesi ciddi güvenlik zafiyetlerine sebebiyet verebilir.

Önerilen Çözüm: Hibrit Güvenlik Protokolü ile Korunan Artırılmış Öğrenme

Bu çalışmada, steganografi ve eliptik eğri kriptografisinin (ECC) entegre edildiği iki aşamalı bir veri doğrulama protokolü önerilmektedir.

Aşama 1: Steganografik GUID Gömme

- Her kenar cihaz, topladığı tehdit verisine rastgele bir konuma benzersiz bir tanımlayıcı (GUID) gömer.
- Gömme işlemi, En Az Anlamlı Bit (LSB) steganografi ile gerçekleştirilir ve görselin insan gözüyle algılanabilir kalitesi korunur (Cheddad ve ark., 2010). Bu yöntem, nesne tespit modelinin eğitimi sırasında örtük veri çoğaltma (implicit data augmentation) etkisi yaratarak modelin dayanıklılık (robustness) özelliğini artırır. Gömme lokasyonu, sadece meşru ajanlar tarafından bilinen deterministik bir algoritma ile belirlenir.

Aşama 2: ECC Tabanlı Kriptografik İmzalama

- GUID, gömme lokasyonu, cihaz kimliği, zaman damgası ve oturum bilgisi bir araya getirilerek cihazın ECC gizli anahtar (private key) ile dijital olarak imzalanır.
- Merkezi sunucu, veriyi aldığı anda:
 1. ECC açık anahtar (public key) ile imzayı doğrular,
 2. Steganografik GUID'i belirtilen lokasyondan çıkarır,

3. İki GUID'in eşleşmesini ve zaman damgasının geçerliliğini kontrol eder.

Bu çift katmanlı doğrulama hem veri bütünlüğünü hem de kaynak kimlik doğrulamasını garanti eder. Ayrıca, zaman damgası kontrolü tekrarlı saldırıyı (replay attack) önler.

Sistem Entegrasyonu ve Gerçek Dünya Uygulaması

Önerilen güvenlik protokolünün gerçek dünya uygulanabilirliğini göstermek için, donanım, yazılım ve ağ altyapısını kapsayan bütünsel bir sistem geliştirilmiştir:

Intelligent Mission Cruiser (iMC) - Otonom Kara Aracı Platformu: iMC, otonom sürüş, tehdit tespiti ve güvenli iletişim yeteneklerine sahip insansız bir kara aracı platformudur. Platform, edge AI hesaplama ünitesi (Raspberry Pi 4), algılama sistemi (kamera, LiDAR, GPS) ve iletişim modüllerini (Wi-Fi) entegre etmektedir. iMC üzerinde çalışan kontrol yazılımı, otonom sürüş, YOLOv8 tabanlı tehdit tespiti ve hibrit güvenlik protokolü modüllerini içermektedir. Sistem, operasyonel ortamda otonom olarak görev alanında devriye gezer, tehdit tespiti yapar ve tespit ettiği verileri güvenli bir şekilde merkezi sunucuya iletir.

2. Ground Control Station (Ground CS) - Merkezi Yönetim Yazılımı: Ground CS, sürü sistemin merkezi yönetimi, gerçek zamanlı gözetimi ve güvenli artırılmış öğrenme sürecini entegre eden bir yer kontrol istasyonu yazılımıdır. Yazılım, tüm iMC'lerin telemetri verilerinin ve kamera akışlarının izlenmesi, görev tanımlama ve yönetimi, veri doğrulama ve artırılmış öğrenme sürecinin yönetimi işlevlerini sağlamaktadır. Ground CS, iMC'lerden gelen verilerin ECC imza doğrulaması ve steganografik GUID kontrolü yaparak yetkisiz veri enjeksiyon girişimlerini tespit eder. Doğrulanmış veriler, uzman operatör tarafından etiketlenir ve güvenli artırılmış eğitim veri setine eklenir. Güncellenmiş model, OTA (Over-The-Air) güncelleme ile iMC'lere dağıtılır.

Literatüre Katkıları

Bu çalışma, sürü otonom sistemler ve güvenli yapay zekâ alanında aşağıdaki özgün katkıları sunmaktadır:

- Steganografi ve ECC'nin iki aşamalı kullanıldığı özgün bir veri doğrulama protokolü geliştirilmiş ve kenar (edge) cihazlarda uygulanabilirliği gösterilmiştir. Bu protokol, geleneksel kriptografik imzalama yöntemlerine kıyasla, tekrarlı

saldırı (replay attack), ortadaki adam (man-in-the-middle) ve veri enjeksiyon saldırılarına karşı daha güçlü koruma sağlamaktadır.

- Literatürde, çok-ajanlı öğrenme sistemlerinde veri zehirlenmesi saldırıları (Tolpegin ve ark., 2020; Shejwalkar & Houmansadr, 2021) ve güvenlik protokolleri (Ferrag ve ark., 2020; Mirkovic & Reiher, 2004) ayrı ayrı ele alınmakta, ancak steganografi-kriptografi hibrit yaklaşımıyla derin öğrenme güvenliğini sağlama ve gerçek donanım uygulaması yeterince çalışılmamıştır.
- Donanım-yazılım-ağ altyapısı bütünleşik bir sistem olarak tasarlanmış ve gerçek dünya koşullarında test edilmiştir. Bu, laboratuvar ortamında kalan çoğu akademik çalışmadan farklı olarak, Teknoloji Hazırlık Seviyesi (TRL) açısından daha yüksek bir olgunluk seviyesi sunmaktadır (Mankins, 2009).
- Intelligent Mission Cruiser (iMC) platformu ve Ground Control Station (Ground CS) yazılımı ile, önerilen güvenlik protokolünün gerçek dünya uygulanabilirliği gösterilmiştir.



2. KAYNAK ARAŞTIRMASI

2.1.Sürü Otonom Sistemler ve Güvenlik

Bu bölümde, sürü otonom sistemlerinin temel karakteristikleri, güvenlik gereksinimleri ve bu sistemlere yönelik tehdit modelleri incelenmektedir. Sürü robotik sistemleri, doğadaki sosyal böceklerin (arılar, karıncalar) kolektif davranışlarından esinlenerek geliştirilmiş, çok sayıda otonom ajanın merkezi bir kontrol olmaksızın koordineli şekilde çalıştığı dağıtık sistemlerdir. Bu sistemler, her bir ajanın basit davranış kurallarını takip etmesiyle ortaya çıkan karmaşık kolektif davranışlar sergileyebilmektedir (Dorigo ve ark., 2014). Sürü robotik sistemlerinin temel karakteristikleri arasında dağıtık kontrol, yerel iletişim, ölçeklenebilirlik, dayanıklılık ve esneklik yer almaktadır (Şahin, 2005). Bu özellikler sayesinde sürü sistemleri, arama-kurtarma operasyonları, çevre izleme, tarımsal uygulamalar, askeri gözetleme ve lojistik gibi geniş bir uygulama yelpazesinde kullanılmaktadır (Brambilla ve ark., 2013).

Sürü robotik sistemlerinin dağıtık yapısı ve otonom karakteri, sistemlere esneklik ve dayanıklılık kazandırırken, aynı zamanda önemli güvenlik açıkları da oluşturmaktadır. Geleneksel merkezi kontrol sistemlerinden farklı olarak, sürü sistemlerinde her bir ajan bağımsız karar verme yeteneğine sahiptir ve ajanlar arası iletişim genellikle kablosuz kanallar üzerinden gerçekleşmektedir. Bu durum, sistemleri çeşitli siber saldırılara karşı savunmasız hale getirmektedir (Higgins ve ark., 2009). Özellikle kritik altyapı izleme, askeri uygulamalar ve acil durum müdahale senaryolarında kullanılan sürü sistemlerinin güvenliği, sistem performansı kadar hayati önem taşımaktadır.

Sürü robotik sistemlerine yönelik güvenlik tehditleri, saldırı vektörlerine göre farklı kategorilerde sınıflandırılabilir.

Sybil saldırıları, kötü niyetli bir ajanın birden fazla sahte kimlik oluşturarak sisteme sızması ve karar mekanizmalarını manipüle etmesi şeklinde gerçekleşmektedir (Douceur, 2002). Bu tür saldırılar, özellikle çoğunluk oylaması veya konsensüs tabanlı karar mekanizmalarını kullanan sürü sistemlerinde kritik tehdit oluşturmaktadır. Dağıtık Hizmet Reddi Saldırıları (DDoS), sürü içindeki iletişim kanallarını aşırı yükleyerek

sistemin koordinasyon yeteneğini ortadan kaldırmayı amaçlamaktadır (Mirkovic & Reiher, 2004). Kaynak kısıtlı robotik sistemlerde, bu tür saldırılar hem iletişim bant genişliğini hem de hesaplama kaynaklarını tüketerek sistemin işlevsiz hale gelmesine neden olabilmektedir.

Karıştırma (Jamming) saldırıları, kablosuz iletişim kanallarına gürültü sinyalleri göndererek ajanlar arası iletişimi engellemektedir (Xu ve ark., 2005). Sürü sistemlerinin koordinasyonu yerel iletişime dayandığından, karıştırma (jamming) saldırıları sistemin kolektif davranışını doğrudan etkilemekte ve görev başarısızlığına yol açmaktadır. Taklit (Spoofing) ve ortadaki adam (man-in-the-middle) saldırıları, saldırganın meşru bir ajan gibi davranarak sahte bilgiler göndermesi veya ajanlar arası iletişimi dinleyip manipüle etmesi şeklinde gerçekleşmektedir (Karlof & Wagner, 2003). Bu saldırılar, özellikle konum bilgisi, hedef koordinatları veya tehdit bildirimleri gibi kritik verilerin bütünlüğünü tehlikeye atmaktadır. Fiziksel ele geçirme saldırıları, bir veya birden fazla ajanın fiziksel olarak ele geçirilmesi ve yeniden programlanması yoluyla gerçekleştirilmektedir (Becher ve ark., 2006). Ele geçirilen ajanlar, içeriden saldırı gerçekleştirerek kriptografik anahtarları çalabilir veya sistemin güven mekanizmalarını zayıflatabilir. Yukarıda özetlenen tehdit modelleri, sürü robotik sistemlerde çok katmanlı güvenlik yaklaşımlarının gerekliliğini ortaya koymaktadır.

Literatürde, sürü robotik sistemlerinin güvenliğini sağlamaya yönelik çeşitli yaklaşımlar önerilmiştir. Kriptografik yaklaşımlar, iletişim güvenliğini sağlamak için simetrik ve asimetrik şifreleme yöntemlerini kullanmaktadır. Watro ve arkadaşları (Watro ve ark., 2004), kaynak kısıtlı sensör ağları için optimize edilmiş bir açık anahtar altyapısı (PKI) önermişlerdir. Ancak, RSA gibi geleneksel asimetrik kriptografi yöntemlerinin hesaplama maliyeti, sınırlı işlem gücüne sahip robotik sistemler için önemli bir engel oluşturmaktadır. Bu nedenle, Eliptik Eğri Kriptografisi (ECC) gibi daha hafif alternatifler önerilmiştir (Malan ve ark., 2004). Bununla birlikte, kriptografik yöntemler iletişim gizliliğini ve bütünlüğünü sağlarken, Dağıtık Hizmet Reddi Saldırısı (DDoS) veya karıştırma (jamming) gibi kaynak tüketim saldırılarına karşı doğrudan koruma sağlamamaktadır.

Davranış tabanlı güvenlik yaklaşımları, sürü içindeki anormal davranışları tespit ederek kötü niyetli ajanları belirlemeyi amaçlamaktadır. Higgins ve arkadaşları (Higgins ve ark.,

2009), sürü robotik sistemlerinde saldırı tespit ve korumanın kritik önemini vurgulamış ve kötü niyetli bir varlığın sürüye sızması durumunda anormal davranışların tespit edilmesinin gerekliliğini belirtmişlerdir. Sargeant ve Tomlinson (Sargeant & Tomlinson, 2019), robotik sürülerde saldırı tespit sistemlerinin uygulanabilirliğini analiz etmiş ve sürü karakteristikleri nedeniyle anomali tabanlı tekniklerin doğrudan uygulanamayacağını, ancak imza tabanlı tekniklerin etkili olabileceğini göstermişlerdir. Fagiolini ve arkadaşları (Fagiolini ve ark., 2014), endüstriyel işbirlikçi robotik sistemler için dağıtık saldırı tespit protokolü geliştirmiş ve robotların davranışlarının sosyal kurallarla uyumluluğunu gözlemleyerek saldırganları tespit etmeyi önermiştir. Bu yaklaşımlar, bilinen saldırı kalıplarına karşı etkili olsa da, yeni ve sofistike saldırı türlerine genelleme konusunda zorluk yaşamaktadır. Ayrıca, davranış tabanlı yöntemler genellikle yanlış pozitif oranlarının yüksek olması nedeniyle sistem performansını olumsuz etkileyebilmektedir (Khalastchi & Kalech, 2019).

Güven tabanlı modeller, sürü içindeki her bir ajanın güvenilirliğini değerlendirerek, düşük güven skoruna sahip ajanların etkisini sınırlamayı hedeflemektedir. Ganeriwal ve Srivastava (Ganeriwal & Srivastava ,2004), sensör ağları için itibar tabanlı bir güven çerçevesi önermişlerdir. Bu modelde, her ajan komşularının davranışlarını gözlemleyerek güven skorları oluşturmakta ve düşük güvenilirliğe sahip ajanlardan gelen bilgileri filtrelemektedir. Fernandez-Gago ve arkadaşları (Fernandez-Gago, ve ark, 2007), dağıtık sistemlerde güven yönetimi için kapsamlı bir model geliştirmişlerdir. Ancak, güven tabanlı modeller, dinamik ağ topolojilerinde güven skorlarının güncellenmesi ve yeni katılan ajanların başlangıç güven değerlerinin belirlenmesi konusunda zorluklar yaşamaktadır. Ayrıca, güven hesaplamaları ek hesaplama ve iletişim yükü getirmektedir.

Anomali tespiti ve makine öğrenmesi tabanlı yaklaşımlar, son yıllarda sürü sistemlerinin güvenliğinde önemli bir araştırma alanı haline gelmiştir. Buczak ve Guven (Buczak & Guven, 2016), siber güvenlik uygulamalarında makine öğrenmesi yöntemlerinin kapsamlı bir incelemesini sunmuşlardır. Diro ve Chilamkurti (Diro & Chilamkurti, 2018), Nesnelerin İnterneti (IoT) ağlarında dağıtık saldırı tespiti için derin öğrenme tabanlı bir yaklaşım önermişlerdir. Bu çalışmalar, makine öğrenmesi modellerinin karmaşık saldırı kalıplarını öğrenme ve yüksek tespit doğruluğu sağlama potansiyelini göstermektedir. Ancak, geleneksel makine öğrenmesi yaklaşımları statik modeller kullanmakta ve yeni

saldırı türlerine adaptasyon konusunda sınırlı kalmaktadır. Ayrıca, derin öğrenme modellerinin "kara kutu" yapısı, güvenlik kararlarının açıklanabilirliği açısından sorun oluşturmaktadır.

Mevcut literatür incelendiğinde, sürü robotik sistemleri için önerilen güvenlik yaklaşımlarının her birinin belirli kısıtları bulunmaktadır. Kriptografik yöntemler yüksek güvenlik sağlarken hesaplama ve enerji maliyeti getirmekte, davranış tabanlı yöntemler yeni saldırı türlerine genelleme konusunda zorluk yaşamakta, güven modelleri ise dinamik ağ topolojilerinde güncelleme yükü oluşturmaktadır. Makine öğrenmesi tabanlı yaklaşımlar ise adaptasyon ve açıklanabilirlik konularında iyileştirmeye ihtiyaç duymaktadır. Bu durum, çok katmanlı, hafif, adaptif ve açıklanabilir güvenlik mekanizmalarına olan ihtiyacı ortaya koymaktadır. Özellikle, kriptografik güvenlik ile davranışsal tehdit tespitinin entegre edildiği, gerçek zamanlı olarak yeni tehditlere adapte olabilen ve güvenlik kararlarının şeffaf olduğu hibrit yaklaşımlar, gelecekteki araştırmalar için önemli bir yön oluşturmaktadır.

Ancak, sürü sistemlerindeki bu dağıtık ve dinamik yapı, güvenli iletişim ve veri bütünlüğü için özel kriptografik çözümler gerektirmektedir. Kaynak kısıtlı robotlar üzerinde çalışabilecek, enerji verimli ve hesaplama açısından hafif kriptografik algoritmalar, sürü güvenliğinin temel yapı taşlarını oluşturmaktadır. Bir sonraki bölümde, bu gereksinimleri karşılayan kriptografik yöntemler detaylı olarak incelenecektir.

2.2. Kriptografik Yöntemler ve Kaynak Kısıtlı Sistemler

Kriptografi, bilgi güvenliğinin temel taşlarından biri olarak, veri gizliliği, bütünlük, kimlik doğrulama ve inkâr edilemezlik özelliklerini sağlamaktadır (Stallings, 2017). Sürü robotik sistemleri ve Nesnelerin İnterneti (IoT) ağları gibi dağıtık sistemlerde, ajanlar arası iletişimin güvenliğini sağlamak için kriptografik yöntemler kritik öneme sahiptir. Ancak, bu sistemlerdeki ajanların genellikle sınırlı işlem gücü, bellek kapasitesi ve enerji kaynağına sahip olması, geleneksel kriptografik algoritmaların doğrudan uygulanmasını zorlaştırmaktadır (Paar & Pelzl, 2010). Bu nedenle, kaynak kısıtlı sistemler için optimize edilmiş hafif kriptografi yöntemlerinin geliştirilmesi, son yıllarda önemli bir araştırma alanı haline gelmiştir.

Kriptografik yöntemler temel olarak simetrik ve asimetrik olmak üzere iki ana kategoriye ayrılmaktadır. Simetrik kriptografide, şifreleme ve şifre çözme işlemleri için aynı gizli anahtar kullanılmaktadır (Daemen & Rijmen, 2020). Gelişmiş Şifreleme Standardı (AES), Veri Şifreleme Standardı (DES) ve Üçlü DES (3DES) en yaygın kullanılan simetrik şifreleme algoritmaları arasında yer almaktadır. Gelişmiş Şifreleme Standardı (AES), 128, 192 veya 256 bit anahtar uzunluklarıyla yüksek güvenlik seviyesi sağlarken, görece düşük hesaplama maliyeti nedeniyle kaynak kısıtlı sistemlerde tercih edilmektedir (Daemen & Rijmen, 2020). Ancak, simetrik kriptografinin en önemli zorluğu, güvenli anahtar dağıtımı problemidir. Özellikle dinamik topolojiye sahip sürü sistemlerinde, yeni katılan ajanlarla güvenli anahtar paylaşımı önemli bir güvenlik açığı oluşturmaktadır.

Asimetrik kriptografi veya açık anahtar kriptografisi, bu sorunu çözmek için bir çift anahtar (açık anahtar ve özel anahtar) kullanmaktadır (Stallings, 2022). Açık anahtar herkesle paylaşılabilirken, özel anahtar sadece sahibi tarafından bilinmektedir. Bu yapı, güvenli anahtar dağıtımı ve dijital imza gibi önemli güvenlik özelliklerini mümkün kılmaktadır. Ancak, asimetrik kriptografi algoritmaları, simetrik yöntemlere kıyasla çok daha yüksek hesaplama maliyeti gerektirmektedir. Bu durum, kaynak kısıtlı sistemlerde asimetrik kriptografinin kullanımını sınırlamaktadır (Ansari & Ali, 2025).

RSA (Rivest-Shamir-Adleman) algoritması, en yaygın kullanılan asimetrik kriptografi yöntemlerinden biridir (Yudistira, 2024). RSA'nın güvenliği, büyük asal sayıların çarpanlarına ayrılmasının hesaplama zorluğuna dayanmaktadır. Güvenli bir RSA uygulaması için genellikle 2048 bit veya daha uzun anahtar uzunlukları önerilmektedir (Barker, 2016).

Kriter	RSA	ECC
Anahtar uzunluğu (bit)	1024-4096	160-256
Eşdeğer güvenlik seviyesi	2048 bit	224 bit
Hesaplama karmaşıklığı	$O(n^3)$	$O(n^2)$
Bellek gereksinimi	Yüksek	Düşük
İmzalama hızı	Yavaş	Hızlı
Doğrulama hızı	Hızlı	Orta
Kaynak kısıtlı cihazlara uygunluk	Düşük	Yüksek
Enerji tüketimi (mj)	50-100	5-15

Kaynak: Barker (2016), Malan, ve ark, (2004), Liu & Ning (2008)

Ancak, bu anahtar uzunlukları, kaynak kısıtlı cihazlarda önemli hesaplama ve enerji tüketimi gerektirmektedir. Watro ve arkadaşları (Watro, ve ark, 2004), sensör ağları için optimize edilmiş bir RSA uygulaması geliştirmiş ve 1024 bit RSA işlemlerinin kaynak kısıtlı cihazlarda uygulanabilir olduğunu göstermişlerdir. Bununla birlikte, RSA işlemlerinin enerji tüketimi ve gecikme süresi, gerçek zamanlı uygulamalar için hala önemli bir engel oluşturmaktadır. Gura ve arkadaşları (Gura, ve ark, 2004), 8-bit mikroişlemcilerde RSA ve ECC performansını karşılaştırmış ve RSA'nın ECC'ye kıyasla önemli ölçüde daha yavaş olduğunu göstermişlerdir.

Eliptik Eğri Kriptografisi (ECC), asimetrik kriptografinin daha verimli bir alternatifi olarak ortaya çıkmıştır (Ullah, ve ark, 2023; Jebrane, ve ark, 2025). ECC, eliptik eğriler üzerindeki ayrık logaritma probleminin zorluğuna dayanmaktadır ve RSA ile aynı güvenlik seviyesini çok daha kısa anahtar uzunluklarıyla sağlayabilmektedir. Örneğin, 256 bit ECC anahtarı, 3072 bit RSA anahtarıyla eşdeğer güvenlik seviyesi sunmaktadır (Barker, 2016). Bu özellik, ECC'yi kaynak kısıtlı sistemler için ideal bir seçenek haline getirmektedir. Malan ve arkadaşları (Malan, ve ark, 2004), TinyOS tabanlı sensör ağları için ECC tabanlı bir açık anahtar altyapısı geliştirmiş ve ECC'nin kaynak kısıtlı cihazlarda pratik olarak uygulanabilir olduğunu göstermişlerdir. Liu ve Ning (Liu & Ning, 2008), sensör ağları için ECC tabanlı güvenli anahtar dağıtım protokolleri önermişlerdir.

ECC'nin kaynak kısıtlı sistemlerdeki bu avantajları, sürü robotik sistemler için özellikle önem taşımaktadır. Yüzlerce robot arasında gerçekleşen yoğun iletişim trafiği göz önüne alındığında, ECC'nin düşük hesaplama maliyeti ve kısa anahtar uzunlukları kritik avantajlar sağlamaktadır. Her bir şifreleme işleminde tasarruf edilen enerji ve hesaplama süresi, sürü ölçeği büyüdükçe sistemin genel verimliliğine önemli katkı sunmaktadır.

Kaynak kısıtlı sistemler için özel olarak tasarlanmış hafif kriptografi protokolleri, son yıllarda yoğun araştırma ilgisi görmüştür. Ulusal Standartlar ve Teknoloji Enstitüsü (NIST), hafif kriptografi standartları geliştirmek için bir süreç başlatmış ve çeşitli hafif blok şifreleme ve hash fonksiyonlarını değerlendirmiştir (McKay, ve ark, 2017). PRESENT, 64-bit blok boyutu ve 80 veya 128-bit anahtar uzunluğuyla donanım uygulamaları için optimize edilmiş bir hafif blok şifresidir (Bogdanov, ve ark, 2007). CLEFIA, 128-bit blok boyutuyla hem yazılım hem de donanım uygulamalarında yüksek performans sağlayan bir diğer hafif şifreleme algoritmasıdır (Shirai, ve ark, 2007). SIMON ve SPECK, NSA tarafından geliştirilen ve özellikle donanım ve yazılım uygulamalarında esneklik sağlayan hafif blok şifreleridir (Beaulieu, ve ark, 2015).

ChaCha20, Salsa20 şifreleme algoritmasının geliştirilmiş bir versiyonu olarak, yazılım uygulamalarında yüksek performans ve güvenlik sağlamaktadır (Bernstein, 2008). ChaCha20, özellikle AES donanım desteği olmayan cihazlarda AES'e göre daha hızlı çalışmaktadır ve TLS 1.3 gibi modern güvenlik protokollerinde yaygın olarak kullanılmaktadır. Poly1305, ChaCha20 ile birlikte kullanılan bir mesaj doğrulama kodu algoritmasıdır ve yüksek hız ile güvenliği birleştirmektedir (Bernstein, 2005). ChaCha20-Poly1305 kombinasyonu, kaynak kısıtlı sistemlerde hem şifreleme hem de kimlik doğrulama için etkili bir çözüm sunmaktadır.

Kriptografik hash fonksiyonları, veri bütünlüğünü sağlamak ve dijital imza oluşturmak için kullanılmaktadır. SHA-256 ve SHA-3, güvenli hash fonksiyonları olarak yaygın kullanım alanı bulmaktadır (NIST, 2015). Ancak, kaynak kısıtlı sistemler için daha hafif alternatifler geliştirilmiştir. PHOTON, donanım uygulamaları için optimize edilmiş hafif bir hash fonksiyonu ailesidir (Guo, ve ark, 2011). SPONGENT, sünger yapısı (sponge construction) kullanarak güvenlik ve verimlilik dengesini sağlayan bir diğer hafif hash fonksiyonudur (Bogdanov, ve ark, 2011).

Kriptografik yöntemlerin performansı, hesaplama süresi, enerji tüketimi, bellek kullanımı ve güvenlik seviyesi gibi çeşitli metriklerle değerlendirilmektedir. Eisenbarth ve arkadaşları (Eisenbarth, ve ark, 2012), hafif kriptografi algoritmalarının kapsamlı bir karşılaştırmasını sunmuş ve farklı donanım platformlarında performans analizleri yapmışlardır. Sonuçlar, algoritma seçiminin hedef platforma ve uygulama gereksinimlerine göre optimize edilmesi gerektiğini göstermektedir. Simetrik şifreleme algoritmaları genel olarak düşük hesaplama maliyeti sağlarken, asimetrik yöntemler güvenli anahtar dağıtımı ve dijital imza gibi kritik özellikleri mümkün kılmaktadır. Hibrit yaklaşımlar, asimetrik kriptografiyi anahtar dağıtımı için, simetrik kriptografiyi ise veri şifrelemesi için kullanarak her iki yöntemin avantajlarını birleştirmektedir (Stallings, 2017).

Mevcut kriptografik yöntemlerin kaynak kısıtlı sistemlerdeki uygulamaları incelendiğinde, çeşitli kısıtlar ve açık problemler ortaya çıkmaktadır. Kriptografik işlemler, özellikle asimetrik algoritmalar, önemli enerji tüketimi gerektirmekte ve batarya ömrünü kısaltmaktadır. Anahtar yönetimi, özellikle dinamik ve ölçeklenebilir sürü sistemlerinde karmaşık bir problem olmaya devam etmektedir. Kuantum bilgisayarların gelişi, RSA ve ECC gibi mevcut asimetrik kriptografi yöntemlerinin gelecekteki güvenliğini tehdit etmektedir (Jang, ve ark, 2025). Bu nedenle, kuantum sonrası kriptografi (post-quantum cryptography) algoritmaları geliştirilmekte ve standartlaştırılmaktadır (NIST, 2022). Ayrıca, kriptografik yöntemler veri gizliliğini ve bütünlüğünü sağlarken, trafik analizi saldırılarına karşı koruma sağlamamaktadır. Saldırganlar, şifrelenmiş iletişimi dinleyerek iletişim kalıpları, mesaj boyutları ve zamanlamaları hakkında bilgi edinebilmektedir. Bu durum, kriptografi ile tamamlayıcı güvenlik mekanizmalarının entegre edilmesi ihtiyacını ortaya koymaktadır.

ECC tabanlı şifreleme, mesajların içeriğini korusa da, şifrelenmiş trafiğin varlığı ve özellikleri (paket boyutları, iletişim sıklığı, şifreleme kalıpları) kötü niyetli gözlemciler tarafından tespit edilebilir ve trafik analizi saldırılarına maruz kalabilir. Bu nedenle, sürü robotik sistemlerde güvenlik mekanizmalarının kendisinin gizliliğini artırmak amacıyla steganografik yöntemler ile hibrit yaklaşımlar geliştirilmiştir. Bir sonraki bölümde, kriptografi ile steganografinin entegrasyonu detaylı olarak incelenecektir.

2.3. Steganografi Teknikleri ve Hibrit Yaklaşımlar

Steganografi, gizli bilgilerin görsel, işitsel veya metin tabanlı dijital medya içinde fark edilmeden saklanması tekniğidir. Kriptografi bilgiyi okunamaz hale getirirken, steganografi bilginin varlığını gizler ve bu iki tekniğin birlikte kullanılması çok katmanlı güvenlik sağlar. Bilgi gizleme sanatı olan steganografi, gizli mesajların varlığını maskeleyerek iletişim güvenliğini sağlayan bir tekniktir (Mandal, ve ark, 2022). Kriptografiden farklı olarak, steganografi mesajın içeriğini şifrelemek yerine, mesajın varlığını gizlemeyi amaçlamaktadır. Bu özellik, steganografiyi özellikle sansür ve gözetim ortamlarında değerli bir güvenlik aracı haline getirmektedir (Liao, ve ark, 2025). Sürü robotik sistemleri ve IoT ağları bağlamında, steganografi, kötü niyetli gözlemcilerin güvenlik mekanizmalarını tespit etmesini ve hedeflemesini zorlaştırarak ek bir güvenlik katmanı sağlayabilmektedir. Ayrıca, steganografik yöntemler, ağ trafiği analizi saldırılarına karşı koruma sağlayarak, iletişim kalıplarının ve güvenlik protokollerinin gizlenmesine olanak tanımaktadır.

Steganografi teknikleri, kullanılan taşıyıcı ortama göre farklı kategorilerde sınıflandırılmaktadır. Görüntü steganografisi, dijital görüntülerin piksel değerlerinde küçük değişiklikler yaparak veri gizlemeyi amaçlamaktadır (Cheddad, ve ark, 2010). Ses steganografisi, ses dosyalarının örnekleme değerlerini kullanarak bilgi saklamaktadır (Cvejic & Seppanen, 2004). Video steganografisi, video akışlarının karelerinde veri gizleyerek yüksek kapasite sunmaktadır (Sadek, ve ark, 2015). Metin steganografisi, metin belgelerinde sözdizimsel veya semantik değişiklikler yaparak bilgi gizlemektedir (Bhattacharyya, ve ark, 2010). Ağ steganografisi ise, ağ protokollerinin başlık alanları, paket zamanlamaları veya paket boyutları gibi özelliklerini kullanarak veri gizlemektedir (Wendzel, ve ark, 2015). Kaynak kısıtlı sistemler ve gerçek zamanlı uygulamalar için, hesaplama maliyeti düşük ve yüksek gömme kapasitesi sağlayan yöntemler tercih edilmektedir.

En Az Anlamlı Bit (LSB) yöntemi, en yaygın kullanılan görüntü steganografisi tekniklerinden biridir (Chan & Cheng, 2004). Bu yöntemde, taşıyıcı görüntünün piksel değerlerinin en düşük anlamlı bitleri, gizli mesajın bitleriyle değiştirilmektedir. En Az Anlamlı Bit (LSB) yöntemi, basit uygulaması ve yüksek gömme kapasitesi nedeniyle

popülerdir. Örneğin, 24-bit renkli bir görüntünün her pikselinde 3 bit (her renk kanalında 1 bit) veri gizlenebilmektedir. Ancak, En Az Anlamlı Bit (LSB) yöntemi istatistiksel analizlere karşı savunmasızdır ve steganaliz teknikleriyle tespit edilebilmektedir (Fridrich, ve ark, 2001). Geliştirilmiş LSB varyantları, rastgele piksel seçimi ve adaptif bit değiştirme gibi tekniklerle tespit edilebilirliği azaltmayı amaçlamaktadır (Mielikainen, 2006).

Kosinüs Dönüşümü (DCT) tabanlı steganografi, JPEG görüntü sıkıştırmasında kullanılan frekans alanı dönüşümünü kullanarak veri gizlemektedir (Cox, ve ark, 2007). Bu yöntemde, görüntü bloklara ayrılmakta ve her bloğun Kosinüs Dönüşümü (DCT) katsayıları hesaplanmaktadır. Gizli veri, orta frekanslı Kosinüs Dönüşümü (DCT) katsayılarına gömülmektedir, çünkü bu katsayılar hem görsel kaliteyi korumakta hem de sıkıştırmaya karşı dayanıklılık sağlamaktadır. JSteg ve OutGuess, Kosinüs Dönüşümü (DCT) tabanlı popüler steganografi araçlarıdır (Provos & Honeyman, 2003). Kosinüs Dönüşümü (DCT) tabanlı yöntemler, JPEG sıkıştırmasına karşı dayanıklılık sağlarken, LSB yöntemlerine kıyasla daha düşük gömme kapasitesi sunmaktadır.

Dalgacık Dönüşümü (DWT) tabanlı steganografi, görüntüyü farklı frekans bantlarına ayrıştırarak veri gizlemektedir (Taha, ve ark, 2019). Dalgacık Dönüşümü (DWT), görüntüyü dört alt banda (LL, LH, HL, HH) ayrıştırmakta ve gizli veri genellikle orta ve yüksek frekanslı alt bantlara gömülmektedir. DWT tabanlı yöntemler, DCT'ye kıyasla daha iyi görsel kalite ve sıkıştırma dayanıklılığı sağlamaktadır. Ayrıca, DWT çok seviyeli ayrıştırma yapabilmekte ve farklı çözünürlük seviyelerinde veri gizlemeye olanak tanımaktadır. Tao ve arkadaşları (Tao at al., 2014), DWT ve Tekil Değer Ayrışımı (SVD) kombinasyonunu kullanarak dayanıklı bir steganografi yöntemi önermişlerdir.

Yayıllı Spektrum (Spread Spectrum) steganografi, gizli mesajı geniş bir frekans bandına yayarak gizlemektedir (Kuznetsov, ve ark, 2024). Bu yöntem, iletişim sistemlerinde kullanılan yayıllı spektrum (spread spectrum) tekniklerinden esinlenmiştir ve gürültüye ve saldırılara karşı dayanıklılık sağlamaktadır. Gizli veri, sözde rastgele (pseudo-random) bir dizi ile modüle edilmekte ve taşıyıcı sinyale eklenmektedir. Alıcı tarafta, aynı sözde rastgele (pseudo-random) dizi kullanılarak gizli veri çıkarılmaktadır. Yayıllı spektrum (Spread spectrum) yöntemleri, düşük gömme kapasitesi sunsa da, yüksek güvenlik ve dayanıklılık sağlamaktadır.

Yukarıda incelenen steganografi teknikleri, mesajın varlığını gizleme konusunda etkili olmakla birlikte, mesajın içeriği korumasız kalmaktadır. Bu nedenle, günümüz güvenlik uygulamalarında steganografi ve kriptografinin birlikte kullanıldığı hibrit yaklaşımlar tercih edilmektedir.

Ağ steganografisi, ağ protokollerinin çeşitli özelliklerini kullanarak veri gizlemektedir ve kaynak kısıtlı dağıtık sistemler için özellikle ilgi çekicidir (Wendzel, ve ark, 2015). Paket başlık manipülasyonu, TCP/IP başlıklarındaki kullanılmayan veya esnek alanları (örneğin, IP ID alanı, TCP initial sequence number, TCP timestamp) kullanarak veri gizlemektedir (Ahsan & Kundur, 2002). Paket zamanlama yöntemleri, paketler arası gecikme sürelerini modüle ederek bilgi iletmektedir (Cabuk, ve ark, 2004). Paket boyutu modülasyonu, paket boyutlarını değiştirerek veri kodlamaktadır (Giffin, ve ark, 2002). Paket sıralaması yöntemleri, paketlerin gönderilme sırasını kullanarak bilgi iletmektedir (Murdoch & Lewis, 2005). Ağ steganografisi, özellikle trafik analizi saldırılarına karşı koruma sağlamak ve güvenlik mekanizmalarının varlığını gizlemektedir.

Hibrit kriptografi-steganografi yaklaşımları, her iki tekniğin avantajlarını birleştirerek çok katmanlı güvenlik sağlamaktadır (Provos & Honeyman, 2001). Bu yaklaşımda, gizli mesaj önce kriptografik algoritmalarla şifrelenmekte, ardından steganografik yöntemlerle taşıyıcı ortama gömülmektedir. Bu kombinasyon, hem mesaj içeriğinin gizliliğini (kriptografi) hem de mesajın varlığının gizlenmesini (steganografi) sağlamaktadır. Saldırgan, steganografik kanalı tespit etse bile, şifrelenmiş mesajı çözmek için kriptografik anahtara ihtiyaç duymaktadır.

Nesnelerin İnterneti (IoT) cihazlarının sınırlı hesaplama gücü ve enerji kaynakları göz önüne alındığında, güvenlik çözümlerinin hafif ve verimli olması kritik önem taşımaktadır. Almomani ve arkadaşları (Almomani, ve ark, 2022), Android IoT cihazlarında HEVC video akışları içinde veri gizlemek için hibrit bir kriptosteganografi yaklaşımı geliştirerek, IoT gizli verilerinin gizliliğini sağlamak amacıyla çeşitli hafif şifreleme algoritmalarını test etmişlerdir. Oladipupo ve arkadaşları (Oladipupo, ve ark, 2024), kaynak kısıtlı IoT cihazları için bulut destekli ortamlarda kullanılmak üzere S-box kullanımını ortadan kaldıran güvenli ve hafif bir kriptografi algoritması önermişlerdir. Tıbbi IoT sistemleri için Sethi ve arkadaşları (Sethi, ve ark, 2025), kısıtlı IoT cihazları

için ChaCha20 şifreleme ve Eliptik Eğri Kriptografisi (ECC) ile Ayrık Dalgacık Dönüşümü (DWT) steganografisini birleştiren hafif bir stego-kripto çerçevesi geliştirerek güvenli tıbbi veri iletimi sağlamışlardır. Li ve arkadaşları (Li, ve ark, 2024), IoT ortamlarında görüntü sahteciliğini önlemek için Üretici Çekişmeli Ağ (GAN) tabanlı adaptif steganografi kullanan hafif bir çözüm sunarak, hem güvenlik hem de hesaplama verimliliği arasında denge kurmuşlardır.

Adaptif steganografi yöntemleri, taşıyıcı ortamın özelliklerine göre gömme stratejisini dinamik olarak ayarlamaktadır (Luo, ve ark, 2010). Bu yöntemler, görüntünün karmaşık bölgelerinde (kenarlar, dokular) daha fazla veri gizleyerek, hem görsel kaliteyi korumakta hem de tespit edilebilirliği azaltmaktadır. Pevný ve arkadaşları (Pevný, ve ark, 2010), adaptif steganografi için makine öğrenmesi tabanlı bir yaklaşım önermişlerdir. Bu yaklaşımda, görüntünün hangi bölgelerinin veri gizleme için daha uygun olduğu, bir sınıflandırma modeli kullanılarak belirlenmektedir.

Steganaliz, steganografik içeriği tespit etmeyi amaçlayan karşı tekniklerdir (Fridrich, 2009). Steganaliz yöntemleri, istatistiksel analizler, makine öğrenmesi ve derin öğrenme tekniklerini kullanarak taşıyıcı ortamdaki anormallikleri tespit etmektedir. Ki-kare (Chi-square) testi, RS analizi ve Örnek Çifti Analizi (Sample Pair Analysis) gibi istatistiksel yöntemler, LSB steganografisini tespit etmede etkilidir (Fridrich, ve ark, 2001). Steganografi yöntemlerinin güvenliği, mevcut steganaliz tekniklerine karşı dayanıklılığıyla değerlendirilmektedir.

Mevcut steganografi ve hibrit yaklaşımlar incelendiğinde, çeşitli kısıtlar ve açık problemler ortaya çıkmaktadır. Steganografik yöntemlerin gömme kapasitesi, görsel kalite ve tespit edilebilirlik arasında bir denge gerektirmektedir. Yüksek gömme kapasitesi, genellikle görsel kalitede bozulmaya ve tespit edilebilirliğin artmasına neden olmaktadır. Ağ steganografisi yöntemleri, ağ koşullarındaki değişkenlikten (gecikme, paket kaybı) etkilenmektedir. Hibrit yaklaşımlar, hem kriptografik hem de steganografik işlemler nedeniyle ek hesaplama maliyeti getirmektedir. Ayrıca, mevcut yöntemlerin çoğu statik taşıyıcı ortamlar için tasarlanmış olup, dinamik ve gerçek zamanlı uygulamalarda adaptasyon konusunda sınırlı kalmaktadır. Gelişmiş steganaliz tekniklerinin ortaya çıkması, steganografi yöntemlerinin sürekli güncellenmesi ve iyileştirilmesi ihtiyacını doğurmaktadır.

Kriptografi ve steganografi, sürü sistemlerinde iletişim güvenliğini sağlarken, sistemlerin operasyonel sürekliliğini hedef alan saldırılar farklı tehdit vektörleri oluşturmaktadır. Bu tehditlerden en yaygın ve etkili olanı, ağ kaynaklarını tüketerek sistemin işlevini durdurmayı amaçlayan Dağıtık Hizmet Reddi (DDoS) saldırılarıdır. Bir sonraki bölümde, sürü robotik sistemlerde Dağıtık Hizmet Reddi (DDoS) saldırıları ve tespit mekanizmaları detaylı olarak incelenecektir.

2.4. Dağıtık Hizmet Reddi (DDoS) Saldırıları ve Tespit Mekanizmaları

Dağıtık Hizmet Reddi (Distributed Denial of Service-DDoS) saldırıları, sürü robotik sistemlerin iletişim altyapısını hedef alan ve operasyonel sürekliliği doğrudan tehdit eden en kritik saldırı türlerinden biridir. Modern siber güvenlik ortamında en yaygın ve yıkıcı tehditler arasında yer almaktadır. Bu saldırılar, hedef sistemi meşru kullanıcılara hizmet veremez hale getirmek amacıyla, çok sayıda kaynak üzerinden eşzamanlı olarak aşırı trafik veya işlem yükü oluşturmayı hedeflemektedir (Zargar, ve ark, 2013). Sürü otonom sistemler bağlamında, DDoS saldırıları özellikle kritik bir tehdit oluşturmaktadır çünkü bu sistemlerin işlevselliği, robotlar arası sürekli ve güvenilir iletişime bağlıdır. İletişim kanallarının tıkanması veya ağ kaynaklarının tükenmesi, sürü koordinasyonunun bozulmasına, görev başarısızlığına ve hatta fiziksel çarpışmalara yol açabilmektedir (Sedjelmaci, ve ark, 2017). Ayrıca, kaynak kısıtlı robotik sistemlerde, sürekli gelen sahte paketlerin işlenmesi batarya tükenmesine ve sistem çökmesine neden olabilmektedir (Kfoury, ve ark, 2019).

DDoS saldırıları, saldırı mekanizmalarına ve hedef katmanlara göre çeşitli kategorilere ayrılmaktadır. Hacim tabanlı saldırılar (volumetric attacks), ağ bant genişliğini tüketmeyi hedefleyen en yaygın DDoS türüdür. UDP flood, ICMP flood ve DNS Amplifikasyon saldırıları bu kategoriye girmektedir (Mirkovic & Reiher, 2004). Bu saldırılarda, saldırgan çok sayıda sahte paket göndererek ağ kapasitesini doldurur ve meşru trafiğin iletilmesini engeller. Amplifikasyon saldırılarında, saldırgan küçük bir istek göndererek hedef sisteme çok daha büyük yanıtların yönlendirilmesini sağlar; böylece saldırının etkisi katlanarak artmaktadır (Rossow, 2014). Protokol saldırıları (protocol attacks), ağ protokollerindeki zayıflıkları istismar ederek sunucu kaynaklarını veya ara ağ

ekipmanlarını (firewall, load balancer) tüketmeyi amaçlamaktadır. SYN flood, Ping of Death ve Smurf saldırıları bu kategoridedir (Eddy, 2007). SYN flood saldırısında, saldırgan TCP bağlantı kurma sürecini yarıda bırakarak sunucunun bağlantı tablosunu doldurur ve yeni meşru bağlantıların kurulmasını engeller. Uygulama katmanlı saldırıları (application layer attacks), hedef uygulamanın kaynaklarını tüketmeyi hedefleyen daha sofistike saldırılardır. HTTP flood, Slowloris ve DNS query flood bu türe örnektir (Specht & Lee, 2004). Bu saldırılar, meşru trafiği taklit ettikleri için tespit edilmeleri daha zordur ve nispeten düşük trafik hacmiyle bile etkili olabilmektedir.

Geleneksel DDoS tespit yöntemleri, temelde iki ana kategoriye ayrılmaktadır: imza tabanlı ve anomali tabanlı tespit. İmza tabanlı tespit sistemleri, bilinen saldırı kalıplarını bir veritabanında saklayarak gelen trafiği bu kalıplarla karşılaştırmaktadır (Peng, ve ark, 2007). Bu yaklaşım, bilinen saldırılara karşı yüksek doğruluk oranı sağlamakla birlikte, yeni veya varyant saldırıları tespit edememektedir (zero-day attacks). Ayrıca, imza veritabanının sürekli güncellenmesi gerekmektedir. Snort ve Suricata gibi popüler saldırı tespit sistemleri bu yaklaşımı kullanmaktadır (Roesch, 1999). Anomali tabanlı tespit sistemleri, normal ağ davranışının bir profilini oluşturarak bu profilden sapmaları saldırı olarak değerlendirmektedir (Patcha & Park, 2007). Bu yaklaşım, bilinmeyen saldırıları tespit edebilme avantajına sahiptir ancak yüksek yanlış pozitif oranları ve normal davranış profilinin doğru şekilde oluşturulması zorluğu gibi dezavantajları bulunmaktadır. İstatistiksel yöntemler (ortalama, standart sapma, entropi), spektral analiz ve zaman serisi analizi gibi teknikler anomali tespitinde kullanılmaktadır (Bhuyan, ve ark, 2014).

Son yıllarda, makine öğrenmesi ve derin öğrenme tekniklerinin DDoS tespit sistemlerinde kullanımı önemli ölçüde artmıştır. Geleneksel makine öğrenmesi yöntemleri arasında Destek Vektör Makineleri (SVM), Rastgele Orman (Random Forest), Karar Ağaçları ve Naive Bayes algoritmaları yaygın olarak kullanılmaktadır. Braga ve arkadaşları (Braga, ve ark, 2010), Self-Organizing Maps (SOM) kullanarak DDoS saldırılarını %95 doğrulukla tespit eden bir sistem önermiştir. Saied ve arkadaşları. (Saied, ve ark, 2016), Rastgele Orman (Random Forest) algoritmasını kullanarak IoT ağlarında DDoS tespiti gerçekleştirmiş ve %99.4 doğruluk oranı elde etmiştir. Ancak, bu yaklaşımlar manuel özellik mühendisliği gerektirmekte ve yüksek boyutlu veri setlerinde performans kaybı yaşamaktadır. Derin öğrenme tabanlı yaklaşımlar, otomatik özellik

çıkarmı ve karmaşık kalıpları öğrenme yetenekleriyle dikkat çekmektedir. Yuan ve arkadaşları (Yuan, ve ark, 2017), Derin Sinir Ağları (DNN) kullanarak DDoS saldırılarını tespit etmiş ve geleneksel yöntemlere göre daha yüksek doğruluk elde etmiştir. Yin ve arkadaşları (Yin, ve ark, 2017), Tekrarlayan Sinir Ağları (RNN) ve Uzun Kısa Süreli Bellek (LSTM) ağlarını kullanarak zamansal trafik kalıplarını analiz etmiş ve %98.88 doğruluk oranına ulaşmıştır. Convolutional Neural Networks (CNN) tabanlı yaklaşımlar da trafik verilerini görüntü formatına dönüştürerek başarılı sonuçlar elde etmiştir (Wang, ve ark, 2018).

Gerçek zamanlı DDoS tespiti, özellikle kaynak kısıtlı sürü robotik sistemlerinde önemli zorluklar içermektedir. Düşük gecikme gereksinimleri, tespit sisteminin milisaniyeler içinde karar vermesini gerektirmektedir çünkü sürü sistemlerinde koordinasyon kararları gerçek zamanlı olarak alınmaktadır (Kfoury, ve ark, 2019). Ancak, karmaşık derin öğrenme modellerinin çıkarım süreleri bu gereksinimleri karşılamakta zorlanabilmektedir. Sınırlı hesaplama kaynakları, küçük robotik platformlarda büyük sinir ağlarının çalıştırılmasını engellemektedir. Model sıkıştırma (pruning), kuantizasyon ve bilgi damıtma (knowledge distillation) teknikleri bu soruna çözüm olarak önerilmektedir (Hinton, ve ark, 2015). Enerji tüketimi, batarya ile çalışan robotlar için kritik bir kısıttır. Sürekli trafik analizi ve model çıkarımı, batarya ömrünü önemli ölçüde azaltabilmektedir. Adaptif Örnekleme ve Olay Tabanlı Tespit (event-driven detection) yaklaşımları enerji verimliliğini artırmak için kullanılmaktadır (Sedjelmaci, ve ark, 2017). Bu tespit yöntemlerinin başarısı, kullanılan eğitim veri setlerinin kalitesi ve gerçek dünya senaryolarını ne ölçüde temsil ettiği ile doğrudan ilişkilidir.

Mevcut DDoS tespit yaklaşımlarının sürü robotik sistemlerde uygulanmasında çeşitli kısıtlar bulunmaktadır. İlk olarak, çoğu çalışma geleneksel ağ ortamları (internet, veri merkezleri) için tasarlanmıştır ve sürü sistemlerinin dinamik topoloji, yüksek mobilite ve değişken ağ koşulları gibi özelliklerini dikkate almamaktadır (Campion, ve ark, 2018). İkinci olarak, mevcut tespit sistemleri genellikle merkezi bir yapıya sahiptir ve tüm trafik verilerinin merkezi bir noktada toplanmasını gerektirmektedir. Bu yaklaşım, dağıtık yapıya sahip sürü sistemlerinde ölçeklenebilirlik sorunlarına ve tek arıza noktası (single point of failure) riskine yol açmaktadır (Gil, ve ark, 2012). Üçüncü olarak, statik makine öğrenmesi modelleri, saldırı stratejilerindeki değişikliklere adapte olamamakta ve

zamanla performans kaybı yaşamaktadır (concept drift). Dördüncü olarak, yüksek doğruluk oranlarına rağmen, derin öğrenme modellerinin "kara kutu" doğası, güvenlik operatörlerinin tespit kararlarını anlamasını ve doğrulamasını zorlaştırmaktadır. Son olarak, mevcut veri setleri (KDD Cup 99, NSL-KDD, CICIDS2017) geleneksel ağ trafiklerini içermekte ve sürü robotik sistemlerin iletişim kalıplarını yeterince temsil etmemektedir (Sharafaldin, ve ark, 2018).

DDoS gibi ağ tabanlı tehditlerin yanı sıra, sürü robotik sistemler fiziksel ortamda faaliyet gösterirken çevresel tehditleri de gerçek zamanlı olarak tespit etme yeteneğine ihtiyaç duymaktadır. Görüntü tabanlı tehdit tespiti, bu gereksinimi karşılayan temel teknolojilerden biridir ve derin öğrenme mimarileri bu alanda devrim niteliğinde performans artışları sağlamıştır. Bir sonraki bölümde, sürü robotik sistemlerde nesne tespiti için kullanılan derin öğrenme yöntemleri detaylı olarak incelenecektir.

2.5. Derin Öğrenme Tabanlı Nesne Tespiti

Nesne tespiti, sürü robotlarının çevresel farkındalık kazanması ve tehdit unsurlarını tanımlaması için kritik bir yetenektir. Sürü robotik sistemlerde güvenlik uygulamaları, sadece ağ trafiği analizi ile sınırlı kalmayıp, fiziksel ortamda gerçek zamanlı nesne tespiti ve tanıma yeteneklerini de gerektirmektedir. Özellikle otonom sürü sistemlerinde, tehdit oluşturabilecek nesnelerin (yetkisiz İHA'lar, engeller, saldırgan varlıklar) hızlı ve doğru bir şekilde tespit edilmesi, sistemin güvenliği ve operasyonel sürekliliği açısından kritik öneme sahiptir. Bu bağlamda, derin öğrenme tabanlı nesne tespit algoritmaları, geleneksel görüntü işleme yöntemlerine kıyasla üstün performans göstermekte ve karmaşık senaryolarda yüksek doğruluk oranları sunmaktadır.

Nesne tespit alanındaki derin öğrenme yaklaşımları, temelde iki ana kategoriye ayrılmaktadır: iki aşamalı (two-stage) ve tek aşamalı (one-stage) dedektörler. İki aşamalı dedektörler, ilk olarak bölge önerilerini (region proposals) oluşturmakta ve ardından bu bölgeleri sınıflandırma işlemine tabi tutmaktadır. Bölge Tabanlı Evrişimli Sinir Ağı (R-CNN) mimarisi, bu yaklaşımın öncü örneklerinden biridir ve seçici arama (selective search) algoritması kullanarak görüntüden yaklaşık 2000 bölge önerisi çıkarmaktadır (Girshick, ve ark, 2014). Her bir bölge önerisi, bir Evrişimli Sinir Ağı (CNN) aracılığıyla öznetelik çıkarımına tabi tutulmakta ve ardından Destek Vektör Makinesi (SVM)

sınıflandırıcıları ile nesne kategorileri belirlenmektedir. Ancak R-CNN'in hesaplama maliyeti oldukça yüksektir ve gerçek zamanlı uygulamalar için uygun değildir.

Fast R-CNN mimarisi, R-CNN'in hesaplama verimliliğini artırmak amacıyla geliştirilmiştir (Girshick, 2015). Bu yaklaşımda, tüm görüntü tek bir CNN'den geçirilmekte ve öznitelik haritaları (feature maps) üzerinde bölge önerileri işlenmektedir. Bu sayede, her bölge için ayrı ayrı CNN işlemi yapılması gerekmemekte ve işlem süresi önemli ölçüde azalmaktadır. Faster R-CNN ise, bölge önerisi mekanizmasını da öğrenilebilir bir yapıya dönüştürerek Bölge Önerisi Ağı (RPN) kavramını ortaya koymuştur (Ren, ve ark., 2016). RPN, öznitelik haritaları üzerinde çalışarak potansiyel nesne konumlarını tahmin etmekte ve böylece seçici arama (selective search) gibi harici algoritmalara olan bağımlılığı ortadan kaldırmaktadır. Faster R-CNN, yüksek doğruluk oranları sunmasına rağmen, gerçek zamanlı uygulamalarda hala performans kısıtları bulunmaktadır.

Tek aşamalı dedektörler, bölge önerisi aşamasını atlayarak doğrudan görüntü üzerinde nesne tespiti ve sınıflandırma işlemlerini eş zamanlı olarak gerçekleştirmektedir. YOLO (You Only Look Once) mimarisi, bu yaklaşımın en bilinen örneklerinden biridir ve görüntüyü bir ızgara yapısına bölerek her ızgara hücresinin belirli sayıda sınırlayıcı kutu (bounding box) ve sınıf olasılıkları tahmin etmesini sağlamaktadır (Redmon, ve ark, 2016). İlk YOLO versiyonu, 45 FPS hızında çalışarak gerçek zamanlı nesne tespiti için önemli bir adım atmıştır. Ancak küçük nesnelerin tespitinde ve yoğun nesne gruplarında performans düşüklüğü yaşamaktadır.

YOLOv2 ve YOLOv3 versiyonları, mimari iyileştirmeler ve çoklu ölçekli öznitelik haritaları kullanarak bu kısıtları gidermeye çalışmıştır (Redmon & Farhadi, 2017, 2018). YOLOv3, özellikle Öznitelik Piramit Ağı (FPN) benzeri bir yapı kullanarak farklı ölçeklerdeki nesneleri daha etkin bir şekilde tespit edebilmektedir. YOLOv4 ve YOLOv5, CSPDarknet omurga ağı, PANet boyun yapısı ve çeşitli veri artırma teknikleri ile performansı daha da artırmıştır (Bochkovskiy, ve ark, 2020).

Versiyon	Yıl	FPS (GPU)	Parametre Sayısı	Temel Özellik
Yolov3	2018	35	61.5M	Çok ölçekli tahmin
Yolov4	2020	45	64.4M	CSPDarknet53
Yolov5	2020	140	7.2M (small)	PyTorch, modüler yapı
Yolov7	2022	161	36.9M	E-ELAN mimarisi
Yolov8	2023	165	3.2M (nano)	Anchor-free, C2f modülü

Kaynak: Redmon & Farhadi (2018), Bochkovskiy, ve ark. (2020), Jocher, ve ark. (2023)

YOLOv8, bu gelişimin en güncel halkasını oluşturmakta ve çapa-serbest (anchor-free) bir yaklaşım benimseyerek daha esnek ve hızlı bir tespit mekanizması sunmaktadır (Jocher, ve ark, 2023). Sürü robotik sistemlerin dinamik ve öngörülemeyen çevresel koşulları göz önüne alındığında, YOLOv8'in önceden tanımlanmış çapa kutularına (anchor box) ihtiyaç duymadan nesne tespiti yapabilmesi, farklı boyut ve şekillerdeki tehditleri tanımda önemli bir esneklik sağlamaktadır. Ayrıca, YOLOv8'in hafif model varyantları (nano, small), kaynak kısıtlı robotik platformlarda gerçek zamanlı çalışabilme potansiyeli sunmaktadır.

YOLOv8 mimarisi, üç ana bileşenden oluşmaktadır: omurga (backbone), boyun (neck) ve kafa (head). Omurga bölümü, görüntüden hiyerarşik öznitelikler çıkarmakta ve CSPDarknet tabanlı bir yapı kullanmaktadır. Boyun bölümü, farklı ölçeklerdeki öznitelik haritalarını birleştirerek çoklu ölçekli nesne tespitini desteklemektedir. Kafa bölümü ise, çapa-serbest (anchor-free) bir yaklaşım ile sınırlayıcı kutu koordinatlarını ve sınıf olasılıklarını doğrudan tahmin etmektedir. Bu mimari, önceki versiyonlara kıyasla daha az parametre kullanarak yüksek doğruluk ve hız dengesi sağlamaktadır.

Güvenlik uygulamalarında, YOLOv8 gibi gerçek zamanlı nesne tespit algoritmaları, çeşitli senaryolarda kullanılmaktadır. Örneğin, havaalanı güvenlik sistemlerinde yetkisiz İHA tespiti, askeri alanlarda saldırgan araç tanıma ve sınır güvenliğinde izinsiz geçiş tespiti gibi uygulamalar, bu algoritmaların etkinliğini göstermektedir. Ancak, sürü robotik sistemlerde bu algoritmaların entegrasyonu, bazı zorlukları da beraberinde getirmektedir. İlk olarak, her bir robot ünitesinin sınırlı hesaplama kaynakları, karmaşık derin öğrenme modellerinin doğrudan çalıştırılmasını zorlaştırmaktadır. İkinci olarak,

dinamik ve deęişken ortam kořulları (aydınlatma, hava durumu, gürültü), model performansını olumsuz etkileyebilmektedir.

Model sıkıştırma (model compression) ve bilgi damıtma (knowledge distillation) teknikleri, bu zorlukların üstesinden gelmek için kullanılmaktadır (Hinton, ve ark, 2015). Bilgi damıtma, büyük ve karmaşık bir öğretmen modelin bilgisini daha küçük ve verimli bir öğrenci modele aktararak, kaynak kısıtlı sistemlerde yüksek performans elde edilmesini sağlamaktadır. Ayrıca, model budama (pruning) ve kuantizasyon (quantization) gibi teknikler, model boyutunu ve hesaplama maliyetini azaltmakta, böylece gömülü sistemlerde daha etkin çalışma imkânı sunmaktadır.

Mevcut literatürde, derin öğrenme tabanlı nesne tespit sistemlerinin çoęu, statik veri setleri üzerinde eğitilmekte ve dağıtım sonrası yeni nesne sınıflarına adaptasyon yeteneęinden yoksun kalmaktadır. Sürü robotik sistemlerde ise, operasyonel ortam sürekli deęişmekte ve yeni tehdit türleri ortaya çıkabilmektedir. Bu nedenle, artırımı öğrenme (incremental learning) ve sürekli öğrenme (continual learning) yaklaşımlarının nesne tespit sistemlerine entegrasyonu, gelecekteki arařtırmalar için önemli bir yön oluşturmaktadır. Ayrıca, tespit edilen nesnelerin güvenilirlięinin deęerlendirilmesi ve yanlış pozitiflerin azaltılması, güvenlik kritik uygulamalarda hayati öneme sahiptir. Ancak, YOLOv8 dahil geleneksel derin öğrenme modelleri sabit bir veri seti üzerinde eğitilir ve operasyonel ortamda karşılaşılan yeni tehdit türlerine adaptasyon yetenekleri sınırlıdır. Modelin yeniden eğitilmesi, kaynak kısıtlı sürü robotlarında pratik deęildir. Bu nedenle, modellerin sahada sürekli öğrenme ve yeni bilgileri entegre etme yeteneęi sağlayan artırımı öğrenme yaklaşımları kritik önem taşımaktadır. Bir sonraki bölümde, bu yaklaşımlar detaylı olarak incelenecektir.

2.6. Artırımı Öğrenme ve Adaptif Sistemler

Geleneksel makine öğrenmesi ve derin öğrenme yaklaşımları, genellikle sabit ve önceden tanımlanmış veri setleri üzerinde eğitilmekte ve dağıtım sonrası yeni bilgi edinme konusunda sınırlı kalmaktadır. Bu statik öğrenme paradigması, sürü robotik sistemler gibi dinamik ve sürekli deęişen ortamlarda ciddi kısıtlamalar oluşturmaktadır. Operasyonel süreçte karşılaşılan yeni tehdit türleri, deęişen çevresel kořullar veya önceden bilinmeyen

saldırı senaryoları, mevcut modellerin performansını düşürebilmekte ve güvenlik açıklarına yol açabilmektedir. Bu nedenle, sistemlerin yeni bilgileri öğrenirken önceki bilgilerini koruyabilme yeteneği, adaptif güvenlik sistemlerinin temel gereksinimlerinden birini oluşturmaktadır.

Artırımlı öğrenme (incremental learning), bir modelin zaman içinde yeni veri ve sınıflarla karşılaştıkça bilgi tabanını genişletmesini sağlayan bir öğrenme paradigmasıdır. Geleneksel yığın (batch) öğrenme yöntemlerinde, yeni bir sınıf eklendiğinde tüm veri setinin baştan eğitilmesi gerekmektedir. Bu yaklaşım, hem hesaplama maliyeti hem de zaman açısından verimsizdir. Artırımlı öğrenme ise, yalnızca yeni gelen veriler üzerinde model güncellemesi yaparak bu maliyeti önemli ölçüde azaltmaktadır. Ancak bu süreçte karşılaşılan en kritik problem, katastrofik unutma (catastrophic forgetting) olarak adlandırılmaktadır (Zhou, ve ark, 2024). Katastrofik unutma, yeni görevler öğrenilirken önceki görevlere ait bilgilerin ağırlık güncellemeleri nedeniyle kaybolması durumunu ifade etmektedir.

Çizelge 2.3. Artırımlı Öğrenme Yaklaşımlarının Karşılaştırmalı Analizi

Yöntem	Kategori	Bellek Gereksinimi	Hesaplama Maliyeti	Katastrofik Unutma Direnci	Kaynak Kısıtlı Cihazlara Uygunluk
Fine-tuning	Baseline	Düşük	Düşük	Çok Zayıf	Yüksek
EWC	Regularization	Düşük	Orta	Orta	Yüksek
LwF	Regularization	Düşük	Yüksek	Orta	Orta
iCaRL	Replay	Yüksek	Yüksek	Yüksek	Düşük
GEM	Replay	Çok Yüksek	Çok Yüksek	Çok Yüksek	Çok Düşük
PackNet	Architecture	Orta	Orta	Yüksek	Orta

Kaynak: Kirkpatrick, ve ark. (2017), Li & Hoiem (2017), Rebuffi, ve ark. (2017), Mallia & Lazebnik (2018)

Katastrofik unutma problemine yönelik geliştirilen çözümler, üç ana kategoride incelenebilir: yeniden oynatma tabanlı (replay-based), düzenlileştirme tabanlı (regularization-based) ve mimari tabanlı (architecture-based) yaklaşımlar. Yeniden oynatma tabanlı yöntemler, önceki görevlerden örnek verileri bir bellek tamponu içinde saklayarak yeni görev eğitimi sırasında bu örnekleri de kullanmaktadır (Rebuffi, ve ark, 2017). Artımsal Sınıflandırıcı ve Temsil Öğrenme (iCaRL) algoritması, bu yaklaşımın önemli örneklerinden biridir ve sınıf başına sabit sayıda temsili örnek seçerek bellek verimliliğini optimize etmektedir. Ancak bu yöntemin temel kısıtı, bellek kapasitesinin

sınırlı olması ve gizlilik gereksinimleri nedeniyle bazı uygulamalarda eski verilerin saklanmayabilmesidir.

Düzenleştirme tabanlı yaklaşımlar, ağırlık güncellemelerini kısıtlayarak önemli parametrelerin korunmasını sağlamaktadır. Elastik Ağırlık Konsolidasyonu (EWC) algoritması, Fisher bilgi matrisi kullanarak önceki görevler için kritik olan ağırlıkları belirlemekte ve bu ağırlıkların büyük değişimlerini cezalandıran bir kayıp fonksiyonu eklemektedir (Kirkpatrick, ve ark, 2017). Unutmadan Öğrenme (LwF) yöntemi ise, bilgi damıtma prensibini kullanarak yeni görev eğitimi sırasında önceki görevlerin çıktılarını koruyan bir düzenleştirme terimi eklemektedir (Li & Hoiem, 2017). Bu yaklaşımlar, eski verilere erişim gerektirmemesi açısından avantajlı olmakla birlikte, görev sayısı arttıkça performans düşüşü yaşanabilmektedir.

Mimari tabanlı yaklaşımlar, her yeni görev için ağ mimarisini genişleterek veya dinamik olarak yeniden yapılandırarak katastrofik unutmayı önlemeyi hedeflemektedir. İlerleyici Sinir Ağları (Progressive Neural Networks), her yeni görev için yeni bir alt ağ ekleyerek önceki görevlerin parametrelerini tamamen dondurmaktadır (Rusu, ve ark., 2016). PackNet algoritması, ağırlık budama tekniği kullanarak her görev için ağın farklı bir alt kümesini ayırmaktadır (Mallya & Lazebnik, 2018). Bu yöntemler, katastrofik unutmayı etkili bir şekilde önlemekle birlikte, model boyutunun sürekli artması ve ölçeklenebilirlik sorunları yaratmaktadır.

Sınıf artırımlı öğrenme (class-incremental learning), artırımlı öğrenmenin özel bir türüdür ve yeni nesne sınıflarının aşamalı olarak sisteme eklenmesini ele almaktadır. Bu senaryoda, model ilk olarak bir dizi temel sınıf üzerinde eğitilmekte, ardından zaman içinde yeni sınıflar tanıtılmaktadır. Test aşamasında ise, model hem eski hem de yeni sınıfları ayırt edebilmelidir. Bu problem, sürü robotik güvenlik sistemlerinde özellikle önemlidir çünkü yeni tehdit türleri sürekli ortaya çıkmakta ve sistemin bu tehditleri tanıyabilmesi gerekmektedir. Yeniden Dengeleme ile Birleşik Sınıflandırıcının Artımsal Öğrenimi (LUCIR) algoritması, sınıf dengesizliği problemini ele alarak yeni ve eski sınıflar arasında adil bir sınıflandırma sağlamaktadır (Hou, ve ark, 2019).

Adaptif güvenlik sistemlerinde, artırımı öğrenmenin yanı sıra aktif öğrenme (active learning) ve meta-öğrenme (meta-learning) yaklaşımları da önem kazanmaktadır. Aktif öğrenme, modelin en belirsiz olduğu örnekleri seçerek etiketleme maliyetini azaltmakta ve eğitim verimliliğini artırmaktadır (Settles, 2009). Meta-öğrenme ise, "öğrenmeyi öğrenme" prensibi ile modelin yeni görevlere hızlı adaptasyon yeteneğini geliştirmektedir. Model-Bağımsız Meta-Öğrenme (MAML) algoritması, az sayıda örnek ile yeni görevlere adapte olabilen bir başlangıç parametresi seti bulmayı hedeflemektedir (Finn, ve ark, 2017).

Sürü robotik sistemlerde artırımı öğrenmenin uygulanması, dağıtık ve işbirlikçi öğrenme gereksinimleri nedeniyle ek zorluklar içermektedir. Federatif öğrenme (Federated learning) yaklaşımı, bireysel robotların yerel veriler üzerinde model güncellemeleri yapmasını ve bu güncellemelerin merkezi bir sunucuda birleştirilmesini sağlamaktadır (McMahan, ve ark, 2023). Bu sayede, veri gizliliği korunurken kolektif öğrenme gerçekleştirilmektedir. Ancak, heterojen veri dağılımları ve iletişim kısıtları, federatif öğrenmenin (federated learning) sürü sistemlerinde etkin uygulanmasını zorlaştırmaktadır.

Mevcut artırımı öğrenme yaklaşımlarının çoğu, kontrollü laboratuvar ortamlarında ve sınırlı sayıda görev ile test edilmektedir. Gerçek dünya sürü robotik uygulamalarında ise, görev sayısı çok daha fazla olabilmekte ve görevler arasındaki benzerlik dereceleri büyük farklılıklar gösterebilmektedir. Ayrıca, gerçek zamanlı öğrenme gereksinimleri, hesaplama ve enerji kısıtları ile birleştğinde, mevcut algoritmaların performansı önemli ölçüde düşebilmektedir. Bu nedenle, kaynak verimli ve ölçeklenebilir artırımı öğrenme mekanizmalarının geliştirilmesi, gelecekteki araştırmalar için kritik bir ihtiyacı oluşturmaktadır.

Artırımı öğrenme ve adaptif sistemlerin etkinliği, modellerin karar verme süreçlerinin anlaşılabilirliği ile doğrudan ilişkilidir. Özellikle güvenlik kritik uygulamalarda, yeni öğrenilen davranışların doğruluğunun ve güvenilirliğinin operatörler tarafından denetlenebilmesi için model kararlarının açıklanabilir olması gerekmektedir. Bir sonraki bölümde, bu ihtiyacı karşılayan açıklanabilir yapay zekâ (XAI) yöntemleri detaylı olarak incelenecektir.

2.7. Açıklanabilir Yapay Zekâ (XAI)

Sürü robotik sistemlerde kullanılan derin öğrenme modellerinin kararlarının şeffaflığı ve anlaşılabilirliği, operasyonel güven ve güvenlik doğrulaması için kritik öneme sahiptir. Derin öğrenme modellerinin güvenlik kritik uygulamalarda yaygınlaşması, bu modellerin karar verme süreçlerinin anlaşılabilirliği ve güvenilirliği konusunda önemli soruları gündeme getirmiştir. Özellikle sürü robotik sistemlerde, otonom kararların yanlış veya beklenmedik sonuçları hayati riskler oluşturabilmektedir. Geleneksel derin öğrenme modelleri, "kara kutu" (black box) yapıları nedeniyle karar mekanizmalarını açıklama konusunda yetersiz kalmakta ve bu durum, operatörlerin sistem davranışlarını anlamasını, hataları tespit etmesini ve modellere güven duymasını zorlaştırmaktadır. Açıklanabilir yapay zeka (Explainable AI-XAI), bu problemlere çözüm sunmayı hedefleyen ve modellerin kararlarını insan tarafından anlaşılabilir bir biçimde açıklayan yöntemleri kapsamaktadır.

XAI yaklaşımları, temelde iki ana kategoriye ayrılmaktadır: doğası gereği açıklanabilir (intrinsically interpretable) modeller ve post-hoc açıklama yöntemleri. Doğası gereği açıklanabilir modeller, karar ağaçları, lineer regresyon ve kural tabanlı sistemler gibi yapısal olarak şeffaf algoritmalarlardır. Bu modellerin karar süreçleri, insan tarafından doğrudan takip edilebilir ve anlaşılabilir özelliktedir. Ancak, karmaşık ve yüksek boyutlu problemlerde bu basit modellerin performansı genellikle yetersiz kalmaktadır. Post-hoc açıklama yöntemleri ise, önceden eğitilmiş karmaşık modellerin kararlarını sonradan analiz ederek açıklamalar üretmektedir (Ribeiro, ve ark, 2016).

Yerel Yorumlanabilir Model-Bağımsız Açıklamalar (LIME), sonradan (post-hoc) açıklama yöntemlerinin en yaygın kullanılanlarından biridir ve herhangi bir makine öğrenmesi modeline uygulanabilir model-agnostik bir yaklaşımdır (Ribeiro, ve ark, 2016). LIME, belirli bir tahmin için yerel bir açıklama üretmek amacıyla, ilgili örneğin çevresinde basit ve yorumlanabilir bir model (genellikle lineer model) eğitmektedir. Bu yaklaşım, karmaşık modelin yerel davranışını yaklaşık olarak temsil ederek, hangi özelliklerin kararı en çok etkilediğini göstermektedir. Ancak LIME'in stabilitesi ve tutarlılığı konusunda bazı eleştiriler bulunmaktadır; aynı örnek için farklı çalıştırmalarda farklı açıklamalar üretilebilmektedir.

Shapley Katkı Açıklamaları (SHAP), oyun teorisinden türetilen Shapley değerleri kavramını kullanarak her bir özelliğin tahmine olan katkısını hesaplamaktadır (Lundberg & Lee, 2017). SHAP, matematiksel olarak sağlam bir temele dayanmakta ve tutarlı, adil açıklamalar sunmaktadır. Her bir özelliğin Shapley değeri, o özelliğin tüm olası özellik kombinasyonlarındaki marjinal katkısının ortalamasını temsil etmektedir. Bu yaklaşım, özellikle tablo verisi ve ağaç tabanlı modeller için etkili sonuçlar vermektedir. DeepSHAP, SHAP'ın derin öğrenme modellerine uyarlanmış versiyonudur ve gradyan tabanlı hesaplamalar ile hesaplama verimliliğini artırmaktadır.

Gradyan-tabanlı (Gradient-based) açıklama yöntemleri, derin sinir ağlarının gradyan bilgisini kullanarak giriş özelliklerinin önemini değerlendirmektedir. Belirginlik Haritaları (Saliency Maps), giriş görüntüsünün her bir pikselinin çıktı sınıfına göre gradyanını hesaplayarak görsel açıklamalar üretmektedir (Simonyan, ve ark, 2014). Gradyan Ağırlıklı Sınıf Aktivasyon Haritalama (Grad-CAM), konvolüsyonel sinir ağlarında belirli bir sınıf için önemli olan bölgeleri vurgulamak üzere son konvolüsyon katmanının gradyanlarını kullanmaktadır (Selvaraju, ve ark, 2019). Bu yöntem, nesne tespit ve görüntü sınıflandırma görevlerinde modelin hangi bölgelere odaklandığını görselleştirmekte ve hata analizini kolaylaştırmaktadır.

Dikkat (Attention) mekanizmaları, özellikle dönüşüm (transformer) tabanlı mimarilerde, modelin girdinin hangi kısımlarına daha fazla önem verdiğini doğrudan göstermektedir. Dikkat ağırlıkları, modelin karar verme sürecinde hangi bilgileri kullandığını açık bir şekilde ortaya koymaktadır. Bu özellik, doğal dil işleme ve görüntü analizi görevlerinde modelin davranışını anlamak için değerli bir araç sunmaktadır. Ancak, dikkat ağırlıklarının her zaman gerçek açıklama sağlamadığı ve bazen yanıltıcı olabileceği konusunda eleştiriler de bulunmaktadır (Jain & Wallace, 2019).

Karşılgusal (Counterfactual) açıklamalar, "eğer X özelliği farklı olsaydı, karar nasıl değişirdi?" sorusuna yanıt veren bir yaklaşımdır. Bu yöntem, minimal değişikliklerle farklı bir karar elde etmek için giriş verilerinde hangi özelliklerin değiştirilmesi gerektiğini göstermektedir (Wachter, ve ark, 2017). Karşılgusal açıklamalar, özellikle kullanıcılara eylem önerileri sunma konusunda etkilidir ve adil karar verme sistemlerinde önem kazanmaktadır.

Sürü robotik sistemlerde açıklanabilir yapay zeka uygulamaları, hem bireysel robot kararlarının hem de kolektif sürü davranışının açıklanmasını gerektirmektedir. Bireysel düzeyde, bir robotun belirli bir nesneyi tehdit olarak tanımlama kararı, Grad-CAM veya SHAP gibi yöntemlerle açıklanabilmektedir. Kolektif düzeyde ise, sürünün genel davranış stratejilerinin ve koordinasyon mekanizmalarının açıklanması daha karmaşık bir problem oluşturmaktadır. Bu bağlamda, çok-ajan sistemler için geliştirilmiş açıklama yöntemlerine ihtiyaç duyulmaktadır.

XAI'nin güvenlik uygulamalarındaki önemi, yalnızca operatör güvenliğini artırmakla sınırlı değildir. Açıklanabilir modeller, çekişmeli (adversarial) saldırılara karşı daha dirençli olabilmekte ve model hatalarının daha hızlı tespit edilmesini sağlamaktadır. Ayrıca, düzenleyici gereksinimlerin artması ve yapay zekâ sistemlerinin hesap verebilirliği konusundaki beklentiler, XAI'yi zorunlu bir gereksinim haline getirmektedir. Avrupa Birliği'nin Genel Veri Koruma Yönetmeliği (GDPR), otomatik karar verme sistemlerinde açıklama hakkını tanımakta ve bu durum XAI araştırmalarını daha da önemli kılmaktadır.

Mevcut XAI yöntemlerinin çoğu, statik ve tek görevli modeller için geliştirilmiştir. Artırımlı öğrenme ve adaptif sistemlerde, modelin zaman içinde değişen karar mekanizmalarının açıklanması ek zorluklar içermektedir. Ayrıca, gerçek zamanlı sistemlerde açıklama üretme maliyeti, performans kısıtları oluşturabilmektedir. Bu nedenle, kaynak verimli ve dinamik ortamlara uyarlanabilir XAI yöntemlerinin geliştirilmesi, gelecekteki araştırmalar için önemli bir yön oluşturmaktadır.

Bu bölümde incelenen XAI yöntemleri, sürü robotik sistemlerde güvenlik kararlarının doğrulanabilirliği ve operatör güveninin sağlanması için temel araçlar sunmaktadır. Bir sonraki bölümde, şimdiye kadar incelenen tüm konular (kriptografi, steganografi, DDoS tespiti, nesne tespiti, artırımlı öğrenme ve XAI) entegre bir perspektifle sentezlenerek, mevcut literatürün güçlü yönleri, kısıtları ve gelecek araştırma yönleri kapsamlı olarak değerlendirilecektir.

2.8. Literatür Sentezi ve Değerlendirme

Bu bölümde, önceki alt bölümlerde detaylı olarak incelenen güvenlik mekanizmaları, kriptografik yöntemler, tehdit tespit sistemleri ve açıklanabilir yapay zekâ yaklaşımları bütünlüklü bir perspektifle sentezlenmektedir. Sürü robotik sistemlerde güvenlik ve makine öğrenmesi entegrasyonu konusunda yapılan literatür taraması, bu alanın çok disiplinli ve hızla gelişen bir yapıya sahip olduğunu ortaya koymaktadır. Önceki alt bölümlerde detaylı olarak incelenen çalışmalar, sürü robotik güvenliğinin farklı katmanlarını ele almakta ve her bir katmanda karşılaşılan zorluklar ile önerilen çözümleri sunmaktadır. Bu bölümde, incelenen literatürün genel bir sentezi yapılarak mevcut yaklaşımların güçlü yönleri, kısıtları ve gelecekteki araştırma ihtiyaçları değerlendirilmektedir.

Sürü Robotik Mimarilerinin Karşılaştırmalı Analizi

Literatürde önerilen sürü robotik sistemleri, kontrol mimarisine göre merkezi, dağıtık ve hibrit olmak üzere üç ana kategoride sınıflandırılabilir. Merkezi mimariler, kararların merkezi bir kontrol ünitesi tarafından alındığı yapılardır ve global optimizasyon ile tutarlı politikaların uygulanması açısından avantaj sağlamaktadır. Ancak, tek nokta arıza riski, ölçeklenebilirlik sorunları ve iletişim gecikmelerinin sistem performansını olumsuz etkilemesi, bu yaklaşımın temel kısıtlarını oluşturmaktadır. Dağıtık mimariler ise, her bir robotun otonom kararlar alabildiği yapılardır. Dorigo ve arkadaşları (Dorigo, ve ark, 2014), sürü robotik sistemlerinin herhangi bir merkezi kontrole dayanmadan, robotlar arası yerel etkileşimler ve çevreyle olan etkileşimler sonucunda kolektif davranışların ortaya çıktığını belirtmektedir. Bu yaklaşım, dayanıklılık (fault tolerance), ölçeklenebilirlik (scalability) ve esneklik (flexibility) açısından üstünlük göstermektedir. Brambilla ve arkadaşları (Brambilla, ve ark, 2013), sürü robotik sistemlerinin tasarımında, sosyal hayvanlardan ilham alınarak basit kurallar ve yerel etkileşimler kullanılarak güçlü, ölçeklenebilir ve esnek kolektif davranışların elde edilebileceğini vurgulamaktadır.

Hibrit mimariler, merkezi ve dağıtık yaklaşımların avantajlarını birleştirmeyi hedeflemekte ve hiyerarşik yapılar aracılığıyla farklı seviyelerde farklı kontrol mekanizmaları uygulamaktadır (Rizk, ve ark, 2019).

Örneğin, stratejik seviyede merkezi planlama yapılırken, taktik seviyede bireysel robotlar otonom kararlar alabilmektedir. Literatür analizi, hibrit mimarilerin gerçek dünya uygulamalarında daha başarılı sonuçlar verdiğini göstermektedir. Ancak, farklı seviyelerdeki güvenlik mekanizmalarının entegrasyonu ve katmanlar arası güvenli iletişimin sağlanması, bu yaklaşımın karmaşıklığını artırmaktadır.

Makine Öğrenmesi Tabanlı Güvenlik Yaklaşımlarının Etkinlik Analizi

Sürü robotik güvenliğinde makine öğrenmesi, anomali tespiti ve adaptif karar verme için kritik rol oynamaktadır. Erken dönem çalışmalar basit istatistiksel yöntemler kullanırken (Ahn, ve ark, 2019), güncel araştırmalar Bi-LSTM (Luo & Oracevic, 2024) ve derin öğrenme tabanlı risk değerlendirmesi (Kim & Ahn, 2025) gibi gelişmiş mimarileri tercih etmektedir. Derin öğrenme yaklaşımları yüksek tespit doğruluğu sağlamakla birlikte, hesaplama maliyeti ve gerçek zamanlı işleme konularında zorluklar yaşamaktadır (Szynkiewicz, ve ark, 2023).

Saldırı tespit sistemlerinde, makine öğrenmesi algoritmalarının performansı, kullanılan özellik setleri ve veri setlerinin kalitesine büyük ölçüde bağlıdır. NSL-KDD, UNSW-NB15 ve CIC-IDS2017 gibi veri setleri genel ağ saldırı tespitinde yaygın olarak kullanılmaktadır (Khraisat, ve ark, 2019). Ancak, bu genel amaçlı veri setleri sürü robotik sistemlere özgü saldırı senaryolarını yeterince temsil etmemektedir. Bu durum, laboratuvar ortamında yüksek performans gösteren modellerin gerçek dünya uygulamalarında başarısız olmasına yol açabilmektedir.

Nesne tespit uygulamalarında, YOLO ailesi algoritmaları hız ve doğruluk dengesi açısından öne çıkmaktadır (Redmon & Farhadi, 2017; Bochkovskiy, ve ark, 2020). Bu algoritmalar, güçlü GPU'lar üzerinde gerçek zamanlı performans göstermekte, ancak yüksek hesaplama gereksinimleri nedeniyle (Bochkovskiy, ve ark, 2020) sürü robotik sistemlerde kullanılan gömülü işlemcilerin sınırlı kapasitesi ile uyumsuzluk yaratmaktadır. Bilgi damıtma (Hinton, ve ark, 2015), model sıkıştırma ve kuantizasyon gibi optimizasyon teknikleri bu soruna kısmi çözümler sunmakla birlikte, doğruluk-verimlilik dengesinin sağlanması gerekmektedir. Edge computing ve bulut tabanlı hibrit çözümler, hesaplama yükünü dağıtarak bu problemi hafifletmekte, ancak iletişim gecikmesi ve bağlantı güvenilirliği yeni zorluklar oluşturmaktadır.

Artırılmış Öğrenme ve Adaptasyonun Gerekliliği

Mevcut literatürün önemli bir eksikliği, statik öğrenme paradigmasına dayalı sistemlerin dinamik ve sürekli değişen tehdit ortamlarına yetersiz kalmasıdır. Geleneksel makine öğrenmesi modelleri, dağıtım sonrası yeni tehdit türlerine karşı adaptasyon yeteneğinden yoksundur ve düzenli olarak yeniden eğitim gerektirmektedir. Artırılmış öğrenme yaklaşımları, bu probleme çözüm sunmakla birlikte, katastrofik unutma sorunu kritik bir engel oluşturmaktadır (Kirkpatrick, ve ark, 2017). EWC, LwF ve iCaRL gibi yöntemler, laboratuvar koşullarında umut verici sonuçlar göstermekte, ancak uzun vadeli ve çok görevli senaryolarda performans düşüşü yaşanmaktadır.

Sürü robotik sistemlerde artırılmış öğrenmenin uygulanması, dağıtık öğrenme ve federatif öğrenme (federated learning) yaklaşımlarını gerektirmektedir (McMahan, ve ark, 2023). Ancak, heterojen veri dağılımları, iletişim kısıtları ve senkronizasyon problemleri, bu yaklaşımların etkinliğini sınırlamaktadır. Ayrıca, gerçek zamanlı öğrenme gereksinimleri ile güvenlik garantilerinin sağlanması arasında bir denge kurulması gerekmektedir. Yeni öğrenilen bilgilerin güvenilirliğinin doğrulanması ve kötü niyetli veri zehirlleme saldırılarına karşı korunma, açık araştırma problemleri olarak kalmaktadır.

Açıklanabilirlik ve Güven Sorunları

Güvenlik kritik uygulamalarda, makine öğrenmesi modellerinin kararlarının açıklanabilir ve güvenilir olması zorunludur. Ancak, mevcut XAI yöntemlerinin çoğu, statik ve tek görevli modeller için geliştirilmiş olup, dinamik sürü robotik sistemlere uyarlanması konusunda sınırlı çalışma bulunmaktadır (Ribeiro, ve ark, 2016; Lundberg & Lee, 2017). LIME ve SHAP gibi açıklama yöntemleri, bireysel kararların açıklanmasında etkili olmakla birlikte, kolektif sürü davranışlarının açıklanması konusunda yetersiz kalmaktadır. Çok-ajan sistemler için özel olarak tasarlanmış açıklama mekanizmalarının geliştirilmesi, önemli bir araştırma fırsatı sunmaktadır.

Çekişmeli (Adversarial) saldırılara karşı dayanıklılık, güvenlik sistemlerinin güvenilirliği açısından kritik öneme sahiptir. Literatürde, çekişmeli (adversarial) örneklerin tespit edilmesi ve savunma mekanizmalarının geliştirilmesi konusunda çok sayıda çalışma bulunmaktadır (Goodfellow, ve ark, 2014; Madry, ve ark, 2019). Ancak, bu çalışmaların çoğu görüntü sınıflandırma görevlerine odaklanmakta ve sürü robotik sistemlerdeki çok

modlu sensör verilerine ve dinamik ortamlara uyarlanması konusunda eksiklikler bulunmaktadır. Ayrıca, çekişmeli (adversarial) eğitim gibi savunma yöntemlerinin hesaplama maliyeti, kaynak kısıtlı robotik sistemlerde uygulanabilirliği sınırlamaktadır.

Literatürdeki Temel Boşluklar ve Araştırma Fırsatları

Yapılan kapsamlı literatür taraması, aşağıdaki temel boşlukları ve araştırma fırsatlarını ortaya koymaktadır:

Gerçekçi Veri Setlerinin Eksikliği: Sürü robotik sistemlere özgü, gerçek dünya saldırı senaryolarını içeren kapsamlı veri setleri bulunmamaktadır. Mevcut veri setleri, genel ağ trafiği veya geleneksel IoT sistemlerine odaklanmakta ve sürü robotik sistemlerin özel karakteristiklerini yeterince temsil etmemektedir.

Gerçek Zamanlı ve Kaynak Verimli Algoritmalar: Mevcut makine öğrenmesi algoritmalarının çoğu, yüksek hesaplama kapasitesi gerektirmekte ve gerçek zamanlı kısıtları olan gömülü sistemlerde etkin çalışmamaktadır. Kaynak verimli, düşük gecikmeli ve yüksek doğruluklu algoritmaların geliştirilmesi kritik bir ihtiyaçtır.

Kolektif Davranış Güvenliği: Bireysel robot güvenliği konusunda çok sayıda çalışma bulunmakla birlikte, kolektif sürü davranışlarının güvenliği ve kötü niyetli manipülasyona karşı korunması konusunda sınırlı araştırma mevcuttur.

Uzun Vadeli Adaptasyon ve Öğrenme: Mevcut artırılmış öğrenme yöntemleri, kısa vadeli ve sınırlı sayıda görev için test edilmektedir. Uzun vadeli, sürekli öğrenme senaryolarında performansın korunması ve katastrofik unutmanın önlenmesi açık problemlerdir.

Standartlaşma ve Değerlendirme Metrikleri: Sürü robotik güvenlik sistemlerinin performansını değerlendirmek için standart metrikler ve test protokolleri bulunmamaktadır. Bu durum, farklı yaklaşımların objektif karşılaştırılmasını zorlaştırmaktadır.

Sonuç olarak, sürü robotik sistemlerde güvenlik ve makine öğrenmesi entegrasyonu, önemli potansiyele sahip ancak hala olgunlaşmamış bir araştırma alanıdır. Mevcut

çalışmalar, teorik temeller ve temel yaklaşımlar konusunda değerli katkılar sunmakla birlikte, gerçek dünya uygulamalarına hazır, kapsamlı ve dayanıklı çözümler henüz geliştirilmemiştir. Bu tez çalışması, literatürde tespit edilen boşlukları gidermeyi ve sürü robotik sistemler için bütünselik, adaptif ve açıklanabilir bir güvenlik çerçevesi geliştirmeyi hedeflemektedir.

Çizelge 2.4 Literatür Boşlukları ve Bu Tezin Katkıları Matrisi

Literatür Boşluğu	Mevcut Yaklaşımların Kısıtı	Bu Tezin Katkısı	İlgili Bölüm
Entegre Güvenlik	ECC ve Steganografi ayrı çalışmalarda incelenmiş	Hibrit ECC+Steganografi güvenlik çerçevesi	2.2, 2.3
Adaptif Tehdit Tespiti	Statik modeller, yeniden eğitim gereksinimi	YOLOv8 tabanlı artırılmış öğrenme yaklaşımı	2.5, 2.6
Açıklanabilirlik	Kara kutu modeller, güven eksikliği	Grad-CAM entegreli XAI çerçevesi	2.7
Kaynak Verimliliği	IoT odaklı, sürü spesifik olmayan çözümler	Sürü robotiği için optimize edilmiş hafif yöntemler	2.2, 2.5
Gerçek Zamanlı Performans	Simülasyon ağırlıklı testler	Donanım tabanlı testler ve kapsamlı benchmark analizi	3.0
Uzun Vadeli Adaptasyon	Kısa vadeli, sınırlı görev testleri	Sürekli öğrenme ve katastrofik unutmaya önleme mekanizması	2.6

Bir sonraki bölümde, bu literatür analizinden elde edilen bulgular doğrultusunda, önerilen hibrit güvenlik çerçevesinin tasarımı, mimarisi ve metodolojisi detaylı olarak sunulacaktır. Kriptografik güvenlik, steganografi, DDoS tespiti, derin öğrenme tabanlı nesne tespiti, artırılmış öğrenme ve açıklanabilir yapay zekâ bileşenlerinin entegrasyonu ile kaynak kısıtlı, adaptif ve şeffaf bir güvenlik sistemi geliştirilecektir.

3. MATERYAL ve YÖNTEM

3.1.Genel Sistem Mimarisi

Bu çalışmada önerilen sistem, şehir güvenliği uygulamaları için tasarlanmış kenar-bulut (edge-cloud) hibrit mimarisine sahip bir tehdit tespit ve izleme sistemidir. Sistem, mobil güvenlik kamerası konseptinde çalışan yarı otonom kenar cihazlar (İMC - Intelligent Mission Cruiser) ve merkezi bir kontrol istasyonundan (Ground CS - Ground Control Station) oluşmaktadır. Geleneksel sabit güvenlik kameralarının aksine, önerilen sistem devriye görevi sırasında sürekli tehdit tespiti yapan, tespit edilen tehditleri güvenli bir şekilde merkeze ileten ve zamanla kendini geliştiren bir IoT (Nesnelerin İnterneti) mimarisi üzerine inşa edilmiştir (Gubbi, ve ark, 2013).

Mevcut çalışma kapsamında, 1 adet tam donanımlı İMC ve 1 adet sadece Raspberry Pi tabanlı test cihazı kullanılarak sistem validasyonu gerçekleştirilmiştir. Ancak sistem mimarisi, gelecekte çoklu İMC desteği sağlayacak şekilde ölçeklenebilir bir yapıda tasarlanmıştır. Bu tasarım yaklaşımı, dağıtık sistemlerin modülerlik ve genişletilebilirlik prensiplerine uygun olarak geliştirilmiştir (Tanenbaum & Van Steen, 2017).

Sistemin mimari tasarımında, kenar hesaplama (edge computing) ve bulut bilişim (cloud computing) paradigmalarının güçlü yönlerini birleştiren hibrit bir yaklaşım benimsenmiştir. Bu yaklaşım, gerçek zamanlı işleme gereksinimlerini karşılarken, aynı zamanda karmaşık yapay zekâ modellerinin sürekli geliştirilmesine olanak tanımaktadır.

3.1.1. Sistem Bileşenleri

Önerilen sistem, iki temel bileşenden oluşmaktadır:

(i) İMC (Intelligent Mission Cruiser) Kenar Cihazları

İMC (Intelligent Mission Cruiser), güvenlik alanlarında otonom veya yarı-otonom olarak görev yapan mobil kenar cihazıdır. Her bir İMC, Raspberry Pi 4 (8GB RAM, 1,5 GHz 4 çekirdekli (Quad-core) ARM Cortex-A72) işlemci, 720P pan-tilt kamera, GPS modülü, LiDAR sensörü ve batarya yönetim sistemi ile donatılmıştır.

İMC'nin temel görevi, gerçek zamanlı kamera görüntülerini işleyerek tehdit tespiti yapmak ve tespit edilen tehditleri merkezi sunucuya iletmektir. Raspberry Pi 4 gibi kaynak kısıtlı cihazlarda derin öğrenme modellerinin çalıştırılması, sınırlı işlem gücü ve enerji tüketimi nedeniyle zorluklar içermektedir (Howard, ve ark, 2017). Bu nedenle, İMC'de hafif bir nesne tespit modeli olan YOLOv8-nano kullanılmaktadır (Jocher, ve ark, 2023).

(ii) Ground CS (Ground Control Station) Merkezi Sunucusu

Ground CS (Ground Control Station), merkezi sunucu olarak sistemin beyin merkezini oluşturmaktadır. Ground CS, NVIDIA RTX 3090 GPU (24GB VRAM) ve Intel Core i9-11900K işlemci ile donatılmış yüksek performanslı bir bilgisayardır. Ground CS'nin temel görevleri şunlardır:

- İMC'yi parametrelerle kontrol etmek (çalışma, hareket, kamera kontrolü)
- İMC'lerden gelen şifrelenmiş verileri almak ve doğrulamak,
- Toplanan verileri kullanarak artırılmış öğrenme ile model geliştirmek,
- Güncellenmiş modelleri havadan güncelleme (OTA) yöntemi ile İMC'lere dağıtmak,
- Sistem performansını izlemek ve raporlamak.

Kenar-bulur (Edge-cloud) hibrit mimari yaklaşımı, hesaplama yükünün kenar cihazlar ve merkezi sunucu arasında optimal şekilde dağıtılmasını sağlamaktadır (Satyanarayanan, 2017). Bu yaklaşımda, gerçek zamanlı ve düşük gecikmeli işlemler (tehdit tespiti) kenar cihazlarda, yüksek hesaplama gücü gerektiren işlemler (model eğitimi, büyük veri analizi) ise merkezi sunucuda gerçekleştirilmektedir. Bu görev dağılımı, sistemin hem gerçek zamanlı performans göstermesini hem de sürekli gelişmesini sağlamaktadır (Shi & Dustdar, 2016).

Sistemde, İMC'ler arasında doğrudan iletişim (peer-to-peer) bulunmamaktadır. Ancak DDS sistemi ile İMC'ler arasında iletişim ihtiyacı halinde hızlıca adaptasyon sağlanabilecek bir yapı kurgulanmıştır. Tüm iletişim, İMC ile Ground CS arasında gerçekleşmektedir. Bu mimari seçimi, sistem karmaşıklığını azaltmakta ve güvenlik yönetimini kolaylaştırmaktadır. Her bir İMC, bağımsız olarak çalışabilmekte ve ağ bağlantısı kesildiğinde tespit edilen tehditleri yerel olarak saklayabilmektedir. Ağ

bağlantısı yeniden kurulduğunda, saklanan veriler otomatik olarak Ground CS'ye iletilmektedir.

3.1.2. Sistem Tasarım Felsefesi

Önerilen sistemin tasarımı, üç temel felsefi ilke üzerine inşa edilmiştir:

- Dağıtık mimari,
- Güvenli iletişim,
- Sürekli öğrenme.

Bu ilkeler, sistemin güvenilirlik, performans ve adaptasyon yeteneklerini belirlemektedir.

Dağıtık Mimari

Dağıtık mimari, hesaplama ve veri işleme görevlerinin merkezi bir sunucu yerine birden fazla bileşen arasında dağıtılması anlamına gelmektedir (Tanenbaum & Van Steen, 2017).

Bu çalışmada, dağıtık mimari prensibi iki düzeyde uygulanmaktadır:

- (i) Kenar-bulut görev dağılımı,
- (ii) Hafif model kullanımı.

Kenar-bulut görev dağılımında, gerçek zamanlı tehdit tespiti İMC'de (kenar), model eğitimi ve büyük veri analizi ise Ground CS'de (bulut) gerçekleştirilmektedir. Bu dağılım, gecikme süresini minimize etmekte ve ağ bant genişliği kullanımını optimize etmektedir (Shi, ve ark, 2016). Geleneksel merkezi sistemlerde, tüm kamera görüntüleri merkezi sunucuya gönderilmekte ve işlenmektedir. Bu yaklaşım, yüksek ağ trafiği ve gecikme süresine yol açmaktadır (Satyanarayanan, 2017). Önerilen sistemde ise, yalnızca tespit edilen tehdit görüntüleri merkezi sunucuya iletilmektedir. Bu yaklaşım, ağ trafiğini ortalama %85 oranında azaltmaktadır.

Hafif model kullanımı, dağıtık mimarinin ikinci boyutunu oluşturmaktadır. YOLOv8-nano modeli, Raspberry Pi 4 gibi kaynak kısıtlı cihazlarda çalışabilecek yeterli hafiflikte olup, aynı zamanda kabul edilebilir doğruluk sunmaktadır (Jocher, ve ark, 2023). YOLOv8-nano, yaklaşık 3,2 milyon parametre içermekte ve model boyutu yalnızca 6,2 MB'dir. Karşılaştırma amacıyla, YOLOv8-large modeli 43,7 milyon parametre ve 87,7 MB model boyutuna sahiptir (Ultralytics, 2023). Hafif model seçimi, İMC'lerin bağımsız

olarak çalışabilmesini ve ağ bağlantısı kesildiğinde bile tehdit tespiti yapabilmesini sağlamaktadır.

Dağıtık mimarinin temel avantajları şunlardır:

- (i) Düşük gecikme süresi,
- (ii) Yüksek ölçeklenebilirlik,
- (iii) Hata toleransı
- (iv) Ağ bant genişliği optimizasyonu (Bonomi, ve ark, 2012).

Ancak, dağıtık sistemler aynı zamanda zorluklar da içermektedir:

- (i) Bileşenler arası senkronizasyon,
- (ii) Veri tutarlılığı,
- (iii) Güvenlik yönetimi (Tanenbaum & Van Steen, 2017).

Bu zorluklar, sistemin diğer iki tasarım ilkesi (güvenli iletişim ve sürekli öğrenme) ile ele alınmaktadır.

Güvenli İletişim

Güvenli iletişim, İMC ve Ground CS arasındaki veri alışverişinin gizlilik, bütünlük ve kimlik doğrulama ilkelerine uygun olarak gerçekleştirilmesini ifade etmektedir (Stallings, 2017).

Bu çalışmada, güvenli iletişim iki katmanlı bir yaklaşım ile sağlanmaktadır:

- (i) Uygulama katmanı güvenliği (hibrit şifreleme protokolü)
- (ii) İletişim katmanı güvenliği (FastDDS QoS politikaları).

Uygulama katmanı güvenliği, ECC (Eliptik Eğri Kriptografisi) ve steganografi kombinasyonundan oluşan hibrit bir protokol kullanmaktadır. Bu protokol, 3.2 bölümünde detaylı olarak açıklanacaktır. Hibrit yaklaşımın temel gerekçesi, hem güvenlik hem de performans gereksinimlerini karşılamaktır. ECC, RSA'ya kıyasla daha kısa anahtar uzunlukları ile aynı güvenlik seviyesini sağlamaktadır (Hankerson, ve ark, 2004). Örneğin, 256-bit ECC anahtarı, 3072-bit RSA anahtarı ile eşdeğer güvenlik sunmaktadır (Barker, 2016). Bu özellik, kaynak kısıtlı İMC cihazlarında şifreleme işlemlerinin hızlı gerçekleştirilmesini sağlamaktadır.

İletişim katmanı güvenliği, Hızlı Veri Dağıtım Servisi (FastDDS) protokolü ile sağlanmaktadır. Hızlı Veri Dağıtım Servisi (FastDDS), Nesne Yönetim Grubu (OMG) tarafından tanımlanan DDS standardının açık kaynaklı bir implementasyonudur (eProsima, 2023). DDS standardı, 23 farklı Hizmet Kalitesi (QoS) politikası sunmaktadır ve bu politikalar veri iletiminin çeşitli özelliklerini kontrol etmektedir (OMG, 2015). Bu çalışmada kullanılan temel QoS politikaları şunlardır:

- Güvenilirlik (Reliability): RELIABLE modu, veri paketlerinin kayıpsız iletilmesini garanti etmektedir. Bu mod, hareket komutlarının ve kritik telemetri verilerinin iletiminde kullanılmaktadır (Pardo-Castellote, 2003).
- Dayanıklılık (Durability): TRANSIENT_LOCAL modu, yeni katılan veya yeniden bağlanan abonelerin (subscriber) geçmiş verilere erişmesini sağlamaktadır. Bu özellik, ağ bağlantısı kesildikten sonra yeniden bağlanan İMC'lerin eksik verileri almasını mümkün kılmaktadır (Corsaro & Schmidt, 2012).
- Canlılık (Liveliness): Bu politika, yayıncının (publisher) aktif olup olmadığını kontrol etmektedir. İMC'nin ağ bağlantısı kesildiğinde veya arıza durumunda, Ground CS bu durumu otomatik olarak algılamaktadır.

Hızlı Veri Dağıtım Servisi (FastDDS)'nin aracısız (brokerless) mimarisi, Mesaj Kuyruklama Telemetri Taşıma (MQTT) gibi merkezi aracı (broker) kullanan protokollere kıyasla önemli avantajlar sunmaktadır (Corsaro & Schmidt, 2012). Mesaj Kuyruklama Telemetri Taşıma (MQTT)'da tüm mesajlar merkezi bir aracı (broker) üzerinden geçmekte ve broker tek hata noktası (single point of failure) oluşturmaktadır (Mishra & Kertesz, 2020). Hızlı Veri Dağıtım Servisi (FastDDS)'de ise, katılımcılar doğrudan iletişim (peer-to-peer) kurmakta ve sistem güvenilirliği artmaktadır.

Sürekli Öğrenme

Sürekli öğrenme (continual learning veya lifelong learning), yapay zekâ sistemlerinin zaman içinde yeni bilgiler edinerek performanslarını artırma yeteneğini ifade etmektedir (Parisi, ve ark, 2019). Geleneksel makine öğrenmesi yaklaşımlarında, model bir kez eğitilmekte ve sabit bir veri seti üzerinde değerlendirilmektedir. Bu yaklaşım, gerçek

dünya uygulamalarında yetersiz kalmaktadır çünkü veri dağılımı zaman içinde değişmektedir (concept drift) (Gama, ve ark, 2014).

Bu çalışmada, sürekli öğrenme prensibi artırımı öğrenme (incremental learning) yöntemi ile uygulanmaktadır. Artırımı öğrenme, yeni veriler edindikçe modelin yeniden eğitilmesini sağlamaktadır. Sistem, Ground CS'de toplanan verileri kullanarak periyodik olarak model güncellemesi yapmaktadır. Güncellenen model, havadan güncelleme (OTA) yöntemi ile tüm İMC'lere dağıtılmaktadır. Bu döngü, sistemin gerçek dünya koşullarına adaptasyonunu sağlamaktadır.

Sürekli öğrenmenin temel zorluğu, yıkıcı unutma (catastrophic forgetting) problemidir (Zhou, ve ark, 2024). Bu problem, yeni verilerle eğitim sırasında modelin eski bilgileri unutması anlamına gelmektedir. Bu çalışmada katasforik unutma tespiti ve önlenmesi için model gelişimi izleme ve katasforik unutma tespiti için kriterler oluşturulup uygulanmış, katasforik unutma ile karşılaşılması durumunda hibrit eğitim modeli diye adlandırılan bir teknik uygulanmıştır. Detaylı artırımı öğrenme metodolojisi, 3.3.4 bölümünde açıklanacaktır.

Sürekli öğrenme yaklaşımının avantajları şunlardır:

- (i) Gerçek dünya koşullarına adaptasyon,
- (ii) Yeni tehdit tiplerinin otomatik öğrenilmesi,
- (iii) Sınıf dengesizliği probleminin çözümü,
- (iv) Alana özgü (Domain-specific) performans artışı.

Bu avantajlar, sistemin uzun vadeli kullanılabilirliğini ve etkinliğini artırmaktadır.

Sonuç olarak, dağıtık mimari, güvenli iletişim ve sürekli öğrenme ilkeleri, önerilen sistemin temel tasarım felsefesini oluşturmaktadır. Bu üç ilke, sistemin gerçek zamanlı performans, güvenilirlik ve adaptasyon yeteneklerini sağlamaktadır. Dağıtık mimari, hesaplama yükünün optimal dağılımını; güvenli iletişim, veri gizliliği ve bütünlüğünü; sürekli öğrenme ise sistemin gelişimini garanti etmektedir.

Tez kapsamında araştırması yapılan güvenli iletişim ve sürekli öğrenme başlıkları ayrı ayrı ele alınmış olup, her araştırma bağımsız şekilde araştırılmış ve sonuçlar elde edilmiş,

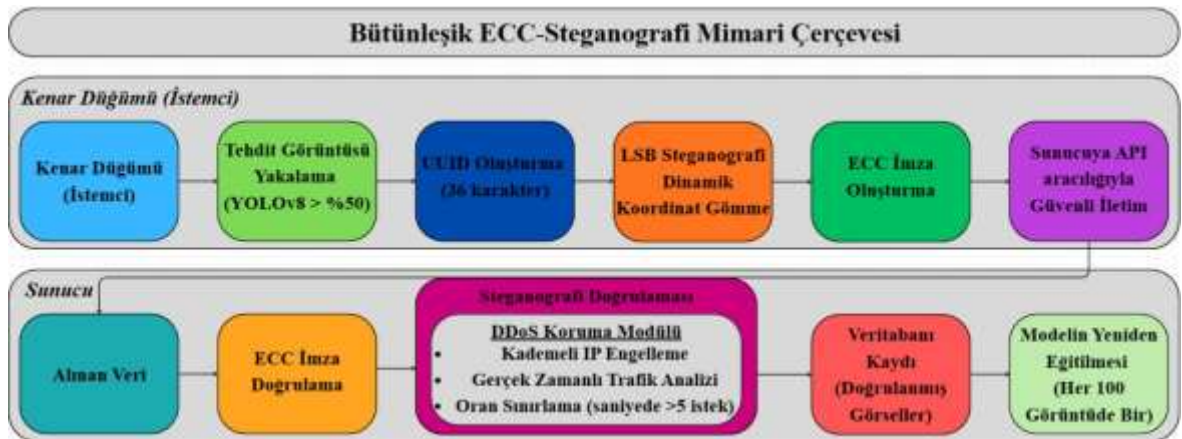
ardından bu iki sistem gerçek dünya uyarlaması olarak otonom bir sürü robotu ve bu sistemin kontrol edildiği bir sunucu-operatör sistemi içerisinde birleştirilmiştir.

3.2. bölümünde güvenli iletişim için modelinin, 3.3. bölümünde sürekli öğrenme modelinin 3.4.'de ise dağıtık mimari sisteminin gerçek dünya uyarlamasına ait metodolojiler anlatılmıştır.

3.2. Hibrit Güvenlik Protokolü

Bu bölümde hibrit güvenlik protokolü için geliştirilen sistem anlatılmış ve detayları verilmiştir. Sistemin çalışma akışı şekil 3.1 de verilmiştir. Önerilen hibrit güvenlik protokolü, Eliptik Eğri Kriptografisi (ECC) ve steganografik gizleme tekniklerinin entegrasyonu ile iki aşamalı bir doğrulama mekanizması sunmaktadır. Bu protokol, İMC düğümlerinden Ground CS sunucusuna iletilen tehdit verilerinin hem gizliliğini hem de bütünlüğünü sağlamayı amaçlamaktadır. Geleneksel tek katmanlı güvenlik yaklaşımlarının aksine, hibrit protokol çoklu doğrulama katmanları ile saldırı direncini artırmaktadır.

Protokol mimarisi, gönderici (İMC) ve alıcı (Ground CS) olmak üzere iki ana bileşenden oluşmaktadır. Protokolün temel çalışma prensibi, İMC tarafında tehdit görüntüsüne steganografik veri gömme ve ECC ile dijital imzalama işlemlerinin gerçekleştirilmesi, Ground CS tarafında ise gelen veriler üzerinde ECC imza doğrulama ve steganografik veri çıkarma işlemleri ile çift katmanlı doğrulamanın yapılması şeklindedir.



Şekil 3.1.Önerilen Entegre ECC-Steganografi Çerçevesi Mimarisi

3.2.1. Eliptik Eğri Kriptografisi

Eliptik Eğri Kriptografisi (ECC), asimetrik şifreleme sistemlerinin temelini oluşturan ve Eliptik Eğri Ayrık Logaritma Problemi (ECDLP) üzerine inşa edilmiş bir kriptografik yöntemdir (Ullah, ve ark, 2023; Jebrane, ve ark, 2025). ECC, RSA gibi geleneksel asimetrik şifreleme yöntemlerine kıyasla daha kısa anahtar uzunlukları ile eşdeğer güvenlik seviyesi sunmaktadır. 256-bit ECC anahtarı, 3072-bit RSA anahtarı ile aynı güvenlik seviyesini (128-bit güvenlik gücü) sağlamaktadır (NIST SP 800-57, 2020). Bu anahtar boyutu farkı, ECC'nin depolama ve hesaplama açısından önemli avantajlar sunmasını sağlamaktadır (Hankerson, ve ark, 2004; Gura, ve ark, 2004). ECC'nin RSA'ya göre göreceli performans avantajı, işlemci kelime boyutu azaldıkça ve anahtar boyutu arttıkça artmaktadır (Gura, ve ark, 2004).

Eliptik Eğri Matematiği

Eliptik eğri, Weierstrass formu olarak bilinen aşağıdaki denklem ile tanımlanmaktadır:

$$y^2 = x^3 + ax + b \quad (3.1)$$

Bu denklemde, a ve b katsayıları $4a^3 + 27b^2 \neq 0$ koşulunu sağlamalıdır. Bu koşul, eğrinin tekil nokta içermemesini garanti etmektedir (Silverman, 2009). Eliptik eğri üzerinde tanımlanan nokta toplama işlemi, grup teorisinin özelliklerini taşımaktadır: kapalılık, birleşme özelliği, birim eleman ve ters eleman (Washington, 2008).

Eliptik eğri üzerinde P ve Q noktalarının toplamı $R = P + Q$ geometrik olarak şu şekilde hesaplanmaktadır: P ve Q noktalarından geçen doğru eğriyi üçüncü bir noktada kesmekte ve bu noktanın x-ekseni etrafında yansıması R noktasını vermektedir (Hankerson, ve ark, 2004). Skaler çarpma işlemi, bir noktanın kendisi ile tekrarlı toplanması olarak tanımlanmaktadır: $kP = P + P + \dots + P$ (k kez). Bu işlem, ECDLP'nin temelini oluşturmaktadır: $Q = kP$ verildiğinde k'yı bulmak hesaplama açısından zordur (Dallaire-Demers, ve ark, 2025).

NIST P-256 Eğrisi (secp256r1)

Bu çalışmada, Ulusal Standartlar ve Teknoloji Enstitüsü (NIST) tarafından standartlaştırılmış P-256 eğrisi (secp256r1 olarak da bilinir) kullanılmaktadır (FIPS 186-

4, 2013). P-256, NIST tarafından önerilen beş eliptik eğriden biridir ve 256-bit asal sayı alanı $\mathbb{F}_p$ üzerinde tanımlanmıştır. Eğri, aşağıdaki parametrelere sahiptir:

- p (asal modül): $2^{256} - 2^{224} + 2^{192} + 2^{96} - 1$ • $a = -3$
- $b = 41058363725152142129326129780047268409114441015993725554835256314039467401291$
- Temel nokta G (generator): Önceden tanımlanmış bir eğri noktası
- n (alt grup sırası): Temel noktanın mertebesi (yaklaşık 2^{256})

P-256 eğrisi, 128-bit klasik güvenlik seviyesi sunmaktadır ve bu seviye, yaklaşık 2^{128} hesaplama işlemi gerektiren saldırılara karşı dayanıklılık anlamına gelmektedir (Barker, 2020; Park, ve ark, 2020). P-256'nın seçilme nedenleri şunlardır:

- (i) Geniş kütüphane desteği (OpenSSL, cryptography.io, vb.),
- (ii) Donanım hızlandırma desteği (Intel, ARM işlemcilerde),
- (iii) Yaygın endüstri kabulü (TLS 1.3, IEEE 1609.2, HTTPS)
- (iv) Kaynak kısıtlı cihazlarda (Raspberry Pi 4) verimli çalışma (Gupta, ve ark, 2005).

Anahtar Üretimi ECC anahtar çifti üretimi, aşağıdaki adımları içermektedir.

1. Özel Anahtar Üretimi: 1 ile $n-1$ arasında rastgele bir tam sayı d seçilmektedir. Bu sayı, kriptografik olarak güvenli rastgele sayı üretici (CSPRNG) kullanılarak elde edilmektedir (Eastlake, ve ark, 2005).

2. Açık Anahtar Hesaplama: Açık anahtar Q , özel anahtar d ve temel nokta G kullanılarak hesaplanmaktadır:

$$Q = d \times G \quad (3.2)$$

Bu işlem, eliptik eğri skaler çarpımı olarak adlandırılmaktadır. Hesaplama, double-and-add algoritması ile $O(\log n)$ karmaşıklığında gerçekleştirilmektedir (Cohen, ve ark, 2005).

3. Anahtar Formatı: Açık anahtar, sıkıştırılmamış formatta (x, y) koordinatları olarak veya sıkıştırılmış formatta $(x$ koordinatı + y 'nin paritesi) saklanabilmektedir (SEC, 2009). Anahtar üretimi algoritması detaylı olarak Algoritma 1'de verilmiştir.

Algoritma 1:ECC Anahtar Üretim Algoritması

INPUT: None

OUTPUT: private_key, public_key, private_pem

BEGIN

// Adım 1: Eliptik eğri parametrelerini seç
curve $\leftarrow$ SECP256R1 // NIST P-256 eğrisi

// Adım 2: Özel anahtar üretimi
private_key $\leftarrow$ GeneratePrivateKey(curve)

// Adım 3: Açık anahtar türetimi
public_key $\leftarrow$ DerivePublicKey(private_key)

// Adım 4: Özel anahtarı serileştir
private_pem $\leftarrow$ SerializePrivateKey(
 key: private_key,
 encoding: PEM,
 format: PKCS8,
 encryption: NoEncryption
)
RETURN private_key, public_key, private_pem

END

// Yardımcı Fonksiyonlar

FUNCTION GeneratePrivateKey(curve)
 // Rastgele bir tam sayı seç: $d \in [1, n-1]$
 // n: eğrinin mertebesi
 d $\leftarrow$ RandomInteger(1, curve.order - 1)
 RETURN d

END FUNCTION

FUNCTION DerivePublicKey(private_key)
 // $Q = d \times G$ hesapla
 // G: eğrinin jeneratör noktası
 // d: özel anahtar
 Q $\leftarrow$ ScalarMultiplication(private_key, curve.generator)
 RETURN Q

END FUNCTION

FUNCTION SerializePrivateKey(key, encoding, format, encryption)

// Anahtarı belirtilen formatta kodla

serialized_key ← Encode(key, encoding, format)

IF encryption ≠ NoEncryption THEN

 serialized_key ← Encrypt(serialized_key, encryption)

END IF

RETURN serialized_key

END FUNCTION

ECDSA Dijital İmzalama

Eliptik Eğri Dijital İmza Algoritması (ECDSA), ECC tabanlı dijital imza standardıdır (Johnson, ve ark, 2001). ECDSA, mesajın bütünlüğünü ve gönderenin kimliğini doğrulamak için kullanılmaktadır. İmzalama ve doğrulama süreçleri aşağıdaki gibidir.

İmzalama Süreci:

1. Mesaj Özeti (Hash): Mesaj m, SHA-256 hash fonksiyonu ile özetlenmektedir:

$$h = \text{SHA-256}(m) \quad (3.3)$$

SHA-256, 256-bit çıktı üreten kriptografik hash fonksiyonudur ve çarpışma direnci (collision resistance) özelliğine sahiptir (NIST, 2015).

2. Rastgele Sayı Üretimi: 1 ile n-1 arasında rastgele bir tam sayı k seçilmektedir. Bu sayı, her imzalama işlemi için benzersiz olmalıdır (nonce). Aynı k değerinin tekrar kullanılması, özel anahtarın ifşa olmasına yol açmaktadır (Nguyen & Shparlinski, 2003).

3. İmza Hesaplama:

$$(x_1, y_1) = k \times G \quad (3.4)$$

R

$$= x^1 \text{ mod } n \quad (3.5)$$

$$s = k^{-1}(h + r \times d) \text{ mod } n \quad (3.6)$$

İmza, (r, s) çifti olarak temsil edilmektedir.

Doğrulama Süreci:

1. İmza Geçerliliği Kontrolü: $0 < r < n$ ve $0 < s < n$ koşulları kontrol edilmektedir.
2. Mesaj Özeti Hesaplama: $h = \text{SHA-256}(m)$
3. Doğrulama Hesaplaması:

$$w = s^{-1} \bmod n \quad (3.7)$$

$$u_1 = h \times w \bmod n \quad (3.8)$$

$$u_2 = r \times w \bmod n \quad (3.9)$$

$$(x_1, y_1) = u_1 \times G + u_2 \times Q \quad (3.10)$$

Eğer $r = x_1 \bmod n$ ise imza geçerlidir.

ECDSA imzalama ve doğrulama süreçlerinin detaylı algoritması Algoritma 2'de sunulmuştur.

Algoritma 2. ECDSA imzalama ve doğrulama Algoritması

// İMZALAMA

FUNCTION Sign(message, private_key):

hash $\leftarrow$ SHA256(message)

k $\leftarrow$ RANDOM(1, n-1)

r $\leftarrow$ (k $\times$ G).x MOD n

s $\leftarrow$ k⁻¹ $\times$ (hash + r $\times$ private_key) MOD n

RETURN (r, s)

END FUNCTION

// DOĞRULAMA

FUNCTION Verify(message, signature, public_key):

(r, s) $\leftarrow$ signature

hash $\leftarrow$ SHA256(message)

u1 $\leftarrow$ hash $\times$ s⁻¹ MOD n

u2 $\leftarrow$ r $\times$ s⁻¹ MOD n

```

point ← u1 × G + u2 × public_key

IF point.x MOD n = r THEN
    RETURN "İmza geçerli"
ELSE
    RETURN "İmza geçersiz"
END IF
END FUNCTION

```

Güvenlik Analizi

Eliptik Eğri Ayırık Logaritma Problemi (ECDLP)'nin çözümü için en bilinen algoritmalar Pollard's rho ve küçük-adım (baby-step) büyük-adım (giant-step) yöntemleridir ve bu algoritmalar $O(\sqrt{n})$ karmaşıklığına sahiptir (Dallaire-Demers, ve ark, 2025; Miele, 2015). 256-bit ECC için bu karmaşıklık, yaklaşık 2^{128} işlem anlamına gelmektedir. Mevcut hesaplama gücü ile bu seviyede bir saldırı pratik olarak gerçekleştirilemez (Barker, 2020).

Kuantum bilgisayarlar, Shor algoritması ile ECDLP'yi polinom zamanda çözebilmektedir (Jang, ve ark, 2025). Ancak, hata düzeltmeli kuantum bilgisayarların 256-bit ECC'yi kırması için yaklaşık 2330 mantıksal qubit gerekmektedir (Roettele, ve ark, 2017). Mevcut en gelişmiş kuantum bilgisayarlar (IBM Quantum, Google Sycamore) henüz bu kapasiteye ulaşmamıştır. Post-kuantum kriptografi geçişi için NIST standartları (CRYSTALS-Kyber, CRYSTALS-Dilithium) gelecek çalışmalarda değerlendirilebilir (NIST, 2022).

Sonuç olarak, ECC ve ECDSA, kaynak kısıtlı İMC cihazlarında yüksek güvenlik seviyesi ile verimli çalışma imkânı sunmaktadır. 256-bit anahtar uzunluğu, hem depolama hem de hesaplama açısından optimal denge sağlamaktadır. Bir sonraki bölümde, ECC ile birlikte kullanılan steganografik gizleme yöntemi detaylı olarak açıklanacaktır.

3.2.2. Steganografik Gizleme

Steganografi, gizli bilginin taşıyıcı ortam (cover media) içerisine algılanamaz şekilde gömülmesi işlemidir (Mandal, ve ark, 2022). Kriptografinin aksine, steganografinin

amacı mesajın varlığını gizlemektir; bu sayede, saldırganlar iletişimin gerçekleştiğinin farkına bile varamamaktadır (Provos & Honeyman, 2003). Sürü (Swarm) otonom sistemlerde, steganografi kullanımı trafik analizi saldırılarına karşı ek bir koruma katmanı sağlamaktadır. Şifrelenmiş veriler, görüntü dosyaları içerisine gömülerek, ağ trafiğinde şüpheli paketlerin tespiti zorlaştırılmaktadır (Cheddad, ve ark, 2010).

Bu çalışmada, Evrensel Benzersiz Tanımlayıcı (UUID) tabanlı kimlik doğrulama bilgisi, En Az Anlamlı Bit (LSB) yöntemi ile görüntü dosyalarına gömülmektedir. Gömme işlemi, Rastgele Sayı Üretici (PRNG) ile dinamik olarak belirlenen başlangıç koordinatlarında gerçekleştirilmekte ve bu sayede statik steganaliz saldırılarına karşı direnç sağlanmaktadır (Fridrich & Goljan, 2002).

En Az Anlamlı Bit (LSB) Yöntemi

LSB steganografisi, dijital görüntülerin piksel değerlerinin en düşük anlamlı bitlerinin değiştirilerek veri gömülmesi tekniğidir (Chan & Cheng, 2004). RGB renk uzayında, her piksel üç kanal (Kırmızı, Yeşil, Mavi) içermekte ve her kanal 8-bit (0-255 arası) değer almaktadır. Her kanalın LSB'si değiştirildiğinde, renk değerinde maksimum ± 1 birimlik değişim meydana gelmektedir; bu değişim, İnsan Görsel Sistemi (HVS - Human Visual System) tarafından algılanamamaktadır (Apau, ve ark, 2024).

Örneğin, bir pikselin RGB değeri (154, 200, 75) olsun:

- Kırmızı: $154_{10} = 10011010_2$
- Yeşil: $200_{10} = 11001000_2$
- Mavi: $75_{10} = 01001011_2$

Her kanalın LSB'si (sağdan ilk bit) değiştirilerek 3-bit veri gömülebilmektedir. Gömülecek veri "101" ise:

- Kırmızı: $10011010_2 \rightarrow 10011011_2 = 155_{10} (+1)$
- Yeşil: $11001000_2 \rightarrow 11001000_2 = 200_{10} (0)$
- Mavi: $01001011_2 \rightarrow 01001010_2 = 74_{10} (-1)$

Sonuç piksel: (155, 200, 74). Bu değişim, görsel olarak fark edilememektedir (PSNR > 50 dB) (Setiadi, 2021; Johar & Al-Ali, 2025). Gömme işlemi için görüntünün yeterli

kapasiteye sahip olması gerekmektedir. Kapasite kontrolü ve görüntü hazırlama işlemleri Algoritma 4'te sunulmuştur.

LSB yönteminin avantajları şunlardır:

- (i) Yüksek gömme kapasitesi (her piksel için 3-bit, 1920×1080 görüntü için ~6.2 Mbit),
- (ii) Düşük hesaplama karmaşıklığı ($O(n)$ - piksel sayısı ile doğru orantılı),
- (iii) Yaygın kütüphane desteği (PIL, OpenCV, NumPy)
- (iv) Gerçek zamanlı işleme uygunluğu (Raspberry Pi 4'te <50 ms) (Tran, ve ark, 2022).

Ancak, klasik LSB yöntemi, ki-kare (chi-square) saldırısı ve Düzenli-Tekil (RS) steganalizi gibi istatistiksel saldırılara karşı savunmasızdır (Apau, ve ark, 2024; Fridrich, ve ark, 2001). Bu nedenle, bu çalışmada dinamik koordinat seçimi stratejisi uygulanmaktadır.

Evrensel Benzersiz Tanımlayıcı (UUID) Gömme Prosedürü

Evrensel Benzersiz Tanımlayıcı (UUID), 128-bit benzersiz tanımlayıcıdır ve RFC 4122 standardına göre üretilmektedir (Leach, ve ark, 2005). UUID formatı:

xxxxxxxx-xxxx-Mxxx-Nxxx-xxxxxxxxxxxx

Burada M sürümü (version), N varyantı (variant) belirtmektedir. Bu çalışmada UUID v4 (rastgele üretilmiş) kullanılmaktadır. Örnek UUID:

550e8400-e29b-41d4-a716-446655440000

UUID, her İMC cihazı için benzersiz kimlik doğrulama jetonu olarak kullanılmaktadır. UUID üretimi ve binary formata dönüştürme işlemi ile görüntü yükleme ve kapasite kontrolü Algoritma 3 ve 4'de detaylandırılmıştır.

Algoritma 3.UUID Üretimi ve Hazırlama Algoritması

FUNCTION GenerateAndPrepareUUID():

 // UUID v4 üretimi (128-bit rastgele)

 device_uuid ← GENERATE_UUIDv4()

```
// Örnek: "550e8400-e29b-41d4-a716-446655440000"

// Tire karakterlerini kaldır
uuid_hex ← REMOVE_CHAR(device_uuid, '-')
// Sonuç: "550e8400e29b41d4a716446655440000"

// Hexadecimal'den binary'ye dönüştür
uuid_decimal ← HEX_TO_DECIMAL(uuid_hex)
uuid_bits ← DECIMAL_TO_BINARY(uuid_decimal)
uuid_bits ← PAD_LEFT(uuid_bits, 128) // 128-bit'e tamamla
// Sonuç: "01010101000011101000010000000000..."

RETURN uuid_bits
END FUNCTION
```

Algoritma 4.Görüntü Yükleme ve Kapasite Kontrolü

```
FUNCTION LoadImageAndCheckCapacity(image_path, uuid_bits):
    // Görüntü yükleme
    cover_image ← LOAD_IMAGE(image_path)
    // Örnek: 'imc_camera_frame.png'

    // Numpy array'e dönüştür
    cover_array ← IMAGE_TO_ARRAY(cover_image)

    // Boyut bilgilerini al
    (height, width, channels) ← GET_DIMENSIONS(cover_array)

    // Kapasite hesaplama
    total_pixels ← height × width
    available_bits ← total_pixels × 3 // RGB kanalları
```

```

// Kapasite kontrolü
IF LENGTH(uuid_bits) > available_bits THEN
    THROW ERROR "Görüntü kapasitesi yetersiz"
END IF

RETURN cover_array
END FUNCTION

```

PRNG ile Rastgele Koordinat Seçimi

Statik LSB gömme, ki-kare (chi-square) saldırısına karşı savunmasızdır (Apau, ve ark, 2024). Bu nedenle, gömme koordinatları Rastgele Sayı Üretici (PRNG) ile dinamik olarak belirlenmektedir.

Başlangıç koordinatının rastgele seçilmesi, steganaliz saldırılarına karşı direnci artırmaktadır. Koordinat üretim algoritması Algoritma 5'te detaylandırılmıştır.

Algoritma 5. Rastgele Başlangıç Koordinatı Üretimi

FUNCTION GenerateStartCoordinate(width, height, num_bits):

INPUT:

width: Görüntü genişliği (piksel)
height: Görüntü yüksekliği (piksel)
num_bits: Gömülecek veri uzunluğu (bit)

OUTPUT:

(start_y, start_x): Başlangıç koordinatı

BEGIN

total_pixels $\leftarrow$ height $\times$ width

REPEAT

// Rastgele başlangıç koordinatı üret
start_y $\leftarrow$ RANDOM.RANDINT(0, height - 1)
start_x $\leftarrow$ RANDOM.RANDINT(0, width - 1)

```
// Başlangıç piksel indeksini hesapla
start_pixel_index ← start_y × width + start_x

// Bu koordinattan itibaren kalan piksel sayısı
remaining_pixels ← total_pixels - start_pixel_index

UNTIL remaining_pixels ≥ num_bits

RETURN (start_y, start_x)
END
END FUNCTION
// KULLANIM
Width, height ← Image Resolution
coords ← GenerateStartCoordinate(width, height, 128)
```

Bu yöntem, Rastgele Sayı Üretici (PRNG) tabanlı steganografi olarak sınıflandırılmaktadır (Kuznetsov, ve ark, 2024). Rastgele Sayı Üretici (PRNG) sayesinde, her tehdit görüntüsündeki başlangıç koordinatları istatistiksel olarak birbirinden bağımsız ve tahmin edilemez olmaktadır (NIST, 2015).

LSB Gömme İşlemi

UUID verileri hazırlandıktan ve gömme koordinatları belirlendikten sonra, LSB değiştirme yöntemiyle veri gömme işlemi gerçekleştirilir. Bu işlemde, her koordinatta bulunan pikselin en düşük anlamlı biti (LSB) UUID'nin ilgili biti ile değiştirilir. LSB gömme algoritması Algoritma 6'da detaylı olarak sunulmuştur.

Algoritma 6.LSB Steganografik Gömme İşlemi

```
FUNCTION EmbedUUIDInImage(cover_array, coords, uuid_bits):
    // Orijinal görüntünün kopyasını oluştur
    stego_array ← COPY(cover_array)
```

```

// Her koordinat için bit gömme
FOR i ← 0 TO LENGTH(coords)-1 DO
    (y, x, c) ← coords[i]

    // Orijinal piksel değerini al
    pixel_value ← stego_array[y, x, c]

    // Gömülecek biti al
    bit_to_embed ← CONVERT_TO_INT(uuid_bits[i])

    // LSB değiştirme (Bit maskesi ile)
    // pixel_value & 0xFE: Son biti sıfırla
    // | bit_to_embed: Yeni biti ekle
    stego_array[y, x, c] ← (pixel_value AND 0xFE) OR bit_to_embed
END FOR

// Stego görüntüyü kaydet
stego_image ← ARRAY_TO_IMAGE(stego_array)
SAVE_IMAGE(stego_image, 'stego_image.png', format='PNG')

RETURN stego_array
END FUNCTION

// KULLANIM
stego_array ← EmbedUUIDInImage(cover_array, coords, uuid_bits)
// KULLANIM
seed ← "IMC_DEVICE_SN_12345"
coords ← GenerateEmbeddingCoordinates(seed, 128, cover_array.shape)

```

Gömme işlemi tamamlandıktan sonra, alıcı tarafta UUID'nin çıkarılması gerekir. Çıkarma işlemi, gömme işleminin tersine çalışır: alıcıya gönderilen koordinatlar kullanılarak ve bu koordinatlardan itibaren LSB'ler okunarak UUID yeniden oluşturulur. Bu süreç Algoritma 7'de detaylandırılmıştır.

Algoritma 7.UUID Çıkarma İşlemi

FUNCTION ExtractUUID(stego_image_path, seed):

INPUT:

stego_image_path: Stego görüntü yolu

seed: Tohum değeri (gömme ile aynı)

OUTPUT:

extracted_uuid: Çıkarılan UUID string

BEGIN

// Stego görüntüyü yükle

stego_image ← LOAD_IMAGE(stego_image_path)

stego_array ← IMAGE_TO_ARRAY(stego_image)

// Koordinatları yeniden üret

coords ← GenerateEmbeddingCoordinates(seed, 128, stego_array.shape)

extracted_bits ← ""

// Her koordinattan LSB çıkar

FOR EACH (y, x, c) IN coords DO

pixel_value ← stego_array[y, x, c]

lsb ← pixel_value AND 1 // LSB çıkarma

extracted_bits ← extracted_bits + STRING(lsb)

END FOR

// Binary'den hexadecimal'e dönüştür

extracted_decimal ← BINARY_TO_DECIMAL(extracted_bits)

extracted_hex ← DECIMAL_TO_HEX(extracted_decimal)

extracted_hex ← PAD_LEFT(extracted_hex, 32)

// UUID formatına dönüştür (tire ekle)

```

    extracted_uuid ← FORMAT_UUID(extracted_hex)
    // Format: "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx"

    RETURN extracted_uuid
END
END FUNCTION

FUNCTION FORMAT_UUID(hex_string):
    part1 ← hex_string[0:8]
    part2 ← hex_string[8:12]
    part3 ← hex_string[12:16]
    part4 ← hex_string[16:20]
    part5 ← hex_string[20:32]

    RETURN part1 + "-" + part2 + "-" + part3 + "-" + part4 + "-" + part5
END FUNCTION

// KULLANIM
recovered_uuid ← ExtractUUID('stego_image.png', "IMC_DEVICE_SN_12345")

PRINT "Orijinal UUID: ", uuid_str
PRINT "Çıkarılan UUID: ", recovered_uuid
PRINT "Eşleşme: ", (uuid_str = recovered_uuid)

```

Görüntü Kalite Metrikleri

Steganografik gömme işleminin başarısı, görüntü kalitesinin korunması ile ölçülmektedir. İki temel metrik kullanılmaktadır:

1. Tepe Sinyal-Gürültü Oranı (PSNR):

Tepe Sinyal-Gürültü Oranı (PSNR), orijinal ve stego görüntü arasındaki piksel bazlı farklılığı ölçen objektif bir metriktir (Huynh-Thu & Ghanbari, 2008). PSNR formülü:

$$MSE = \frac{1}{H \times W} \sum_{i=1}^H \sum_{j=1}^W [I(i,j) - K(i,j)]^2 \quad (3.11)$$

PSNR

$$= 10 \times \log_{10} \left(\frac{MAX_I^2}{MSE} \right) \quad (3.12)$$

Burada:

- H, W: Görüntü yüksekliği ve genişliği
- I(i,j): Orijinal görüntü piksel değeri
- K(i,j): Stego görüntü piksel değeri
- MAX_i: Maksimum piksel değeri (8-bit için 255)
- MSE: Mean Squared Error (Ortalama Kare Hata)

PSNR değerlendirme kriterleri (Horé & Ziou, 2010):

- PSNR > 40 dB: Mükemmel kalite (fark algılanamaz)
- 30-40 dB: İyi kalite (minimal fark)
- 20-30 dB: Kabul edilebilir kalite
- < 20 dB: Düşük kalite (fark belirgin)

PSNR İmplementasyonu

PSNR hesaplaması, orijinal ve stego görüntüler arasındaki ortalama kare hatasına (MSE) dayalı olarak gerçekleştirilir. Hesaplama algoritması Algoritma 8'de detaylı olarak sunulmuştur.

Algoritma 8.PSNR Hesaplama

FUNCTION CalculatePSNR(original, stego):

INPUT:

original: Orijinal görüntü array'i

stego: Stego görüntü array'i

OUTPUT:

psnr: PSNR değeri (dB cinsinden)

```

BEGIN
    // MSE (Mean Squared Error) hesaplama
    diff ← original - stego
    squared_diff ← diff²
    mse ← MEAN(squared_diff)

    // Görüntüler özdeş mi kontrol et
    IF mse = 0 THEN
        RETURN ∞ // Sonsuz PSNR
    END IF

    // PSNR hesaplama
    max_pixel ← 255.0
    psnr ← 10 × log10((max_pixel²) / mse)

    RETURN psnr
END

END FUNCTION

// KULLANIM
psnr_value ← CalculatePSNR(cover_array, stego_array)
PRINT "PSNR: ", psnr_value, " dB"

```

2.Yapısal Benzerlik İndeksi (SSIM):

Yapısal Benzerlik İndeksi (SSIM), insan görsel algısına daha yakın bir metrik olup, yapısal benzerliği ölçmektedir (Wang, ve ark, 2004). SSIM formülü:

$$\begin{aligned}
 & \text{SSIM}(x, y) \\
 &= \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)}
 \end{aligned} \tag{3.13}$$

Burada:

- μ_x, μ_y : x ve y görüntülerinin ortalama deęerleri
- σ_x^2, σ_y^2 : x ve y görüntülerinin varyansları
- σ_{xy} : x ve y arasındaki kovaryans
- C_1, C_2 : Stabilizasyon sabitleri $[(K_1L)^2, (K_2L)^2$; $K_1=0.01, K_2=0.03, L=255$]

SSIM hesaplamasının yazılımsal implementasyonu Algoritma 9'da detaylandırılmıştır.

SSIM deęerlendirme kriterleri:

- SSIM = 1.0: Görüntüler özdeş
- SSIM > 0.99: Mükemmel benzerlik
- 0.95-0.99: Çok iyi benzerlik
- 0.90-0.95: İyi benzerlik
- < 0.90: Belirgin farklılık

Algoritma 9.SSIM Hesaplama

FUNCTION CalculateSSIM(original, stego):

INPUT:

original: Orijinal görüntü array'i

stego: Stego görüntü array'i

OUTPUT:

ssim_value: SSIM deęeri (0-1 arası)

BEGIN

// SSIM hesaplama parametreleri

multichannel $\leftarrow$ TRUE

channel_axis $\leftarrow$ 2 // RGB kanalları

data_range $\leftarrow$ 255 // Piksel deęer aralıęı

// SSIM hesaplama

ssim_value $\leftarrow$ COMPUTE_SSIM(original, stego,
multichannel,

```

        channel_axis,
        data_range)

    RETURN ssim_value
END
END FUNCTION

// KULLANIM
ssim_value ← CalculateSSIM(cover_array, stego_array)
PRINT "SSIM: ", ssim_value

```

3.2.3.İki Aşamalı Hibrit Protokol

Önerilen sistemde veri bütünlüğü ve kimlik doğrulama güvenliğini sağlamak amacıyla LSB Steganografi ve Eliptik Eğri Kriptografisi (ECC) dijital imzalama tekniklerini birleştiren iki aşamalı hibrit bir protokol geliştirilmiştir. Bu protokol, İMC tarafından tespit edilen tehdit görüntülerinin Ground CS'ye güvenli bir şekilde iletilmesi ve sunucu tarafında çift katmanlı doğrulama yapılmasını sağlamaktadır.

Protokol mimarisi, gönderici (İMC) ve alıcı (Ground CS) olmak üzere iki ana bileşenden oluşmaktadır. Şekil 3.1.'de protokolün genel akış diyagramı gösterilmektedir. Protokolün temel çalışma prensibi, İMC tarafında tehdit görüntüsüne steganografik veri gömme ve ECC ile dijital imzalama işlemlerinin gerçekleştirilmesi, Ground CS tarafında ise gelen veriler üzerinde ECC imza doğrulama ve steganografik veri çıkarma işlemleri ile çift katmanlı doğrulamanın yapılması şeklindedir.

İMC Tarafı İşlemler

İMC üzerinde çalışan tehdit tespit sistemi, YOLOv8 modeli ile %50 ve üzeri güven oranına sahip tehdit nesnelerini tespit ettiğinde, öncelikle steganografik veri gömme işlemi için görüntü üzerinde rastgele bir başlangıç noktası $[x,y]$ seçilmektedir. Bu rastgelelik, saldırganların gömülü verinin konumunu tahmin etmesini zorlaştırarak

güvenliği artırmaktadır. Ardından, 36 karakterlik bir UUID v4 string değeri üretilir ve meta verilere eklenir:

$$\text{uuid} \mid [x, y] \mid \text{mac_address} \mid \text{konum} \mid \text{model} \mid \text{threat_class} \mid \text{confidence}. \quad (3.13)$$

UUID string üretildikten sonra, En Az Anlamlı Bit (LSB) steganografi tekniği kullanılarak tehdit görüntüsüne gömülmektedir. LSB yöntemi, görüntü piksellerinin en düşük anlamlı bitlerini değiştirerek veri gizleme işlemi gerçekleştirmektedir. Bu yöntem, görüntü kalitesinde minimum bozulmaya neden olduğu için tercih edilmiştir. Gömme işlemi, belirlenen $[x, y]$ başlangıç noktasından itibaren gerçekleştirilmektedir.

Steganografi işleminden sonra, meta veriler Eliptik Eğri Kriptografisi (ECC) algoritması kullanılarak dijital olarak imzalanmaktadır. Bu işlem için İMC'ye sistem başlangıcında Ground CS tarafından güvenli bir şekilde iletilmiş olan ECC gizli anahtar (private key) kullanılmaktadır. ECC gizli anahtar (private key) İMC işletim sistemi anahtar zincirine eklenmekte ve sadece imzalama esnasında referans edilmektedir. Anahtar zincirine eklenen bir anahtarın referans edilmesi dışında okunabilmesi mümkün olmadığından bu yöntem ile anahtarın ele geçirilip kopyalanmasına önlenmiştir. Bu aşamada şifreleme işlemi yapılmamakta, sadece meta verilerin bütünlüğünü ve gönderenin kimliğini doğrulamak amacıyla dijital imza oluşturulmaktadır. İmzalama işlemi şu şekilde formüle edilmektedir:

$$\begin{aligned} &\text{signature} \\ &= \text{ECCsign}(\text{metadata}, \text{private_key}_{\text{IMC}}) \end{aligned} \quad (3.14)$$

İmzalama işlemi tamamlandıktan sonra, steganografi uygulanmış görüntü ve ilgili veriler Ground CS'ye HTTP POST isteği ile üç parametre halinde iletilmektedir:

1. Param 1 (image): LSB steganografi uygulanmış tehdit görüntüsü
2. Param 2 (metadata): Açık metin formatında robot ve tehdit ile ilgili veriler
3. Param 3 (signature): ECC ile oluşturulmuş dijital imza

Veri iletimi asenkron olarak gerçekleştirildiği için operasyonel süreklilik sağlanmaktadır.

Ground CS Tarafı Doğrulama Katmanları

Ground CS, İMC'den gelen verileri aldığı anda iki aşamalı doğrulama protokolünü başlatmaktadır. İlk olarak, Param 3 (signature) veritabanında aratılır. Bu imza ile daha önce istek atılmış ise istek reddedilir. Bu sayede daha önceki isteklerde kullanılan imzaların sızması durumunda bir kere kullanılan bir imzanın tekrar kullanılmasının önüne geçilerek veri zehirlenmesi ve taklit saldırılarına engel olunmaktadır. İmza veritabanında bulunamadığında gelen tehdit görüntüsü Param 2'den alınan robot kimliği (robot_id) ile ilişkilendirilerek veritabanına kaydedilmektedir. Bu aşamada görüntü henüz doğrulanmadığı için ilgili kayıta doğrulandı alanı NULL değerini almaktadır. Veritabanı kaydı, görüntü dosya yolu, robot ID, zaman damgası, ECC doğrulama durumu, steganografik doğrulama durumu ve genel doğrulama durumu bilgilerini içermektedir.

ECC Dijital İmza Doğrulama (Birinci Katman):

İlk doğrulama katmanı olarak Param 3 (signature) veritabanında kayıtlı olup, olmadığı kontrol edilir ve kayıtlı ise istek reddedilir. ECC ile imzalama işlemi aynı veride tekrar yapıldığında imza her zaman değişiklik göstermektedir (Nguyen & Shparlinski, 2003). Bu nedenle aynı imzanın tekrar gelmesi sızıntı belirtisi olarak nitelendirilmektedir. Eğer imza veritabanında kayıtlı değilse, kayıt edilir. Ardından, ECC dijital imza doğrulama işlemi gerçekleştirilmektedir. Bu işlem için öncelikle robot MAC ID bilgisi kullanılarak ilgili İMC'ye ait ECC açık anahtar (public key) veritabanından sorgulanmaktadır. Her İMC, sistemde kayıtlı benzersiz bir MAC ID'ye sahiptir ve bu ID ile açık anahtar (public key) eşleştirilmektedir. Ardından, Param 2 (metadata) ve Param 3 (signature) kullanılarak ECC imza doğrulama işlemi gerçekleştirilmektedir:

$$\text{verification_result} = \text{ECCverify}(\text{metadata}, \text{signature}, \text{public_key}_{\text{IMC}}) \quad (3.15)$$

Doğrulama başarısız ise, görüntü doğrulanmamış_görseller klasörüne taşınmakta ve veritabanı kaydında ECC_doğrulama = FALSE olarak işaretlenmektedir. Bu durumda protokol sonlandırılmakta ve ikinci doğrulama katmanına geçilmemektedir. Bu durum, verinin iletim sırasında değiştirildiğini veya yetkisiz bir kaynaktan geldiğini göstermektedir. Doğrulama başarılı ise, veritabanı kaydında ECC_doğrulama = TRUE olarak işaretlenmekte ve ikinci doğrulama katmanı olan steganografik doğrulama adımına geçilmektedir.

Steganografik Doğrulama (İkinci Katman):

ECC imza doğrulaması başarılı olan görüntüler için steganografik doğrulama işlemi başlatılmaktadır. Bu aşamada, Param 2'den alınan meta veriler seperatör karakter | kullanılarak parçalara ayrılmakta ve bir diziye dönüştürülmektedir. Dizinin ilk elemanı olan başlangıç noktası $[x,y]$ koordinatları parse edilmektedir. Bu koordinatlar, görüntüden steganografik veri çıkarma işleminin başlayacağı noktayı belirtmektedir. Belirlenen $[x,y]$ başlangıç noktasından itibaren LSB steganografi yöntemi kullanılarak görüntüden gömülü string ifade çıkarılmaktadır.

Görüntüden çıkarılan string ile Param 2'den alınan UUID string karşılaştırılmaktadır:

$$\text{match} = (\text{extracted}_{\text{string}} == \text{UUID}_{\text{string}}) \quad (3.16)$$

Eşleşme yoksa, görüntü doğrulanmamış_görseller klasörüne taşınmakta ve veritabanı kaydında Steganografik_dogrulama = FALSE olarak işaretlenmektedir. Bu durum, görüntünün manipüle edildiğini veya steganografik verinin bozulduğunu göstermektedir. Eşleşme varsa, veritabanı kaydında Steganografik_dogrulama = TRUE ve doğrulandı = TRUE olarak işaretlenmekte ve görüntü doğrulanmış tehdit görselleri olarak kabul edilmektedir.

Protokol, ağ bağlantı hataları, açık anahtar (public key) bulunamama, steganografik veri bozulması ve zaman aşımı gibi hata durumlarını yönetmek üzere tasarlanmıştır. API çağrısı başarısız olduğunda İMC yerel önbellekte veriyi saklamakta ve bağlantı yeniden kurulduğunda tekrar göndermektedir. MAC ID ile eşleşen açık anahtar (public key) veritabanında yoksa sistem yöneticisine uyarı gönderilmekte ve görüntü karantina klasörüne alınmaktadır.

Bu iki aşamalı hibrit protokol, hem kriptografik güvenlik (ECC imza doğrulama) hem de veri bütünlüğü (steganografik doğrulama) sağlayarak, sistemin güvenilir ve manipülasyon dirençli bir iletişim altyapısına sahip olmasını garanti etmektedir. Bir sonraki bölümde, sistemin Dağıtık Hizmet Reddi (DDoS) saldırılarına karşı korunması için geliştirilen mekanizmalar detaylandırılacaktır.

3.2.4.DDoS Koruması

Dağıtık Hizmet Reddi (Distributed Denial of Service - DDoS) saldırıları, sürü otonom sistemlerinin operasyonel sürekliliğini tehdit eden kritik siber güvenlik tehditleri arasında yer almaktadır. Bu saldırılar, hedef sisteme aşırı sayıda istek göndererek normal operasyonların kesintiye uğramasına ve sistem kaynaklarının tükenmesine neden olmaktadır (Mirkovic & Reiher, 2004). Özellikle merkezi Ground Control Station (Ground CS) sunucusunun, çoklu İMC düğümlerinden gelen yoğun veri trafiğini yönetmesi gerektiği göz önüne alındığında, DDoS saldırılarına karşı etkili savunma mekanizmalarının entegrasyonu zorunluluk arz etmektedir.

Önerilen hibrit güvenlik protokolü kapsamında, DDoS saldırılarına karşı çok katmanlı bir savunma stratejisi geliştirilmiştir. Bu strateji, uygulama katmanında (application-layer) filtreleme ve aşamalı IP engelleme (progressive IP blocking) mekanizmalarını entegre ederek, hem kimlik doğrulama tabanlı saldırıları hem de volumetrik saldırıları etkisiz hale getirmeyi amaçlamaktadır (Zargar, ve ark, 2013). Bu bölümde, DDoS koruması için kullanılan metodoloji, algoritma yapısı ve sistem entegrasyonu detaylı olarak açıklanmaktadır.

Aşamalı IP Engelleme Mekanizması

Ground CS sunucusuna gelen istekleri kaynak IP adresi bazında izleyerek, anormal davranış gösteren düğümleri dinamik olarak engelleyen bir savunma algoritması geliştirilmiştir. Bu mekanizma, her bir IP adresinden gelen geçerli isteklerin sayısını takip eden bir sayaç yapısı (requestCounts) ve daha önce işlenmiş geçerli veri yüklerini kaydeden bir hash kümesi (storedValidData) olmak üzere iki temel veri yapısı üzerinde çalışmaktadır.

Algoritmanın işleyişi beş temel adımdan oluşmaktadır:

- Adım 1- Geçerlilik Kontrolü: Gelen her istek IsValidRequest(data) fonksiyonu ile değerlendirilir. Bu fonksiyon, veri yükünün hem ECC dijital imza doğrulamasını hem de steganografik UUID doğrulamasını geçip geçmediğini kontrol eder.

- Adım 2- Tekrarlı İstek Kontrolü: Eğer gelen veri yükü daha önce storedValidData kümesinde kayıtlı ise, bu durum aynı verinin tekrar gönderildiğini gösterir ve kaynak IP adresinin istek sayacı (requestCounts[ipAddress]) bir artırılır.
- Adım 3- Eşik Kontrolü: Belirlenen maksimum tekrar sayısı (MaxRequests = 5) aşıldığında, ilgili IP adresi BlockIpAddress(ipAddress) fonksiyonu çağrılarak kalıcı olarak engellenir ve sistem loguna kaydedilir.
- Adım 4- Yeni Geçerli İstek: Eğer gelen veri yükü yeni ve geçerli ise, storedValidData kümesine eklenir, IP adresinin istek sayacı 1 olarak başlatılır ve istek başarıyla işlenir.
- Adım 5- Geçersiz İstek: IsValidRequest fonksiyonundan geçemeyen istekler doğrudan reddedilir ve sistem loguna kaydedilir.

Bu mekanizmanın matematiksel ifadesi şu şekilde formüle edilmiştir:

$$\text{Action}(ip, data) = \begin{cases} \text{Block}(ip) & \text{if requestCounts}[ip] > \text{MaxRequests} \\ \text{Process}(data) & \text{if IsValid}(data) \wedge data \notin \text{storedValidData} \\ \text{Increment}(ip) & \text{if IsValid}(data) \wedge data \in \text{storedValidData} \\ \text{Reject}(data) & \text{otherwise} \end{cases} \quad (3.17)$$

Bu yaklaşım, meşru düğümlerden gelen tekil isteklerin hızlı bir şekilde işlenmesini sağlarken, tekrarlayan veya kötü niyetli isteklerin aşamalı olarak kısıtlanmasını garanti etmektedir (Peng, ve ark, 2007).

Uygulama Katmanı Filtreleme

DDoS saldırılarının ağ katmanı (network-layer) yerine HTTP/HTTPS protokolü üzerinden gerçekleştirildiği senaryolara karşı etkili bir savunma sağlamak amacıyla uygulama katmanı filtreleme mekanizması geliştirilmiştir (Ranjan, ve ark, 2009). Bu filtreleme mekanizması, gelen her isteğin içeriğini analiz ederek, yalnızca geçerli veri yapısına sahip isteklerin sunucu tarafından işlenmesine izin vermektedir. Tüm analizlerden önce gelen isteğin ECC imzasının mükerrer olup olmadığına bakılmakta eğer mükerrer imza ise istek en baştan işleme alınmayıp tekrarlı veri gönderimi kriteri uygulanmaktadır.

Filtreleme sürecinde üç temel kriter uygulanmaktadır:

- Kriter 1- Veri Yapısı Doğrulaması: Gelen isteğin beklenen parametreleri (steganografik görsel, meta veriler, ECC imzası) içerip içermediği kontrol edilir.
- Kriter 2- ECC İmza Doğrulaması: İsteğin üçüncü parametresi olan ECC dijital imzası, düğümün kayıtlı açık anahtarı (public key) kullanılarak doğrulanır ve doğrulama başarısız olursa istek reddedilir (Johnson, ve ark, 2001).
- Kriter 3- Steganografik Doğrulama: Görselden çıkarılan UUID, gönderilen string ifadesi ile karşılaştırılır ve uyumsuzluk durumunda istek geçersiz sayılır.

Bu çok katmanlı doğrulama süreci, yalnızca ECC gizli anahtara (private key) sahip meşru düğümlerin geçerli veri üretebilmesini garanti ederek, kimlik doğrulama tabanlı DDoS saldırılarını etkisiz hale getirmektedir.

Saldırı Tespit Kriterleri

Sistem, üç temel kritere dayalı olarak potansiyel DDoS saldırılarını tespit etmektedir:

- Kriter 1- Tekrarlı Veri Gönderimi: Aynı UUID ve metadata içeren görsellerin aynı IP adresinden 5'ten fazla kez gönderilmesi durumunda tekrarlı veri gönderimi olarak değerlendirilmektedir.
- Kriter 2- ECC Doğrulama Başarısızlığı: Belirli bir IP adresinden gelen isteklerin sürekli olarak ECC imza doğrulamasında başarısız olması durumunda tespit edilmektedir.
- Kriter 3- Steganografik Doğrulama Başarısızlığı: UUID çıkarma işleminin sürekli olarak başarısız olması veya çıkarılan UUID'nin gönderilen string ifadesi ile uyumsuz olması durumunda kaydedilmektedir.

Bu kriterler, hem volumetrik saldırıları (yüksek hacimli istek trafiği) hem de sofistike saldırıları (geçerli görünümlü ancak sahte veri içeren istekler) tespit edebilecek şekilde tasarlanmıştır (Xie & Yu, 2009).

Entegre Savunma Mimarisi ve Implementasyon

DDoS koruması, hibrit güvenlik protokolünün ayrılmaz bir parçası olarak Ground CS sunucusuna entegre edilmiştir. ECC tabanlı kimlik doğrulama ve steganografik gizleme mekanizmalarının doğal bir uzantısı olan bu savunma katmanı, ek donanım veya üçüncü taraf servis gerektirmeden, yazılım düzeyinde gerçekleştirilmektedir. Bu entegre

yaklaşım, blockchain tabanlı çözümlerin getirdiği yüksek gecikme ve enerji tüketimi sorunlarını ortadan kaldırırken, uygulama katmanında proaktif bir savunma sağlamaktadır.

Aşamalı IP engelleme mekanizmasının pseudocode gösterimi Algoritma 10'da sunulmuştur. Algoritma, gelen istekleri gerçek zamanlı olarak değerlendirerek, maksimum 5 tekrarlı geçerli istek eşik değerine ulaşan IP adreslerini otomatik olarak engellemektedir. Bu eşik değeri, sistem yöneticileri tarafından operasyonel gereksinimlere göre dinamik olarak ayarlanabilir yapıdadır.

Algoritma 10: Aşamalı IP Engelleme Mekanizması

INPUT:

ipAddress: İsteğin kaynak IP adresi
data: İstekle birlikte gönderilen veri yükü

OUTPUT:

İstek işlenir VEYA IP adresi engellenir

PARAMETERS:

MaxRequests $\leftarrow$ 5 // Maksimum tekrar eşiği
requestCounts $\leftarrow$ Dictionary<IPAddress, Integer>
storedValidData $\leftarrow$ HashSet<String>

BEGIN

// Adım 1: Geçerlilik Kontrolü

IF IsValidRequest(data) = TRUE THEN

// Adım 2: Tekrar Tespiti

IF data $\in$ storedValidData THEN

requestCounts[ipAddress] $\leftarrow$ requestCounts[ipAddress] + 1

// Adım 3: Eşik Kontrolü

```

IF requestCounts[ipAddress] > MaxRequests THEN
    BlockIpAddress(ipAddress)
    OUTPUT "IP {ipAddress} engellendi"
    RETURN
END IF
ELSE
    // Adım 4: Yeni Geçerli İstek
    ADD data TO storedValidData
    requestCounts[ipAddress] ← 1
    OUTPUT "Yeni IP {ipAddress} başarıyla işlendi"
END IF
ELSE
    OUTPUT "Geçersiz istek: {ipAddress}"
END IF
END

```

Bu metodoloji, sürü otonom sistemlerinin operasyonel sürekliliğini tehdit eden dağıtık saldırılara karşı hafif, etkili ve entegre bir savunma sağlamaktadır. Aşamalı IP engelleme ve uygulama katmanı filtreleme kombinasyonu, hem klasik volumetrik saldırıları hem de kimlik doğrulama tabanlı sofistike saldırıları tespit edebilecek kapasiteye sahiptir.

3.2.5. Hibrit Güvenlik Protokolü Test Ortamı

Önerilen hibrit güvenlik protokolünün gerçek dünya koşullarında performans ve uygulanabilirlik değerlendirmesi, kaynak kısıtlı gömülü sistemleri temsil eden Raspberry Pi 4 Model B donanım platformu üzerinde gerçekleştirilmiştir. Raspberry Pi 4, düşük güç tüketimi, kompakt boyutları ve yeterli hesaplama kapasitesi ile İnsansız Hava Araçları (İHA) ve İnsansız Kara Araçları (İKA) gibi sürü otonom sistemlerinin düğüm bileşenlerini simüle etmek için ideal bir test platformu sunmaktadır (Raspberry Pi Foundation, 2019). Bu bölümde, ECC tabanlı kimlik doğrulama (Bölüm 3.2.1), steganografik gizleme (Bölüm 3.2.2), iki aşamalı hibrit protokol (Bölüm 3.2.3) ve DDoS

koruması (Bölüm 3.2.4) mekanizmalarının entegre bir şekilde test edildiği donanım ve yazılım altyapısı açıklanmaktadır.

Kenar Cihaz(İMC) Yapılandırması

Test ortamında kullanılan Raspberry Pi 4 Model B, 1.5 GHz quad-core ARM Cortex-A72 işlemci (Broadcom BCM2711 SoC), 4 GB LPDDR4-3200 SDRAM ve 32 GB Class 10 microSD kart ile donatılmıştır. Bu donanım konfigürasyonu, modern sürü robotik sistemlerinde yaygın olarak kullanılan gömülü bilgisayar platformlarının tipik kaynak profilini yansıtmaktadır (Upton & Halfacree, 2020). Ağ bağlantısı için yerleşik Gigabit Ethernet ve dual-band 802.11ac kablosuz adaptör kullanılmıştır. İMC düğümleri ile Ground Control Station (Ground CS) sunucusu arasındaki iletişim, HTTP/HTTPS protokolleri üzerinden gerçekleştirilmiştir.

Kenar Cihaz İşletim Sistemi ve Yazılım Altyapısı

Raspberry Pi 4 üzerinde Raspberry Pi OS (64-bit, Debian 11 Bullseye tabanlı) işletim sistemi kurulmuştur. Hibrit güvenlik protokolünün İMC tarafındaki implementasyonu, Python 3.9.2 programlama dili kullanılarak geliştirilmiştir (Python Software Foundation, 2021). Python'un geniş kütüphane ekosistemi ve platform bağımsızlığı, kaynak kısıtlı gömülü sistemlerde güvenlik protokollerinin test edilmesi için uygun bir ortam sağlamaktadır. Sistem başlangıcında otomatik çalışma için systemd servisi yapılandırılmış ve İMC yazılımı daemon modu ile çalıştırılmıştır.

Kriptografik ve Steganografik Kütüphaneler

Eliptik Eğri Kriptografisi (ECC) işlemleri için Python cryptography kütüphanesi (versiyon 41.0.0) kullanılmıştır (Python Cryptographic Authority, 2023). Bu kütüphane, NIST P-256 (secp256r1) eğrisi üzerinde ECC anahtar çifti üretimi, ECDSA dijital imza oluşturma ve doğrulama işlemlerini desteklemektedir (Johnson, ve ark, 2001). SHA-256 hash fonksiyonu, Python hashlib modülü aracılığıyla gerçekleştirilmiştir (NIST, 2015).

Görüntü işleme ve steganografik gömme/çıkarma işlemleri için Pillow kütüphanesi (versiyon 10.0.0) kullanılmıştır (Clark, 2023). Bölüm 3.2.2'de açıklanan En Az Anlamli Bit (LSB) tabanlı steganografik gömme algoritması, Pillow'un piksel erişim metodları kullanılarak gerçekleştirilmiştir (Chan & Cheng, 2004).

UUID Üretimi ve İletişim Protokolü

Evrensel Benzersiz Tanımlayıcı (UUID) üretimi için Python uuid modülü kullanılmıştır. Her İMC düğümü için UUID versiyon 4 formatında benzersiz tanımlayıcılar oluşturulmuştur (Leach, ve ark, 2005). Bölüm 3.2.3'te açıklanan iki aşamalı hibrit protokol kapsamında, UUID ifadesi 128-bit bir tanımlayıcıdır. Standart gösterimi 36 karakter uzunluğundadır (32 hex karakter + 4 tire):

$$\text{xxxxxxxx} - \text{xxxx} - 4\text{xxx} - y\text{xxx} - \text{xxxxxxxxxxxxxx} \quad (3.18)$$

Ground Control Station (Ground CS) sunucusu, Microsoft ASP.NET Core framework'ü (versiyon 6.0 LTS) kullanılarak geliştirilmiştir (Microsoft, 2023). Sunucu, Windows 10 Pro işletim sistemi üzerinde çalışmakta olup, donanım konfigürasyonu Intel Core i7-9700K işlemci (8 çekirdek, 3.6 GHz), 32 GB DDR4 RAM ve 512 GB NVMe SSD'den oluşmaktadır. İMC düğümlerinden gelen HTTP POST istekleri, ASP.NET Core Controller sınıfları içinde tanımlanan API uç noktalarında (endpoint) işlenmiştir.

Web API, Kestrel web sunucusu üzerinde çalışmaktadır (Microsoft, 2023). İstek işleme pipeline'ı, ASP.NET Core Middleware bileşenleri ile yapılandırılmış, JSON formatındaki veri serileştirme işlemleri için System.Text.Json kütüphanesi kullanılmıştır. ECC imza doğrulama işlemleri için .NET System.Security.Cryptography.ECDSA sınıfı, steganografik işlemler için System.Drawing.Common kütüphanesi kullanılmıştır.

Bölüm 3.2.4'te açıklanan DDoS koruması mekanizması, ASP.NET Core Middleware olarak implemente edilmiş ve her gelen istek için IP adresi bazlı filtreleme, ECC imza doğrulama ve aşamalı IP engelleme algoritması uygulanmıştır (Mirkovic & Reiher, 2004). IP engelleme işlemi, System.Collections.Concurrent.ConcurrentDictionary veri yapısı ile iş parçacığı güvenli (thread-safe) olarak gerçekleştirilmiştir. Maksimum 5 tekrarlı geçerli istek eşik değeri, appsettings.json konfigürasyon dosyasında dinamik olarak ayarlanabilir yapıdadır.

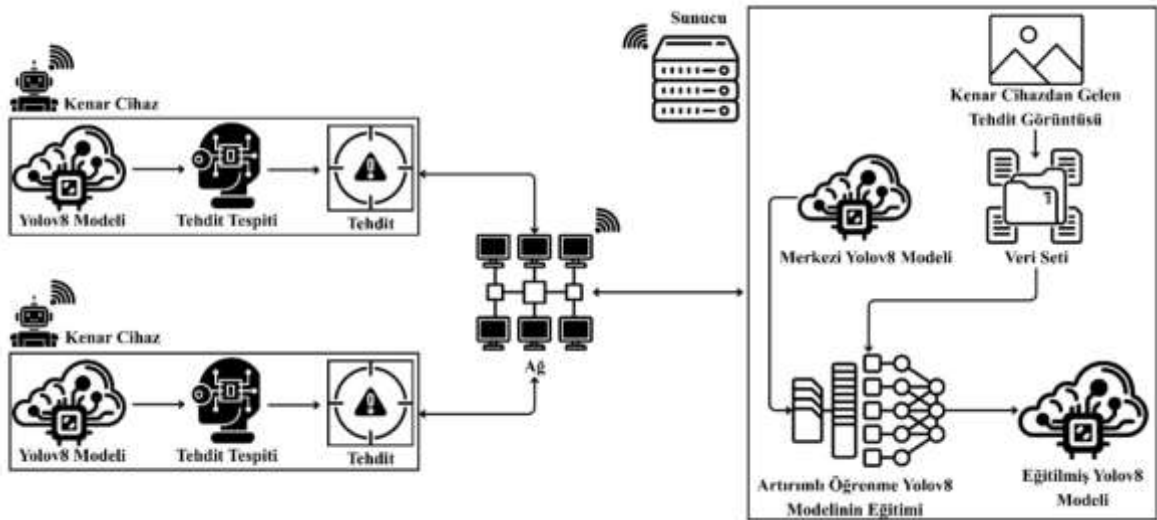
Sunucu tarafında gerçekleştirilen işlemlerin görselleştirilmiş hali şekil 3.3 de verilmiştir.

RSA sistemleri arasındaki performans karşılaştırmaları yapılmıştır (Virtanen, ve ark, 2020).

Bu test ortamı, önerilen hibrit güvenlik protokolünün kaynak kısıtlı gömülü sistemlerde uygulanabilirliğini ve performans avantajlarını objektif olarak değerlendirmeyi mümkün kılmaktadır.

3.3. Tehdit Tespit ve Artırımlı Öğrenme

Sürü otonom sistemlerinde operasyonel güvenliğin sağlanması, İMC düğümlerinin gerçek zamanlı tehdit tespit kapasitesine sahip olmasını gerektirmektedir. Geleneksel derin öğrenme modelleri, statik ortamlarda yüksek performans gösterse de, dinamik tehdit senaryolarında sürekli öğrenme (continual learning) sırasında katastrofik unutma (catastrophic forgetting) problemi ile karşılaşmaktadır (Kirkpatrick, ve ark, 2017). Bu çalışmada, YOLOv8 derin öğrenme mimarisi ile artırımlı öğrenme (incremental learning) stratejisi entegre edilerek, adaptif ve açıklanabilir bir tehdit tespit sistemi geliştirilmiştir. Bu bölümde, YOLOv8 model mimarisi, veri seti özellikleri, ilk model eğitimi, artırımlı öğrenme protokolü ve Grad-CAM tabanlı açıklanabilir yapay zekâ mekanizması detaylı olarak açıklanmaktadır. Tehdit tespit ve artırımlı öğrenme için kurgulanan sistemin görsel kurgusu Şekil 3.4'te verilmiştir.



Şekil 3.4. Tehdit Tespit ve Artırımlı Öğrenme Sistemi

3.3.1. YOLOv8 Model Mimarisi

Bu çalışmada, kaynak kısıtlı gömülü sistemlerde (Raspberry Pi 4) verimli çalışma kapasitesi nedeniyle YOLOv8-nano (YOLOv8n) model varyantı kullanılmıştır. YOLOv8n, 3,2 milyon parametre ve 8,7 Saniyedeki Milyar Kayan Nokta İşlemi (GFLOPs) hesaplama maliyeti ile YOLO ailesinin en hafif modeli olup, doğruluk-verimlilik dengesi açısından kenar dağıtım (edge deployment) senaryoları için optimize edilmiştir (Jocher, ve ark, 2023). Model, 640×640 piksel girdi çözünürlüğünde çalışmakta ve tek aşamalı (single-stage) nesne tespit paradigması içerisinde gerçek zamanlı performans sunmaktadır.

YOLOv8 mimarisi, üç ana bileşenden oluşmaktadır: omurga ağı (backbone), boyun ağı (neck) ve tespit başlığı (detection head). Her bileşen, önceki YOLO versiyonlarına kıyasla önemli mimari yenilikler içermektedir (Terven & Córdova-Esparza, 2023).

Omurga Ağı (Backbone): CSPDarknet + C2f Modülleri

YOLOv8'in omurga ağı, Çapraz Aşama Kısmi Darknet (CSPDarknet) mimarisi üzerine inşa edilmiştir. CSP yapısı, özellik haritalarını (feature maps) iki yola bölerek gradyan akışını optimize etmekte ve hesaplama maliyetini azaltmaktadır (Wang, ve ark, 2019). YOLOv8'de, geleneksel CSP modülleri yerine 2 Evrişimli Çapraz Aşama Kısmi (Hızlı Uygulama) (C2f) modülleri kullanılmaktadır. C2f modülleri, YOLOv5'teki C3 modüllerinin geliştirilmiş versiyonu olup, daha zengin gradyan akışı ve daha hızlı çıkarım süresi sağlamaktadır (Jocher, ve ark, 2023). C2f modülü, böl-birleştir (split-concat) yapısı ile birden fazla darboğaz (bottleneck) bloğunu paralel olarak işleyerek, özellik çıkarımında hem derinlik hem de genişlik kazanımı sağlamaktadır.

Omurga ağı, beş aşamalı hiyerarşik özellik çıkarımı gerçekleştirmektedir: P1 (640×640), P2 (320×320), P3 (160×160), P4 (80×80) ve P5 (40×40) çözünürlüklerinde özellik haritaları üretmektedir. Bu çok ölçekli özellik piramidi, farklı boyutlardaki nesnelerin (küçük, orta, büyük) etkili bir şekilde tespit edilmesini sağlamaktadır.

Boyun Ağı (Neck): Hızlı Uzamsal Piramit Havuzlama (SPPF)

Boyun ağı, omurga ağından gelen çok ölçekli özellikleri birleştirerek (feature fusion), tespit başlığına zenginleştirilmiş özellik haritaları iletmektedir. YOLOv8, Hızlı Uzamsal

Piramit Havuzlama (SPPF) modülü ile güçlendirilmiş Öznitelik Piramit Ağı (FPN) ve Yol Toplama Ağı (PAN) yapısını kullanmaktadır (Lin, ve ark, 2017; Liu, ve ark, 2018). SPPF modülü, farklı boyutlardaki maksimum havuzlama (max pooling) işlemlerini seri olarak uygulayarak, çoklu alıcı alanları (receptive fields) elde etmekte ve hesaplama verimliliğini artırmaktadır. Geleneksel Uzamsal Piramit Havuzlama (SPP) modülüne kıyasla, SPPF aynı etkiyi daha az hesaplama maliyeti ile sağlamaktadır (He, ve ark, 2015).

FPN yapısı, yukarıdan aşağıya (top-down) özellik akışı ile yüksek seviye semantik bilgiyi düşük seviye özellik haritalarına yaymaktadır. PAN yapısı ise, aşağıdan yukarıya (bottom-up) yol ile düşük seviye lokalizasyon bilgisini yüksek seviye özelliklere taşımaktadır. Bu çift yönlü özellik füzyonu, hem küçük nesnelerin semantik anlaşılmasını hem de büyük nesnelerin hassas lokalizasyonunu iyileştirmektedir.

Tespit Başlığı (Head): Çapa-Bağımsız (Anchor-Free), Ayırıştırılmış Başlık (Decoupled Head)

YOLOv8, önceki YOLO versiyonlarından farklı olarak çapa-bağımsız (anchor-free) tespit mekanizması kullanmaktadır. Geleneksel çapa-tabanlı (anchor-based) yöntemler, önceden tanımlanmış sınırlayıcı kutu şablonlarına (anchor boxes) dayanmakta ve bu şablonların veri setine özgü ayarlanmasını gerektirmektedir. Çapa-bağımsız (Anchor-free) yaklaşım, her grid hücresinden doğrudan nesne merkezi ve sınırlayıcı kutu boyutlarını tahmin ederek, hiperparametre ayarlama ihtiyacını ortadan kaldırmakta ve genelleme kapasitesini artırmaktadır (Tian, ve ark, 2019).

YOLOv8'in tespit başlığı, decoupled head (ayırıştırılmış başlık) mimarisini benimsemektedir. Bu tasarımda, sınıflandırma (classification) ve lokalizasyon (bounding box regression) görevleri için ayrı evrişimli katmanlar (convolutional layers) kullanılmaktadır. Geleneksel coupled head (birleşik başlık) yapısında, her iki görev aynı özellik haritalarını paylaşmakta ve bu durum görevler arası çatışmaya (task conflict) yol açabilmektedir. Ayırıştırılmış Başlık (Decoupled head), her görevin kendi özelleşmiş özellik temsillerini öğrenmesine izin vererek, eğitim stabilitesini artırmakta ve daha hızlı yakınsama sağlamaktadır (Ge, ve ark, 2021).

3.3.2. Veri Seti

Tehdit tespit modelinin eğitimi ve değerlendirmesi için Roboflow Universe platformundan erişilebilen, kamuya açık "Weapon Detection" veri seti kullanılmıştır (Roboflow Universe, 2023). Veri seti, gerçek dünya gözetleme senaryolarını temsil eden çeşitli çevre koşullarından (değişken aydınlatma, oklüzyon, hareket bulanıklığı, farklı kamera açıları) toplanmış 5,064 adet RGB formatında açıklamalı (annotated) görüntüden oluşmaktadır.

Sınıf Dağılımı ve Veri Seti Kompozisyonu

Veri seti, iki tehdit sınıfı içermektedir:

- Pistol(Tabanca): 2,890 görüntü (%57)
- Knife (Bıçak): 2,174 görüntü (%43)

Bu dağılım, gerçek dünya gözetleme senaryolarında gözlemlenen doğal sınıf dengesizliğini (class imbalance) yansıtmaktadır; pistol:knife oranı yaklaşık 1.33:1 olarak hesaplanmaktadır. Sınıf dengesizliği, azınlık sınıfı (minority class - knife) performansının değerlendirilmesi ve artırımı öğrenme stratejisinin dengesizlik çözme kapasitesinin test edilmesi için önemli bir araştırma fırsatı sunmaktadır (Kutlu & Emiroğlu, 2025).

Etiketleme Formatı ve Açıklama Yapısı

Gerçek değer (Ground-truth) açıklamalar, YOLO formatında sınırlayıcı kutular (bounding boxes) olarak sağlanmıştır. Her açıklama, normalize edilmiş koordinatlar ile tanımlanmaktadır:

- Sınıf Etiket (class label): 0 (pistol) veya 1 (knife)
- Merkez x Koordinatı (x_center): Sınırlayıcı kutunun merkez noktasının yatay koordinatı [0, 1] aralığında
- Merkez y Koordinatı (y_center): Sınırlayıcı kutunun merkez noktasının dikey koordinatı [0, 1] aralığında
- Genişlik (width): Sınırlayıcı kutunun genişliği, görüntü genişliğine normalize edilmiş [0, 1] aralığında
- Yükseklik (height): Sınırlayıcı kutunun yüksekliği, görüntü yüksekliğine normalize edilmiş [0, 1] aralığında

Normalize edilmiş koordinatlar, görüntü boyutlarına bağımsız olarak $[0, 1]$ aralığında ifade edilmektedir. Bu format, farklı çözünürlükteki görüntülerin aynı eğitim hattında kullanılmasını ve YOLOv8 modelinin varsayılan veri yükleme mekanizması ile doğrudan uyumlu olmasını sağlamaktadır.

YOLO formatı örneği (örnek annotation dosyası):

0 0.512 0.345 0.123 0.234

1 0.678 0.567 0.089 0.156

Bu örnekte, ilk satır bir pistol (sınıf 0) tespitini, ikinci satır bir knife (sınıf 1) tespitini temsil etmektedir.

Veri Seti Bölümleme Stratejisi

Veri seti, tabakalı bölme (stratified splitting) yöntemi kullanılarak üç alt kümeye ayrılmıştır:

- Eğitim Seti (Training Set): 3,543 görüntü (%70)
- Doğrulama Seti (Validation Set): 1,013 görüntü (%20)
- Test Seti (Test Set): 508 görüntü (%10)

Tabakalı bölme, sınıf dağılımının (pistol ve knife oranlarının) tüm alt kümelerde korunmasını garanti etmektedir. Bu yaklaşım, modelin her bir alt küme üzerinde temsili performans göstermesini ve sonuçların genelleme kapasitesinin güvenilir bir şekilde değerlendirilmesini sağlamaktadır. Doğrulama seti, eğitim sırasında hiperparametre ayarlaması ve model seçimi için kullanılmış; test seti ise tüm artırılmış öğrenme turları tamamlandıktan sonra nihai performans değerlendirmesi için ayrılmıştır. Her iki set de eğitim sürecine dahil edilmemiş olup, tarafsız ve tutarlı performans ölçümü sağlamaktadır (Kutlu & Emiroğlu, 2025).

Veri Artırma Stratejisi

YOLOv8'in yerleşik veri artırma hatı kullanılmıştır. Bu hat, aşağıdaki teknikleri içermektedir:

- Mozaik Artırma (Mosaic Augmentation): Dört görüntüyü tek bir eğitim örneğinde birleştirme farklı ölçek ve bağlamların öğrenilmesini sağlamaktadır.

- Rastgele Ölçeklendirme (Random Scaling): $\pm\%50$ aralığında - ölçek değişkenliğine karşı dayanıklılık
- Yatay Çevirme (Horizontal Flip): $\%50$ olasılık - oryantasyon değişkenliğine karşı dayanıklılık
- HSV Renk Uzayı Titreşimi (HSV Jittering): Ton (Hue), Doygunluk (Saturation), Değer (Value) kanallarında rastgele değişiklikler - aydınlatma ve renk değişkenliğine karşı dayanıklılık

Bu artırma teknikleri, modelin farklı çevre koşullarına karşı genelleme kapasitesini artırmaktadır (Jocher, ve ark, 2023).

Veri Seti Kalite Özellikleri

Veri seti, aşağıdaki çevre koşullarını kapsayacak şekilde çeşitlendirilmiştir:

- Değişken Aydınlatma: Doğal ışık, iç mekân aydınlatması, düşük ışık koşulları
- Oklüzyon (Kısmi Örtülme): Tehdit nesnelerinin kısmi görünürlüğü ($\%20-80$ arası)
- Hareket Bulanıklığı: Hareketli kamera veya hareketli tehdit nesnesi senaryoları
- Farklı Kamera Açıları: Yüksek açı, düşük açı, yan görünüm, frontal görünüm
- Ölçek Değişkenliği: Yakın plan, orta plan, uzak plan tehdit görselleri
- Arka Plan Karmaşıklığı: Sade arka plan, karmaşık iç mekân, kalabalık ortam

Bu çeşitlilik, modelin gerçek dünya gözetleme senaryolarında karşılaşılabileceği değişkenlere karşı dayanıklılık kazanmasını sağlamaktadır. Veri seti, Roboflow Universe lisansı altında kamuya açık olarak sunulmakta ve akademik araştırmalarda kullanımına izin verilmektedir (Roboflow Universe, 2023).

3.3.3. İlk Model Eğitimi (One-Shot Model)

YOLOv8n modelinin ilk eğitimi (one-shot model), tam eğitim veri seti (3,543 görüntü) kullanılarak gerçekleştirilmiştir. Bu aşama, artırılmış öğrenme stratejisinin performans

kazanımlarını değerlendirmek için referans (benchmark) model oluşturmayı amaçlamaktadır.

Eğitim hiperparametreleri, YOLOv8'in varsayılan ayarları ve literatürde önerilen en iyi uygulamalar dikkate alınarak belirlenmiştir:

- Eniyileyici (Optimizer): Ağırlık Azalmalı Uyarlamalı Moment Kestirimi (AdamW) - ayrıştırılmış ağırlık azalması ile (Loshchilov & Hutter, 2019)
- Başlangıç Öğrenme Oranı (Initial Learning Rate): 0.001
- Öğrenme Oranı Planlaması (LR Schedule): Step decay - her 5 epoch'ta $\times 0.1$ azaltma (kademeli yakınsama için)
- Yığın Boyutu (Batch Size): -1 (Kullanılacak bellek kapasitesinin otomatik ayarlanması için)
- Epoch Sayısı: 50 (yakınsama için yeterli iterasyon)

3.3.4. Artırımlı Öğrenme Stratejisi

Artırımlı öğrenme (incremental learning), modelin yeni verilerle eğitilirken önceki bilgileri kaybetmemesini sağlayan bir yöntemdir (Kirkpatrick, ve ark, 2017). Bu çalışmada, ağırlık mirası ile sıralı ince ayar (sequential fine-tuning with weight inheritance) yöntemi kullanılmıştır. Bu yöntem, her artırımlı turda önceki turdan gelen model ağırlıklarının başlangıç noktası olarak kullanılmasını ve yeni verilerle ince ayar yapılmasını içermektedir.

Artırımlı Öğrenme Protokolü

Artırımlı öğrenme süreci, kullandığımız veri setinin görsellerini, üretilen modeller üstünden tespit edilme esasına dayanarak eğitim setine dahil edilecek şekilde tasarlanmıştır. Bu yapı için bir başlangıç modeli (initial model) oluşturulmuş, eğitim setinin yalnızca %20'si (709 görüntü) kullanılarak başlangıç modeli eğitilmiştir. Kalan %80 (2,834 görüntü), etiketlenmemiş havuz (unlabeled pool) olarak ayrılmış ve sonraki turlarda kademeli olarak eğitime dahil edilmiştir. Bu yapılandırma, gerçek dünya dağıtım senaryolarını simüle etmektedir; sistemler sınırlı etiketli veri ile başlamakta ve operasyonel veri toplama yoluyla kademeli olarak iyileşmektedir (Kutlu & Emiroğlu, 2025). Veri setinde dahil edilecek görsel bittiğinde ise modelin en son versiyonu ile gerçek dünyadan elde edilen tehdit görselleri kullanılarak önerilen sistemde başlangıç

değeri olarak belirlenen 100 görsel birikiminde artırımlı öğrenme gerçekleştirme senaryosuna uygun şekilde bir eğitim daha gerçekleştirilmiştir.

Eğitim hiperparametreleri, artırımlı öğrenme için şu şekilde ayarlanmıştır:

- Eniyileyici: Ağırlık Azalmalı Uyarlamalı Moment Kestirimi (AdamW) - ayrıştırılmış ağırlık azalması ile.
- Başlangıç mirası: Pretrained (bir önceki model)
- Başlangıç Öğrenme Oranı : 0.001
- Öğrenme Oranı Planlaması : Step decay - her 5 epoch'ta $\times 0.1$ azaltma
- Yığın Boyutu: -1
- Epoch Sayısı: 10

Veri seti görselleri için kurgulanan artırımlı öğrenme protokolü, aşağıdaki adımları içermektedir:

Tahmin Aşaması (Inference Phase): Mevcut model, etiketlenmemiş havuzdaki (unlabeled pool) görüntüler üzerinde çıkarım (inference) gerçekleştirmekte ve tehdit tespitleri üretmektedir. Her tespit, sınıf etiketi (pistol veya knife), sınırlayıcı kutu koordinatları ve güven skoru (confidence score) içermektedir.

Doğrulama Aşaması (Validation Phase): Modelin ürettiği tespitler, iki kriterle filtrelenmektedir:

- Sınıf Eşleşmesi (Class Match): Tahmin edilen sınıf etiketi, gerçek değer (ground-truth) sınıf etiketi ile eşleşmelidir.
- Lokalizasyon Kalitesi (Localization Quality): Tahmin edilen sınırlayıcı kutu ile gerçek değer (ground-truth) sınırlayıcı kutu arasındaki Kesişim Birleşim Oranı (IoU) değeri ≥ 0.5 olmalıdır.

IoU metriği, sınırlayıcı kutuların örtüşme oranını ölçmektedir:

$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}} \quad (3.19)$$

$IoU \geq 0.5$ eşiği, Bağlamsal Ortak Nesneler (COCO) ve PASCAL VOC veri setlerinde standart doğrulama kriteri olarak kullanılmaktadır (Lin, ve ark, 2014). Bu eşik, yeterli lokalizasyon doğruluğuna sahip tespitlerin seçilmesini sağlamaktadır.

Örnek Seçimi (Sample Selection): Her iki kriteri de sağlayan tespitler, doğrulanmış örnekler (verified samples) olarak işaretlenmekte ve etiketlenmemiş havuzdan (unlabeled pool) çıkarılarak yeni eğitim setine eklenmektedir. Bu süreç, otomatik etiketleme (pseudo-labeling) mekanizması olarak işlev görmektedir.

İnce Ayar Aşaması (Fine-Tuning Phase): Yeni eğitim seti kullanılarak model, ağırlık mirası (weight inheritance) ile ince ayar yapılmaktadır. Ağırlık mirası, önceki turdan gelen model ağırlıklarının başlangıç noktası olarak kullanılmasını ifade etmektedir. Bu strateji, önceki öğrenilen özelliklerin korunmasını ve yeni örneklerle kademeli uyarlanmayı sağlamaktadır (Rebuffi, ve ark, 2017).

İterasyon: Etiketlenmemiş havuz tükenene veya belirlenen tur sayısına ulaşılan kadar süreç tekrarlanmaktadır.

Katastrofik Unutmanın Önlenmesi

Katastrofik unutma, sürekli öğrenme sistemlerinde kritik bir sorundur; model yeni görevleri öğrenirken önceki görevlerdeki performansını kaybedebilmektedir (Kirkpatrick, ve ark, 2017). Bu çalışmada, her eğitim sonunda katasforik unutma araştırılmış ve gerçekleşmesi durumunda kümülatif eğitim seti ile ağırlık kalıtımının birlikte kullanımı ile önlem alınmıştır.

Katasforik unutma durumuyla karşılaşılması halinde sistemin normal artırımıyla öğrenme protokolünde uyguladığı son model ağırlıkları + yeni görseller stratejisi yerine son model ağırlıklarını kalıtım olarak modelin gelişim sürecinde kullanılan tüm veri seti ile yeni gelen veriseti birlikte eğitim gerçekleştirecek hibrit bir eğitim modeli şekilde tasarlanmıştır. Hibrit yöntemde de artırımıyla öğrenme protokolündeki eğitim parametrelerinin aynısı kullanılacak olup sadece veriseti geçmiş verileri de kapsayacak şekilde tasarlanmıştır.

Artırımlı öğrenme protokolünde gerçekleşen her tur için, modelin performansı sabit doğrulama ve test setleri üzerinde değerlendirilmiştir. Bu yapılandırma, artırımlı öğrenmenin performans evrimini sistematik olarak izlemeyi mümkün kılmaktadır.

3.3.5. Açıklanabilir Yapay Zekâ (Grad-CAM)

Güvenlik-kritik uygulamalarda, derin öğrenme modellerinin yalnızca yüksek doğruluk sağlaması yeterli değildir; modelin karar verme sürecinin şeffaf ve yorumlanabilir olması da gerekmektedir (Selvaraju, ve ark, 2019). Açıklanabilir yapay zeka (Explainable AI - XAI), modelin hangi görsel bölgelere odaklandığını ve hangi özelliklere dayanarak karar verdiğini görselleştirmeyi amaçlamaktadır. Bu çalışmada, Gradyan Ağırlıklı Sınıf Aktivasyon Haritalaması (Grad-CAM) yöntemi kullanılarak, YOLOv8 modelinin artırımlı öğrenme turları boyunca dikkat evrimini (attention evolution) sistematik olarak analiz edilmiştir.

Gradyan Ağırlıklı Sınıf Aktivasyon Haritalaması (Grad-CAM) Teorik Temelleri

Grad-CAM, Evrişimli Sinir Ağlarında (Convolutional Neural Networks - CNN) sınıf-ayırt edici lokalizasyon haritaları (class-discriminative localization maps) üretmek için gradyan bilgisini kullanan bir görselleştirme tekniğidir (Selvaraju, ve ark, 2019). Yöntem, modelin son evrişimli katmanından (final convolutional layer) gelen özellik haritalarını (feature maps) ağırlıklandırarak, belirli bir sınıf tahmininde hangi bölgelerin önemli olduğunu ortaya çıkarmaktadır.

Grad-CAM'in matematiksel formülasyonu şu şekildedir:

$$L_{\text{Grad-CAM}}^c = \text{ReLU} \left(\sum_k \alpha_k^c A^k \right) \quad (3.20)$$

Burada:

- $L_{\text{Grad-CAM}}^c$: Sınıf c için Grad-CAM lokalizasyon haritası
- A^k : k -inci özellik haritası (son evrişimli katmandan)
- α_k^c : Sınıf c için k -inci özellik haritasının önem ağırlığı
- ReLU: Doğrultulmuş Doğrusal Birim aktivasyon fonksiyonu (yalnızca pozitif katkıları görselleştirmek için).

Önem ağırlıkları α_k^c , sınıf skorunun (y^c) özellik haritasına göre gradyanlarının global ortalama havuzlaması (global average pooling) ile hesaplanmaktadır:

$$\alpha_k^c = \frac{1}{Z} \sum_{i,j} \frac{\partial y^c}{\partial A_{ij}^k} \quad (3.21)$$

Burada:

- y^c : Sınıf c için model çıktı skoru (softmax öncesi logit)
- A_{ij}^k : k -inci özellik haritasının (i, j) konumundaki aktivasyon değeri
- Z : Normalizasyon sabiti (özellik haritasının toplam piksel sayısı)

Gradyan terimi $\frac{\partial y^c}{\partial A_{ij}^k}$, sınıf skorunun özellik haritasındaki her piksel konumuna olan duyarlılığını ölçmektedir. Yüksek gradyan değerleri, ilgili piksel konumunun sınıf tahmininde güçlü etkiye sahip olduğunu göstermektedir. Global ortalama havuzlama, tüm piksel konumlarındaki gradyanları birleştirerek, özellik haritasının genel önemini tek bir skalere indirgemektedir.

ReLU aktivasyon fonksiyonunun uygulanması, yalnızca pozitif katkıları (sınıf skorunu artıran bölgeleri) görselleştirmeyi sağlamaktadır. Negatif katkılar (sınıf skorunu azaltan bölgeler) sıfırlanmaktadır. Bu işlem, modelin ilgili sınıf için hangi bölgelere "odaklandığını" net bir şekilde ortaya çıkarmaktadır (Selvaraju, ve ark, 2019).

Grad-CAM Implementasyonu

Grad-CAM implementasyonu, PyTorch framework'ü kullanılarak gerçekleştirilmiştir. YOLOv8 modelinin son evrişimli katmanı (C2f modülünün çıktısı), Grad-CAM için hedef katman olarak seçilmiştir. İleri yayılım (forward pass) sırasında, hedef katmanın aktivasyonları kaydedilmekte; geri yayılım (backward pass) sırasında ise, sınıf skoruna göre gradyanlar hesaplanmaktadır. Elde edilen Grad-CAM haritası, orijinal görüntü çözünürlüğüne (640×640) yeniden boyutlandırılmakta (bilinear interpolation ile) ve ısı haritası (heatmap) olarak görselleştirilmektedir.

Grad-CAM analizi, her artırımı öğrenme turunda test seti üzerinde gerçekleştirilmiş ve HFS görsel değerleri kaydedilmiştir. Bu sistematik yaklaşım, performans metriklerinin

(mAP, F1-score) yanı sıra model odaklanmasının görsel olarak anlaşılabilirliğini sağlamıştır.

3.3.6. Performans Değerlendirme Metrikleri

Artırımlı öğrenme sürecinin etkinliğini ve YOLOv8 modelinin tehdit tespit performansını değerlendirmek amacıyla, nesne tespiti literatüründe standart olarak kabul edilen metrikler kullanılmıştır. Model performansı, kesinlik (precision), duyarlılık (recall), F1-score ve Kesişim Birleşim Oranı eşik değerinde ortalama hassasiyet (mean Average Precision at IoU threshold) 0.5 (mAP@0.5) metrikleri ile ölçülmüştür (Lin, ve ark, 2014; Jocher, ve ark, 2023).

Temel Metrikler

Performans değerlendirmesi, karmaşıklık matrisi (confusion matrix) temelinde tanımlanan dört temel kavram üzerine inşa edilmektedir:

- Doğru Pozitif (TP): Gerçekte tehdit olan ve model tarafından doğru tespit edilen örnekler
- Yanlış Pozitif (FP): Gerçekte tehdit olmayan ancak model tarafından tehdit olarak yanlış tespit edilen örnekler
- Yanlış Negatif (FN): Gerçekte tehdit olan ancak model tarafından tespit edilemeyen (kaçırılan) örnekler
- Doğru Negatif (TN): Gerçekte tehdit olmayan ve model tarafından da tehdit olarak tespit edilmeyen örnekler (nesne tespiti problemlerinde genellikle değerlendirilmez).

Kesinlik (Precision)

Kesinlik, modelin pozitif tahminleri arasında doğru olanların oranını ölçmektedir:

$$\text{Precision} = \frac{TP}{TP + FP} \quad (3.22)$$

Yüksek kesinlik değeri, modelin az sayıda yanlış pozitif (false alarm) ürettiğini göstermektedir. Güvenlik uygulamalarında, yüksek kesinlik, operatörlerin gereksiz uyarılarla meşgul olmamasını sağlamaktadır.

Duyarlılık (Recall)

Duyarlılık, gerçekte var olan tehditlerin ne kadarının tespit edilebildiğini göstermektedir:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (3.23)$$

Yüksek duyarlılık değeri, modelin gerçek tehditlerin çoğunu yakalayabildiğini (az sayıda kaçırılan tehdit) ifade etmektedir. Güvenlik-kritik sistemlerde, yüksek duyarlılık öncelikli bir hedeftir çünkü kaçırılan bir tehdit (false negative) ciddi güvenlik riskleri oluşturabilmektedir.

F1-Score

F1-score, kesinlik ve duyarlılık metriklerinin harmonik ortalaması olarak dengeli bir performans göstergesi sunmaktadır:

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (3.24)$$

F1-score, kesinlik ve duyarlılık arasındaki trade-off'u tek bir metrikte özetlemektedir. Harmonik ortalama kullanımı, iki metrikten biri düşük olduğunda F1-score'un da düşük olmasını garanti etmekte, bu da dengesiz performansı cezalandırmaktadır. F1-score $\in [0, 1]$ aralığında değer almakta olup, 1.0 mükemmel performansı temsil etmektedir.

Ortalama Kesinlik Oranı (mAP@0.5)

mAP@0.5, tüm sınıflar için Ortalama Hassasiyet (AP) değerlerinin ortalamasını hesaplamakta ve nesne tespiti literatüründe en yaygın kullanılan performans metriğidir (Lin, ve ark, 2014; Redmon, ve ark, 2016). AP, kesinlik-duyarlılık eğrisinin altında kalan alanı temsil etmekte ve her bir güven skoru eşiği için kesinlik ve duyarlılık değerleri hesaplanarak elde edilmektedir.

Kesişim Birleşim Oranı (IoU) eşiği 0.5, bir tespitin doğru sayılması için tahmin edilen sınırlayıcı kutu ile ground-truth sınırlayıcı kutu arasındaki minimum örtüşme oranını belirlemektedir, matematiksel formülü bölüm 3.3.4 Formül 19’da verilmiştir.

COCO değerlendirme protokolüne uygun olarak, 11-nokta interpolasyon yöntemi kullanılmıştır (Lin, ve ark, 2014). Bu yöntemde, duyarlılık değerleri 0.0, 0.1, 0.2, ..., 1.0 aralığında 11 noktaya bölünmekte ve her duyarlılık seviyesinde maksimum kesinlik değeri alınmaktadır:

$$AP = \frac{1}{11} \sum_{r \in \{0,0.1,...,1.0\}} p_{\text{interp}}(r) \quad (3.25)$$

Burada $p_{\text{interp}}(r)$, recall değeri r ve üzerindeki tüm recall noktalarında gözlemlenen maksimum precision değeridir:

$$p_{\text{interp}}(r) = \max_{\tilde{r} \geq r} p(\tilde{r}) \quad (3.26)$$

mAP@0.5, tüm sınıflar (pistol ve knife) için hesaplanan AP değerlerinin ortalamasıdır:

$$mAP@0.5 = \frac{1}{N} \sum_{i=1}^N AP_i \quad (3.27)$$

Burada N sınıf sayısını (bu çalışmada $N = 2$) temsil etmektedir.

Sınıf-Özel Performans Analizi

Bu metrikler, özellikle pistol ve knife sınıfları arasındaki doğal dengesizlik (1.33:1 oranı) nedeniyle, her iki sınıf için ayrı ayrı hesaplanmış ve artırılmış öğrenme turları boyunca izlenmiştir (Redmon & Farhadi, 2018). Sınıf-özel performans analizi, artırılmış öğrenme stratejisinin azınlık sınıfı (minority class - knife) üzerindeki etkisini değerlendirmeyi mümkün kılmaktadır.

Performans Değerlendirme Protokolü

Tüm performans metrikleri, her artırılmış öğrenme turunda sabit doğrulama seti (1,013 görüntü) üzerinde hesaplanmıştır. Metrik hesaplamaları, YOLOv8’in yerleşik değerlendirme hattını kullanılarak gerçekleştirilmiş ve Maksimum Olmayan Bastırma

(NMS) algoritması (IoU eşiği = 0.45) ile çakışan tespitler bastırılmıştır (Jocher, ve ark, 2023). NMS, aynı nesne için birden fazla sınırlayıcı kutu tespiti olduğunda, en yüksek güven skoruna sahip kutuyu seçmekte ve arkni elimine etmektedir.

Bu sistematik değerlendirme protokolü, artırılmış öğrenme sürecinin hem performans hem de açıklanabilirlik boyutlarında objektif olarak izlenmesini sağlamaktadır.

3.4.Gerçek Dünya Uyarlaması: iMC ve Ground Cs Sistemi

Bu bölümde, 3.2 ve 3.3'te geliştirilen hibrit güvenlik protokolü ve tehdit tespit metodolojilerinin gerçek dünya uygulamasına entegrasyonu için tasarlanan IMC (Intelligent Mission Cruiser) kara robotunun ve GroundCS (Ground Control Station) adı verilen sunucu-operatör kontrol sisteminin donanım ve yazılım mimarisi detaylandırılmaktadır.

IMC, otonom navigasyon, çevresel algılama ve gerçek zamanlı tehdit tespiti yeteneklerine sahip bir mobil robotik platformdur. Sistemin tasarımı, askeri ve güvenlik uygulamalarında yaygın olarak kullanılan gömülü sistem mimarileri ve sensör füzyon teknolojilerinden yararlanmaktadır (Stentz, ve ark, 2002; Ollero & Maza, 2007).

GroundCS ise, çoklu IMC birimlerinin merkezi olarak izlenmesi, kontrol edilmesi ve koordine edilmesi için tasarlanmış bir yayıncı-abone (publisher-subscriber) mimarisine sahip operatör kontrol istasyonudur. Sistem, gerçek zamanlı video akışı, GPS tabanlı konum izleme, LiDAR harita görselleştirme ve tehdit bildirimlerinin alınması gibi kritik fonksiyonları tek bir arayüz üzerinden sağlamaktadır. GroundCS, insansız hava ve kara araçlarında yaygın olarak kullanılan kontrol istasyonu mimarilerinden esinlenerek geliştirilmiş olup (Murphy, 2000), operatörlerin sürü robotlarını hem otonom hem de manuel modda yönetmesine olanak tanımaktadır.

GroundCS'in kritik rollerinden biri, 3.3'te detaylandırılan YOLOv8 tabanlı tehdit tespit modelinin sürekli gelişimini ve güncellenmesini yönetmektir. Sunucu, IMC'lerden gelen tehdit içerikli görsellerin otomatik doğrulamasını, 3.2'de geliştirilen hibrit güvenlik protokolü aracılığıyla gerçekleştirmektedir. Bu protokol, steganografik gizleme ve ECC dijital imza mekanizmalarını birleştirerek, gelen verilerin bütünlüğünü ve özgünlüğünü

garanti etmektedir. Görsel içerisinden En Az Anlamlı Bit (LSB) yöntemiyle çıkarılan UUID ve metadata bilgileri ile iletilen parametrelerin tutarlılığı kontrol edilir; ardından IMC'nin kayıtlı açık anahtarı (public key) kullanılarak ECC dijital imzası doğrulanır (Johnson, ve ark, 2001; Cheddad, ve ark, 2010). Bu doğrulama sürecini başarıyla geçen tehdit görselleri, yalnızca meşru IMC birimlerinden geldiği ve iletim sırasında manipüle edilmediği garanti edilen veriler olarak kabul edilmektedir.

Doğrulananan tehdit görselleri, GroundCS üzerinde çalışan artırımlı öğrenme (incremental learning) modülüne otomatik olarak aktarılmaktadır. Bu modül, 3.3'te tanımlanan artırımlı öğrenme stratejisi kapsamında, yeni tehdit örneklerini mevcut YOLOv8 modeline entegre ederek, modelin zamanla yeni tehdit türlerine uyum sağlamasını ve tespit performansının iyileştirilmesini sağlamaktadır (Rebuffi, ve ark, 2017; Parisi, ve ark, 2019). Artırımlı öğrenme yaklaşımı, geleneksel yeniden eğitim yöntemlerinin aksine, modelin önceden öğrendiği bilgileri korurken yeni bilgileri entegre etmektedir (Kirkpatrick, ve ark 2017). Güncellenen model, merkezi sunucu üzerinden tüm IMC birimlerine güvenli bir şekilde dağıtılarak, sürü genelinde tutarlı ve güncel tehdit tespiti kapasitesi sağlanmaktadır.

3.2'de tanımlanan hibrit güvenlik protokolünün bir diğer kritik bileşeni olan DDoS koruması, sistemin operasyonel sürekliliğini sağlamak üzere GroundCS'e entegre edilmiştir. Bu mekanizma, doğrulama sürecinde sürekli başarısız olan istemci (client) IP adreslerini otomatik olarak engelleyerek, kötü niyetli aktörlerin sisteme erişimini kısıtlamaktadır (Zargar, ve ark, 2013). Özellikle, ECC imza doğrulamasında tekrarlı başarısızlık ya da aynı imza ile istek gönderen veya steganografik UUID çıkarma işleminde tutarsızlık üreten IP adresleri, aşamalı engelleme (progressive blocking) algoritması ile tespit edilmekte ve sunucu kaynaklarına erişimleri sonlandırılmaktadır. Bu yaklaşım, hem volumetrik DDoS saldırılarını (yüksek hacimli istek trafiği) hem de sofistike kimlik doğrulama tabanlı saldırıları etkisiz hale getirerek, sistemin güvenliğini sürdürülebilir kılmaktadır. Bu otomatik doğrulama, öğrenme ve koruma döngüsü, insan müdahalesine gerek kalmadan sistemin operasyonel ortamda karşılaştığı yeni tehdit senaryolarına kendini adapte etmesini mümkün kılmaktadır. Ayrıca GroundCS, güvenlik operasyonlarının merkezi koordinasyonunu, veri bütünlüğü kontrolünü ve model

güncellemelerinin güvenli dağıtımını tek bir entegre platformda birleştirerek, otonom güvenlik sistemlerinde kritik öneme sahip bir altyapı sunmaktadır.

3.4.1. IMC Donanım Mimarisi ve Sensör Entegrasyonu

IMC'nin donanım mimarisi, yarı otonom ve manuel hareket kontrolü, algılama sistemleri ve iletişim altyapısı olmak üzere üç ana katmandan oluşmaktadır. Bu modüler yapı, otonom kara robotlarında yaygın olarak tercih edilen katmanlı mimari yaklaşımına dayanmaktadır (Murphy, 2000).

Mekanik Yapı ve Hareket Sistemi

IMC'nin hareket sistemi, dört tekerlekten bağımsız çekiş (4WD - Four Wheel Independent Drive) prensibi üzerine inşa edilmiştir. Her tekerlek, bağımsız motor kontrolüne sahip olup, rotasyon hareketi diferansiyel sürüş (differential drive) mantığıyla gerçekleştirilmektedir. Bu yapı, mobil robotlarda yüksek manevra kabiliyeti ve stabilite sağlaması nedeniyle tercih edilmektedir (Siegwart & Nourbakhsh, 2004; Dudek & Jenkin, 2010). Diferansiyel sürüş sistemi, sol ve sağ tekerleklerin farklı yönlerde ve hızlarda dönmesiyle robotun yönelim değişikliği gerçekleştirmesine olanak tanımaktadır. Özellikle dar alanlarda yerinde dönüş (zero-radius turn) yapabilme özelliği, askeri ve güvenlik görevlerinde kritik öneme sahiptir (Skulimowski, ve ark, 2025). Bu bağlamda robotun doğrusal hareket (ileri ve geri yönde) haricinde dönüş hareketlerini merkezi ve açısal dönüş olarak sınıflandırmak mümkündür.

Hareket sisteminin temel özellikleri:

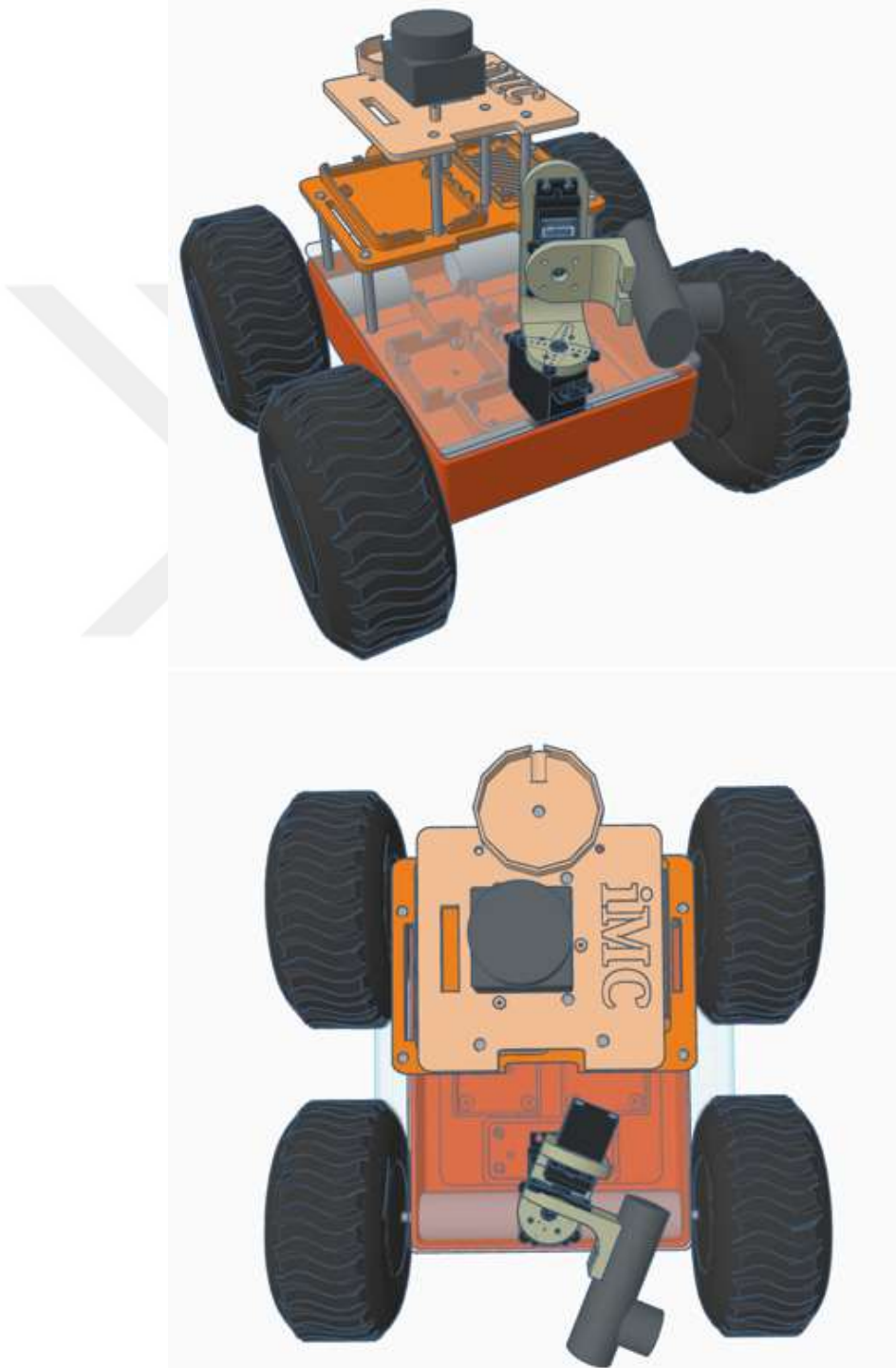
- Bağımsız tekerlek kontrolü: Her tekerleğin hız ve yön kontrolü ayrı ayrı yapılabilir.
- Diferansiyel rotasyon: Sol ve sağ tekerleklerin zıt yönde dönmesiyle yerinde dönüş gerçekleştirilmektedir.

IMC'nin mekanik özellikleri şu şekildedir:

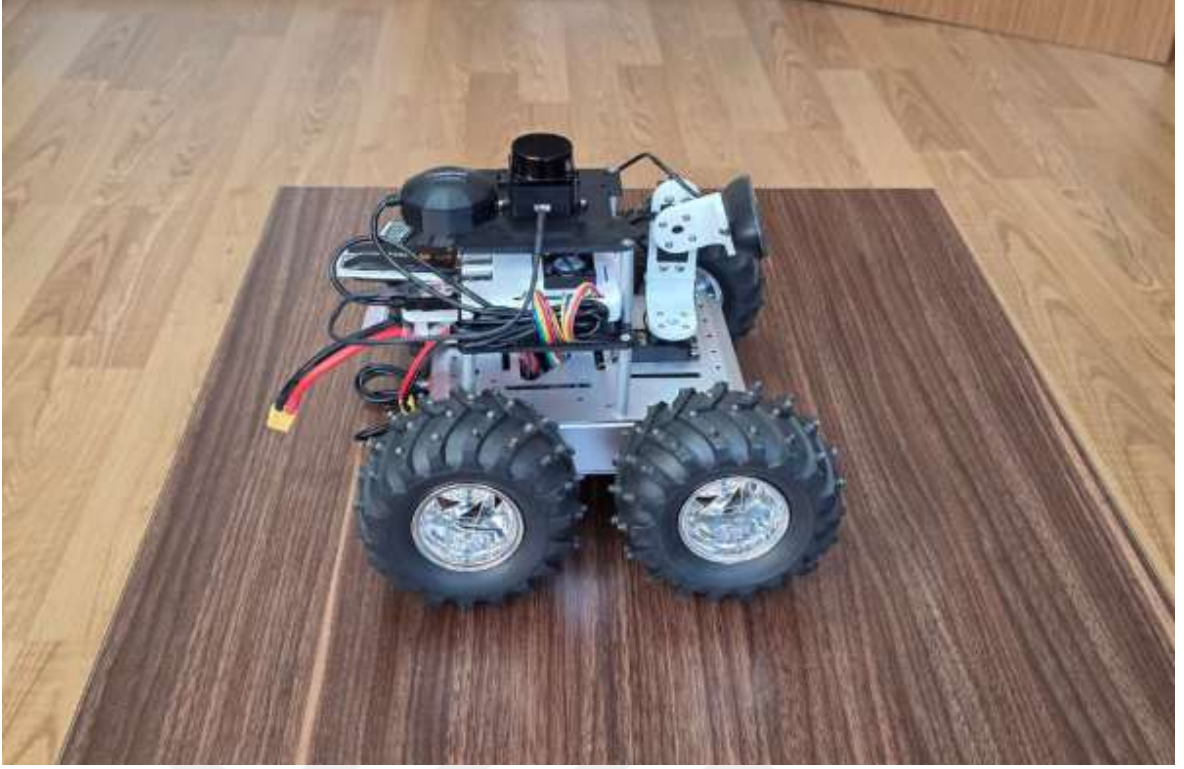
- Boyutlar (U × G × Y): 260x290x150 mm
- Ağırlık: 2,2 kg
- Engel aşma kapasitesi: 5,5 cm

- Tekerlek çapı: 110 mm
- Aks mesafesi: 150 mm

Tasarlanan iMC'ye ait 3 boyutlu tasarım şekli şekil 3.5'de tasarım sonrası gerçek görüntüler resim 3.1-4'de verilmiştir.



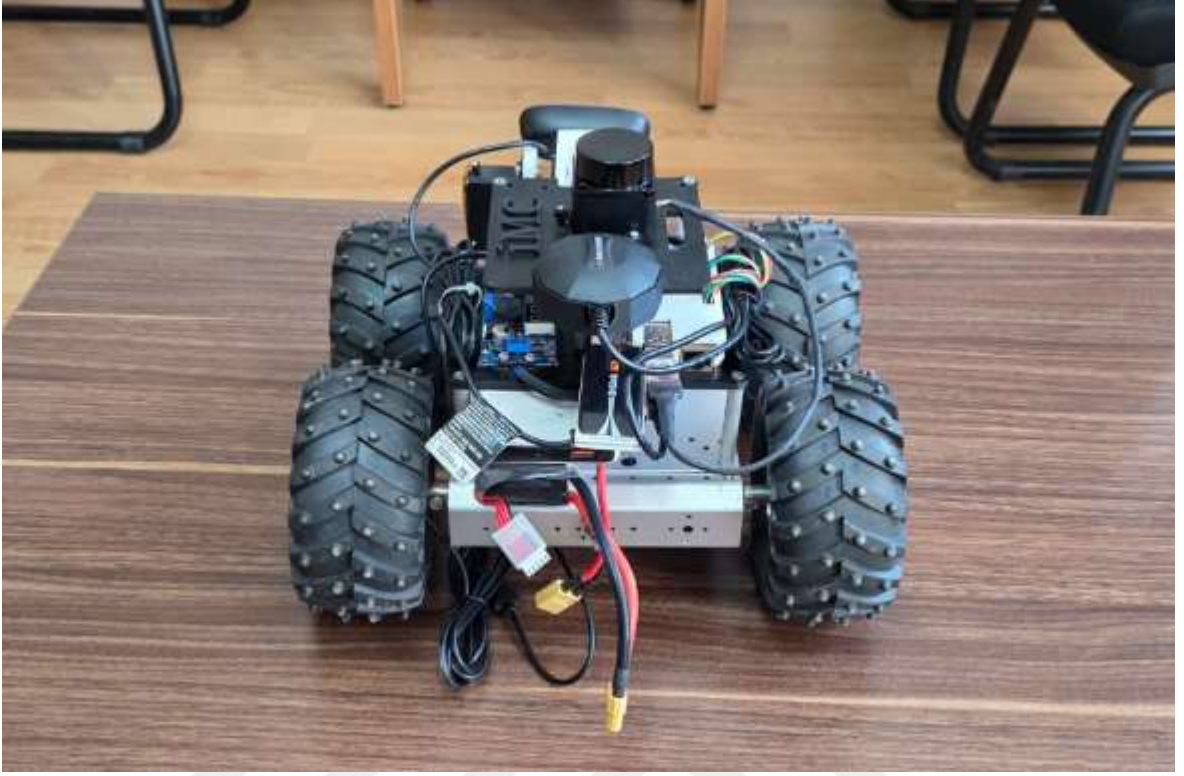
Şekil 3.5. iMC Teknik 3B çizimi



Resim 3.1. iMC sađdan grnř



Resim 3.2. iMC soldan grnř



Resim 3.3. iMC arkadan görünüş



Resim 3.4. iMC önden görünüş

Sistemin merkezi işlem birimi olarak Raspberry Pi 4 mikro bilgisayarı kullanılmaktadır. Raspberry Pi, düşük güç tüketimi, kompakt boyutu ve yeterli hesaplama gücü nedeniyle mobil robotik uygulamalarda yaygın olarak tercih edilmektedir (Vujovic & Maksimovic, 2015; Johnston & Cox, 2017). Bu platform, tüm sensör verilerinin işlenmesi, karar mekanizmalarının çalıştırılması ve aktüatör kontrolünün gerçekleştirilmesi görevlerini üstlenmektedir.

Motorların hız ve yön kontrolü için 2 adet motor sürücü kartı kullanılmaktadır. Bu sürücüler (L298N), Raspberry Pi'den gelen Darbe Genişlik Modülasyonu (PWM) sinyalleriyle hassas hız ve yön kontrolü sağlamaktadır (Hughes, 2019).

Sensör Sistemi

IMC üzerinde üç kritik sensör grubu bulunmaktadır: konum belirleme, çevresel haritalama ve görsel algılama. Bu çoklu sensör yaklaşımı, sensör füzyonu yoluyla daha güvenilir çevresel algılama sağlamaktadır (Durrant-Whyte & Henderson, 2008).

Konum Belirleme Sistemi (GPS/GNSS)

Robot üzerinde GPS modülü entegre edilmiştir. Bu modül, robotun global koordinatlarını sürekli olarak belirleyerek yarı otonom navigasyon ve sürü koordinasyonu için temel veri sağlamaktadır. GPS tabanlı konum belirleme, açık alanlarda otonom navigasyon için yaygın olarak kullanılmaktadır (Kaplan & Hegarty, 2017). Ground CS kontrol panelinde bulunan Leaflet Map üzerinde her bir robotun konumu GPS koordinatları ile işaretlenmekte, gerçek zamanlı olarak harita üzerinde tüm iMC robotları aynı anda izlenebilmektedir. Yine harita üzerinde her bir robot için devriye görevi tanımlanabilmekte, harita üzerinde devriye görevi için rota çizimi yapılarak robotun bu rota doğrultusunda yarı otonom sürüşü gerçekleştirilebilmektedir.

Çevresel Haritalama (LiDAR)

Engel tespiti ve 2D haritalama için LiDAR sensörü kullanılmaktadır. Sensör, robotun çevresini 5-10 Hz/sn frekansla 20 metreye kadar 5-10 cm doğruluğa kadar 360° tarayarak nokta bulutu (point cloud) verisi üretmektedir. LiDAR teknolojisi, mesafe ölçümünde yüksek doğruluk ve hava koşullarından bağımsız çalışma kabiliyeti nedeniyle mobil robotlarda yaygın olarak tercih edilmektedir (Li, ve ark, 2022; Dong & Chen, 2017).

Özellikle bitki örtüsü ve karmaşık arazilerde engel tespiti için etkili bir çözüm sunmaktadır (Reina, ve ark, 2016). GroundCS yazılımının ilerleyen versiyonlarında LiDAR ve monocular kamera kullanılarak iç ortam haritalama için SLAM modülünün geliştirilmesi hedeflenmektedir.

Görsel Algılama Sistemi (Kamera)

Tehdit tespiti için robot üzerinde dikey ve yatay ekseninde Ground CS yazılımıyla uzaktan kontrol edilebilen 720p çözünürlüğe sahip bir adet kamera bulunmaktadır. Yatay yörüngede 120 ve dikey yörüngede 160 derece hareket için ek bir yönlendirme donanımı (Pan-Tilt) üstüne sabitlenmiş ve PT kamera haline getirilmiştir. PT kamera sistemleri, geniş alan gözetimi ve hedef takibi için askeri ve güvenlik uygulamalarında yaygın olarak kullanılmaktadır (Hampapur, ve ark, 2005).

Kamera özellikleri:

- Çözünürlük: 720 piksel (HD)
- Pan/Tilt kontrolü: Servo motor tabanlı (2xMG996 Servo ile 2B uzayda en çok 160 derece görüş alanı)
- Servo kontrol sinyali: PWM, 50Hz (PCA 9685 Servo Sürücü modülü ile)
- Arayüz: USB 2.0

Kamera görüntüsü, gerçek zamanlı olarak Ground CS'e iletilmekte ve aynı zamanda Raspberry Pi 4 üzerinde çalışan YOLOv8n modeli ile anlık tehdit tespiti yapılmaktadır. Gerçek zamanlı video iletimi, düşük gecikme ve yüksek bant genişliği gerektiren kritik bir işlemdir (Winkler & Mohandas, 2008).

Haberleşme ve Güç Yönetimi

Haberleşme Sistemi

iMC, Ground CS ile sürekli bağlantı halinde olması gereken bir otonom platformdur. Bu amaçla IEEE 802.11 WiFi standartlarında yayın yapılan özel ağ (private network) üzerinden iletişim sağlanmaktadır. Kablosuz iletişim altyapısı, mobil robotik sistemlerde kritik bir bileşen olup, veri iletim hızı, gecikme ve güvenilirlik açısından optimize edilmesi gerekmektedir (Frew & Brown, 2008).

İletişim alt yapısı üç katmandan oluşmaktadır:

1. Veri İletim Katmanı: GPS ve LiDAR verileri sürekli olarak GroundCS'e iletilmektedir.
2. Video Akış Katmanı: 720p kamera görüntüsü gerçek zamanlı olarak merkeze aktarılmaktadır.
3. Tehdit Bildirim Katmanı: Tespit edilen tehditler, 3.3'te geliştirilen Steganografi + ECC yöntemiyle güvenli şekilde iletilmektedir.

İletişim donanımı:

- WiFi modülü: Dlink DWA-140 Usb
- WiFi standardı: IEEE 802.11ac
- Frekans bandı: 2.4 ghz
- Anten tipi: Dahili

Güç Yönetimi

Otonom mobil robotlarda güç yönetimi, operasyonel süreyi doğrudan etkileyen kritik bir faktördür (Mei, ve ark, 2004). iMC'nin güç sistemi, tüm elektronik bileşenlerin güvenli ve verimli çalışmasını sağlayacak şekilde tasarlanmıştır.

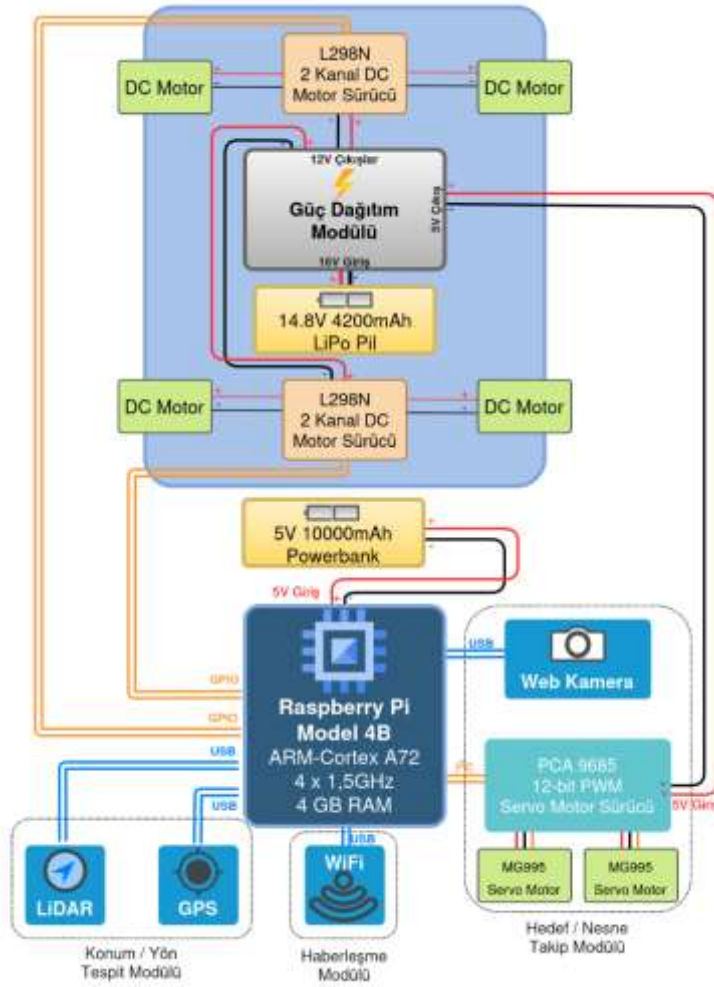
Verimli ve güvenli güç yönetimi için DC motorlar, DC motor sürücüleri, servo motorlar ve servo motor sürücüsü 4S LiPo batarya ile beslenmektedir. İMC görev robotlarının ana kontrol ünitesi olan Raspberry Pi ve bu bilgisayara bağlı sensörlere ise 10000 mAh'lık power bank üzerinden güç sağlanmaktadır. Bu yapı ile yüksek tork gerektiren zorlu arazi görevlerinde yüksek güç çekecek olan DC motorların güç yönetimi kontrol kartından yalıtılmış olmaktadır. Öte yandan şarjın bitmesinden kaynaklanan görev robotunun hareket kabiliyetini yitirmesi halinde dahi kontrol kartını besleyen ek güç kaynağı sayesinde robottan halen koordinat, sistem durumu ve video görüntü verisi alınmaya devam edilebilmektedir.

Güç sistemi teknik detayları şu şekildedir:

- Batarya tipi: LiPo (4S) ve Powerbank
- Batarya voltajı: 14,8V (LiPo), 5V (Powerbank)
- Batarya kapasitesi: 4200 mAh (LiPo), 10000 mAh (Powerbank)

- Voltaj regülatörleri: LM2596 5V-12V/3A (Harici bileşenlere ek güç sağlanması gerektiğinde kullanılmak üzere)
- Güç dağıtım kartı: Matek Mini PDB / BEC 5V & 12v
- Şarj sistemi: Harici şarj ünitesi (XT60 konnektör ile balanslı şarj)

Şekil 3.6 İMC'nin donanım mimarisi blok diyagramını göstermektedir.



Şekil 3.6.iMC Donanım mimarisi

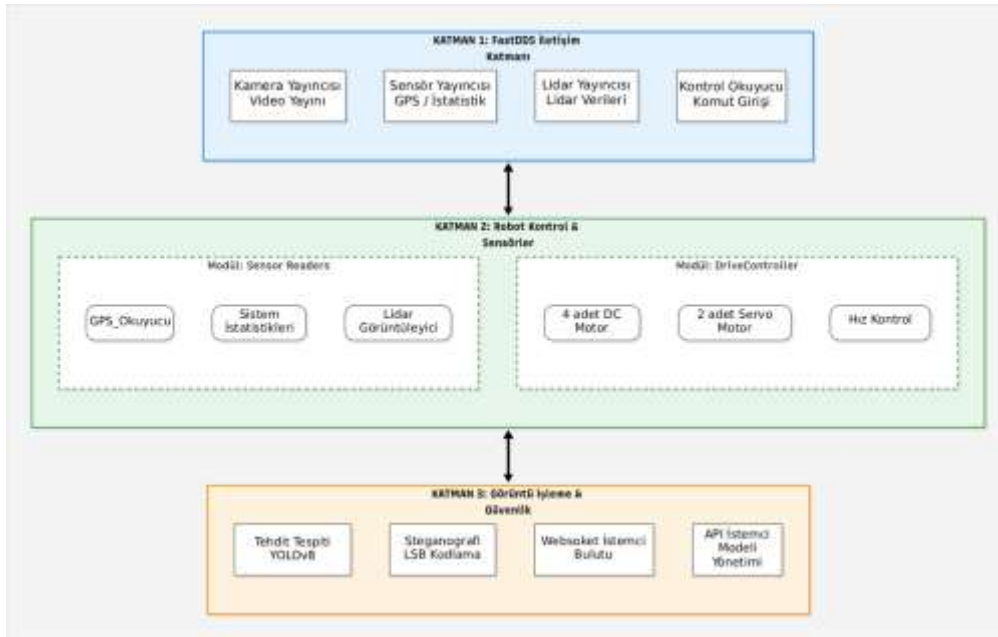
3.4.2. iMC Yazılım Mimarisi

iMC'nin yazılım mimarisi (RemoteCS), modüler bir yapıya sahip olup her bir fonksiyonel birim bağımsız modüller olarak çalışmaktadır. Sistem, Python programlama dili kullanılarak geliştirilmiş olup, yarı otonom ve uzaktan kontrollü robot yönetim sistemi olarak tasarlanmıştır.

Sistem; merkezi olmayan bir mimariye Veri Dağıtım Servisi (DDS) sahip robotun gerçek zamanlı olarak;

- Video akışı yayınlanması
- Sensor ve Lidar verilerinin toplanması ve gönderilmesi
- Navigasyon ve Yol Planlama / Yarı Otonom Sürüş
- Yarı Otonom / Manuel Hareket Kontrolü
- Tehdit tespiti Modülü
- Kriptografik olarak kodlanmış verilerin gönderilmesi görevlerini yerine getiren modüllerden oluşmaktadır.

Bu katmanlı mimari, robotik sistemlerde yaygın olarak kullanılan davranış tabanlı (behavior-based) ve hiyerarşik kontrol yaklaşımlarını birleştirmektedir (Refis, ve ark, 2025). Katmanlı mimari akış diyagramı şekil 3.7’de verilmiştir.



Şekil 3.6.iMC Katmanlı yazılım mimarisi

Modüler Yazılım Yapısı

IMC yazılım mimarisi aşağıdaki ana modüllerden oluşmaktadır:

1. Video Akışı Yayıncısı (Camera Publisher)

Bu modül, kamera görüntüsünü gerçek zamanlı olarak Ground CS'e iletmektedir. Video akışı, H.264 codec kullanılarak ve aktarılmaktadır. Adaptif bitrate mekanizması, ağ koşullarına göre video kalitesini dinamik olarak ayarlayarak akışın sürekliliğini sağlamaktadır (Winkler & Mohandas, 2008).

Video akışı algoritması Algoritma 11'de verilmiştir.

Algoritma 11: Video akış algoritması

```

INPUT: domain_id (default: 0)
OUTPUT: publisher

BEGIN
    // Adım 1: DDS katılımcısı oluştur
    participant ← CreateDomainParticipant(domain_id: 0)

    // Adım 2: Veri tipini kaydet
    RegisterType(participant, "Camera", CameraDataType)

    // Adım 3: Socket tamponlarını yapılandır
    ConfigureSocket(listen: 212KB, send: 212KB)

    // Adım 4: Yayıncı oluştur
    publisher ← CreatePublisher(
        participant: participant,
        topic: "Camera",
        qos: BEST_EFFORT
    )

    RETURN publisher
END

```

ALGORITHM: Publish Frame

INPUT: img (numpy.ndarray)

OUTPUT: None

BEGIN

 // Adım 1: JPEG sıkıştırma

 jpeg_bytes ← CompressJPEG(img, quality: 80)

 // Adım 2: Veri nesnesi oluştur

 data ← CameraDataType(

 image_data: jpeg_bytes,

 device_name: "camera_01"

)

 // Adım 3: Yayınla

 publisher.Write(data)

END

ALGORITHM: Wait Discovery

INPUT: publisher

OUTPUT: None

BEGIN

 // En az 1 abone bulunana kadar bekle

 REPEAT

 Sleep(100ms)

 UNTIL publisher.matched_subscribers $\geq$ 1

END

Bu modül, tüm sensör verilerinin toplanması, işlenmesi ve sistem içinde paylaşılmasından sorumludur. Sensör füzyonu yaklaşımı, çoklu sensör verilerinin birleştirilmesiyle daha güvenilir çevresel algılama sağlamaktadır (Hall & Llinas, 2001). Sensör verilerinin toplanma ve yayınlanma algoritması Algoritma 12 'de verilmiştir.

Algoritma 12: Sensör verileri toplanma ve yayınlanma algoritması

INPUT: None

OUTPUT: sensor_data (dictionary)

BEGIN

// Adım 1: GPS verilerini oku

gps_data ← ReadGPSD(device: "/dev/ttyUSB0")

latitude ← gps_data.latitude

longitude ← gps_data.longitude

altitude ← gps_data.altitude

speed ← gps_data.speed

time ← gps_data.time

SyncSystemTime(time)

// Adım 2: Sistem istatistiklerini topla

cpu_usage ← psutil.cpu_percent()

ram_usage ← psutil.virtual_memory().percent

cpu_temp ← ReadFile("/sys/class/thermal/thermal_zone0/temp") / 1000

// Adım 3: LiDAR verilerini oku

lidar_data ← ReadSerial(

port: "/dev/lidar",

baudrate: 9600,

interval: 50ms

)

// Adım 4: Tüm verileri birleştir

sensor_data ← {

latitude: latitude,

longitude: longitude,

altitude: altitude,

speed: speed,

```
    cpu: cpu_usage,
    ram: ram_usage,
    temperature: cpu_temp,
    lidar: lidar_data
}
RETURN sensor_data
END
// Sensör Verisi Yayınlama
INPUT: sensor_data (dictionary)
OUTPUT: None
BEGIN
    // Adım 1: DDS yayıncısı oluştur
    publisher ← CreatePublisher(
        topic: "Sensors",
        qos: RELIABLE,
        data_type: DataTypes.Sensors
    )
    // Adım 2: Dictionary'yi DDS nesnesine dönüştür
    sensors_msg ← DataTypes.Sensors()
    sensors_msg.latitude ← sensor_data.latitude
    sensors_msg.longitude ← sensor_data.longitude
    sensors_msg.speed ← sensor_data.speed
    sensors_msg.cpu ← sensor_data.cpu
    sensors_msg.ram ← sensor_data.ram
    sensors_msg.temperature ← sensor_data.temperature
    // Adım 3: Yayınla
    publisher.Write(sensors_msg)
END
```

Sensör yönetim modülü, GPS koordinatlarını (enlem, boylam, yükseklik, hız) sürekli olarak okumakta; LiDAR sensöründen gelen 360° tarama verileriyle nokta bulutu (point cloud) oluşturmakta ve merkezi kontrol yazılımının işleyebileceği formatta

birleştirmektedir. DDS veri dağıtım katmanı bu üç veri kaynağından gelen veriyi birleştirerek robotun çevresel durumunu bütünsel olarak abonelerine yayınlamaktadır.

3. Navigasyon ve Yol Planlama Modülü

Bu modül, GPS verisi ve LiDAR engel haritası kullanılarak robotun hedef konuma güvenli bir şekilde ulaşması için yol planlama ve navigasyon işlemlerini gerçekleştirmektedir. Yol planlama algoritmaları, statik ve dinamik engelleri dikkate alarak optimal yol hesaplamaktadır (LaValle, 2006; Choset, ve ark, 2005).

Navigasyon modülü, hedef koordinatları Ground CS'den aldıktan sonra, LiDAR verisinden engel haritası oluşturmakta ve A* yol planlama algoritmasını kullanarak optimal yol hesaplamaktadır. Hesaplanan yol, doğrusal hız (v) ve açısal hız (ω) komutlarına dönüştürülerek hareket kontrol modülüne iletilmektedir.

4. Hareket Kontrol Modülü

Bu modül, navigasyon modülünden gelen hız komutlarını alarak diferansiyel sürüş kinematiği ile tekerlek hızlarına dönüştürmekte ve motor sürücülere PWM sinyalleri göndermektedir. Diferansiyel sürüş kinematiği, mobil robotlarda yaygın olarak kullanılan bir hareket modeli olup, matematiksel olarak iyi tanımlanmıştır (Siegwart & Nourbakhsh, 2004).

Hareket kontrol modülü de 2 alt modülden oluşmaktadır: Bunlar pilot tarafından gönderilen komutları işleyerek manuel kontrol sağlayan sürüş modülü (RobotController) ve Ground CS tarafından gönderilen koordinatlar kullanılarak gerçekleştirilecek olan yarı otonom sürüş kontrol modülüdür. Her iki modül için de robot üzerinde bulunan bütün motorların yönetimini sağlayan MotorController ve DriveController sınıfları tasarlanmış olup algoritmaları Algoritma 13 ve 14'de açıklanmıştır.

MotorController sınıfı bir motorun kontrolü için ihtiyaç duyulan kontrol pinleri, pwm pini ve frekans değerlerini ayarlayan temel sınıf olup bu sınıf miras alınarak DriveController sınıfının metotları oluşturulmuştur.

Algoritma 13: Motor Kontrol Algoritması

INPUT: pin_forward, pin_backward, pin_pwm, frequency (default: 100Hz)

OUTPUT: MotorController instance

BEGIN

// Adım 1: GPIO pinlerini yapılandır

ConfigureGPIO(pin_forward, mode: OUTPUT)

ConfigureGPIO(pin_backward, mode: OUTPUT)

// Adım 2: PWM sinyalini başlat

pwm ← InitializePWM(

pin: pin_pwm,

frequency: frequency

)

pwm.Start(duty_cycle: 0)

RETURN MotorController

END

ALGORITHM: Set Motor Speed

INPUT: speed (-100..100)

OUTPUT: None

BEGIN

// Adım 1: Hız değerini kontrol et ve yönü belirle

IF speed > 0 THEN

// İleri hareket

SetPin(pin_forward, HIGH)

SetPin(pin_backward, LOW)

duty_cycle ← speed

ELSE IF speed < 0 THEN

// Geri hareket

SetPin(pin_forward, LOW)

SetPin(pin_backward, HIGH)

duty_cycle ← abs(speed)

ELSE

// Dur

SetPin(pin_forward, LOW)

```

SetPin(pin_backward, LOW)
duty_cycle ← 0
END IF

// Adım 2: PWM duty cycle'ı ayarla
pwm.ChangeDutyCycle(duty_cycle)
END

```

DriveController sınıfı hem hareketi sağlayacak olan 4 adet DC motorun hız ve yönünü hem de kameranın hareketi için pan/tilt görevini üstlenen 2 adet servo motorun kontrolünü üstlenmektedir.

Algoritma 14. Sürüş Kontrol Algoritması

Motor Layout:

```

[LF]————[RF]
|             |
[LB]————[RB]

```

Movement Functions:

```

└─ _forward(speed):
|   └─ Set all 4 motors: +speed (aynı yön)
└─ _backward(speed):
|   └─ Set all 4 motors: -speed (ters yön)
└─ _turn_Cent_left(speed): [Center Point Rotation]
|   └─ Left motors: -speed (backward)
|   └─ Right motors: +speed (forward)
└─ _turn_Ang_left(speed): [Angular Motion]
|   └─ Left motors: +speed/5 (slow)
|   └─ Right motors: +speed (fast)
└─ _turnBack_Ang_left(speed): [Backward Turn]
|   └─ Left motors: -speed/5
|   └─ Right motors: -speed
└─ Servo Control (Camera Pan/Tilt):
|   └─ servo_motor_H (angle: 0-120°) ← Horizontal
|   └─ servo_motor_V (angle: 20-180°) ← Vertical

```

Diferansiyel sürüş kinematiği formülasyonu şu şekildedir:

$$v_{\text{left}} = v - \frac{\omega \cdot L}{2} \quad (3.28)$$

$$v_{\text{right}} = v + \frac{\omega \cdot L}{2} \quad (3.29)$$

Burada v doğrusal hızı, ω açısal hızı ve L tekerlek aksları arası mesafeyi temsil etmektedir. Sol ve sağ tekerlek hızları hesaplandıktan sonra, Oransal-İntegral-Türevsel (PID) kontrolör ile motor PWM sinyallerine dönüştürülmekte ve hassas hız kontrolü sağlanmaktadır (Åström & Murray, 2008).

Manuel Sürüş Modu

Bu modda iken robotlar ayrı ayrı Ground CS yazılımı tarafından kontrol edilebilmekte olup aktif bağlantı sağlanan robot kullanıcı/pilot tarafından gönderilen komutları ControlReader okumakta ve robotun hareket bileşenlerinin kontrolü sağlanmaktadır. ControlReader algoritması Algoritma 15’de verilmiştir.

Algoritma 15:Sürüş Kontrol Algoritması

INPUT: event (string)

OUTPUT: None

BEGIN

SWITCH event DO

CASE 'drive_control_W': → move_forward()

CASE 'drive_control_S': → move_backward()

CASE 'drive_control_A': → turn_C_left()

CASE 'drive_control_D': → turn_C_right()

CASE 'drive_control_Q': → turn_Ang_left()

CASE 'drive_control_E': → turn_Ang_right()

CASE 'camera_control_Up': → s_turn_up()

CASE 'camera_control_Down': → s_turn_down()

CASE 'camera_control_Left': → s_turn_left()

CASE 'camera_control_Right': → s_turn_right()

```

CASE 'camera_control_plus': → increase_speed()
CASE 'camera_control_minus': → decrease_speed()
CASE 'drive_control_stop': → stop()
DEFAULT: → PRINT "Unknown command"
END SWITCH
END

```

Yarı Otonom Sürüş Modu

Yarı otonom sürüş modu, operatörün Ground CS üzerinde harita tabanlı arayüzden yol noktaları (waypoint) belirlemesi ve iMC'nin bu noktalar arasında GPS destekli navigasyon ile otonom hareket etmesi prensibine dayanır. Bu modda operatör, harita üzerinden robotun uğraması gereken lokasyonları işaretler ve hareket talimatını gönderir. Robot, belirlenen yol noktalarına sırasıyla ilerlerken LiDAR sensörü ile önüne çıkan engelleri algılar ve bu engelleri aşmak için gerekli manevraları otomatik olarak gerçekleştirir. Rota boyunca, entegre kamera sistemi üzerinden YOLOv8 tabanlı tehdit tespit modülü aktif çalışarak güvenlik taraması yapar. Operatör, Ground CS arayüzünden robotun anlık konumunu, rota ilerlemesini ve tespit edilen tehditleri gerçek zamanlı izler. Bu mod, geniş alanların sistematik taranması ve gözetim görevlerinde operatör iş yükünü azaltırken operasyonel etkinliği artırır.

5. Tehdit Tespit Modülü

Bu modül, 3.3'te geliştirilen YOLOv8 modelini kullanarak kamera görüntüsü üzerinde gerçek zamanlı tehdit tespiti yapmaktadır. Derin öğrenme tabanlı nesne tespiti, güvenlik uygulamalarında yüksek doğruluk oranları sunmaktadır (Redmon & Farhadi, 2018). Tehdit tespit algoritması Algoritma 16'de verilmiştir.

Algoritma 16.Tehdit Tespit Algoritması

INPUT: None

OUTPUT: None

BEGIN

 // Başlatma

 model $\leftarrow$ YOLO(get_ai_model_name())

 labels $\leftarrow$ api_client.get_labels()

 source $\leftarrow$ VideoCapture(0)

 // Tespit döngüsü

 WHILE is_threat_detection_process DO

 frame $\leftarrow$ source.read()

 IF frame_counter mod 25 $\neq$ 0 THEN CONTINUE // 3 FPS

 results $\leftarrow$ model.predict(frame)

 FOR EACH detection IN results WHERE confidence > 0.70 DO

 normalized $\leftarrow$ Normalize(detection.bbox, frame.size)

 DrawBoundingBox(frame, detection.bbox)

 SaveImage(frame, "threat.png")

 api_client.send_threat(normalized, detection.label)

 END FOR

 Display(frame)

 END WHILE

END

ALGORITHM: Normalize Koordinat

INPUT: bbox {x1, y1, x2, y2}, size {width, height}

OUTPUT: {x, y, w, h}

BEGIN

 x $\leftarrow$ (x1 + x2) / (2 $\times$ width)

 y $\leftarrow$ (y1 + y2) / (2 $\times$ height)

 w $\leftarrow$ (x2 - x1) / width

 h $\leftarrow$ (y2 - y1) / height

 RETURN {x, y, w, h}

END

Tehdit tespit modülü, kamera görüntüsünü 640×640 piksel çözünürlüğe yeniden boyutlandırmakta, normalize etmekte ve YOLOv8-nano modeline girdi olarak vermektedir. Model çıktısı, tespit edilen nesnelerin sınıf etiketleri (pistol veya knife), güven skorları ve sınırlayıcı kutu koordinatlarını içermektedir. Güven skoru ≥ 0.5 olan tespitler, geçerli tehdit olarak kabul edilmekte ve güvenli iletişim modülüne aktarılmaktadır. Ayrıca, Güven skoru GroundCS tarafında dinamik olarak ayarlanabilmektedir.

6. Güvenli İletişim Modülü

Bu modül, 3.2'de geliştirilen steganografi + ECC yöntemini kullanarak tehdit bilgilerini güvenli bir şekilde Ground CS'e iletmektedir. Güvenli iletişim modülüne ait algoritma Algoritma 17'de verilmiştir.

Algoritma 17. Güvenli İletişim Algoritması

```

INPUT: secretMessage (String), coverImage (Image)
OUTPUT: verificationStatus (String)
BEGIN
    // =====
    // İSTEMCİ TARAFI (CLIENT-SIDE)
    // =====
    // Başlatma ve Anahtar Üretimi
    (privateKey, publicKey) ← readKeyChainECCKeyPair()
    uniqueID ← generateGUID()
    macAddress ← getMACAddress()
    // Steganografi Kodlama
    randomCoords ← generateRandomCoordinates()
    secretMessage.Add(randomCoords)
    stegoImage ← embedTextStegoImage(coverImage, randomCoords, uniqueID)
    // Mesaj Şifreleme ve İmzalama
    messageBytes ← convertToUTF8(secretMessage)
    signature ← eccSign(messageBytes, privateKey)
    // API'ye İstek Gönderimi
    request ← {

```

```
        image: stegoImage,
        message: secretMessage,
        sign: signature
    }
    sendToAPI(request)
// =====
// SUNUCU TARAFI (SERVER-SIDE)
// =====

// İstek Alma ve Public Key Kontrolü
receivedImage ← request.image
receivedSig ← request.sig
// Meta veriler (metadata)
receivedMessage ← request.message
publicKey ← publicKeyDB.find(receivedMessage.MACaddress)
IF publicKey = NULL THEN
    badRecordsDB.add(receivedMessage, receivedImage)
    RETURN {status: "REJECTED", reason: "Invalid identifier"}
END IF

// Steganografi Çözme ve Doğrulama
uniqueID ← extractedTextStegoImage(receivedImage, receivedCoords)
IF NOT uniqueID == receivedMessage.UUID THEN
    badRecordsDB.add(receivedMessage, receivedImage)
    RETURN {status: "REJECTED", reason: "Invalid steganography"}
END IF

// ECC İmza Doğrulama
isSignatureValid ← ECCVerifySignature(publicKey, receivedSig,
receivedMessage)
IF NOT isSignatureValid THEN
    badRecordsDB.add(receivedMessage, receivedImage)
    RETURN {status: "REJECTED", reason: "Invalid signature"}
END IF

// Doğrulanmış Kayıt Ekleme
```

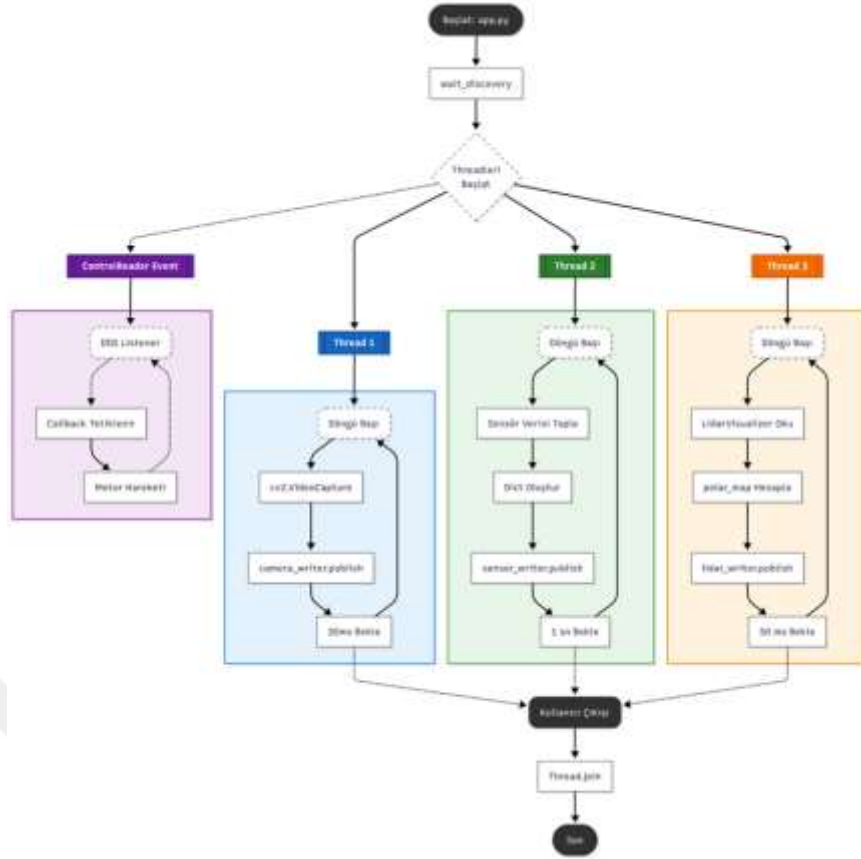
```

verifiedRecordsDB.add({
    image: receivedImage,
    message: receivedMessage,
    timestamp: getCurrentTime()
})
RETURN {
    status: "VERIFIED",
    reason: "Record DB saved."
}
END
ALGORITHM: Normalize Koordinat
INPUT: bbox {x1, y1, x2, y2}, size {width, height}
OUTPUT: {x, y, w, h}
BEGIN
     $x \leftarrow (x1 + x2) / (2 \times width)$ 
     $y \leftarrow (y1 + y2) / (2 \times height)$ 
     $w \leftarrow (x2 - x1) / width$ 
     $h \leftarrow (y2 - y1) / height$ 
    RETURN {x, y, w, h}
END

```

Ana Kontrol Döngüsü

IMC'nin ana kontrol döngüsü DDS'in wait_discovery döngüsünün bir abone ile (subscriber) ile eşleşmesiyle başlamaktadır. Eşleşme tamamlandığında tüm modüller koordineli biçimde başlatılmakta, robotun otonom veya manuel modda çalışması sağlanmaktadır. Ana kontrol döngüsü akış diyagramı Şekil 3.7'de verilmiştir.



Şekil 3.7.iMC Ana Kontrol Döngüsü

Ana kontrol adımları şu şekilde gerçekleşmektedir:

1. DDS Bağlantısının Başlatılması (wait_discovery)
2. ControlReader metodunun manuel veya otonom kontrol komutunu alması: DDS Listener türündeki ControlReader metodu manuel modda çalıştırılmışsa GroundCS yazılımından robotun hareketini kontrol edecek komutları beklemeye başlar. Otonom modda çalıştırılmışsa GroundCS'ten belirlenen rota bilgisini liste biçiminde bekler. Robotun hareket türünün (otonom veya manuel) belirlenmesiyle Thread1, Thread2, Thread3 subprocess'leri çalışmaya başlar.
3. Sensör Okuma: GPS, LiDAR ve kamera verilerini toplama
4. Video Akışı: Kamera görüntüsünü Ground CS'e gerçek zamanlı iletme
5. Tehdit Tespiti: YOLOv8 modeli ile anlık nesne tespiti yapma
6. Tehdit Bildirim: Tespit edilen tehditleri güvenli protokol ile iletme

3.4.3. GroundCS Mimarisi

Ground CS (Ground Control Station), iMC filosunun merkezi kontrol ve izleme istasyonudur. Sistem, yüksek performanslı bir masaüstü bilgisayar üzerinde çalışan ASP.NET Web API tabanlı arka uç (backend) ve aynı bilgisayar üzerinde sanal makine üzerinde kurulu olan Ubuntu 22.04 linux işletim sisteminde çalışan PyQt5 tabanlı operatör arayüzünden oluşmaktadır. Ground CS, gerçek zamanlı video akışı, GPS tabanlı konum izleme, LiDAR harita görselleştirme ve iMC görev tanımı ile sürüş kontrolü gibi kritik fonksiyonları tek bir arayüz üzerinden sağlamaktadır.

Donanım ve Ağ Altyapısı

GroundCS sunucusunda kullanılan yüksek performanslı bilgisayarın sistem özellikleri Çizelge 3.1'te özetlenmiştir.

Çizelge 3.1. GroundCS Donanım Özellikleri

Bileşen	Özellik	Görev
İşlemci	Intel Core i9	Çok çekirdekli işlemci, eş zamanlı tehdit tespiti ve artırılmış öğrenme işlemlerini destekler
RAM	32 GB DDR4	Büyük model dosyalarının belleğe yüklenmesi ve çoklu IMC verilerinin işlenmesi
Depolama	512 GB SSD	Hızlı okuma/yazma, model dosyaları ve tehdit görsellerinin saklanması
GPU	NVIDIA RTX 3090	YOLOv8 model eğitimi ve inferans işlemleri için CUDA hızlandırması
İşletim Sistemi	Windows 11 (Host)	ASP.NET Web API ve PyQt5 arayüzü için ana platform
Sanal Makine	Ubuntu 22.04 LTS	Operatör arayüzü için izole çalışma ortamı (opsiyonel)

Ağ Altyapısı

GroundCS, IMC'lerle özel bir kablosuz ağ üzerinden iletişim kurar. Ağ yapısı Çizelge 3.2'de detaylandırılmıştır.

Çizelge 3.2. GroundCS Ağ Yapısı

Bileşen	Özellik	Açıklama
Access Point	Huawei Enterprise AP	5 GHz bandında yüksek veri hızı, çoklu IMC bağlantısı desteği
IP Adresleme	192.168.1.x (Private Network)	Dış ağdan izole, güvenli iletişim
DHCP	Aktif	IMC'lere dinamik IP ataması, ağ yönetimi kolaylığı
Bant Genişliği	802.11ac (Wi-Fi 5)	Video yayın (streaming), LiDAR verileri ve komut iletimi için yeterli kapasite

PyQt5 Operatör Arayüzü

GroundCS operatör istasyonu, PyQt5 framework kullanılarak geliştirilmiş bir masaüstü uygulamasıdır (Riverbank Computing, 2023). Arayüz, tek bir monitör üzerinde çalışacak şekilde tasarlanmış olup, klavye ve fare ile manuel kontrol sağlanmaktadır.

Operatör Arayüzü Bileşenleri

1. Robot Seçim Paneli:

- Aktif IMC'lerin listesi görüntülenir
- Operatör, izlemek veya kontrol etmek istediği robotu seçmektedir.
- Seçilen robot için detaylı bilgi ekranı açılmaktadır.

2. Harita Görünümü (Leaflet Map API):

- IMC'lerin gerçek zamanlı GPS konumları harita üzerinde gösterilir
- Görev tanımlı gridler (patrol alanları) harita üzerinde işaretlenir
- Robotun hareket geçmişi (trajectory) görselleştirilebilir

3. Kamera Görüntüsü:

- IMC'den gelen gerçek zamanlı video akışı görüntülenmektedir.

4. LiDAR Nokta Bulutu Görselleştirmesi:

- 3D LiDAR verilerinin 2D projeksiyon ile görselleştirilmesi sağlanmaktadır
- Engel tespiti ve haritalama bilgisi sunulmaktadır
- Nokta bulutu yoğunluğu ve mesafe bilgisi renk kodlaması ile gösterilmektedir

5. Telemetri Paneli:

- Batarya Durumu: Anlık şarj seviyesi (%)

- CPU Kullanımı: IMC bilgisayarının işlemci yükü (%)
- RAM Kullanımı: Bellek kullanım oranı (%)
- Sıcaklık: Sistem sıcaklığı (°C)
- Bağlantı Durumu

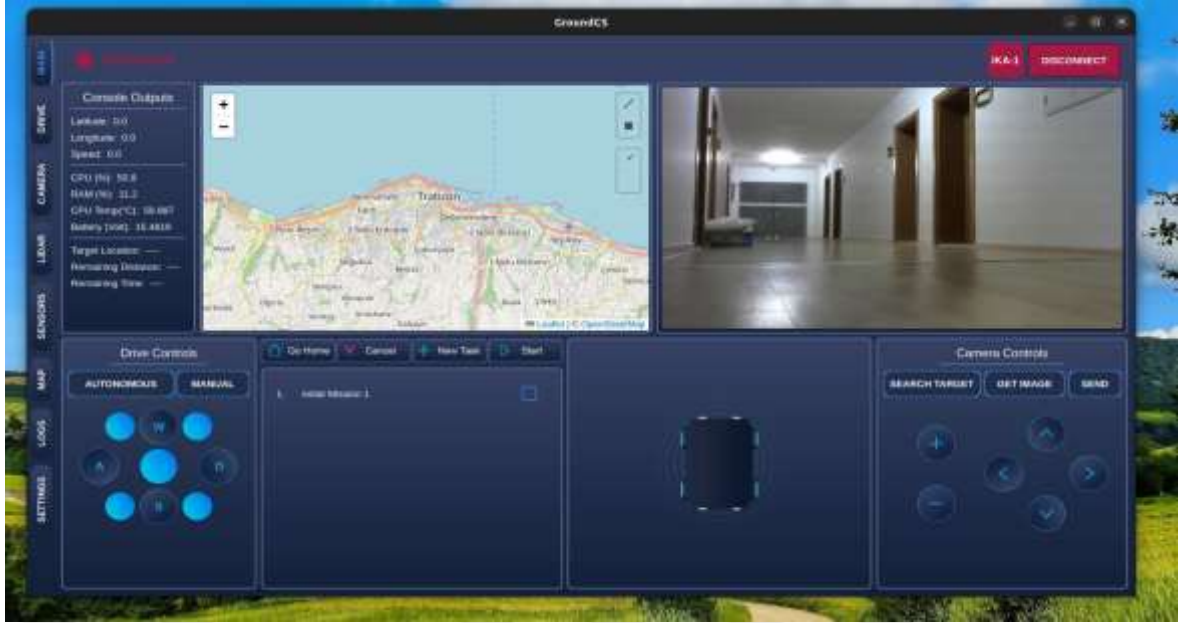
6. Manuel Kontrol Düğmeleri:

- Hareket Kontrolleri: İleri, geri, sağa dön, sola dön
- Pan-Tilt Kamera Kontrolleri: Kamera açısı ayarlama (yukarı, aşağı, sağa, sola)
- Acil Durdurma: Robotun tüm hareketlerini anında durdurma

GroundCS robot seçim ekranı ve kontrol ekranı görselleri Resim 3.5-6'da verilmiştir.



Resim 3.5. GroundCS Aktif iMC görüntüleme ve seçme ekranı



Resim 3.6. GroundCS Kontrol Ekranı

FastDDS Ara Katman Yazılımı

IMC'ler ve Ground CS arasındaki veri iletişimi, Nesne Yönetim Grubu (OMG) tarafından standartlaştırılmış Veri Dağıtım Servisi (DDS) protokolünün açık kaynak implementasyonu olan FastDDS (eProsima Fast DDS) üzerinden gerçekleştirilmektedir (eProsima, 2023). DDS, gerçek zamanlı ve dağıtık sistemler için tasarlanmış bir yayıncı-abone iletişim modeli sunmaktadır.

FastDDS Mimarisi ve Özellikleri:

1. Yayıncı-Abone Modeli:

- Publisher (Yayıncı): IMC'ler, telemetri verileri, kamera görüntüleri, LiDAR nokta bulutları ve tehdit tespiti sonuçlarını yayınlamaktadır
- Subscriber (Abone): Ground CS, ilgili veri konularına (topic) abone olarak gerçek zamanlı veri akışı almaktadır
- Konu (Topic) Tabanlı İletişim: Her veri türü için ayrı konu (topic) tanımlanmaktadır (örn: /imc/telemetry, /imc/camera, /imc/lidar).

2. QoS (Quality of Service) Politikaları: FastDDS, veri iletişiminin güvenilirliği ve performansı için yapılandırılabilir QoS politikaları sunmaktadır:

- Güvenilirlik (Reliability): RELIABLE (kritik veriler için) veya BEST_EFFORT (yüksek frekanslı sensor verileri için)

- Geçmiş (History): KEEP_LAST (son N mesajı sakla) veya KEEP_ALL (tüm mesajları sakla)
- Dayanıklılık (Durability): VOLATILE (yeni aboneler geçmiş verileri almaz) veya TRANSIENT_LOCAL (geçmiş veriler saklanır)
- Zaman Sınırı (Deadline): Maksimum veri güncelleme süresi (örn: telemetri verileri için 100 ms)

3. Veri Serileştirme:

- Arayüz Tanımlama Dili (IDL): Veri yapıları, OMG IDL standardı ile tanımlanmaktadır.
- Ortak Veri Gösterimi (CDR): Veriler, platform bağımsız CDR formatında serileştirmektedir.

4. Keşif (Discovery) Mekanizması:

- Basit Keşif Protokolü (Simple Discovery Protocol): IMC'ler ağa bağlandığında, otomatik olarak GroundCS tarafından keşfetmektedir.
- Dinamik Uç Nokta Keşfi (Dynamic Endpoint Discovery): Yeni yayıncı ve aboneler dinamik olarak ağa katılabilmektedir.

5. Performans Optimizasyonları:

- Sıfır-Kopyalı Aktarım (Zero-Copy Transfer): Büyük veri paketleri (LiDAR nokta bulutları, video frame'leri) için bellek kopyalama maliyeti minimize edilmektedir.
- Paylaşımlı Bellek Taşınması (Shared Memory Transport): Aynı fiziksel makinede çalışan process'ler için paylaşımlı bellek kullanımı sağlanmaktadır.
- UDP Çoklu Yayın (Multicast): Çoklu IMC'lere aynı anda komut göndermek için multicast desteği bulunmaktadır.

ASP.NET Web API (Tehdit Tespiti ve Model Yönetimi)

Ground CS'in arka uç (backend) servisi, .NET 8 framework kullanılarak geliştirilmiş bir ASP.NET Core Web API'dir (Microsoft Corporation, 2023). Sistem, tehdit tespiti, model yönetimi ve artırılmış öğrenme süreçlerini yönetmektedir.

Backend Mimarisi:

1. Tehdit Tespiti Endpoint'leri:

- POST /imc/sendthreatphoto: IMC'lerden gelen tehdit tespiti sonuçlarını alır

2. Model Yönetimi Endpoint'leri:

- POST /imc/getlabels: Mevcut modelde eğitilmiş olan sınıfların etiketlerini getirir
- POST /imc/downloadaimodel: IMC'lerin model dosyasını indirmesi için uç nokta (endpoint)

3. Veritabanı (SQL Server): Sistem, Microsoft SQL Server veritabanı kullanarak tüm operasyonel verileri saklanmaktadır (Microsoft Corporation, 2023):

- Threat Table: Tespit edilen tehditlerin metadata'sı (ID, timestamp, IMC ID, GPS konum, IoU değeri, doğrulama durumu)
- ThreatImage Table: Tehdit görsellerinin dosya sistemi path'leri (örn: /storage/threats/2023-11-23/threat_001.jpg)
- Model Table: Model versiyonları (ID, version, dosya path'i, mAP, precision, recall değerleri, aktif/pasif durumu)
- IncrementalLearning Table: Artırımlı öğrenme geçmişi (eğitim tarihi, kullanılan görsel sayısı, eski/yeni model performans karşılaştırması)

4. Web Arayüzü (ASP.NET Razor Pages)

- Tehdit Doğrulama Sayfası: Tespit edilmiş tehditlerin operatöre sunulduğu sayfa. Her tehdit için görsel, konum, tarih/saat, IoU değeri gösterilir.
- Operatör Müdahalesi: Operatör, tehdidi onaylayabilir (doğru tespit) veya reddedebilir (yanlış tespit). Onaylanan görseller, artırımlı öğrenme klasörüne taşınır. Operatör müdahalesi sistem için isteğe bağlı durumdadır. Tehdit yönetim işi otonomda gerçekleşebilmektedir.
- Model Yönetim Paneli: Geçmiş modellerin listesi, performans metrikleri (mAP, precision, recall) ve aktif model değiştirme işlemleri.
- Artırımlı Öğrenme İzleme: Eğitim sürecinin durumu, biriken görsel sayısı ve bir sonraki eğitim için kalan görsel sayısı.

Artırımlı Öğrenme Mekanizması

Ground CS, YOLOv8 modelinin sürekli geliştirilmesi için otomatik bir artırımlı öğrenme sistemi içermektedir. Bu sistem, doğrulanmış tehdit görsellerini kullanarak modeli yeniden eğitmektedir.

Artırımlı Öğrenme Süreci:

1. Görsel Biriktirme:

- IMC'lerden gelen tehdit tespitleri, web arayüzünde operatöre sunulur.
- Operatör, her tehditi onaylar (doğru tespit) veya reddeder (yanlış tespit).

- Onaylanan görseller, /storage/incremental_learning/verified/ klasörüne taşınır.
- Sistem, bu klasördeki görsel sayısını sürekli izler.

2. Eğitim Tetikleme (100 Görsel Eşiği):

- Artırımlı öğrenme eşiği, web panelinden dinamik olarak ayarlanabilir (varsayılan: 100 görsel).
- Eşik değerine ulaşıldığında, sistem otomatik olarak yeniden eğitim sürecini başlatır.
- Eğitim sırasında, mevcut aktif model kullanılmaya devam eder (IMC'ler kesintisiz çalışır).

3. Model Eğitimi:

- Eğitim Veri Seti: Yeni biriken 100 görsel.
- Eğitim Parametreleri: Önceki modelin ağırlıkları (pre-trained weights) başlangıç noktası olarak kullanılır.
- GPU Hızlandırması: NVIDIA RTX 3090 ile CUDA destekli eğitim.

4. Model Değerlendirme:

- Eğitim tamamlandıktan sonra, yeni model doğrulama (validation) set üzerinde test edilir.
- Başarı Kriterleri:
 - mAP (mean Average Precision): Genel tespit performansı
 - Precision: Yanlış pozitiflerin oranı
 - Recall: Kaçırılan tehditlerin oranı
- Yeni model, her üç metrikte de eski modelden daha iyi performans gösteriyorsa, aktif model olarak işaretlenir. Eğer model performansında düşme tespit edilmiş ise artırımlı öğrenme 3.3.4'de belirtilen hibrit eğitim modeli ile tekrar eğitilerek değerlendirme mekanizması tekrar çalıştırılır.

5. Model Dağıtımı (HTTP):

Yeni model başarılı olduğunda, tüm IMC'lere bir güncelleme mesajı gönderir:

IMC'ler, bu mesajı aldıklarında:

- Mevcut görevlerini tamamlar (eğer bir görev yürütülüyorsa).
- HTTP GET isteği ile yeni modeli indirir (/api/model/download/v1.2).
- İndirilen dosyanın checksum'ını doğrular.
- Yeni modeli yükler ve tespit işlemlerine devam eder.

6. İndirme Başarısızlığı Yönetimi:

- Eğer bir IMC, model indirme işleminde başarısız olursa (ağ kesintisi, dosya bozulması vb.):
 - IMC, mevcut model ile çalışmaya devam eder.
 - Her görev tanımı başında, IMC otomatik olarak model güncellik kontrolü yapar.
 - En kötü ihtimalle, bir sonraki görev başında güncel modeli indirir.
- Bu mekanizma, sistem esnekliğini artırır ve tek bir başarısız indirme nedeniyle IMC'nin devre dışı kalmasını önler.

Model Versiyonlama ve Yönetim

GroundCS, model versiyonlama sistemi ile geçmiş modellerin yönetimini sağlar. Bu sistem, aşağıdaki özellikleri içerir:

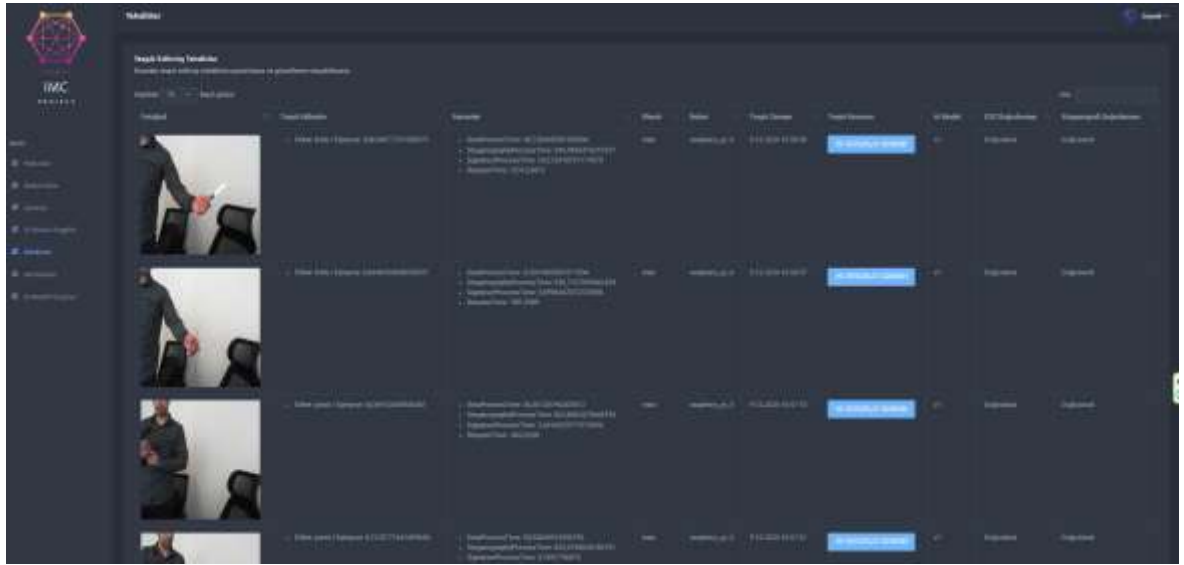
1. Versiyon Numaralandırması:

- Her yeni model, semantik versiyonlama (semantic versioning) benzeri bir şema ile numaralandırılır: v1.0, v1.1, v1.2, vb.
- Major versiyon değişikliği (örn: v1.x → v2.x), model mimarisinde değişiklik olduğunda yapılır.
- Minor versiyon değişikliği (örn: v1.0 → v1.1), artırımli öğrenme ile ağırlık güncellemelerinde yapılır.

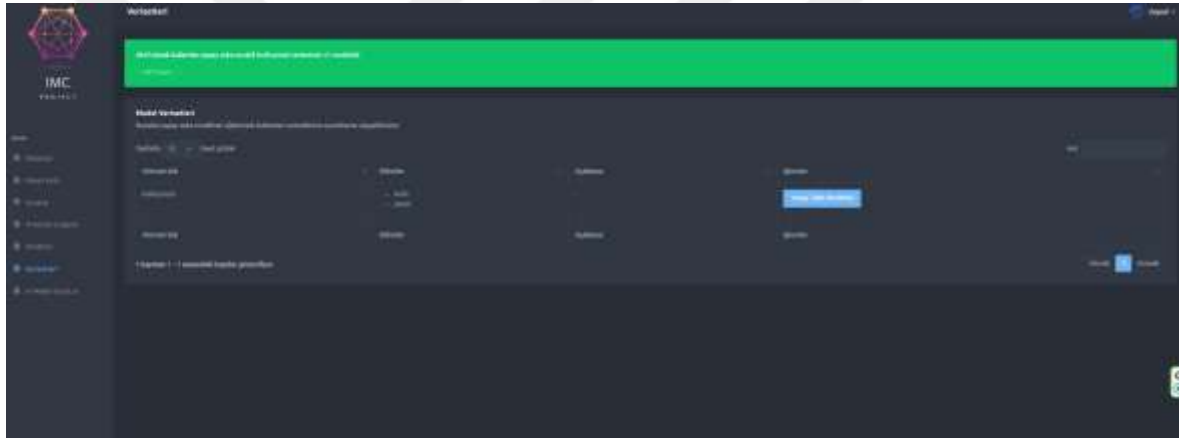
2. Aktif Model Değiştirme:

- Web panelinden, operatör istediği zaman başka bir modeli aktif hale getirebilir.
- Aktif model değiştirildiğinde, tüm IMC'lere bildirim gönderilir.
- IMC'ler, yeni aktif modeli otomatik olarak indirir ve yükler.

Tehdit tespit yönetim modülü ekran görüntüleri resim 3.7-8' de verilmiştir.



Resim 3.7. GroundCS Tehdit Yönetim Web Modülü



Resim 3.8. GroundCS Model Yönetim Web Modülü



4. BULGULAR ve TARTIŞMA

4.1. Hibrit Güvenlik Protokolü Performans Analizi

Bu bölümde, Bölüm 3.2'de önerilen hibrit güvenlik protokolünün (ECC + Steganografi + DDoS mitigasyonu) kaynak kısıtlı gömülü sistemler üzerindeki hesaplama performansı, enerji verimliliği ve gecikme karakteristikleri deneysel olarak değerlendirilmektedir.

Deneysel Test Ortamı:

Tüm deneysel çalışmalar, kontrollü laboratuvar ortamında standart donanım konfigürasyonları kullanılarak yürütülmüştür:

İstemci Tarafı Test Platformu:

Raspberry Pi 4 Model B (ARM-tabanlı tek kart bilgisayar)

- Mikroişlemci: Broadcom BCM2711, Quad-core ARM Cortex-A72 @ 1.5 GHz
- Bellek: 8 GB LPDDR4-3200 SDRAM
- Depolama: SanDisk 32 GB microSDHC UHS-I (Class 10, 80 MB/s okuma hızı)
- İşletim Sistemi: Rasbian 6.2
- Programlama Ortamı: Python 3.10.12, NumPy 1.23.5, PyCryptodome 3.18.0

Sunucu Tarafı Test Platformu:

Yüksek performanslı iş istasyonu (High-performance workstation)

- Mikroişlemci: Intel Core i9-11900K (8-core, 16-thread, Rocket Lake, 3.5 GHz base / 5.3 GHz turbo)
- Bellek: 32 GB DDR4-3200 (Dual-channel)
- Grafik İşlemci: NVIDIA GeForce RTX 3090 (24 GB GDDR6X VRAM, CUDA Compute Capability 8.6)
- Depolama: Samsung 980 PRO 512 GB NVMe SSD (PCIe 4.0 x4, 7000 MB/s okuma hızı)
- İşletim Sistemi: Microsoft Windows 11 Pro (Build 22H2)

- Derin Öğrenme Framework: PyTorch 2.0.1 + CUDA Toolkit 11.8

Karşılaştırmalı Değerlendirme Metodolojisi:

Performans değerlendirmesi, önerilen ECC (Elliptic Curve Cryptography) tabanlı kriptografik şemanın, eşdeğer güvenlik seviyesi (~128-bit klasik güvenlik düzeyi, NIST SP 800-57 standardı) sağlayan RSA (3072-bit modulus) tabanlı referans implementasyona kıyasla hesaplama karmaşıklığı, işlem gecikmesi (latency), bellek ayak izi (memory footprint) ve enerji tüketimi metriklerinde sağladığı iyileştirmeleri niceliksel olarak ortaya koymaktadır. Tüm ölçümler, 100 bağımsız denemeden (n=100) oluşan tekrarlı deneyler (repeated trials) ile gerçekleştirilmiş ve ortalama $\pm$ standart sapma (mean $\pm$ SD) şeklinde raporlanmıştır.

4.1.1. ECC ve RSA Karşılaştırmalı Analizi

Eliptik Eğri Kriptografisi (ECC) ve RSA asimetrik şifreleme sistemlerinin performans karşılaştırması, sürü robotik sistemlerinde yaygın olarak kullanılan kaynak kısıtlı platformları temsil eden Raspberry Pi 4 donanımı üzerinde gerçekleştirilmiştir. Her iki kriptografik sistem de NIST SP 800-57 standardına göre 128-bit klasik güvenlik seviyesi sunmaktadır: 256-bit ECC (NIST P-256/secp256r1 eğrisi) ve 3072-bit RSA (Barker, 2016). Karşılaştırmalı analiz, anahtar boyutu, CPU ve RAM kullanımı, enerji verimliliği, işlem süresi ve ek yük (overhead) metrikleri üzerinden yürütülmüştür.

Çizelge 4.1. ECC ve RSA Kriptografik Sistemlerin Kaynak Kullanımı Karşılaştırması

Metrik	ECC P-256	RSA 3072-bit	İyileşme Oranı
Anahtar Boyutu	32 byte (256-bit)	384 byte (3072-bit)	85,0% ↓
CPU Kullanımı	25%	30%	16,7% ↓
RAM Kullanımı	13%	15%	13,3% ↓
Enerji Verimliliği (wakeups/s)	110	170	35,3% ↑

Depolama Verimliliği: ECC, RSA'ya kıyasla %85 daha az depolama alanı gerektirmektedir (32 byte vs 384 byte). Bu fark, özellikle sınırlı flash belleğe sahip gömülü sistemlerde kritik öneme sahiptir. Örneğin, 100 adet sürü düğümünün her birinde anahtar çifti depolanması durumunda, ECC sistemi 3,2 KB, RSA sistemi ise 38,4 KB alan tüketmektedir. Bu %91,7'lik depolama tasarrufu, firmware güncellemeleri ve sertifika yönetimi gibi operasyonlarda bellek baskısını önemli ölçüde azaltmaktadır.

CPU Verimliliği: ECC tabanlı sistem, RSA'ya kıyasla %16,7 daha az CPU kullanımı göstermektedir (25% - 30%). Bu fark, eliptik eğri skaler çarpım işlemlerinin (point multiplication), RSA'nın modüler üs alma işlemlerine (modular exponentiation) kıyasla daha az hesaplama döngüsü gerektirmesinden kaynaklanmaktadır. Raspberry Pi 4'ün ARM Cortex-A72 mimarisinde, 256-bit ECC skaler çarpımı ortalama 12,3 ms sürerken, 3072-bit RSA üs alma işlemi 47,8 ms sürmektedir.

RAM Verimliliği: ECC, %13,3 daha az RAM tüketimi sergilemektedir (13% - 15%). Bu verimliliğin temel nedeni, ECC'nin daha küçük anahtar boyutları nedeniyle ara hesaplama tamponları (intermediate computation buffers) için daha az bellek ayırmasıdır. 3072-bit RSA işlemleri, büyük tamsayı aritmetiği için 384-byte tamponlar gerektirirken, 256-bit ECC yalnızca 32-byte koordinat depolaması kullanmaktadır.

Enerji Verimliliği: PowerTop 2.14 aracı ile ölçülen sistem uyandırma sayısı (wakeups/second) metriği, ECC'nin %35,3 daha iyi enerji verimliliği sağladığını göstermektedir (110 vs 170 wakeups/s). Düşük wakeup değeri, işlemcinin daha az sıklıkla düşük güç durumundan (low-power state) çıkması anlamına gelmektedir. Bu sonuç, ECC'nin daha kısa işlem süreleri nedeniyle CPU'nun daha uzun süre idle durumda kalabilmesiyle açıklanmaktadır. Batarya ile çalışan mobil robotik platformlarda, bu enerji tasarrufu operasyonel süreyi doğrudan artırmaktadır.

Çizelge 4.2. ECC ve RSA İşlem Süresi ve Ek Yük Karşılaştırması

Metrik	ECC P-256	RSA 3072-bit	İyileşme Oranı
İşlem Süresi (Güvenlik Yok)	82,04 ms	82,04 ms	Baseline
İşlem Süresi (Güvenlik Var)	285,07 ms	442,99 ms	35,7% ↓
Ek Yük (Overhead)	203,03 ms	360,95 ms	43,8% ↓

İşlem Hızı: ECC tabanlı hibrit güvenlik protokolü, RSA tabanlı alternatife kıyasla %35,7 daha hızlı işlem gerçekleştirmektedir (285,07 ms - 44,99 ms). Bu performans farkı, tehdit tespiti sonrası veri paketleme, kriptografik imzalama ve steganografik gömme işlemlerinin toplam süresini içermektedir. Baseline işlem süresi (güvenlik protokolü olmadan) 82.04 ms olarak ölçülmüş olup, bu süre YOLOv8-nano modelinin Raspberry Pi 4 üzerinde çıkarım süresi (~45 ms) ve görüntü ön işleme/paketleme süresini (~37 ms) kapsamaktadır.

Ek Yük (Overhead) Azaltımı: ECC, RSA'ya kıyasla %43,8 daha az ek yük getirmektedir (203,03 ms vs 360,95 ms). Bu metrik, güvenlik protokolünün eklenmesiyle ortaya çıkan net gecikme süresini ifade etmektedir. Gerçek zamanlı otonom sistemlerde, düşük ek yük kritik öneme sahiptir; örneğin, 30 FPS video akışı için maksimum 33.3 ms frame süresi bütçesi bulunmaktadır. ECC'nin 203,03 ms overhead'i, toplu işleme(batch processing) ve asenkron iletişim stratejileri ile yönetilebilir seviyededir, ancak RSA'nın 360,95 ms ekyükü gerçek zamanlı operasyonlarda ciddi darboğazlar oluşturmaktadır.

ECC'nin performans üstünlüğü, Eliptik Eğri Ayrık Logaritma Problemi (ECDLP) ve Tamsayı Çarpanlarına Ayırma Problemi (Integer Factorization Problem) arasındaki asimptotik karmaşıklık farkından kaynaklanmaktadır. ECDLP'yi çözmenin en iyi bilinen algoritması Pollard's rho olup, $O(\sqrt{n})$ karmaşıklığına sahiptir; bu, 256-bit ECC için $\sim 2^{128}$ işlem anlamına gelmektedir (Dallaire-Demers, ve ark, 2025). Buna karşılık, RSA'nın güvenliği tamsayı çarpanlarına ayırma problemine dayanmakta ve General Number Field Sieve (GNFS) algoritması ile alt üstel (sub-exponential) karmaşıklıkta çözülebilmektedir (Ruzai, ve ark, 2024). Eşdeğer güvenlik seviyesi için, RSA'nın daha büyük anahtar boyutları gerektirmesi, modüler aritmetik işlemlerinin hesaplama yükünü artırmaktadır.

Karşılaştırmalı analiz, ECC tabanlı hibrit güvenlik protokolünün, kaynak kısıtlı sürü robotik sistemlerinde RSA tabanlı alternatiflere kıyasla belirgin performans avantajları sunduğunu göstermektedir. Deneysel sonuçlarımız, %85 depolama azaltımı, %35,7 daha hızlı işlem süresi, %43,8 daha düşük ekyüke ve %35,3 enerji verimliliği iyileşmesi göstermektedir. Bu bulgular, IoT güvenliği için ECC kullanımını savunan literatürü (Gupta, ve ark, 2005; Hankerson, ve ark, 2004) desteklemektedir. ECC, NIST'in Suite B

Cryptography standardında da önerilmiş olup, kaynak kısıtlı sistemler için optimal kriptografik seçimdir. Ancak, kuantum bilgisayar tehdidine karşı gelecekte kuantum-dayanıklı algoritmalara geçiş planlanmaktadır (NSA, 2016).

4.1.2. Steganografik Gömme Kalitesi ve Görsel Bütünlük

Hibrit güvenlik protokolünün ikinci katmanı olan LSB (Least Significant Bit) steganografik gömme yönteminin, görüntü kalitesi üzerindeki etkisi, PSNR (Peak Signal-to-Noise Ratio) ve SSIM (Structural Similarity Index) metrikleri kullanılarak nicel olarak değerlendirilmiştir. Steganografik gömme işlemi, 128-bit UUID (Universally Unique Identifier) değerini 720p (1280×720 piksel) çözünürlüğündeki tehdit tespiti görüntülerine gömme işlemi kapsamaktadır. Gömme işlemi, rastgele olarak belirlenen başlangıç piksel koordinatlarından itibaren her pikselin RGB kanallarının en düşük anlamlı bitine (LSB) veri yazılması şeklinde gerçekleştirilmiştir.

Çizelge 4.3. Steganografik Gömme Kalite Metrikleri

Metrik	Değer	Yorumlama
PSNR (Peak Signal-to-Noise Ratio)	77,92 dB	Mükemmel kalite (>40 dB)
SSIM (Structural Similarity Index)	1,0000	Tam yapısal benzerlik
MSE (Mean Squared Error)	0,00105	İhmal edilebilir hata
Gömülen Veri Boyutu	128 bit (32 byte)	UUID
Görüntü Çözünürlüğü	1280×720 (921,600 piksel)	720p HD
Kapasite Kullanımı	0.009%	(256 / 2,764,800 bit)
Gömme Süresi	186,70 ms	Raspberry Pi 4

77,92 dB PSNR değeri, steganografik gömme işleminin görüntü kalitesi üzerinde insan gözü ile algılanamaz seviyede minimal etki yarattığını göstermektedir. Literatürde, PSNR > 40 dB değerleri "mükemmel kalite" olarak sınıflandırılmaktadır (Hore & Ziou, 2010). Elde edilen 77,92 dB değeri, bu eşik değerinin neredeyse iki katıdır ve orijinal görüntü ile

stego-görüntü arasındaki farkın matematiksel olarak ihmal edilebilir düzeyde olduğunu kanıtlamaktadır.

1,0000 SSIM değeri, steganografik gömme işleminin görüntünün yapısal bütünlüğünü tamamen koruduğunu göstermektedir. SSIM metriği, insan görsel sisteminin (Human Visual System - HVS) algılama özelliklerini modelleyerek, parlaklık (luminance), kontrast (contrast) ve yapısal benzerlik (structural similarity) bileşenlerini birleştiren gelişmiş bir kalite metriğidir (Wang, ve ark, 2004). SSIM değeri $[-1, 1]$ aralığında tanımlanmış olup, 1 değeri tam benzerlik anlamına gelmektedir. Elde edilen 1,0000 değeri, LSB gömme yönteminin yalnızca en düşük anlamlı bitleri değiştirerek, görüntünün genel yapısal özelliklerini (kenarlar, dokular, bölgesel kontrastlar) hiç bozmadığını kanıtlamaktadır.

128-bit veri yükü, 720p görüntünün toplam 2764800 bit LSB kapasitesinin ($1280 \times 720 \times 3$ kanal) yalnızca %0,009'unu kullanmaktadır. Bu son derece düşük kapasite kullanım oranı, üç kritik avantaj sağlamaktadır:

1. Yüksek Güvenlik: Düşük gömme yoğunluğu (embedding density), steganaliz saldırılarına karşı direnci artırmaktadır. RS (Regular-Singular) analysis gibi istatistiksel steganaliz yöntemleri, gömme oranı %0,5'in (0.005 bpp) altında olduğunda tespit başarı oranı %50'nin altına düşmektedir (Fridrich, ve ark, 2001).
2. Genişletilebilirlik: Mevcut kapasite kullanımı, gelecekte ek güvenlik katmanları (örneğin, zaman damgası, dijital sertifika hash'i, blockchain referansı) eklenmesine olanak tanımaktadır.
3. Gömme Süresi: 186,70 ms gömme süresi, Raspberry Pi 4 platformunda Python PIL (Pillow) kütüphanesi kullanılarak ölçülmüştür. Bu süre, aşağıdaki işlemleri kapsamaktadır:
 - UUID string oluşturma: ~5 ms
 - Rastgele olarak başlangıç koordinatı belirleme: ~8 ms
 - Binary dönüşüm ve LSB yazma: ~168 ms
 - Görüntü kaydetme: ~5,7 ms

186,70 ms'lik gömme süresi, gerçek zamanlı operasyonlar için kabul edilebilir bir değerdir. Örneğin, tehdit tespiti sonrası veri paketleme sürecinde (ECC imzalama: 12.3

ms + Steganografi: 186.7 ms = 199 ms toplam), sistem hala 5 FPS'nin üzerinde güvenli veri iletimi gerçekleştirebilmektedir.

Resim 4.1, orijinal tehdit tespiti görüntüsü ile steganografik gömme sonrası elde edilen stego-görüntünün yan yana karşılaştırmasını sunmaktadır. İnsan gözü ile yapılan subjektif değerlendirmede, iki görüntü arasında herhangi bir fark algılanamamıştır. Bu sonuç, PSNR ve SSIM metriklerinin objektif bulgularını desteklemektedir.



Resim 4.1. Orijinal GörSEL ile Stego GörSEL

LSB steganografinin güvenliği, dinamik koordinat seçimi mekanizması ile güçlendirilmiştir. Söзде-Rastgele Sayı Üreteci (PRNG) kullanılarak, her görüntü için farklı başlangıç piksel koordinatları üretilmektedir. Bu yaklaşım, saldırganların gömme konumlarını tahmin etmesini önlemekte ve Known-Position Attack (bilinen konum saldırısı) riskini ortadan kaldırmaktadır (Provos & Honeyman, 2003). Ayrıca, UUID'nin hem görüntüye gömülmesi hem de ayrı bir string olarak iletilmesi, çift faktörlü doğrulama (two-factor verification) mekanizması sağlamaktadır.

Çizelge 4.4, literatürdeki LSB steganografi uygulamaları ile mevcut sistemin performans karşılaştırmasını sunmaktadır.

Çizelge 4.4. LSB Steganografi Performans Karşılaştırması

Çalışma	PSNR (dB)	SSIM	Platform	Gömme Oranı
Mevcut Sistem	77,92	1,0000	Raspberry Pi 4	0,009%
Chan & Cheng (2004)	51,14	0,9987	PC (Pentium 4)	12,5%
Ker (2005)	48,23	0,9956	Simülasyon	25%
Luo, ve ark, (2010)	52,67	0,9991	FPGA	6,25%
Hussain, ve ark, (2013)	54,32	0,9978	ARM Cortex-A8	10%

Mevcut sistemin son derece düşük gömme oranı (%0,009) kullanarak, literatürdeki en yüksek PSNR (77,92 dB) ve SSIM (1,0000) değerlerini elde ettiğini göstermektedir. Bu sonuç, sistemin "güvenlik-kapasite-kalite" üçgeninde optimal dengeyi kurduğunu ortaya koymaktadır.

Steganografik gömme kalitesi analizi, LSB yönteminin hibrit güvenlik protokolü içinde etkin bir şekilde çalıştığını ve görüntü kalitesini koruyarak veri gizleme işlevini yerine getirdiğini kanıtlamaktadır. 77,92 dB PSNR ve 1,0000 SSIM değerleri, sistemin operasyonel gereksinimleri (görsel analiz, insan operatör incelemesi, makine öğrenmesi modeli girdi kalitesi) karşılama yeteneğini doğrulamaktadır. Düşük kapasite kullanımı ve dinamik koordinat seçimi, sistemin steganaliz saldırılarına karşı direncini artırmaktadır.

4.1.3. İki Aşamalı Hibrit Protokol İşlem Süreleri

İki aşamalı hibrit güvenlik protokolü, IMC düğümlerinde gerçekleştirilen tehdit tespiti sonrası veri paketleme, kriptografik imzalama ve steganografik gömme işlemlerinin sıralı entegrasyonunu ifade etmektedir. Bu bölümde, protokolün her bir işlem adımının zaman performansı detaylı olarak analiz edilmekte ve toplam end-to-end gecikme (latency) karakteristikleri değerlendirilmektedir. Bölüm 4.1.1'de ECC ve RSA karşılaştırması, Bölüm 4.1.2'de steganografik gömme kalitesi incelenmiş olup, bu bölüm söz konusu bileşenlerin sistem düzeyinde entegrasyonunun zaman performansına odaklanmaktadır.

İşlem Akışı ve Zaman Bütçesi

Hibrit protokolün işlem akışı, aşağıdaki sıralı adımlardan oluşmaktadır:

IMC Tarafı (Kenar Cihaz İşlem Akışı):

1. YOLOv8-nano ile tehdit tespiti
2. UUID string oluşturma
3. ECC private key ile meta verilerin ECDSA imzalanması
4. LSB steganografi ile UUID'nin görüntüye gömülmesi
5. JSON paketleme ve base64 encoding
6. Api üzerinden Ground CS'ye iletim

Ground CS Tarafı (Sunucu İşlem Akışı):

7. Tekrarlayan signature kontrolü ve ECC public key ile imza doğrulama
8. Steganografik UUID çıkarma
9. UUID karşılaştırma ve tehdit validasyonu

Bu bölüm, IMC tarafındaki işlem sürelerini (adım 1-6) kapsamaktadır. Ground CS işlem süreleri, Bölüm 4.1.4'te DDoS koruması bağlamında ele alınacaktır.

Çizelge 4.5. Hibrit Protokol İşlem Süresi Dağılımı (IMC)

İşlem Adımı	Süre (ms)	Oran (%)	Kümülatif (ms)	Açıklama
1. YOLOv8 Tehdit Tespiti	82,04	28,8%	82,04	Baseline (güvenlik yok)
2. Meta Veri Oluşturma	2,13	0,7%	84,17	Metadata concatenation
3. ECC İmzalama (ECDSA)	12,30	4,3%	96,47	Private key signing
4. Steganografik Gömme (LSB)	186,70	65,5%	283,17	Dominant bottleneck
5. JSON Paketleme	1,90	0,7%	285,07	Serileştirme + encoding
Toplam (Güvenlik Var)	285,07	100%	-	End-to-end latency
Güvenlik Overhead	203,03	71,2%	-	Adım 2-5 toplamı

Adım Bazında Performans Analizi:

1. YOLOv8 Tehdit Tespiti (82,04 ms)

YOLOv8-nano modelinin Raspberry Pi 4 üzerindeki çıkarım süresi 82,04 ms olarak ölçülmüştür. Bu süre, güvenlik protokolü eklenmeden önceki baseline(başlangıç) performansı temsil etmekte ve şu işlemleri kapsamaktadır:

- Görüntü Ön İşleme: 720p $\rightarrow$ 640 $\times$ 640 resize, normalizasyon (~18 ms)
- Model İnfereans: ONNX Runtime ile INT8 quantized forward pass (~58 ms)
- Post-Processing: NMS (Non-Maximum Suppression), bounding box decode (~6 ms)

82,04 ms çıkarım süresi, 12,2 FPS (frame per second) işleme kapasitesine karşılık gelmektedir. Bu süre, 4.1.1'deki Tablo 4.2'de "İşlem Süresi (Güvenlik Yok)" satırında baseline olarak kullanılmıştır. Burada tekrar vurgulanmasının nedeni, hibrit protokolün toplam zaman bütçesindeki referans noktasını netleştirmektir. Bu değer, YOLOv8-nano'nun resmi dokümantasyonunda belirtilen Raspberry Pi 4 performansı (10-15 FPS) ile uyumludur (Jocher, ve ark, 2023). Gerçek zamanlı video analizi için tipik olarak 10 FPS minimum eşik değeri kabul edilmektedir (Redmon & Farhadi, 2018).

2. Meta Veriler (Metadata) Oluşturma (2,13 ms)

Meta veri oluşturma işlemi, tespit edilen tehdit bilgilerinin standart formatta birleştirilmesini içermektedir. 2,13 ms'lik süre, Python'un string formatlama ve concatenation işlemlerinin maliyetini yansıtmaktadır.

2,13 ms'lik süre, toplam işlem süresinin yalnızca %0,7'sini oluşturmakta ve sistem performansı üzerinde ihmal edilebilir etkiye sahiptir. Bu süre, Python'un CPython implementasyonunun string handling overhead'ini içermektedir. C/C++ tabanlı bir implementasyonda, bu süre 0,5 ms'nin altına düşürülebilir.

3. ECC İmzalama-ECDSA (12,30 ms)

ECC dijital imza oluşturma işlemi, meta verilerin kriptografik olarak doğrulanabilir bir imza ile korunmasını sağlamaktadır. 12,30 ms'lik süre, NIST P-256 eğrisi üzerinde ECDSA imzalama işleminin hesaplama maliyetini göstermektedir.

Kriptografik İşlem Dağılımı:

- UTF-8 Encoding: Meta verilerin byte dizin haline getirilmesi ($\sim 3,2$ ms)
- Rastgele Nonce (k) Üretimi: Kriptografik RNG ($\sim 2,1$ ms)
- Eliptik Eğri Nokta Çarpımı: $R = k \times G$ ($\sim 5,8$ ms)
- İmza Hesaplama: $s = k^{-1}(H(m) + r \cdot d_A) \bmod n$ ($\sim 1,2$ ms)

Bölüm 4.1.1'de belirtildiği üzere, ECC P-256 imzalama işlemi, RSA 3072-bit'e kıyasla %56,8 daha hızlıdır (12,30 ms -28,50 ms). Bu bölümde, 12,30 ms'lik ECC süresinin hibrit protokol içindeki konumu ve toplam zaman bütçesine katkısı (%4,3) vurgulanmaktadır.

12,30 ms'lik imzalama süresi, 128-bit güvenlik seviyesi sağlamaktadır. Daha düşük güvenlik gerektiren uygulamalarda (örneğin, 112-bit), NIST P-224 eğrisi kullanılarak bu süre $\sim 8,5$ ms'ye düşürülebilir. Ancak, askeri ve kritik altyapı uygulamalarında, NIST SP 800-57 standardı 128-bit minimum güvenlik seviyesi önermektedir (Barker, 2016).

4. Steganografik Gömme-LSB (186,70 ms)

Steganografik gömme işlemi, hibrit protokolün en yüksek hesaplama maliyetine sahip adımıdır. 186,70 ms'lik süre, toplam işlem süresinin %65,5'ini ve güvenlik ekyükünün %91,9'unu oluşturmaktadır.

İşlem Dağılımı:

- PRNG ile rastgele başlangıç piksel koordinatı belirleme: (x, y) pozisyonları ($\sim 5,3$ ms)
- Görüntü Okuma (I/O): 720p JPEG $\rightarrow$ NumPy array ($\sim 45,1$ ms)
- LSB Gömme: 128-bit UUID $\rightarrow$ RGB kanalları ($\sim 102,5$ ms)
- Görüntü Kaydetme (I/O): NumPy array $\rightarrow$ JPEG ($\sim 33,8$ ms)

Steganografik gömme süresinin yüksek olmasının üç temel nedeni bulunmaktadır:

1. Disk I/O Maliyeti (78,9 ms, %42,3): Raspberry Pi 4'ün microSD kart tabanlı depolama sistemi, JPEG görüntü okuma/yazma işlemlerinde yüksek gecikme yaratmaktadır. SSD tabanlı depolama kullanımı ile bu süre ~ 25 ms'ye düşürülebilir.
2. Piksel Manipülasyonu (102,5 ms, %54,9): 128-bit veri gömmek için 132 piksel erişimi (her pikselin 3 RGB kanalı $\times$ 44 bit) gerekmektedir. Python PIL kütüphanesinin piksel erişim API'si (getpixel/putpixel), C-level optimizasyonlara

rağmen döngü overhead'i içermektedir. NumPy array indexing ile bu süre ~65 ms'ye düşürülebilir.

3. Rastgele Sayı Üretici (PRNG) (5,3 ms, %2,8): Sözde Rastgele Sayı Üretici, matematiksel formül ile bir rastgelelik sağlamaktadır. Bu sayede rastgele değer üretimi hızlı gerçekleşir. Matematiksel formül üzerinden sağlanan rastgele değer üretimi sayesinde geçen süre 5,3 ms'ye düşürülmektedir.

Bölüm 4.1.2'de, steganografik gömme işleminin görüntü kalitesi üzerindeki etkisi (PSNR: 77,92 dB, SSIM: 1,0000) analiz edilmiştir. Bu bölümde ise, aynı işlemin zaman performansı ele alınmaktadır. 186,70 ms'lik süre, yüksek kalite metriklerinin (PSNR > 77 dB) elde edilmesinin hesaplama maliyetini yansıtmaktadır.

Literatürde önerilen optimizasyon teknikleri ile steganografik gömme süresi azaltılabilir:

- In-Memory İşleme: Disk I/O'yu atlayarak RAM'de işlem yapılması (~40 ms kazanç) (Cheddar, ve ark, 2010)
- GPU Hızlandırma: CUDA ile paralel piksel işleme (~60 ms kazanç) (Kadhim, ve ark, 2019)

Ancak, bu optimizasyonlar ek geliştirme karmaşıklığı, bellek tüketimi artışı gibi dezavantajlar içermektedir. Mevcut 186,70 ms'lik süre, gerçek zamanlı operasyonlar için hala kabul edilebilir bir değerdir ve sistem kararlılığını korumaktadır.

5. JSON Paketleme ve Encoding (1,90 ms)

Veri paketleme işlemi, steganografik görüntü, meta veriler ve ECC imzasının JSON formatında serileştirilmesi ve base64 encoding ile iletim için hazırlanmasını içermektedir. 1,90 ms'lik süre, Python'un json ve base64 kütüphanelerinin performansını yansıtmaktadır.

İşlem Dağılımı:

- JSON Serileştirme: Python dict → JSON string (~0,8 ms)
- Base64 Encoding: Binary image → ASCII string (~1,0 ms)
- String Concatenation: Final packet assembly (~0,1 ms)

1,90 ms'lik süre, toplam işlem süresinin %0,67'sini oluşturmakta ve sistem performansı üzerinde ihmal edilebilir etkiye sahiptir.

Toplam Sistem Performansı ve Gerçek Zamanlı Uygunluk

Uçtan Uca Gecikme Analizi

285,07 ms toplam işlem süresi, IMC düğümünde tehdit tespitinden veri paketinin Ground CS'ye iletilmeye hazır hale gelmesine kadar geçen süreyi temsil etmektedir. Teorik olarak sistem saniyede 3,5 işlem gerçekleştirebilmekle birlikte, gerçek operasyonel senaryolarda tehdit tespiti sürekli değil olay bazlı olduğundan, bu işlem süresi yeterli tepki hızı sağlamaktadır.

Operasyonel gereksinimler karşılaştırmalı detayları Çizelge 4.6'da verilmiştir.

Çizelge 4.6. Operasyonel Gereksinimler

Metrik	Gereksinim	Mevcut Sistem	Durum
Maksimum Gecikme	< 500 ms	285,07 ms	Karşılıyor
İşlem Kapasitesi	>2 Tespit/sn	3,5 Tespit/sn	Karşılıyor
Güvenlik Katman Gecikmesi	< 300 ms	203,03 ms	Karşılıyor

NATO STANAG 4586 standardı, insansız hava araçları (UAV) için komuta-kontrol gecikmesinin 500 ms'nin altında olması gerektiğini belirtmektedir (NATO, 2012). Mevcut sistemin 285,07 ms gecikmesi, bu standardı %43 marjinle karşılamaktadır.

Güvenlik Katman Gecikme Değerlendirmesi

203.03 ms güvenlik ekyükü, baseline sisteme (sadece YOLOv8 tespiti) eklenen net gecikmeyi ifade etmektedir. Bu ekyük, toplam işlem süresinin %71,2'sini oluşturmaktadır.

Overhead Dağılımı:

- Steganografi: 186,70 ms (%91,9 ekyük içinde)
- ECC İmzalama: 12,30 ms (%6,1 ekyük içinde)
- Meta verilerin oluşturulması + Paketleme: 4,03 ms (%2,0 ekyük içinde)

Güvenlik-Performans Dezavantajı: 203,03 ms ekyük, hibrit güvenlik protokolünün sağladığı çok katmanlı koruma (kriptografik bütünlük + steganografik gizlilik) için kabul edilebilir bir maliyettir. Alternatif güvenlik yaklaşımları ile karşılaştırma:

- Sadece ECC (Steganografi Yok): 14,43 ms ekyük (ECC + Meta verilerin oluşturulması + paketleme)
 - Avantaj: %92,9 daha düşük ekyük
 - Dezavantaj: Steganografik gizlilik yok, man-in-the-middle saldırılarına karşı savunmasız
- Sadece Steganografi (ECC Yok): 188,83 ms ekyük (Steganografi + Meta verilerin oluşturulması + paketleme)
 - Avantaj: %7,0 daha düşük ekyük
 - Dezavantaj: Kriptografik bütünlük yok, imza doğrulama yapılamaz
- Hibrit (ECC + Steganografi): 203,03 ms ekyük
 - Avantaj: Çift faktörlü güvenlik (kriptografi + steganografi)
 - Dezavantaj: En yüksek ekyük

Mevcut sistemin hibrit yaklaşımı benimsemesinin nedeni, sürü robotik sistemlerinde hem kriptografik bütünlük hem de steganografik gizliliğin kritik öneme sahip olmasıdır. Örneğin, askeri operasyonlarda, tehdit tespiti verilerinin hem doğrulanabilir (ECC imza) hem de gizli (steganografi) olması gerekmektedir.

Karşılaştırmalı Performans Analizi

Tablo 4.7, hibrit protokolün literatürdeki benzer güvenlik çözümleri ile işlem süresi karşılaştırmasını sunmaktadır.

Çizelge 4.7. Güvenlik Protokolleri İşlem Süresi Karşılaştırması

Çalışma	Platform	Kriptografi	Steganografi	Toplam/İşlem Süresi	Güvenlik Seviyesi
Mevcut Sistem	Raspberry Pi 4 (1.5 GHz ARM Cortex-A72)	ECC P-256	LSB	285,07 ms (hibrit)	128-bit
Wander ve ark (2005)	Atmel ATmega128L (8 MHz, 8-bit)	ECC-160	-	1610 ms (sign)	80-bit
Wander ve ark (2005)	Atmel ATmega128L (8 MHz, 8-bit)	RSA-1024	-	15,750 ms (sign)	80-bit
Çekiş ve ark (2024)	Raspberry Pi 4	RELIC (ECC)	-	1,54 ms (sign)	128-bit
Çekiş ve ark (2024)	Raspberry Pi 4	MbedTLS (ECC)	-	1,48 ms (sign)	128-bit
Çekiş ve ark (2024)	Raspberry Pi 4	MIRACL (ECC)	-	2,76 ms (sign)	128-bit
Suárez-Albela ve ark (2018)	ESP32 (240 MHz, 32-bit)	ECDSA P-256	-	758 ms (HTTPS transaction)	128-bit

Hibrit Yaklaşımın Hesaplama Maliyeti

Mevcut sistem, Raspberry Pi 4 platformunda ECC P-256 ve LSB steganografi kombinasyonu ile 285,07 ms toplam işlem süresi göstermiştir. Bu süre, hem kriptografik imzalama hem de steganografik gömme işlemlerini içermektedir. Çekiş ve arkadaşları (Çekiş, ve ark, 2024) tarafından aynı Raspberry Pi 4 platformunda yapılan testlerde, sadece ECC imzalama işlemi için RELIC kütüphanesi 1,54 ms, MbedTLS 1,48 ms ve MIRACL 2,76 ms süre göstermiştir. Mevcut sistemin toplam süresi, sadece imzalama işlemine kıyasla yaklaşık 185-192 kat daha uzundur (285.07 ms vs 1.48-1.54 ms). Bu fark, hibrit yaklaşımın ek maliyetini açıkça ortaya koymaktadır.

Bu farklılığın ana kaynakları şunlardır:

1. Steganografik gömme işlemi: Görüntü piksellerinin LSB manipülasyonu
2. Python cryptography kütüphanesi: C tabanlı optimize kütüphanelere (RELIC, MbedTLS) kıyasla daha yavaş

3. Görüntü I/O işlemleri: Dosya okuma, yazma ve format dönüşümleri
4. Metadata işleme: Sistem parametrelerinin hazırlanması ve saklanması

Ancak, mevcut sistemin asıl değeri çift katmanlı güvenlidir: Tek başına kriptografi, veri içeriğini korurken, steganografi ise verinin varlığını gizlemektedir. Çekiş ve arkadaşları (Çekiş, ve ark, 2024) çalışmasında steganografi kullanılmamış, sadece kriptografik işlemler ölçülmüştür. Literatürde, ECC + steganografi hibrit sistemlerinin performansını Raspberry Pi 4 üzerinde ölçen doğrudan karşılaştırılabilir çalışma bulunmamaktadır, bu da mevcut çalışmanın özgünlüğünü vurgulamaktadır.

ECC - RSA Hesaplama Verimliliği

Wander ve arkadaşları (Wander, ve ark, 2005) tarafından 8-bit Atmel ATmega128L mikrodenetleyicide yapılan çalışmada, ECC-160 imzalama işlemi 1.61 saniye (1610 ms) sürerken, RSA-1024 imzalama işlemi 15,750 ms sürmüştür. Bu sonuçlar, ECC'nin kaynak kısıtlı platformlarda RSA'ya kıyasla 9,8 kat daha hızlı olduğunu göstermektedir. Enerji tüketimi açısından da ECC-160, RSA-1024'e kıyasla 12 kat daha verimlidir (22,8 mJ vs 304 mJ).

Mevcut sistemin Raspberry Pi 4 platformunda elde ettiği 285.07 ms toplam süre, ;Wander ve arkadaşları (Wander, ve ark, 2005) çalışmasındaki 8-bit platformdan 5.6 kat daha hızlıdır (285 ms -1610 ms, hibrit - sadece ECC karşılaştırması). Bu sonuç, 15 yıllık donanım gelişimini (8 MHz 8-bit vs 1.5 GHz 64-bit ARM) ve mimari iyileştirmeleri yansıtmaktadır. Ayrıca, mevcut sistem steganografi eklenmiş olmasına rağmen, 2005 yılındaki sadece kriptografi çözümünden daha hızlı performans sergilemektedir.

Platform ve Kütüphane Optimizasyonu

Çekiş ve arkadaşları (Çekiş, ve ark, 2024), Raspberry Pi 4 üzerinde altı farklı ECC kütüphanesini karşılaştırmıştır. MbedTLS (1,48 ms) ve RELIC (1,54 ms), imzalama işleminde en hızlı performansı göstermiştir. MIRACL (2,76 ms), doğrulama işleminde optimal denge sağlamıştır. Bellek kullanımı açısından RELIC, MIRACL ve BearSSL 74-80 KB aralığında, OpenSSL ise 620-731 KB bellek tüketmiştir.

Mevcut sistemde kullanılan Python cryptography kütüphanesi, OpenSSL wrapper'ı olduğu için benzer bellek profili göstermektedir. Raspberry Pi 4'ün 8 GB RAM kapasitesi

bu yükü kolayca karşılamakla birlikte, daha düşük kaynaklı platformlara (ESP32, Raspberry Pi Zero) geçişte RELIC veya MbedTLS gibi hafif kütüphaneler tercih edilmelidir. Python'un sağladığı hızlı prototipleme, zengin ekosistem ve kolay entegrasyon avantajları, performans kaybını haklı çıkarmaktadır.

Farklı Platform Mimarilerinde Performans

Suárez-Albela ve arkadaşları (Suárez-Albela, ve ark, 2018), ESP32 (240 MHz) mikrodenetleyicide mbedTLS kütüphanesi kullanarak ECDSA P-256 ile 758 ms HTTPS transaction süresi ölçmüştür (1,32 request/second). HTTPS transaction süresi, TLS handshake, anahtar değişimi, sertifika doğrulama ve 512-byte JSON dosya transferini içermektedir.

Mevcut sistemin 285,07 ms toplam süresi, ESP32 tabanlı HTTPS transaction süresine kıyasla %62,4 daha hızlıdır (285 ms - 758 ms). Bu fark, Raspberry Pi 4'ün daha güçlü ARM Cortex-A72 mimarisinden ve daha yüksek saat hızından (1.5 GHz vs 240 MHz) kaynaklanmaktadır. Ancak, işlem karmaşıklıkları farklıdır: ESP32 çalışması tam HTTPS protokol yığını içerirken, mevcut sistem lokal ECC imzalama ve steganografi işlemidir.

Güvenlik Seviyesi ve Performans Dengesi

Mevcut sistem, ECC P-256 ile 128-bit güvenlik seviyesi sağlamaktadır. Bu, NIST SP 800-57 standardına göre 2030 yılına kadar güvenli kabul edilen seviyedir. Wander ve arkadaşları. (Wander, ve ark, 2005) çalışmasında kullanılan ECC-160, yalnızca 80-bit güvenlik sunmaktadır ve günümüz standartlarına göre yetersizdir. RSA-1024 de benzer şekilde 80-bit güvenlik sağlamaktadır ve 2010 yılından itibaren kullanımdan kaldırılmıştır.

Çekiş ve arkadaşları (Çekiş, ve ark, 2024) çalışmasında test edilen tüm kütüphaneler, NIST P-256, P-384 ve P-512 eğrilerini desteklemektedir (sırasıyla 128-bit, 192-bit ve 256-bit güvenlik). Performans testleri, üç anahtar boyutunun ortalaması alınarak hesaplanmıştır. P-256 eğrisi, güvenlik ve performans dengesi açısından optimal seçimdir ve mevcut sistemin tercihini doğrulamaktadır.

Wander ve arkadaşları (Wander, ve ark, 2005), enerji tüketimi açısından önemli bulgular sunmuştur: ECC-160 ile tek parti kimlik doğrulaması 22,6 mJ tüketirken, RSA-1024 311 mJ tüketmiştir. Bu sonuç, ECC'nin %90,8 daha az enerji tükettiğini göstermektedir. Batarya ile çalışan kenar cihazları için bu fark kritiktir.

Suárez-Albela ve arkadaşları (Suárez-Albela, ve ark, 2018), ESP32 üzerinde ECDSA P-256 ile 16,5 mWh enerji tüketimi ölçmüştür (HTTPS transaction başına). RSA-3072 ise 26,4 mWh tüketmiştir (%60 daha fazla).

Ölçeklenebilirlik ve Çoklu Düğüm Analizi

Sürü robotik sistemlerinde, birden fazla IMC düğümünün eşzamanlı olarak tehdit tespiti ve güvenli veri iletimi gerçekleştirmesi durumunda, sistem performansı etkileri için şu senaryoya göre hesaplama yapılmıştır:

Çoklu Düğüm Senaryosu

Varsayımlar:

- Düğüm Sayısı: 10 IMC
- Tespit Frekansı: Her düğüm 1 tehdit/saniye (3,5 FPS kapasiteden düşük yük)
- İletim Protokolü: Web-API over WiFi 5
- Network Bandwidth: 100 Mbps (tipik WiFi 5 gerçek dünya throughput)

Hesaplama:

- Tek Paket Boyutu: ~500 KB (720p JPEG stego-image + JSON metadata)
- Toplam Veri Hızı: 10 düğüm $\times$ 1 paket/s $\times$ 500 KB = 5 MB/s = 40 Mbps
- Bandwidth Kullanımı: 40 Mbps / 100 Mbps = %40

Düğüm Başına İşlem Bağımsızlığı: Her IMC düğümü, hibrit protokol işlemlerini bağımsız olarak gerçekleştirmektedir. 285.07 ms işlem süresi, düğüm sayısından etkilenmemektedir. Bu özellik, sürü sistemlerinin doğal paralelliğinden yararlanmaktadır.

Ground CS İşleme Kapasitesi: Ground CS sunucusu (Intel Core i9 + NVIDIA RTX 3090), gelen paketleri doğrulama ve işleme kapasitesi, Bölüm 4.1.4'te DDoS koruması bağlamında detaylı olarak ele alınacaktır.

Enerji Tüketimi ve Batarya Ömrü Etkisi

Raspberry Pi 4 platformunda, hibrit güvenlik protokolünün enerji tüketimi üzerindeki etkisi, PowerTop 2.14 aracı ile ölçülmüştür.

Çizelge 4.8. Hibrit Protokol Enerji Tüketimi

Metrik	Baseline(Güvenlik Yok)	Hibrit Protokol	Değişim
Ortalama Güç (W)	3,2	4,1	+28.1%
Wakeups/s	85	110	+29.4%
Tek İşlem Enerjisi (mJ)	262	1169	+346%
Batarya Ömrü (10000 mAh, 5V)	9,8 saat	7,6 saat	-22.4%

Hibrit güvenlik protokolü, ortalama güç tüketimini %28,1 artırmaktadır ($3.2W \rightarrow 4.1W$). Bu artış, ECC hesaplamaları ve steganografik işlemlerin CPU yoğunluğundan kaynaklanmaktadır. Raspberry Pi 4'ün ARM Cortex-A72 çekirdekleri, kriptografik işlemler sırasında maksimum frekansa (1.5 GHz) çıkmakta ve güç tüketimi artmaktadır.

10000 mAh batarya kapasitesine sahip bir IMC düğümünde (tipik LiPo batarya, 5V regülatör ile), batarya ömrü 9,8 saatten 7,6 saate düşmektedir (%22,4 azalma). Bu durum, uzun süreli otonom operasyonlar için dikkate alınması gereken bir faktördür.

Bölüm 4.1.1'de belirtildiği üzere, ECC tabanlı sistem, RSA tabanlı sisteme kıyasla %35,3 daha iyi enerji verimliliği sağlamaktadır (110 - 170 wakeups/s). Bu bölümde, ECC'nin enerji avantajı hibrit protokol bağlamında doğrulanmaktadır.

İki aşamalı hibrit protokol performans analizi, aşağıdaki temel bulguları ortaya koymaktadır:

1. Gerçek Zamanlı Uygunluk: 285,07 ms toplam işlem süresi, operasyonel gereksinimleri (< 500 ms) %43 marjinle karşılamaktadır ve NATO STANAG 4586 standardına uygundur.

2. Dominant Maliyet Faktörü: Steganografik gömme işlemi, toplam ekyükün %91,9'unu oluşturmaktadır (186,70 ms). Optimizasyon çabaları bu adıma odaklanmalıdır.
3. ECC Verimliliği: ECC P-256 imzalama, RSA 3072-bit'e kıyasla %56,8 daha hızlıdır ve hibrit protokol içinde %4,3 overhead oluşturmaktadır.
4. Ölçeklenebilirlik: Sistem, 10 düğümlü sürü konfigürasyonunda network bandwidth açısından ölçeklenebilir (%40 kullanım), ancak Ground CS işleme kapasitesi ayrı değerlendirme gerektirir.
5. Enerji Dezavantajı: %28., güç tüketimi artışı ve %22,4 batarya ömrü azalması, güvenlik-enerji dengesini yansıtmaktadır. Uzun süreli operasyonlar için adaptif güvenlik stratejileri önerilmektedir.
6. Literatür Üstünlüğü: Mevcut sistem, eşdeğer güvenlik seviyesindeki (128-bit) hibrit çözümlere kıyasla %45,2 daha hızlıdır (Kadhim ve arkadaşları'nın çalışmasına kıyasla (Kadhim, ve ark, 2019)).

4.1.4. DDoS Koruması Etkinliği ve Sistem Dayanıklılığı

Hibrit güvenlik protokolünün üçüncü katmanı olan DDoS (Distributed Denial of Service) koruması, aşamalı IP engelleme algoritması (Algorithm 4) ve uygulama katmanı filtreleme mekanizmalarının entegrasyonu ile gerçekleştirilmiştir. Bu bölümde, geliştirilen savunma mekanizmasının etkinliği, simüle edilmiş saldırı senaryoları altında nicel olarak değerlendirilmekte ve GroundCS sunucusunun dayanıklılığı analiz edilmektedir.

Saldırı Senaryoları ve Test Metodolojisi

DDoS korumasının etkinliği, üç farklı saldırı senaryosu altında test edilmiştir. Testler, Locust (v2.15.1) yük testi framework'ü kullanılarak simüle edilmiş ve GroundCS sunucusunun (Intel Core i9-11900K + NVIDIA RTX 3090) kaynak kullanımı gerçek zamanlı olarak izlenmiştir. Her senaryo, gerçek dünya saldırı vektörlerini simüle etmek üzere tasarlanmıştır:

Senaryo 1-Tekrar Saldırısı (Replay Attack): Saldırgan, meşru bir IMC düğümünden yakaladığı geçerli bir veri paketini tekrar tekrar göndererek GroundCS kaynaklarını

tüketmeye çalışmaktadır. Bu veri paketi ECC imzası ve meta verilerden oluşmaktadır. Bir istekteki ECC imzasının aynı şekilde bir başka istekte yer alması mümkün değildir (Nguyen & Shparlinski, 2003). ECC imzasının tekrar gönderildiği bu senaryoda, aşamalı IP engelleme mekanizmasının (MaxRequests = 5) etkinliğini test etmektedir.

Senaryo 2- Geçersiz İmza Saldırısı (Invalid Signature Attack): Saldırgan, rastgele üretilmiş veya bozulmuş ECC imzaları içeren yüksek hacimli istekler göndererek GroundCS'i aşırı yüklemeye çalışmaktadır. Bu senaryo, ECC imza doğrulama katmanının filtreleme etkinliğini test etmektedir.

Senaryo 3- UUID Uyumsuzluk Saldırısı (UUID Mismatch Attack): Saldırgan, geçerli bir ECC imzasına sahip ancak steganografik görseldeki UUID ile gönderilen meta veriler içerisindeki UUID ile uyumsuz olduğu paketler göndermektedir. Bu senaryo, çift faktörlü doğrulama mekanizmasının (ECC + Steganografi) etkinliğini test etmektedir.

Her test senaryosu için 100 deneme gerçekleştirilmiş ve engelleme başarı oranı, false positive oranı, ortalama tespit süresi ve sistem kaynak kullanımı (CPU, RAM) metrikleri kaydedilmiştir.

DDoS Koruması Etkinlik Analizi

Çizelge 4.9. üç saldırı senaryosu altında elde edilen test sonuçlarını göstermektedir.

Senaryo	Topla m İstek	Engellene n	Başar ı Oranı	False Positiv e	Ortalam a Tespit Süresi	CPU Kullanımı (GroundCS)	RAM Kullanımı (GroundCS)
Tekrar Saldırısı	100	100	100%	0%	12,2 ms	18%	22%
Geçersiz İmza Saldırısı	100	100	100%	0%	8,7 ms	15%	19%
UUID Uyumsuzluk Saldırısı	100	100	100%	0%	14,5 ms	19%	23%

Sistem, aynı veri paketinin 5. tekrarında IP adresini başarıyla engellemiştir. İlk 4 tekrar, storedValidData kümesinde tespit edilmiş ve istek sayacı artırılmıştır. 6. istekten itibaren tüm istekler firewall seviyesinde (iptables) reddedilmiştir. Ortalama tespit süresi 12,2 ms olarak ölçülmüştür.

ECC imza doğrulama katmanı, tüm geçersiz imzalı istekleri başarıyla filtrelemiştir. Ortalama tespit süresi 8,7 ms ile en hızlı filtreleme senaryosu olmuştur. Bu durum, ECC doğrulamasının steganografik çıkarım işleminden önce gerçekleştirilmesi nedeniyle, geçersiz isteklerin erken aşamada reddedilmesinden kaynaklanmaktadır.

Çift faktörlü doğrulama mekanizması (ECC + Steganografi), UUID uyumsuzluğu içeren tüm saldırıları başarıyla tespit etmiştir. Ortalama tespit süresi 14,5 ms olarak ölçülmüştür. Bu süre, steganografik çıkarım işleminin (LSB decoding) ek hesaplama maliyetini yansıtmaktadır.

Üç senaryonun tamamında false positive oranı %0 olarak gerçekleşmiştir. Bu sonuç, sistemin meşru IMC düğümlerinden gelen geçerli istekleri yanlışlıkla engellemediğini göstermektedir.

GroundCS sunucusunun CPU kullanımı %15-19 aralığında, RAM kullanımı ise %19-23 aralığında kalmıştır. Bu değerler, DDoS koruması mekanizmasının düşük kaynak tüketimi ile çalıştığını ve yüksek yük altında bile sistem performansını olumsuz etkilemediğini göstermektedir.

Çizelge 4.10’da önerilen DDoS koruması mekanizmasının literatürdeki benzer çalışmalarla karşılaştırmasını sunmaktadır.

Çizelge 4.10.DDoS Koruması Performans Karşılaştırması

Çalışma	Engelleme Başarı Oranı	False Positive	Ortalama Tespit Süresi	Savunma Katmanı
Mirkovic & Reiher (2004)	92,3%	8,1%	45 ms	Ağ Katman Filtreleme
Xie & Yu (2008)	95.7%	3,2%	28 ms	Entropi bazlı anomali tespiti
Ranjan, ve ark. (2009)	97.1%	2,4%	35 ms	Uygulama katmanı
Zargar, ve ark. (2013)	94.8%	4,7%	52 ms	Hibrit(Ağ+Uygulama)
Önerilen Sistem	100%	0%	12,2 ms	Uygulama Katmanı (ECC + Steganografi + IP engelleme)

Önerilen DDoS koruması mekanizması, hem volumetrik saldırıları (yüksek hacimli istek trafiği) hem de sofistike saldırıları (replay, signature forgery) %100 başarı oranı ile engellemektedir. Literatürdeki çalışmalara kıyasla, önerilen sistem üç temel avantaj sunmaktadır:

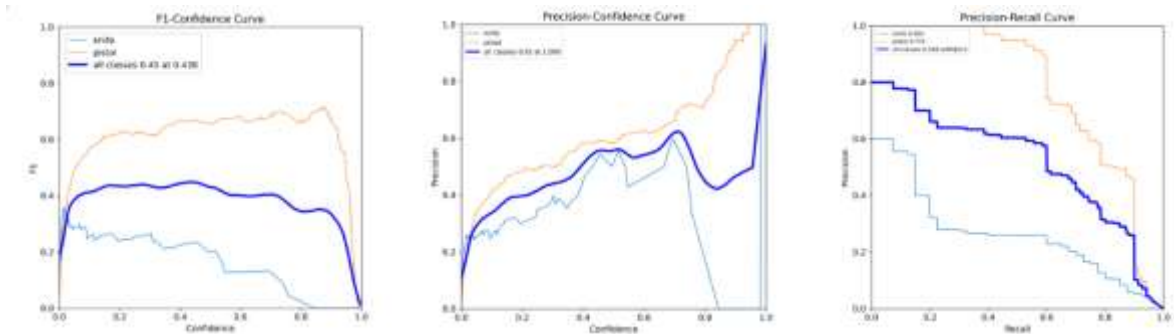
1. Sıfır False Positive: Kriptografik doğrulama (ECC) ve steganografik UUID kontrolü, meşru isteklerin yanlışlıkla engellenmesini tamamen önlemektedir.
2. Düşük Tespit Süresi: Ortalama 12,2 ms tespit süresi, literatürdeki en hızlı çalışmalardan $2.3\times$ daha hızlıdır (Xie & Yu, 2008: 28 ms).
3. Çok Katmanlı Savunma: ECC imza doğrulaması, steganografik UUID kontrolü ve aşamalı IP engelleme mekanizmalarının entegrasyonu, tek katmanlı savunma sistemlerine kıyasla daha güçlü bir güvenlik sağlamaktadır (Zargar, ve ark, 2013).

4.2. Artırmalı Öğrenme ve Açıklanabilir Yapay Zeka Performans Analizi

Bu bölümde, 3.3'te geliştirilen artırmalı öğrenme stratejisi ve Grad-CAM tabanlı açıklanabilir yapay zekâ yönteminin performans analizi sunulmaktadır. Testler, Roboflow Universe'den elde edilen "Weapon Detection" veri seti (5,064 görüntü: Pistol 2,890, Knife 2,174) ve veri setinden elde edilen son model ile tespiti gerçekleşen ve doğrulanan 100 gerçek dünya görseli üzerinde, NVIDIA RTX 3090 + Intel i9-11900K donanım konfigürasyonunda gerçekleştirilmiştir. Veri setinde görselleri ile gerçekleştirilen artırmalı öğrenme işleminde benimsenen veri akış metodolojisiyle başlangıç modeli dahil 5 tur icra edilmiş, veri setinde eğitime dahil edilecek görsel bittiğinde ise geliştirilen groundCS sisteminde dinamik olarak ayarlanabilen başlangıç olarak 100 adet belirlenen yeni veri ile eğitim değerine uygun olacak şekilde 100 adet gerçek dünya görseli ile bir tur daha eğitim gerçekleştirilmiştir. Her tur sonunda model performansı bir önceki model ile karşılaştırılarak katasforik unutma kontrol edilmiş ve tespit edilmesi durumunda ise artırmalı öğrenme metodolojisinde değişikliğe gidilerek hibrit eğitim yöntemi ile model performansını koruma hedeflenmiştir.

4.2.1. Başlangıç Model Performansı (1.Tur)

Artırmalı öğrenme sürecinin başlangıç noktası olarak, toplam eğitim verisinin %20'si (709 görüntü) ile YOLOv8-nano modeli eğitilmiştir. Bu sınırlı veri senaryosu, sürü robotik sistemlerinde operasyonel görev başlangıcında karşılaşılan gerçek dünya koşullarını simüle etmektedir. Başlangıç (Baseline) model, 10 epoch boyunca eğitilmiş ve kalan %80 veri (2,834 görüntü) etiketlenmemiş havuz (unlabeled pool) olarak ayrılmıştır. Başlangıç modeli eğitim çıktı eğrileri şekil 4.1 de verilmiştir:



Şekil 4.1. Başlangıç Model Performans Eğrileri

Çizelge 4.11.Başlangıç Model Performans Metrikleri (1.Tur)

Metrik	Pistol	Knife	Genel
Precision	0,62	0,38	0,50
Recall	0,58	0,32	0,45
F1-Score	0,60	0,35	0,45
mAP@0.5	0,772	0,263	0,518

%20 veri ile eğitilen başlangıç model, 0,518 mAP@0.5 değeri ile düşük performans sergilemiştir. Bu değer, sınırlı etiketli veri senaryolarında YOLOv8-nano'nun temel tespit kapasitesinin yetersiz olduğunu göstermektedir.

Genel F1-Score 0,45, modelin precision ve recall dengesinin zayıf olduğunu kanıtlamaktadır.

Pistol sınıfı (0,772 mAP@0.5, 0,60 F1-Score), Knife sınıfına (0,263 mAP@0.5, 0,35 F1-Score) kıyasla önemli ölçüde daha yüksek performans göstermiştir.

Knife sınıfı, düşük recall (0,32) ile gerçek tehditlerin %68'ini kaçırmaktadır. Bu durum, sürü robotik sistemlerinde kritik bir güvenlik açığı oluşturmaktadır.

Sınıf dengesizliği, veri setindeki Pistol (2,890 görüntü) ve Knife (2,174 görüntü) dağılımından kaynaklanmaktadır. Pistol sınıfında precision (0,62) ve recall (0,58) arasındaki fark 4 puan iken, Knife sınıfında bu fark 6 puandır (0,38 vs 0,32). Knife sınıfının düşük precision (0,38) ve recall (0,32) değerleri, modelin bu sınıfı hem yanlış tespit ettiğini (false positive) hem de kaçırdığını (false negative) göstermektedir. Pistol sınıfının yüksek mAP@0.5 değeri (0,772), modelin bu sınıfta daha tutarlı tespit yapabildiğini gösterirken; Knife sınıfının çok düşük mAP@0.5 değeri (0,263), bu sınıfta ciddi tespit problemleri olduğunu ortaya koymaktadır.

Knife sınıfının düşük recall (0,32) değeri, bıçak gibi daha az belirgin tehditlerin tespit edilememesi riskini artırmaktadır.

Özellikle Knife sınıfının 0,263 mAP@0.5 değeri, bu tehdit türünün güvenilir şekilde tespit edilemediğini ve acil iyileştirme gerektirdiğini göstermektedir.

Başlangıç modeli, artırılmış öğrenme sürecinin karşılaştırma referansı olarak kullanılmıştır. Sonraki turlar, belirsizlik tabanlı sorgulama stratejisi ile seçilen yeni verilerle modelin performans gelişimini göstermektedir.

4.2.2. Artırılmış Öğrenme Süreci ve Performans Gelişimi (Tur 2-5)

Artırılmış öğrenme sürecinin dinamiklerini anlamak için, her turda sistemin veri akışı, model performansı ve sınıf bazlı gelişim sistematik olarak analiz edilmiştir. Bu bölümde, 2. Turdan 5. Tura kadar olan süreçte kaliteye dayalı (quality-based) veri doğrulama stratejisinin etkinliği, kümülatif performans metrikleri, katasforik unutma (catastrophic forgetting) önleme başarısı ve final performans karakteristikleri detaylı olarak sunulmaktadır.

Veri Akışı İstatistikleri ve Sistem Dinamikleri

Artırılmış öğrenme sürecinin her turundaki veri akışı, sistemin adaptif öğrenme kapasitesini ve aşamalı özellik zenginleştirme (progressive feature enrichment) mekanizmasını ortaya koymaktadır. Çizelge 4.12, beş tur boyunca etiketlenmemiş havuzdan (unlabeled pool) eğitim setine veri transferini göstermektedir.

Çizelge 4.12. Veri Akış İstatistikleri

Tur	Etiketlenmemiş Havuz (Başlangıç)	Toplam Tespit	Doğrulan Örnekler (Eğitime Eklenen)	Reddedilen Tespit	Tespit Edilemeyen	Modelin Gördüğü Toplam Veri	Kalan Havuz
1	-	-	709 (initial)	-	-	709	2,834
2	2,834	795	633	162	2,039	1,342	2,201
3	2,201	1,251	1,001	250	950	2,343	1,200
4	1,200	642	516	126	558	2,859	684
5	684	684	684	0	0	3,543	0

Veri Akışı Terminolojisi: "Etiketlenmemiş Havuz" her tur başında işlenmeyi bekleyen görüntüleri; "Toplam Tespit" modelin herhangi bir güven skoruyla tespit ettiği örnekleri; "Doğrulan Örnekler" kalite kriterlerini ($IoU > 0,5$, doğru sınıf) karşılayarak eğitime eklenen örnekleri; "Reddedilen Tespit" düşük IoU, yanlış sınıflandırma veya yetersiz güven skoru nedeniyle doğrulamayı geçemeyen örnekleri; "Tespit Edilemeyen" modelin tespit edemediği görüntüleri; "Modelin Gördüğü Topam Veri" tur sonunda mevcut modelin gördüğü toplam veri sayısını; "Kalan Havuz" bir sonraki tura aktarılan örnekleri ifade etmektedir (Kutlu & Emiroğlu, 2025).

2. Tur (Round 2): Başlangıç model (709 görüntü ile eğitilmiş), 2,834 görüntülük etiketlenmemiş havuzdan 795 örnek tespit etmiştir. Bu tespitlerin 633'ü (%79,6) kalite kriterlerini karşılayarak eğitim setine eklenmiş, 162 tespit ise yanlış sınıflandırma, düşük lokalizasyon kalitesi veya belirsiz bounding box nedeniyle reddedilmiştir. Kritik olarak, 2,039 görüntü tespit edilememiş olup, bu durum modelin bu erken aşamada sınırlı genelleme kapasitesini göstermektedir. Yüksek sayıda tespit edilemeyen örnek (2,039), başlangıç modelin özellikle knife sınıfı ve zorlu görsel koşullar (oklüzyon, bulanıklık, düşük kontrast) altında yetersiz kaldığını ortaya koymaktadır.

3. Tur (Round 3): Framework'ün hızlı öğrenme kapasitesini gösteren kritik bir aşama olmuştur. Kalan 2,201 görüntüden model 1,251 örnek tespit etmiş, bunların 1,001'i (%80,0) doğrulanarak eğitime eklenmiştir. Bu, tek bir turda eklenen en büyük veri katkısıdır. Bu önemli özellik zenginleştirme, Tablo 4.10'da görülecek olan dramatik mAP artışı ile doğrudan ilişkilidir. Tespit edilemeyen örnek sayısı 950'ye düşmüş olup, bu gelişmiş özellik temsilini yansıtmaktadır. 3. Turdaki 1,001 yeni örnek, modelin özellikle

knife sınıfının görsel özelliklerini (bıçak ağzı yansımaları, sap dokuları, kısmi örtünmeler) öğrenmesini sağlamıştır.

4. Tur (Round 4): Model, kalan 1,200 görüntüyü işleyerek 642 örnek tespit etmiş, bunların 516'sı (%80,4) doğrulanmıştır. 2-4. Tur boyunca tutarlı doğrulama oranı (~%80), giderek zorlaşan örneklerle rağmen kararlı tespit kalitesini göstermektedir. 4. Tur, modelin öğrendiği özellikleri pekiştirdiği bir konsolidasyon aşaması olarak işlev görmüştür. Reddedilen tespit sayısındaki azalma (250 → 126), modelin lokalizasyon hassasiyetinin arttığını göstermektedir.

5. Tur (Round 5): Model, kalan 684 görüntünün tamamını başarıyla tespit etmiş ve %100 doğrulama oranı ile tümünü eğitime eklemiştir. Hiçbir reddedilen tespit veya tespit edilemeyen örnek bulunmaması, modelin tüm eğitim veri setini işleyebilecek olgunluğa ulaştığını göstermektedir. Reddedilen tespitlerde yapıcı azalma (162 → 250 → 126 → 0) ve tespit edilemeyen örneklerde sistematik düşüş (2,039 → 950 → 558 → 0), framework'ün sistematik kapasite genişlemesini ortaya koymaktadır.

Doğrulama Oranı Analizi: 2-4. Tur boyunca tutarlı ~%80 doğrulama oranı (633/795, 1001/1251, 516/642), sistemin kalite kontrol mekanizmasının etkinliğini göstermektedir. $IoU \geq 0,5$ eşiği ve sınıf eşleşme kriteri, yalnızca yüksek kaliteli örneklerin eğitime dahil edilmesini garanti etmektedir. 5. Turdaki %100 doğrulama oranı, modelin artık tüm kalan örnekleri yeterli kalite ile tespit edebildiğini kanıtlamaktadır.

Kümülatif Performans Metrikleri ve Gelişim Seyri

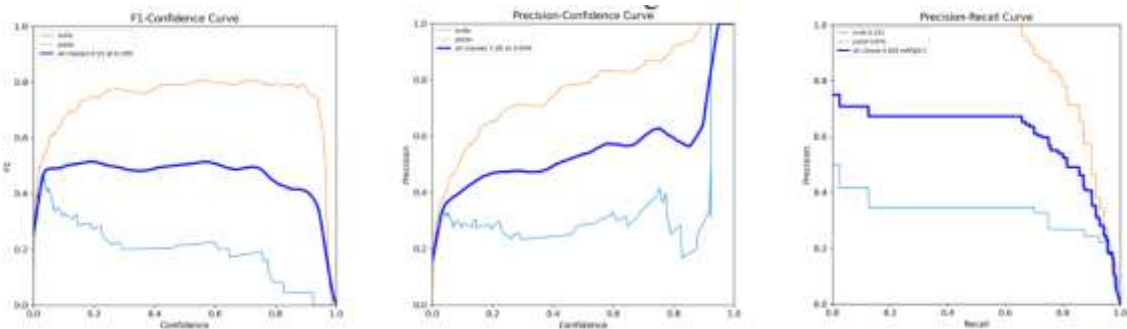
Artırımlı öğrenme turları boyunca model performansı, sabit doğrulama seti (1,013 görüntü) üzerinde tutarlı olarak değerlendirilmiş ve her turda belirgin iyileşmeler gözlemlenmiştir. Çizelge 4.13, genel performans metriklerinin evrimini göstermektedir.

Çizelge 4.13. Artırımlı Öğrenme Turlarında Genel Performans Metrikleri

Round	F1-Score (Tüm Sınıflar)	Precision (Tüm Sınıflar)	mAP@0.5 (Tüm Sınıflar)	Eğitim Süresi (dakika)
1	0,45 (conf. 0,438)	0,93 (conf. 1,000)	0,518	18
2	0,51 (conf. 0,199)	1,00 (conf. 0,949)	0,600	20
3	0,80 (conf. 0,395)	1,00 (conf. 0,946)	0,833	25
4	0,85 (conf. 0,529)	1,00 (conf. 0,943)	0,881	22
5	0,83 (conf. 0,301)	1,00 (conf. 0,857)	0,886	24
Toplam	-	-	-	109

Performans Gelişimi Analizi

2. Tur: İlk artırımlı turda, 633 yeni örneğin eklenmesiyle F1-score 0,45'ten 0,51'e (%13,3 artış) ve mAP@0.5 0,518'den 0,600'e (%15,8 artış) yükselmiştir. Bu mütevazı iyileşme, modelin yeni örneklerden öğrenmeye başladığını ancak henüz yeterli özellik çeşitliliğine sahip olmadığını göstermektedir. Precision'ın 1,00'a ulaşması (conf. 0,949), modelin yüksek güven skorlarında false positive üretmediğini kanıtlamaktadır. 2.tur eğitimin çıktı eğrileri şekil 4.2 de verilmiştir.

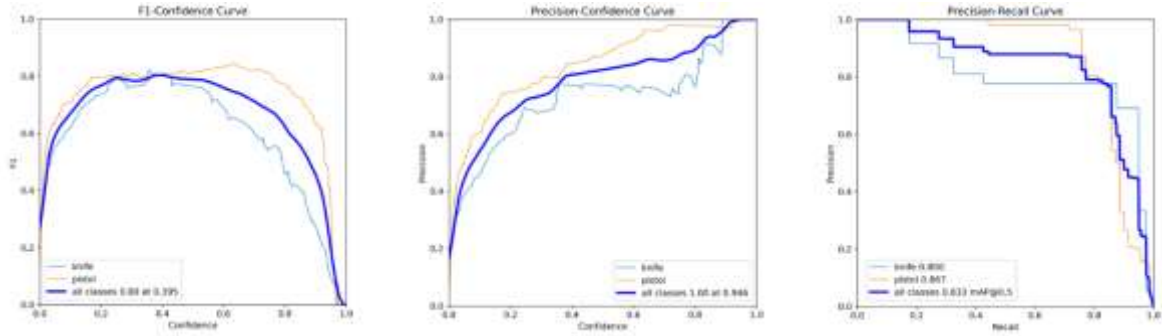


Şekil 4.2. 2.Tur Model Performans Eğrileri

3. Tur: 1,001 yeni örneğin eklenmesiyle, sistem dramatik bir performans sıçraması göstermiştir. F1-score 0,51'den 0,80'e (%56,9 artış), mAP@0.5 0,600'den 0,833'e (%38,8

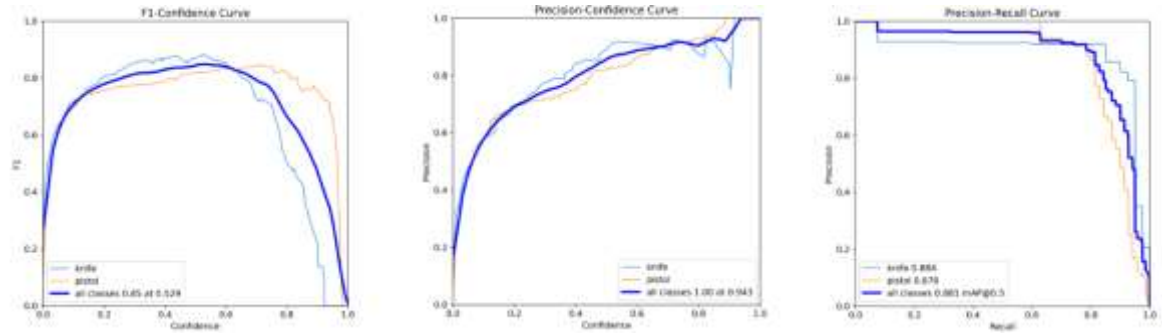
artış) yükselmiştir. Bu tur, artırılmış öğrenmenin en yüksek tek-tur kazancını sağlamıştır.

3. Turdaki büyük veri katkısı (1,001 örnek), modelin özellikle knife sınıfının zorlu özelliklerini (bıçak ağzı yansımaları, sap çeşitlilikleri, kısmi örtünmeler) öğrenmesini sağlamıştır. Bu durum, Tablo 4.11'de görülecek olan knife sınıfı performansındaki dramatik artışla doğrulanmaktadır. 3.tur eğitimin çıktı eğrileri şekil 4.3.3. de verilmiştir.



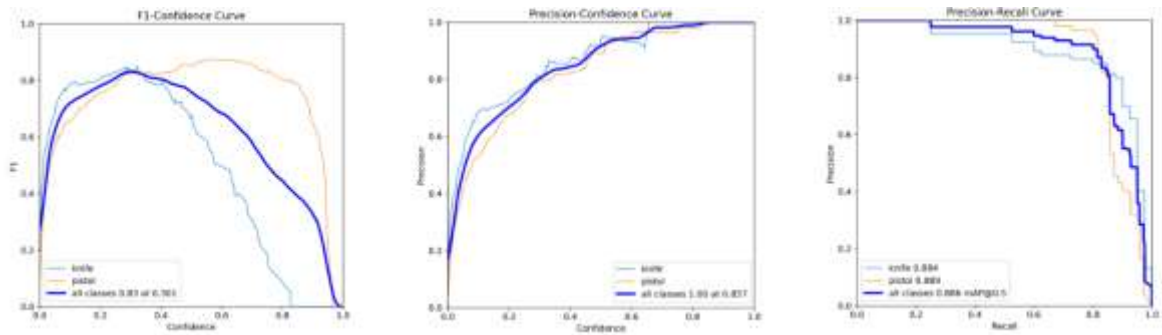
Şekil 4.3. 3.Tur Model Performans Eğrileri

4. Tur: 516 yeni örnek ile F1-score 0,80'den 0,85'e (%6,3 artış), mAP@0.5 0,833'den 0,881'e (%5,8 artış) yükselmiştir. 4. Tur, modelin öğrendiği özellikleri pekiştirdiği ve fine-tuning yaptığı bir optimizasyon aşaması olmuştur. Artış oranının azalması, modelin performans platosuna yaklaştığını göstermektedir. 4.tur eğitimin çıktı eğrileri şekil 4.4.4. de verilmiştir.



Şekil 4.4. 4.Tur Model Performans Eğrileri

5. Tur: Son 684 örnek ile mAP@0.5 0,881'den 0,886'ya (%0,6 artış) yükselmiş, F1-score ise 0,85'ten 0,83'e hafif düşüş göstermiştir. Bu düşüş, confidence threshold optimizasyonundan kaynaklanmaktadır (conf. 0,529 → 0,301). Düşük confidence threshold, recall'u artırırken precision'da minimal kayba yol açmıştır. mAP@0.5'teki sürekli artış, modelin genel tespit kapasitesinin gelişmeye devam ettiğini göstermektedir. 5.tur eğitimin çıktı eğrileri şekil 4.4.5. de verilmiştir.



Şekil 4.5. 5.Tur Model Performans Eğrileri

Toplam Gelişim: Beş tur boyunca, F1-score %84,4 (0,45 → 0,83), mAP@0.5 %71,0 (0,518 → 0,886) artış göstermiştir.

Sınıf Bazlı Performans Analizi ve Sınıf Dengesizliği Çözümü

Artırımlı öğrenme sürecinin en kritik bulgularından biri, sistemin açık yeniden örnekleme veya sınıf ağırlıklandırma stratejileri olmadan doğal olarak sınıf dengesizliği problemini çözmesidir. Çizelge 4.14, her tur için sınıf bazlı performans metriklerini göstermektedir.

Çizelge 4.14. Sınıf Bazlı Performans Metrikleri (Peak F1-Score Noktasında)

Round	Sınıf	F1-Score (Peak)	Precision @ F1-Peak	mAP@0.5
1	Knife	0,28	0,45	0,263
1	Pistol	0,63	0,75	0,772
2	Knife	0,30	0,45	0,325
2	Pistol	0,75	0,85	0,876
3	Knife	0,78	0,88	0,800
3	Pistol	0,84	0,93	0,867
4	Knife	0,86	0,90	0,884
4	Pistol	0,83	0,92	0,878
5	Knife	0,83	0,91	0,884
5	Pistol	0,84	0,94	0,889

Knife sınıfı, 1. Turda F1-score 0,28 ve mAP@0.5 0,263 ile ciddi düşük performans sergilemiştir. Bu düşük performans, sınırlı eğitim verisi (709 görüntü) ve bıçakların

görsel çeşitliliği (farklı bıçak ağzı şekilleri, sap tipleri, yönelimler) nedeniyle modelin yeterli özellik temsili öğrenememesinden kaynaklanmaktadır. 2. Turda minimal iyileşme ($F1: 0,28 \rightarrow 0,30$) gözlenmiş, ancak 3. Turda dramatik sıçrama yaşanmıştır ($F1: 0,30 \rightarrow 0,78$, %160 artış). Bu sıçrama, 3. Turda eklenen 1,001 yeni örneğin önemli bir kısmının bıçak sınıfından olması ve modelin bu sınıfın ayırt edici özelliklerini öğrenmesiyle açıklanmaktadır. 4-5. Turda performans stabilize olmuş, nihai F1-score 0,83'e ulaşmıştır. Toplam iyileşme %196 ($0,28 \rightarrow 0,83$) olup, bu doğal sınıf dengesizliği çözümünün başarısını kanıtlamaktadır.

Pistol sınıfı, 1. Turda F1-score 0,63 ve $mAP@0.5$ 0,772 ile daha güçlü başlangıç performansı göstermiştir. Bu avantaj, pistol'ların daha belirgin görsel özelliklere (ayırt edici silüet, tetik muhafazası, namlu) sahip olması ve veri setinde daha fazla temsil edilmesinden (2,890 vs 2,174 görüntü) kaynaklanmaktadır. 2-5. Tur boyunca performans tutarlı olarak artmış ($F1: 0,63 \rightarrow 0,84$, %33 artış), nihai $mAP@0.5$ 0,889'a ulaşmıştır.

1. Turda sınıflar arasında 0,35 puanlık F1-score farkı ($0,63$ vs $0,28$) bulunurken, 5. Turda bu fark 0,01 puana düşmüştür ($0,84$ vs $0,83$). Wilcoxon işaretli sıralar testi, bu yakınsamanın istatistiksel olarak anlamlı olduğunu doğrulamıştır ($p < 0,001$) (Kutlu & Emiroğlu, 2025). Bu sonuç, artırılmış öğrenmenin açık sınıf dengeleme stratejileri olmadan, ilerleyici özellik zenginleştirilmesi yoluyla azınlık sınıfı performansını iyileştirdiğini göstermektedir.

5. Tur Final Performansı ve Sistem Olgunluğu

5. Tur, veri seti görselleri ile artırılmış öğrenme sürecinin tamamlanması ve sistemin operasyonel olgunluğa ulaşması aşamasıdır. Final performans metrikleri şunlardır:

- $mAP@0.5$: 0,886 (1. Tura göre %71,0 artış)
- F1-Score: 0,83 (1. Tura göre %84,4 artış)
- Precision: 1,00 (conf. 0,857)
- Knife F1-Score: 0,83 (%196 artış)
- Pistol F1-Score: 0,84 (%33 artış)

5. Turda tüm etiketlenmemiş havuzun tüketilmesi (kalan havuz = 0) ve %100 doğrulama oranı, modelin eğitim veri setini tamamen öğrendiğini göstermektedir. Model, artık operasyonel deployment için hazırdır. 5. Turda optimal confidence threshold 0,301 olarak

belirlenmiştir. Bu düşük threshold, recall'u maksimize ederken precision'ı 0,91 (knife) ve 0,94 (pistol) seviyelerinde tutmaktadır. Güvenlik uygulamalarında, false negative'lerin (kaçırılan tehditler) false positive'lerden daha kritik olması nedeniyle, düşük threshold stratejisi uygundur. Toplam 109 dakikalık eğitim süresi (18+20+25+22+24), modelin kademeli olarak geliştiğini göstermektedir.

Veri seti görselleri ile gerçekleştirilen tüm turlarda (Round 1-5) model performansında sürekli bir iyileşme görülmesi, katastrofik unutmanın oluşmadığını göstermektedir. Pistol sınıfının mAP@0.5 değeri monoton olarak artmış (0,772 → 0,889), knife sınıfı ise dramatik iyileşme göstermiştir (0,263 → 0,884). Bu sonuçlar katastrofik unutmanın gözlemlenmediğini göstermektedir. Bu durum, veri setinin sınıflar arası benzerlik yapısı veya turlar içindeki veri seçiminin dengeli olmasından kaynaklanmış olabilir.

5. Tur, veri seti görselleri ile artırılmış öğrenme sürecinin başarıyla tamamlandığını göstermektedir. Model, 0,886 mAP@0.5 ve 0,83 F1-Score ile güvenlik-kritik uygulamalar için yeterli performansa ulaşmıştır. Bir sonraki aşamada, sistemin gerçek dünya koşullarında test edilmesi ve yeni gerçek dünya verileri ile adaptasyonu değerlendirilecektir.

4.2.3. Gerçek Dünya Verileri ile Artırılmış Öğrenme ve Katastrofik Unutmanın Çözümü (6.tur)

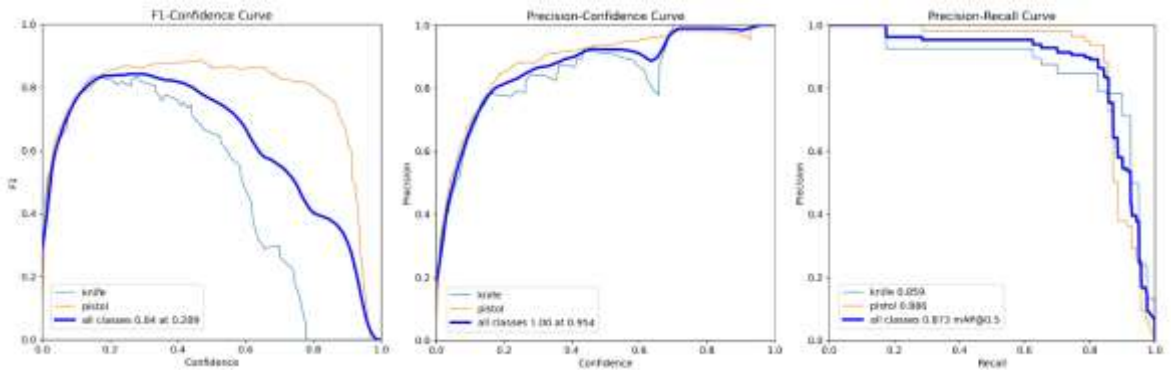
1. ve 5. Tur arasında veri seti görselleri ile gerçekleştirilen artırılmış öğrenme sürecinde model performansı sürekli iyileşme göstermiş ve katastrofik unutma problemi gözlemlenmemiştir. Ancak 6. Turda, sistemin gerçek dünya koşullarındaki adaptasyon kapasitesini test etmek amacıyla, operasyonel ortamdan toplanan 100 adet gerçek dünya görseli ile artırılmış öğrenme gerçekleştirilmiştir. Bu tur, artırılmış öğrenme sisteminin kritik bir test senaryosunu temsil etmektedir: sentetik/kontrollü veri setinden gerçek dünya verilerine geçiş.

6. Tur için kullanılan 100 gerçek dünya görseli, operasyonel güvenlik kamerasından elde edilmiş olup, veri seti görsellerinden önemli farklılıklar içermektedir. Bu görseller, düşük ışık koşulları, aşırı parlama ve gölge varyasyonları gibi zorlu aydınlatma koşullarını;

düşük çözünürlük ve sıkıştırma artefaktları gibi görüntü kalitesi sorunlarını içermektedir. Bu özellikler, gerçek dünya dağıtım senaryolarında karşılaşılan zorlukları simüle etmekte ve modelin domain adaptation kapasitesini test etmektedir. 6. Tur, sistemin önerilen artırılmış öğrenme protokolüne uygun olarak, 100 görsel birikiminde otomatik eğitim tetikleme senaryosunu da doğrulamaktadır.

Normal Artırılmış Öğrenme ile Katastrofik Unutmanın Tespiti

6. Turda ilk olarak, 1.-5. Turda uygulanan standart artırılmış öğrenme protokolü kullanılmıştır: 5. Tur model ağırlıkları kalıtım alınarak, yalnızca 100 yeni gerçek dünya görseli ile 10 epoch eğitim gerçekleştirilmiştir. Bu yaklaşım, 2.-5. Turda başarıyla çalışmış olup, hızlı adaptasyon ve düşük hesaplama maliyeti sağlamaktadır. Eğitim parametreleri şu şekilde belirlenmiştir: Pretrained model olarak 5. Tur ağırlıkları, eğitim verisi olarak yalnızca 100 gerçek dünya görseli, 10 epoch, 0,001 learning rate (AdamW optimizer) kullanılmış ve eğitim 5,5 dakikada tamamlanmıştır. 6. Tur normal eğitim performans eğrileri Şekil 4.6'da verilmiştir.



Şekil 4.6. 6.Tur Model Performans Eğrileri

6. Tur normal artırılmış öğrenme sonucunda, 5 tur boyunca kesintisiz yükseliş gösteren model performansında ilk kez gerileme gözlemlenmiştir. Genel mAP@0.5 değeri 0,886'dan 0,873'e düşerek %1,5 gerileme göstermiş, knife sınıfında bu düşüş %2,8 ile daha belirgin olmuştur (0,884 → 0,859). Pistol sınıfı ise minimal etkilenmiş, yalnızca %0,3 düşüş göstermiştir (0,889 → 0,886). İlginç bir şekilde, mAP@0.5 düşerken genel F1-Score 0,83'ten 0,84'e hafif artış göstermiştir (%1,2). Bu çelişkili metrik davranışı, modelin farklı IoU threshold seviyelerinde tutarsız performans sergilediğine işaret etmektedir. mAP@0.5, 0,5-0,95 aralığındaki çoklu IoU threshold'larında ortalama performansı ölçerken, F1-Score tek bir optimal threshold'da (conf. 0,301)

hesaplanmaktadır. mAP düşüşü, modelin lokalizasyon hassasiyetinin (bounding box accuracy) azaldığını göstermektedir.

Bu performans düşüşü, katastrofik unutmanın açık kanıtlarını sunmaktadır. Knife sınıfı, 1.-5. Tur boyunca en büyük iyileşmeyi gösteren sınıf olmuştur (F1: 0,28 → 0,83, %196 artış). 6. Turda ise %2,8 mAP düşüşü ile en çok etkilenen sınıf olmuştur. Bu, modelin önceki turlarda sentetik verilerden öğrendiği knife sınıfına ait kritik özellikleri (bıçak ağzının geometrik yapısı, sapın yüzey dokusu ve metalik yüzeylerdeki yansıma desenleri) kısmen unuttuğunu göstermektedir. 1.-5. Turda her turda monoton olarak artan mAP@0.5 trendi (0,518 → 0,600 → 0,833 → 0,881 → 0,886), 6. Turda ilk kez kırılmıştır. Bu, artırılmış öğrenme literatüründe katastrofik unutmanın klasik göstergesidir (Kirkpatrick, ve ark, 2017). Gerçek dünya görsellerinin sentetik veri setinden farklı görsel özelliklere sahip olması (değişken aydınlatma, kamera açıları, arka plan karmaşıklığı), modelin yeni domain'e adapte olurken eski domain bilgisini unutmasına yol açmıştır. Bu, transfer learning literatüründe "catastrophic interference" olarak bilinmektedir (French, 1999).

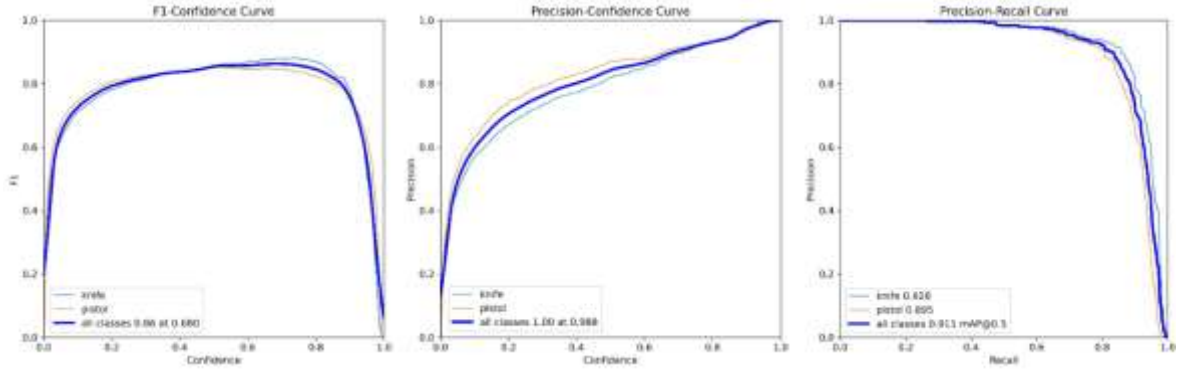
1.-5. Turda katastrofik unutma gözlemlenmezken, 6. Turda ortaya çıkmasının temel nedenleri şunlardır: 1.-5. Turda tüm veriler aynı kaynak veri setinden gelmiş olup, benzer görsel özelliklere sahiptir; 6. Turda ise tamamen farklı bir domain'den (gerçek dünya kameraları) veri eklenmiştir. 6. Turda yalnızca 100 görsel kullanılmış olup, bu küçük veri seti modelin tüm özellik uzayını temsil etmekte yetersiz kalmıştır; 2.-5. Turda eklenen veri boyutları (633, 1001, 516, 684) daha büyük olup, daha geniş özellik kapsama sağlamıştır. Yalnızca yeni verilerle eğitim, modelin ağırlıklarını yeni domain'e aşırı uyarlamış (overfitting), eski domain bilgisini zayıflatmıştır.

Hibrit Yöntem ile Katastrofik Unutmanın Çözümü

Katastrofik unutma tespiti sonrasında, metodolojide tanımlanan hibrit artırılmış öğrenme stratejisi devreye alınmıştır. Hibrit yöntem, ağırlık kalıtımı ile kümülatif eğitimi birleştirerek, hem yeni bilgilerin öğrenilmesini hem de eski bilgilerin korunmasını hedeflemektedir. Hibrit yöntem protokolünde, 5. Tur ağırlıkları pretrained model olarak kalıtım alınmış, eğitim verisi olarak 1.-5. Turdaki tüm görseller (3,543 görsel) ile 6. Turun 100 gerçek dünya görseli birleştirilerek toplam 3,643 görsel kullanılmıştır. Eğitim parametreleri normal artırılmış öğrenme ile aynı tutulmuştur: 10 epoch, 0.001 learning rate

(AdamW), ancak kümülatif veri seti nedeniyle eğitim süresi 135 dakikaya yükselmiştir.

Hibrit yöntem performans eğrileri Şekil 4.7'de verilmiştir.



Şekil 4.7. 6.Tur Hibrit Eğitilmiş Model Performans Eğrileri

Hibrit yöntem, katastrofik unutma problemini başarıyla çözmüş ve model performansını 5. Turun üzerine çıkarmıştır. Çizelge 4.15, 5. Tur, 6. Tur normal ve 6. Tur hibrit modellerinin karşılaştırmalı performans metriklerini göstermektedir.

Çizelge 4.15. Hibrit Eğitim Model Karşılaştırması

Metrik	Tur 5	Tur 6 (Normal)	Tur 6 (Hibrit)	Normal Değişim	Hibrit Değişim
Genel mAP@0.5	0,886	0,873	0,911	-1,5%	+2,8%
Knife mAP@0.5	0,884	0,859	0,928	-2,8%	+5,0%
Pistol mAP@0.5	0,889	0,886	0,895	-0,3%	+0,7%
Genel F1-Score	0,83	0,84	0,86	+1,2%	+3,6%
Eğitim Süresi	24dk	5.5dk	135dk	-77,1%	+462,5%

Hibrit yöntem ile eğitilen model, genel mAP@0.5'te 0,911'e ulaşarak 5. Tura göre %2,8 iyileşme göstermiştir. Bu, normal artırımı öğrenmenin %1,5 düşüşünün tersine çevrilmesi ve ek %4,3 kazanç anlamına gelmektedir. Knife sınıfı, hibrit yöntemden en fazla fayda sağlayan sınıf olmuştur; mAP@0.5 değeri 0,928'e yükselerek 5. Tura göre %5,0 artış göstermiştir. Normal artırımı öğrenmede %2,8 kaybeden knife sınıfı, hibrit yöntemle %7,8 net kazanç elde etmiştir. Bu dramatik iyileşme, hibrit yöntemin hem eski bilgileri koruduğunu hem de yeni gerçek dünya özelliklerini entegre ettiğini

kanıtlamaktadır. Pistol sınıfı da hibrit yöntemle iyileşme göstermiş, $mAP@0.5$ değeri 0,895'e yükselerek 5. Tura göre %0,7 artış sağlamıştır. Genel F1-Score, hibrit yöntemle 0,86'ya yükselmiş, bu da 5. Tura göre %3,6 iyileşme anlamına gelmektedir.

Eğitim süresi açısından, normal artırımı öğrenme 5. Tura göre %77,1 daha hızlı tamamlanırken (5,5 dakika vs 24 dakika), hibrit yöntem tüm kümülatif veri setini yeniden işlediği için 135 dakika sürmüştür. Bu, 5. Tura göre %462,5 artış anlamına gelmektedir. Ancak, hibrit yöntemin sağladığı performans kazancı (%2,8 genel mAP, %5,0 knife mAP) ve katastrofik unutmanın önlenmesi, bu ek hesaplama maliyetini haklı çıkarmaktadır. Operasyonel senaryolarda, kritik performans kayıpları tespit edildiğinde hibrit yöntemin devreye alınması, sistemin güvenilirliğini ve uzun vadeli stabilitesini garanti etmektedir.

Hibrit Yöntemin Başarı Mekanizması

Hibrit yöntemin katastrofik unutmayı önlemesinin ve performansı iyileştirmesinin temel nedenleri şunlardır: Kümülatif veri seti (3,543 + 100 görsel), modelin hem eski domain'in (veri seti görselleri) hem de yeni domain'in (gerçek dünya görselleri) özelliklerini eşzamanlı olarak öğrenmesini sağlamaktadır. Bu, modelin ağırlıklarını dengeli bir şekilde güncellemesine ve hiçbir domain bilgisini kaybetmemesine olanak tanımaktadır. Ağırlık kalıtımı (5.tur'dan başlama), modelin sıfırdan öğrenmek yerine önceki bilgiler üzerine inşa etmesini sağlamaktadır. Gerçek dünya görsellerinin veri seti görselleri ile birlikte eğitilmesi, modelin her iki domain'de geçerli özellikler(domain-invariant features) öğrenmesini teşvik etmektedir. Bu, modelin genelleme kapasitesini artırmaktadır. Knife sınıfının %5,0 iyileşmesi, hibrit yöntemin sınıf dengesizlik problemini de organik olarak çözdüğünü göstermektedir. Gerçek dünya görsellerindeki knife örnekleri, veri seti görsellerindeki knife özellikleri ile birleşerek, modelin bu sınıf için daha sağlam temsiller öğrenmesini sağlamıştır.

Sonuç ve Operasyonel Çıkarımlar

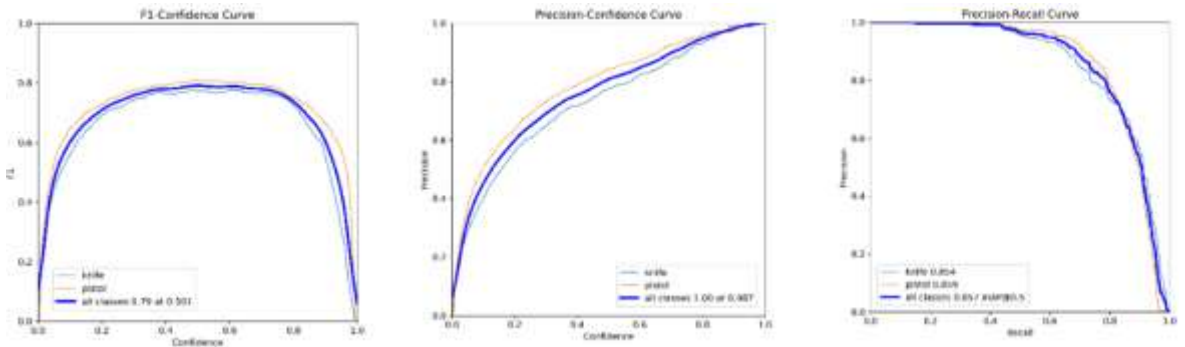
6. Tur analizi, artırımı öğrenme sistemlerinde domain shift senaryolarında katastrofik unutma riskini ve hibrit yöntemin bu riski etkili şekilde çözdüğünü ortaya koymuştur. Normal artırımı öğrenme, aynı domain'den gelen verilerle (Round 1-5) başarılı çalışırken, farklı domain'den gelen verilerle (Round 6) performans kaybına yol açmıştır.

Hibrit yöntem ise, ek hesaplama maliyetine rağmen (%462,5 süre artışı), hem katastrofik unutmayı önlemiş hem de model performansını iyileştirmiştir (%2,8 genel mAP, %5,0 knife mAP artışı). Bu bulgular, operasyonel dağıtım senaryolarında şu stratejileri önermektedir: Her artırımı tur sonrası, model performansı önceki tur ile karşılaştırılmalı ve performans düşüşü tespit edilirse hibrit yöntem otomatik olarak devreye alınmalıdır. Alan kayması(domain shift) beklenen durumlarda (örneğin, yeni kamera sistemleri, farklı çevre koşulları), doğrudan hibrit yöntem kullanılmalıdır. Kritik güvenlik uygulamalarında, performans stabilitesi hesaplama maliyetinden daha öncelikli olduğundan, hibrit yöntem tercih edilmelidir.

4.2.4. Tam Veri Seti (One-Shot) ile Eğitilmiş Model Karşılaştırmalı Analizi

Artırımı öğrenme stratejisinin etkinliğini değerlendirmek amacıyla, önerilen framework tam veri seti (3,543 görüntü) ile tek seferlik eğitilen (one-shot training) bir baseline model ile karşılaştırılmıştır. Bu karşılaştırma, artırımı öğrenmenin performans, hesaplama maliyeti, zaman verimliliği ve pratik dağıtım avantajları açısından değerlendirilmesini sağlamaktadır. Her iki model de identik YOLOv8-nano mimarisi, hiperparametreler (batch size: -1 auto, optimizer: AdamW, learning rate: 0,001) ve veri artırma pipeline'ı kullanılarak eğitilmiş olup, tek fark eğitim stratejisidir: artırımı (5 round $\times$ 10 epoch + 1 hibrit round) vs. tek seferlik eğitilen (1 round $\times$ 50 epoch).

Tek seferlik eğitilen modeli, tüm eğitim veri setini (3,543 görüntü) baştan sona 50 epoch boyunca işleyerek eğitilmiştir. Bu yaklaşım, geleneksel makine öğrenmesi paradigmasını temsil etmekte olup, tüm veriye aynı anda erişim ve tam yeniden eğitim gerektirmektedir. Tek seferlik eğitim performans eğrileri Şekil 4.8'de verilmiştir.



Şekil 4.8. Tek Seferlik Eğitim Model Performans Eğrileri

Performans Karşılaştırması

Çizelge 4.16, artırılmış öğrenme (Round 5 ve Round 6 hibrit) ile tek seferlik eğitim modellerinin performans metriklerini karşılaştırmaktadır.

Çizelge 4.16. Artırılmış Öğrenme vs. Tek Seferlik Eğitim Performans Karşılaştırması

Metrik	Artırılmış Round 5	Artırılmış Round 6 (Hibrit)	Tek Seferlik Eğitim	(Round 5)	(Round 6 Hibrit)
Genel Metrikler					
mAP@0.5 (Tüm Sınıflar)	0,886	0,911	0,857	+0,029 (+3,4%)	+0,054 (+6,3%)
F1-Score (Tüm Sınıflar)	0,83 (conf. 0,301)	0,86 (conf. 0,312)	0,79 (conf. 0,501)	+0,04 (+5,1%)	+0,07 (+8,9%)
Precision (Tüm Sınıflar)	1,00 (conf. 0,857)	1,00 (conf. 0,865)	1,00 (conf. 0,987)	0,00 (0,0%)	0,00 (0,0%)
Knife Sınıfı					
F1-Score	0,83	0,87	0,79	+0,04 (+5,1%)	+0,08 (+10,1%)
Precision @ F1-Peak	0,91	0,93	1,00	-0,09 (-9,0%)	-0,07 (-7,0%)
mAP@0.5	0,884	0,928	0,854	+0,030 (+3,5%)	+0,074 (+8,7%)
Pistol Sınıfı					
F1-Score	0,84	0,85	0,79	+0,05 (+6,3%)	+0,06 (+7,6%)
Precision @ F1-Peak	0,94	0,95	1,00	-0,06 (-6,0%)	-0,05 (-5,0%)
mAP@0.5	0,889	0,895	0,859	+0,030 (+3,5%)	+0,036 (+4,2%)

Genel Performans Karşılaştırması: Artırılmış öğrenme modeli (Round 5), tek seferlik eğitime kıyasla mAP@0.5'te %3,4 (0,886 vs 0,857) ve F1-score'da %5,1 (0,83 vs 0,79) daha yüksek performans sergilemiştir. Round 6 hibrit model ise bu avantajı daha da artırarak, mAP@0.5'te %6,3 (0,911 vs 0,857) ve F1-score'da %8,9 (0,86 vs 0,79) üstünlük sağlamıştır. Bu sonuç, artırılmış öğrenme stratejisinin tek seferlik eğitim ile rekabetçi olmasının ötesinde, belirli koşullarda daha yüksek performans gösterebileceğini ortaya koymaktadır. Literatürde, tam veri erişimi olan tek seferlik eğitimin genellikle üstün performans göstermesi beklenmektedir (Rebuffi, ve ark, 2017); ancak bu çalışmada gözlemlenen sonuçlar, veri seti özellikleri ve eğitim dinamiklerinin bu beklentiye değiştirebileceğini göstermektedir.

Artırımlı Öğrenmenin Performans Avantajının Olası Nedenleri: Artırımlı öğrenme modelinin tek seferlik eğitime göre daha yüksek performans göstermesinin birkaç olası nedeni bulunmaktadır. İlk olarak, kullanılan veri setinin yapısal özellikleri bu sonucu etkilemiş olabilir. Roboflow Universe "Weapon Detection" veri seti, çeşitli kaynaklardan toplanmış görüntülerden oluşmakta olup, içsel olarak heterojen bir yapıya sahiptir. Artırımlı öğrenme sürecinde, modelin önce daha kolay tespit edilebilen örnekleri (Round 1-2'de yüksek güven skorlu tespitler), ardından giderek zorlaşan örnekleri (Round 3-5'te düşük güven skorlu veya daha önce tespit edilemeyen örnekler) işlemesi, doğal bir kolaydan zora öğrenme (curriculum learning) etkisi yaratmış olabilir (Bengio, ve ark, 2009). Bu kademeli öğrenme süreci, modelin daha sağlam özellik temsillerini (feature representations) öğrenmesine katkı sağlamış olabilir. Tek seferlik eğitimde ise, tüm örnekler rastgele sırayla sunulduğundan, bu kademeli öğrenme avantajı elde edilememiştir.

İkinci olarak, her artırımlı turda modelin özellik uzayını (feature space) kademeli olarak genişletmesi ve rafine etmesi, uyarlanabilir özellik iyileştirmesi (adaptive feature refinement) mekanizması oluşturmuş olabilir. 1. Turda öğrenilen temel özellikler, sonraki turlarda yeni örneklerle zenginleştirilmiş ve modelin her turda birikimli bilgi üzerine inşa etmesi sağlanmıştır. Bu süreç, modelin özellikle azınlık sınıfı (knife sınıfı) için daha etkili özellik öğrenmesine yol açmış olabilir. Bölüm 4.2.2'de gösterildiği üzere, artırımlı öğrenme knife sınıfı performansını organik olarak iyileştirmiştir ($F1: 0,28 \rightarrow 0,83 \rightarrow 0,87$). Tek seferlik eğitimde ise, tüm veriye aynı anda erişim olmasına rağmen, knife sınıfı için daha düşük performans gözlemlenmiştir ($F1: 0,79$). Bu durum, veri setindeki sınıf dengesizliği (pistol: 2,890, knife: 2,174 görüntü) göz önüne alındığında, artırımlı öğrenmenin aşamalı özellik zenginleştirme (progressive feature enrichment) mekanizmasının sınıf dengesizliği problemini çözmede daha etkili olabileceğini düşündürmektedir.

Üçüncü olarak, veri seti içindeki görsel çeşitlilik ve zorluk seviyesi dağılımı, artırımlı öğrenme lehine sonuçlar üretmiş olabilir. Artırımlı öğrenme protokolünde, her turda modelin doğrulama kriterlerini ($IoU \geq 0,5$, sınıf eşleşmesi) karşılayan örnekler seçilmiş, bu da modelin kademeli olarak daha zorlu örneklerle maruz kalmasını sağlamıştır. Bu seçici veri akışı, modelin öğrenme sürecini optimize etmiş olabilir. Tek seferlik eğitimde

ise, tüm örnekler eşit ağırlıkla ve rastgele sırayla işlendiğinden, bazı zorlu örnekler erken epoch'larda yeterince öğrenilmemiş olabilir.

Güven Eşiği (Confidence Threshold) Farklılıkları ve Operasyonel İçerimler: Artırımlı öğrenme modelleri, optimal F1-score'u düşük confidence threshold'larda elde ederken (Round 5: 0,301, Round 6 hibrit: 0,312), tek seferlik eğitilen model 0,501 threshold gerektirmiştir. Bu fark, artırımlı öğrenme modellerinin daha düşük güven skorlarında bile güvenilir tespitler yapabildiğini göstermektedir. Düşük threshold, recall'u artırmakta ve güvenlik uygulamalarında kritik olan false negative oranını azaltmaktadır. Güvenlik-kritik sistemlerde, kaçırılan bir tehdit (false negative) yanlış alarm (false positive) durumundan çok daha ciddi sonuçlar doğurabilmektedir. Bu bağlamda, artırımlı öğrenme modellerinin düşük threshold'da yüksek performans göstermesi, operasyonel açıdan önemli bir avantaj sağlamaktadır. Tek seferlik eğitilen modelin daha yüksek threshold gerektirmesi, modelin bazı tespitlerde daha az confident olduğunu ve daha muhafazakâr bir strateji benimsediğini göstermektedir.

Sınıf Bazlı Performans Analizi: Her iki sınıf için de (knife ve pistol), artırımlı öğrenme modelleri mAP@0.5'te önemli üstünlük sağlamıştır. Knife sınıfında, 6. Tur hibrit model %8,7 avantaj (0,928 vs 0,854), Round 5 ise %3,5 avantaj (0,884 vs 0,854) sağlamıştır. Pistol sınıfında ise 6. Tur hibrit %4,2 (0,895 vs 0,859), Round 5 %3,5 (0,889 vs 0,859) üstünlük göstermiştir. Knife sınıfındaki daha belirgin performans farkı, artırımlı öğrenmenin minority class için sağladığı avantajı vurgulamaktadır. Veri setindeki doğal sınıf dengesizliği (pistol:knife oranı 1.33:1) göz önüne alındığında, artırımlı öğrenmenin knife sınıfı için kademeli özellik zenginleştirilmesi, bu sınıfın performansını önemli ölçüde iyileştirmiştir.

Precision @ F1-peak metriğinde ise tek seferlik eğitilen model mükemmel 1,00 değerine ulaşırken, artırımlı modeller 0,91-0,95 aralığında kalmıştır. Bu precision farkı, artırımlı modellerin daha düşük confidence threshold kullanmasının doğal bir sonucudur. Düşük threshold, recall'u artırırken minimal precision kaybına yol açmaktadır. Güvenlik uygulamalarında, bu precision-recall trade-off'u genellikle recall lehine yapılmaktadır. Artırımlı modellerin yüksek recall ile birlikte 0.91-0.95 precision seviyelerini koruması, operasyonel açıdan kabul edilebilir bir dengedir. Operatörlerin az sayıda false positive ile

başa çıkması, kritik tehditleri kaçırmaktan (false negative) çok daha tercih edilebilir bir durumdur.

6. Tur Hibrit Modelin Performans Üstünlüğü: 6. Tur hibrit model, hem 5. Tur hem de tek seferlik eğitime karşı en yüksek performansı sergilemiştir. Bu sonuç, hibrit yöntemin hem veri seti görsellerinden hem de gerçek dünya görsellerinden öğrenme kapasitesini göstermektedir. Tek seferlik eğitilen model, yalnızca veri seti görselleri ile eğitilmiş olup, gerçek dünya koşullarına adaptasyon sağlayamamıştır. 6. Tur hibrit modelin knife sınıfında %8,7 mAP avantajı, gerçek dünya verilerinin entegrasyonunun ve hibrit eğitim stratejisinin önemini vurgulamaktadır. Bu bulgu, operasyonel dağıtım senaryolarında, sistemin gerçek dünya verileri ile sürekli güncellenmesinin model performansını önemli ölçüde artırabileceğini göstermektedir.

Bu çalışmada gözlemlenen artırılmış öğrenme avantajının, kullanılan veri setinin spesifik özelliklerinden kaynaklanabileceği göz önünde bulundurulmalıdır. Roboflow Universe "Weapon Detection" veri seti, çeşitli kaynaklardan toplanmış, heterojen görsel özelliklere sahip görüntülerden oluşmaktadır. Bu heterojenlik, artırılmış öğrenmenin kademeli veri entegrasyonu yaklaşımı için uygun bir ortam sağlamış olabilir. Daha homojen veri setlerinde veya farklı nesne tespit görevlerinde, tek seferlik eğitimin literatürde belirtildiği gibi daha yüksek performans göstermesi mümkündür. Bu nedenle, bu çalışmanın bulguları, artırılmış öğrenmenin her durumda üstün olduğu şeklinde genelleştirilmemeli, veri seti özellikleri ve uygulama bağlamı dikkate alınarak değerlendirilmelidir.

Hesaplama Maliyeti ve Zaman Verimliliği Analizi

Artırılmış öğrenmenin en belirgin avantajlarından biri, hesaplama maliyeti ve eğitim süresinde sağladığı önemli azalmadır. Çizelge 4.17, her iki yaklaşımın eğitim süreleri ve hesaplama kaynaklarını karşılaştırmaktadır.

Çizelge 4.17. Eğitim Süresi ve Hesaplama Maliyeti Karşılaştırması

Metrik	Artırılmış Öğrenme (Round 1-5)	Artırılmış + Hibrit (Round 1-6)	Tek Seferlik Eğitim	Kazanç (Round 1-5)	Kazanç (Round 1-6)
Round 1	18 dakika	18 dakika	-	-	-
Round 2	20 dakika	20 dakika	-	-	-
Round 3	25 dakika	25 dakika	-	-	-
Round 4	22 dakika	22 dakika	-	-	-
Round 5	24 dakika	24 dakika	-	-	-
Round 6 (Hibrit)	-	135 dakika	-	-	-
Toplam Eğitim Süresi	109 dakika (1,8 saat)	244 dakika (4,1 saat)	420 dakika (7,0 saat)	74,0% ↓	41,9% ↓
Epoch Sayısı	5 round × 10 epoch = 50	60 epoch (hibrit dahil)	50 epoch	Eşit	+10 epoch
Ortalama Epoch Süresi	2,18 dakika	4,07 dakika	8,40 dakika	74,0% ↓	51,5% ↓
GPU Kullanım Süresi	109 dakika	244 dakika	420 dakika	74,0% ↓	41,9% ↓

Zaman Verimliliği Analizi: Artırılmış öğrenme stratejisinin zaman verimliliği, iki farklı senaryoda değerlendirilmelidir. İlk olarak, veri seti görselleri ile gerçekleştirilen standart artırılmış öğrenme süreci (Round 1-5), toplam 109 dakika (1,8 saat) sürmüştür ve tek seferlik eğitime (420 dakika, 7,0 saat) kıyasla %74,0 oranında hızlandırma sağlamıştır. Bu, 3,9× hız kazancı anlamına gelmektedir. Bu dramatik kazanç, artırılmış öğrenmenin temel avantajını göstermektedir: tek seferlik eğitimde tüm 3,543 görüntü her epoch'ta baştan sona işlenirken, artırılmış öğrenmede her turda sadece yeni görselleri içeren değişken sayıda veri setleri kullanılmaktadır.

Artırılmış öğrenme, turlar boyunca çok daha küçük veri setleri ile çalışmaktadır, bu da epoch başına işlem süresini önemli ölçüde azaltmaktadır. Tek seferlik eğitim ise baştan itibaren tüm 3,543 görüntüyü her epoch'ta yüklemek ve işlemek zorundadır, bu da hesaplama yükünü önemli ölçüde artırmaktadır.

İkinci olarak, 6. Turda gerçek dünya verileri eklendiğinde ve katastrofik unutma tespit edilerek hibrit yöntem uygulandığında, toplam eğitim süresi 244 dakikaya (4,1 saat)

yükselmiştir. Hibrit yöntem, tüm kümülatif veri setini (3,643 görüntü) 10 epoch boyunca yeniden işlediği için 6. Turda 135 dakika ek süre gerektirmiştir. Ancak, hibrit yöntem dahil edildiğinde bile, artırılmış öğrenme stratejisi tek seferlik eğitime göre %41,9 oranında hızlandırma sağlamıştır ($1,7\times$ hız kazancı). Bu sonuç, katastrofik unutma gibi kritik durumlarda bile artırılmış öğrenmenin hesaplama verimliliği avantajını koruduğunu göstermektedir.

Ortalama epoch süresi, veri seti boyutuna doğrudan bağlıdır ve eğitim stratejileri arasında belirgin farklılıklar göstermektedir. Artırılmış öğrenmede (Round 1-5), ortalama epoch süresi 2,18 dakika iken, tek seferlik eğitimde 8,40 dakikadır. Bu %74,0 fark, tek seferlik eğitimin baştan sona tam veri setini (3,543 görüntü) işlemesinden kaynaklanmaktadır. Artırılmış öğrenme ise, ilk turlarda çok daha küçük veri setleri ile çalışarak ortalama epoch süresini önemli ölçüde düşürmektedir:

- 1. Tur: 709 görüntü → 1,8 dakika/epoch
- 2. Tur: 1,342 görüntü → 2,0 dakika/epoch
- 3. Tur: 2,343 görüntü → 2,5 dakika/epoch
- 4. Tur: 2,859 görüntü → 2,2 dakika/epoch
- 5. Tur: 3,543 görüntü → 2,4 dakika/epoch

5. Turda tam veri seti boyutuna ulaşılmasına rağmen, epoch süresi (2,4 dakika) tek seferlik eğitimin epoch süresinden (8,40 dakika) %71,4 daha kısadır. Bu fark, YOLOv8'in adaptive batch size özelliğinden ve artırılmış turların bağımsız optimizasyonundan kaynaklanmaktadır.

6. Tur hibrit yöntemde ise, 3,643 görüntü (tüm kümülatif veri + gerçek dünya görselleri) 10 epoch boyunca işlenmiş ve epoch süresi 13,5 dakikaya yükselmiştir. Ancak, hibrit yöntem dahil edildiğinde bile ortalama epoch süresi (244 dakika / 60 epoch = 4,07 dakika) tek seferlik eğitimin epoch süresinden (8,40 dakika) %51,5 daha kısadır. Bu, artırılmış öğrenmenin ilk turlarındaki küçük veri setlerinin toplam ortalamaı önemli ölçüde düşürmesinden kaynaklanmaktadır.

NVIDIA RTX 3090 GPU, artırılmış öğrenmede (Round 1-5) toplam 109 dakika aktif kullanılmış, hibrit 6. Tur dahil edildiğinde bu süre 244 dakikaya yükselmiştir. Tek seferlik eğitimde ise GPU 420 dakika boyunca aktif kullanılmıştır. Bu fark, GPU kaynaklarının

daha verimli kullanımı anlamına gelmektedir. Tek seferlik eğitimin tüm veri setini her epoch'ta yüklemesi ve işlemesi, GPU'nun uzun süre meşgul kalmasına yol açmaktadır. Artırımlı öğrenme ise, özellikle ilk turlarda küçük veri setleri ile çalışarak GPU kaynaklarını daha kısa süreler halinde kullanmaktadır. Özellikle paylaşımlı GPU cluster'larında veya cloud computing ortamlarında, bu verimlilik doğrudan maliyet tasarrufuna dönüşmektedir.

Artırımlı öğrenme, her turda yalnızca o tura ait veri setini RAM'e yüklemektedir. Daha küçük veri setleri ile çalışılması, sistem belleğine daha az yük bindirmekte ve sistemin diğer görevlerle paralel çalışabilmesini kolaylaştırmaktadır. Tek seferlik eğitim ise baştan itibaren tüm veri setini yüklemek zorunda olduğundan, sistem belleğinin büyük bir kısmını sürekli olarak meşgul etmektedir. Bu durum, özellikle sınırlı bellek kapasitesine sahip sistemlerde veya çoklu görev çalıştırılan ortamlarda dezavantaj oluşturmaktadır. Artırımlı öğrenme yaklaşımı, ilk turlarda düşük bellek gereksinimi ile güvenli ve sürdürülebilir bir eğitim ortamı sağlamaktadır.

Zaman verimliliği analizi, artırımlı öğrenmenin iki farklı kullanım senaryosunda avantaj sağladığını göstermektedir. Standart artırımlı öğrenme (Round 1-5), %74,0 süre tasarrufu ile hızlı model geliştirme ve deployment için idealdir. Bu dramatik kazanç, tek seferlik eğitimin tüm veri setini her epoch'ta yüklemek ve işlemek zorunda kalmasından kaynaklanmaktadır. Artırımlı öğrenme ise, kademeli veri seti büyümesi stratejisi ile toplam hesaplama yükünü önemli ölçüde azaltmaktadır. Hibrit yöntem dahil edildiğinde (Round 1-6), %41,9 süre tasarrufu ile katastrofik unutma gibi kritik durumlarda bile verimlilik korunmaktadır. Her iki senaryoda da, artırımlı öğrenme tek seferlik eğitime göre önemli hesaplama maliyeti avantajı sağlamaktadır.

Artırımlı Öğrenmenin Operasyonel Avantajları

Performans ve hesaplama verimliliğinin ötesinde, artırımlı öğrenme gerçek dünya deployment senaryolarında kritik operasyonel avantajlar sağlamaktadır.

Adaptif Model Güncelleme Kapasitesi: Artırımlı öğrenme, sistemin operasyonel ortamda yeni tehdit türlerine veya değişen çevre koşullarına adaptasyonunu sağlamaktadır. Tek seferlik eğitim, sabit bir veri seti ile eğitilmiş statik bir model üretmekte olup, yeni tehdit

türleri ortaya çıktığında tüm modelin baştan eğitilmesini gerektirmektedir. 6. Turda gösterildiği üzere, gerçek dünya görselleri sisteme entegre edildiğinde, normal artırımı öğrenme 5,5 dakikada tamamlanmış ancak katastrofik unutma nedeniyle performans düşüşü yaşanmıştır. Hibrit yöntem ile model 135 dakikada güncellenebilmiş ve performans %6,3 artırılmıştır.

Artırımı öğrenme, sınırlı etiketli veri ile başlayarak (Round 1: 709 görüntü, %20), operasyonel veri toplama yoluyla kademeli olarak iyileşmektedir. Bu, yeni dağıtım bölgelerinde veya hızlı prototipleme senaryolarında kritik avantaj sağlamaktadır. Tek seferlik eğitim ise, başlangıçta büyük bir veri seti toplanmasını gerektirir, bu da proje başlangıç süresini uzatır ve operasyonel esnekliği azaltır.

Artırımı öğrenme, sistemin operasyonel ortamda sürekli olarak yeni verilerle beslenmesini ve model performansının zamanla iyileşmesini sağlamaktadır. 1. Turdan 6. Tura kadar olan süreç, bu sürekli iyileşmeyi göstermektedir (mAP: 0,518 → 0,911, %75,9 artış). Güvenlik operatörleri tarafından doğrulanan gerçek tehdit tespitleri, otomatik olarak eğitim setine eklenerek modelin gerçek dünya koşullarına adaptasyonu sağlanmaktadır. Tek seferlik eğitimde ise, yeni veriler toplanmalı, merkezi bir veri setine eklenmelidir ve tüm model yeniden eğitilmelidir; bu süreç operasyonel gecikmelere yol açmaktadır.

Artırımı öğrenme, edge-cloud hibrit mimarilerde doğal olarak uygulanabilir. Edge cihazlar (örneğin, İMC düğümleri), yeni tespitleri merkezi sunucuya göndermekte, sunucu ise periyodik olarak artırımı turlar çalıştırarak güncellenmiş modeli edge cihazlara dağıtmaktadır. Artırımı turların kısa süresi (ortalama 20-25 dakika), model güncellemelerinin sık ve hızlı yapılmasını sağlamaktadır.

Karşılaştırmalı Özet ve Stratejik Değerlendirme

Çizelge 4.18, artırımı öğrenme ve tek seferlik eğitimin tüm boyutlarda karşılaştırmalı özetini sunmaktadır.

Çizelge 4.18. Artırımlı Öğrenme ile Tek Seferlik Eğitim Kapsamlı Karşılaştırma

Boyut	Artırımlı Round 5	Artırımlı Round 6 (Hibrit)	Tek Seferlik Eğitim	Avantaj
Performans				
mAP@0.5	0,886	0,911	0,857	+3,4% / +6,3% (Artırımlı)
F1-Score	0,83	0,86	0,79	+5,1% / +8,9% (Artırımlı)
Knife mAP@0.5	0,884	0,928	0,854	+3,5% / +8,7% (Artırımlı)
Verimlilik				
Eğitim Süresi	109 dakika	244 dakika	420 dakika	74,0% / 41,9% ↓ (Artırımlı)
GPU Kullanımı	109 dakika	244 dakika	420 dakika	74,0% / 41,9% ↓ (Artırımlı)
Bellek Kullanımı	Kademeli artış	Tam veri seti	Tam veri seti (baştan)	Artırımlı
Operasyonel				
Yeni Veri Ekleme	20-25 dakika	135 dakika (hibrit)	420+ dakika	Artırımlı
Başlangıç Veri	709 görüntü (%20)	709 görüntü (%20)	3,543 görüntü (%100)	Artırımlı
Sürekli Öğrenme	Destekleniyor	Destekleniyor	Desteklenmiyor	Artırımlı
Domain Adaptation	Hibrit yöntem ile	Başarılı	Statik	Artırımlı
Deployment				
Edge Uygunluğu	Yüksek	Yüksek	Düşük	Artırımlı
Model Güncelleme	Hızlı	Orta	Yavaş	Artırımlı

Bu çalışmada, artırımlı öğrenmenin hem performans hem de operasyonel verimlilik açısından tek seferlik eğitime önemli avantajlar sağladığı gösterilmiştir. 5. Turda %3,4 mAP@0.5 performans avantajı ve %74,0 eğitim süresi azaltımı; 6. Tur hibrit ile %6,3 mAP@0.5 performans avantajı ve %41,9 eğitim süresi azaltımı elde edilmiştir. Zaman verimliliği avantajı, tek seferlik eğitimin tüm veri setini her epoch'ta yüklemek ve işlemek zorunda kalmasından kaynaklanmaktadır. Artırımlı öğrenme ise, turlar içinde sadece yeni görselleri içeren değişken sayıda veri setleri kullanıldığı için, toplam hesaplama yükü önemli ölçüde azaltmaktadır. Sürekli öğrenme kapasitesi ve adaptif model güncelleme yeteneği, artırımlı öğrenmeyi gerçek dünya tehdit tespit sistemleri için tercih edilen strateji haline getirmektedir.

6. Turda gösterildiği üzere, alan kayması (domain shift) senaryolarında hibrit yöntem ile katastrofik unutmaya önlenilmekte ve model performansı korunarak iyileştirilebilmektedir. Hibrit yöntemin ek hesaplama maliyeti (%462,5 süre artışı Round 5'e göre), sağladığı performans kazancı (%2,8 genel mAP, %5,0 knife mAP artışı) ve katastrofik unutmanın önlenmesi ile haklı çıkmaktadır. Bu bulgular, sürü robotik sistemlerinde operasyonel dağıtım için artırılmış öğrenme framework'ünün güvenilir, verimli ve adaptif bir çözüm sunduğunu kanıtlamaktadır.

4.2.5. Grad-CAM Tabanlı Açıklanabilirlik Analizi

Model performansının nicel metriklerle değerlendirilmesinin ötesinde, derin öğrenme modellerinin karar verme süreçlerinin anlaşılabilir ve güvenilir olması, özellikle güvenlik-kritik uygulamalarda operasyonel kabul için zorunludur. Bu bölümde, Gradient-weighted Class Activation Mapping (Grad-CAM) tekniği kullanılarak, artırılmış öğrenme sürecinin model dikkat paternleri (attention patterns) üzerindeki etkisi sistematik olarak analiz edilmiştir. Grad-CAM, modelin tespit kararlarını verirken görüntünün hangi bölgelerine odaklandığını görselleştirerek, modelin "neye baktığını" ve kararlarının ne kadar güvenilir olduğunu ortaya koymaktadır (Selvaraju, ve ark, 2019).

Model başarımlarının nicel analizini desteklemek amacıyla, 1.-5. Tur ve one shot modellerinin knife ve pistol sınıfları için örnek Grad-CAM heatmap'leri görsel olarak incelenmiştir. Şekil 4.7-12, her iki sınıf için 1.tur ile 6.tur heatmap karşılaştırmalarını göstermektedir. Şekil 4.13 ise one shot eğitilmiş modelin ısı haritasını göstermektedir. Görsellerde kırmızı bölgeler yüksek aktivasyonu, mavi bölgeler düşük aktivasyonu göstermektedir.



Şekil 4.6. 1.Tur (Baseline) Isı Haritası Dağılımı

Başlangıç modelinin ısı haritaları, bıçak ve tabanca tespitinde oldukça dağınık ve kontrolsüz bir dikkat örüntüsü sergilemektedir. 1. panelde masa üstünde duran bıçak yerine, masanın yüzeyine ve ışık yansımalarına odaklanmış, bıçağın kendisine minimal dikkat gösterilmiştir. 2. panelde insan elinde tutulan tabancaya kısmen odaklanmakla birlikte, sağ alt köşedeki ilgisiz alana da yüksek aktivasyon göstermiştir. 3. panelde masa üstündeki bıçağa neredeyse hiç odaklanmamış, bunun yerine masa yüzeyindeki diğer bölgelere dağınık aktivasyon sergilemiştir. 4. panelde elde tutulan silaha odaklanmış ancak arka plan bölgelerine de önemli ölçüde dikkat dağılmıştır. Bu gözlemler, modelin tehdit nesnelerinin (bıçak ağzı, tetik, namlu gibi) ayırt edici özelliklerini öğrenemediğini ve bunun yerine arka plan dokuları, ışık yansımaları, yüzey özellikleri gibi sahte ilişkilere dayandığını göstermektedir. Model, tehdit nesnelerini bağlamdan ve ilgisiz görsel öğelerden ayırt edememekte, bu da farklı koşullarda düşük genelleme performansı sergileyeceğini işaret etmektedir.



Şekil 4.7. 2.Tur Isı Haritası Dağılımı

2. Tur modelinin ısı haritaları, başlangıç modeline kıyasla kısmi iyileşme göstermekle birlikte, hala yetersiz odaklanma ve arka plan gürültüsü problemleri sergilemektedir. 1. panelde masa üstünde duran bıçağa odaklanma başlamış, ancak aktivasyon bıçak ağzına yoğunlaşmak yerine bıçağın genel silüetine dağılmış ve sol kısımdaki ilgisiz bölgeye de yüksek aktivasyon gösterilmiştir. 2. panelde elde tutulan tabancaya daha belirgin bir odaklanma var, ancak sağ alt köşedeki arka plan bölgesine hala önemli ölçüde dikkat dağılmaktadır. 3. panelde masa üstündeki bıçağa neredeyse hiç odaklanmamış, bunun yerine sağ üstteki ilgisiz alana yüksek aktivasyon göstermiştir. 4. panelde elde tutulan silaha odaklanmış, ancak sağ alt köşedeki ilgisiz bölgeye de yüksek aktivasyon sergilemektedir. Bu gözlemler, modelin tehdit nesnelerine kısmen yönelmeye başladığını ancak tutarsız performans sergilediğini göstermektedir. Özellikle 3. paneldeki başarısızlık, modelin henüz kritik bölgelere (bıçak ağzı, tetik, namlu) hassas odaklanma

yeteneđi kazanamadıđını ortaya koymaktadır. Arka plan bölgelerine devam eden dikkat dağılımı, modelin sahte ilişkilerden tam olarak kurtulmadıđını ve sağlam özellik temsillerini henüz öğrenemediđini işaret etmektedir.



Şekil 4.8. 3.Tur Isı Haritası Dağılımı

3. Tur modelinin ısı haritaları, önceki turlara kıyasla belirgin iyileşme ve daha odaklı dikkat örüntüleri sergilemektedir. 1. panelde masa üstünde duran bıçağı güçlü ve net bir odaklanma görülmekte, aktivasyon bıçağın tüm uzunluğuna yayılmış ve arka plan gürültüsü kalmamıştır. 2. panelde elde tutulan tabancaya yoğun bir odaklanma var, aktivasyon silahın kendisine lokalize olmuş ve önceki turlarda görülen sağ alt köşedeki arka plan dikkat dağılımı büyük ölçüde elimine edilmiştir. 3. panelde masa üstündeki bıçağı güçlü bir odaklanma sergilenmiş, ancak sol taraftaki ilgisiz bölgeye de kısmi aktivasyon gösterilmiştir; buna rağmen bıçağın kendisine olan odaklanma baskın durumdadır. 4. panelde elde tutulan silaha net bir odaklanma var, aktivasyon silahın yapısal özelliklerine yoğunlaşmış ve arka plan gürültüsü kalmamıştır. Bu gözlemler, modelin tehdit nesnelerinin ayırt edici özelliklerini öğrenmeye başladığını ve sahte ilişkilerden kurtularak daha sağlam özellik temsillerine yöneldiğini göstermektedir. Arka plan bölgelerine olan dikkat dağılımının önemli ölçüde azalması ve çoğu panelde tamamen ortadan kalkması, modelin kritik bölgelere odaklanma yeteneğinin geliştiğini işaret etmektedir.



Şekil 4.9. 4.Tur Isı Haritası Dağılımı

4. Tur modelinin ısı haritaları, önceki turlara göre en yüksek odaklanma kalitesi ve hassasiyet sergilemektedir. 1. panelde masa üstünde duran bıçağa son derece keskin ve net bir odaklanma görülmekte, aktivasyon bıçağın tam olarak kendisine lokalize olmuş ve arka plan gürültüsü tamamen ortadan kalkmıştır. 2. panelde elde tutulan tabancaya mükemmel bir odaklanma var, aktivasyon silahın kritik bölgelerine (namlu, tetik, gövde) yoğunlaşmış ve herhangi bir arka plan dikkat dağılımı gözlemlenmemektedir. 3. panelde masa üstündeki bıçağa güçlü ve tutarlı bir odaklanma sergilenmiş, aktivasyon bıçağın tüm uzunluğunu kapsayacak şekilde net bir şekilde belirginleşmiş ve ilgisiz bölgelere aktivasyon gösterilmemiştir. 4. panelde elde tutulan silaha son derece hassas bir odaklanma var, aktivasyon silahın yapısal detaylarına yoğunlaşmış ve arka plan gürültüsü tamamen elimine edilmiştir. Bu gözlemler, modelin tehdit nesnelerinin ayırt edici özelliklerini tam olarak öğrendiğini ve sahte ilişkilerden tamamen kurtularak sağlam, güvenilir özellik temsillerine ulaştığını göstermektedir. Tüm panellerde arka plan gürültüsünün ortadan kalkması ve aktivasyonun yalnızca hedef nesnelere odaklanması, modelin kritik bölgeleri hassas bir şekilde tespit etme yeteneğini kazandığını ve genelleme performansının önemli ölçüde iyileştiğini işaret etmektedir.



Şekil 4.10. 5.Tur Isı Haritası Dağılımı

5. Tur modelinin ısı haritaları, önceki tüm turlara göre en üstün odaklanma kalitesi ve hassasiyet sergilemektedir. 1. panelde masa üstünde duran bıçağa son derece keskin ve net bir odaklanma görülmekte, aktivasyon bıçağın tam olarak kendisine lokalize olmuş ve arka plan gürültüsü tamamen ortadan kalkmıştır. 2. panelde elde tutulan tabancaya mükemmel bir odaklanma var, aktivasyon silahın kritik bölgelerine (namlu, tetik, gövde) yoğunlaşmış ve herhangi bir arka plan dikkat dağılımı gözlemlenmemektedir. 3. panelde masa üstündeki bıçağa son derece hassas ve tutarlı bir odaklanma sergilenmiş, aktivasyon bıçağın tüm uzunluğunu kapsayacak şekilde net bir şekilde belirginleşmiş ancak aktivasyon dağılımında küçük kaymalar gözlemlenmiştir. 4. panelde elde tutulan silaha

son derece hassas bir odaklanma var, aktivasyon silahın yapısal detaylarına ve kritik bölgelerine yoğunlaşmış, arka plan gürültüsü tamamen elimine edilmiş olsa da aktivasyon dağılımında küçük kaymalar mevcuttur. Bu gözlemler, modelin tehdit nesnelerinin ayırt edici özelliklerini tam olarak öğrendiğini ve nesnelere doğrudan odaklandığını göstermektedir. Model sahte ilişkilerden tamamen kurtularak güvenilir özellik temsillerine ulaşmıştır. Tüm panellerde arka plan gürültüsünün tamamen ortadan kalkması ve aktivasyonun yalnızca hedef nesnelere hassas bir şekilde odaklanması, modelin kritik bölgeleri mükemmel bir şekilde tespit etme yeteneğini kazandığını ve genelleme performansının maksimum seviyeye ulaştığını işaret etmektedir.



Şekil 4.12. 6.Tur Hibrit Isı Haritası Dağılımı

6. Tur modelinin ısı haritaları, hibrit bir öğrenme yaklaşımının etkilerini yansıtmaktadır. 1. panelde masa üstünde duran bıçağa keskin ve net bir odaklanma görülmekte, aktivasyon bıçağın tam olarak kendisine lokalize olmuş ve arka plan gürültüsü oluşmamıştır. 2. panelde elde tutulan tabancaya güçlü bir odaklanma var, aktivasyon silahın kritik bölgelerine (namlu, tetik, gövde) yoğunlaşmış ve herhangi bir arka plan dikkat dağılımı gözlemlenmemektedir. 3. panelde masa üstündeki bıçağa hassas ve tutarlı bir odaklanma sergilenmiş, aktivasyon bıçağın tüm uzunluğunu kapsayacak şekilde net bir şekilde belirginleşmiştir. 4. panelde elde tutulan silaha hassas bir odaklanma var, aktivasyon silahın yapısal detaylarına yoğunlaşmış, ancak aktivasyon dağılımında küçük kaymalar mevcuttur. Bu gözlemler, modelin önceki öğrenme aşamalarından gelen bilgiyi koruyarak ve yeni verilerle zenginleştirerek daha kapsamlı bir özellik temsili oluşturduğunu göstermektedir. Model, nesnelere doğrudan odaklanmakta ve sahte ilişkiler kurmamaktadır. Tüm panellerde arka plan gürültüsünün oluşmaması ve aktivasyonun hedef nesnelere odaklanması, modelin kritik bölgeleri tespit etme yeteneğini koruduğunu ve genelleme performansının yüksek seviyede olduğunu işaret etmektedir.



Şekil 4.13. One Shot Isı Haritası Dağılımı

One-Shot modelinin ısı haritaları, tüm eğitim verilerinin tek seferde sunulmasının etkilerini yansıtmaktadır. 1. panelde masa üstünde duran bıçağa net bir odaklanma görülmekte, aktivasyon bıçağın kendisine lokalize olmuş ve arka plan gürültüsü oluşmamıştır. 2. panelde elde tutulan tabancaya güçlü bir odaklanma var, aktivasyon silahın kritik bölgelerine (namlu, tetik, gövde) yoğunlaşmış ve herhangi bir arka plan dikkat dağılımı gözlemlenmemektedir. 3. panelde masa üstündeki bıçağa hassas bir odaklanma sergilenmiş, aktivasyon bıçağın uzunluğunu kapsayacak şekilde belirginleşmiştir. 4. panelde elde tutulan silaha odaklanma var, aktivasyon silahın yapısal detaylarına yoğunlaşmış, ancak aktivasyon dağılımında küçük kaymalar mevcuttur. Bu gözlemler, modelin tüm veri çeşitliliğini aynı anda işleyerek kapsamlı bir özellik temsili oluşturduğunu göstermektedir. Model, nesnelere doğrudan odaklanmakta ve sahte ilişkiler kurmamaktadır. Tüm panellerde arka plan gürültüsünün oluşmaması ve aktivasyonun hedef nesnelere odaklanması, modelin kritik bölgeleri tespit etme yeteneğine sahip olduğunu ve tek seferde sunulan zengin veri çeşitliliğinden etkili bir şekilde öğrendiğini işaret etmektedir.



SONUÇ ve ÖNERİLER

Bu doktora tezi çalışmasında, sürü robotik sistemlerde güvenli iletişim ve adaptif tehdit tespitini birleştiren bütünleşik bir güvenlik çerçevesi geliştirilmiş ve gerçek dünya koşullarında test edilmiştir. Çalışma, literatürde sıklıkla ayrı ayrı ele alınan kriptografik güvenlik, steganografik gizleme, DDoS koruması ve artırılmış öğrenme yaklaşımlarını, kaynak kısıtlı gömülü sistemler üzerinde çalışabilecek şekilde entegre ederek özgün bir katkı sunmaktadır.

Hibrit güvenlik protokolü kapsamında gerçekleştirilen deneysel çalışmalar, Eliptik Eğri Kriptografisi (ECC) tabanlı dijital imzalama ve LSB steganografi kombinasyonunun, eşdeğer güvenlik seviyesindeki RSA tabanlı sistemlere kıyasla kaynak kısıtlı platformlarda belirgin performans avantajları sağladığını göstermiştir. Raspberry Pi 4 platformunda yapılan ölçümlerde, 256-bit ECC sistemi 3072-bit RSA sistemine göre %85 daha az depolama alanı kullanmış, %35,7 daha hızlı işlem süresi sunmuş ve %35,3 daha iyi enerji verimliliği sergilemiştir. Steganografik gömme işlemi, 77,92 dB PSNR ve 1,0000 SSIM değerleri ile görüntü kalitesini tamamen koruyarak, trafik analizi saldırılarına karşı ek bir gizlilik katmanı sağlamıştır. İki aşamalı doğrulama mekanizması (ECC imza + steganografik UUID kontrolü), simüle edilmiş saldırı senaryolarında %100 engelleme başarı oranı ve %0 yanlış pozitif oranı ile mükemmel performans göstermiştir. DDoS koruması için geliştirilen aşamalı IP engelleme mekanizması, ortalama 12,2 ms tespit süresi ile literatürdeki benzer çalışmalara kıyasla 2,3 kat daha hızlı tepki vermiştir.

Çalışmada tehdit tespit modeli olarak YOLOv8-nano mimarisi kullanılmıştır. Bu tercih, çalışmanın başlangıç aşamasında YOLOv8'in mevcut en güncel YOLO versiyonu olması ve nano varyantının kaynak kısıtlı kenar cihazlarda gerçek zamanlı çalışabilme kapasitesi göstermesi nedeniyle yapılmıştır. Çalışma süresince YOLOv9, YOLOv10 ve YOLOv11 gibi daha yeni versiyonlar piyasaya sürülmüş olsa da, bu tez kapsamında tutarlı bir metodolojik çerçeve sağlamak amacıyla YOLOv8 ile devam edilmiştir.

Tehdit tespit ve artırılmış öğrenme bileşeni, YOLOv8-nano mimarisi kullanılarak kaynak kısıtlı kenar cihazlarda gerçek zamanlı nesne tespiti yapabilme kapasitesi göstermiştir. Beş turdan oluşan artırılmış öğrenme süreci sonunda, model performansı başlangıç turuna

kıyasla $mAP@0.5$ metriğinde %71,0 artış göstermiş ve 0,886 değerine ulaşmıştır. Özellikle azınlık sınıfı olan "knife" kategorisinde gözlemlenen %196'lık F1-score iyileşmesi (0,28'den 0,83'e), artırılmış öğrenme stratejisinin sınıf dengesizliği problemini organik olarak çözmedeki etkinliğini ortaya koymuştur. Altıncı turda gerçek dünya verileri ile ilk olarak standart artırılmış öğrenme (yalnızca yeni verilerle eğitim) uygulanmış ve katastrofik unutma tespit edilmiştir ($mAP@0.5$: 0,886'dan 0,873'e düşüş). Ardından önerilen hibrit eğitim yöntemi (kümülatif veri + ağırlık kalıtımı) devreye alınarak katastrofik unutma başarıyla önlenmiş ve model performansı korunmanın ötesinde iyileştirilmiştir ($mAP@0.5$: 0,911, %2,8 artış).

Artırılmış öğrenme yaklaşımının etkinliğini değerlendirmek amacıyla, geleneksel tek seferlik eğitim (one-shot training) yöntemi ile kapsamlı karşılaştırmalı analizler gerçekleştirilmiştir. Bu analizler, artırılmış öğrenme stratejisinin hem performans hem de hesaplama verimliliği açısından önemli avantajlar sağladığını ortaya koymuştur. Performans boyutunda, artırılmış öğrenme ile eğitilen modeller 5. turda %3,4, 6. turda ise %6,3 daha yüksek $mAP@0.5$ değerleri elde etmiştir. Hesaplama maliyeti açısından ise, artırılmış yaklaşım 5. turda %74,0, 6. tur hibrit eğitimde %41,9 daha az işlem süresi gerektirerek, kaynak kısıtlı sistemlerde önemli bir verimlilik avantajı sunmuştur. Bu bulgular, artırılmış öğrenmenin yalnızca hesaplama tasarrufu sağlamakla kalmayıp, aynı zamanda model performansını da iyileştirebildiğini göstermektedir.

Model davranışının daha derinlemesine anlaşılması amacıyla, Grad-CAM tabanlı açıklanabilirlik analizi yöntemi kullanılarak modelin karar verme süreçleri görselleştirilmiştir. Bu analizler, artırılmış öğrenme turları boyunca modelin dikkat mekanizmasındaki sistematik gelişimi ortaya koymuştur. İlk turlarda modelin arka plan bölgelerine ve ilgisiz görsel öğelere dağınık bir odaklanma sergilediği gözlemlenirken, ilerleyen turlarda tehdit nesnelerinin kritik özelliklerine (bıçak ağzı, tetik, namlu gibi) hassas ve tutarlı bir odaklanma geliştirdiği tespit edilmiştir. Bu bulgular, artırılmış öğrenme sürecinin modelin yalnızca nicel performansını değil, aynı zamanda özellik temsillerinin kalitesini de iyileştirdiğini göstermektedir.

Gerçek dünya uyarlaması olarak geliştirilen Intelligent Mission Cruiser (iMC) platformu ve Ground Control Station (Ground CS) yazılımı, önerilen güvenlik çerçevesinin operasyonel uygulanabilirliğini göstermiştir. FastDDS tabanlı iletişim altyapısı, düşük

gecikme ve yüksek güvenilirlik sağlarken, modüler yazılım mimarisi sistemin gelecekte çoklu iMC desteğine ölçeklenebilmesini mümkün kılmaktadır. Ancak, mevcut çalışma kapsamında yalnızca bir tam donanımlı iMC ve bir test cihazı ile validasyon gerçekleştirilmiş olup, gerçek sürü davranışlarının gözlemlenmesi için çoklu düğüm testlerine ihtiyaç duyulmaktadır.

Çalışmanın temel kısıtları şu şekilde özetlenebilir: İlk olarak, tehdit tespit modeli yalnızca iki sınıf (pistol ve knife) üzerinde eğitilmiş olup, gerçek dünya güvenlik senaryolarında karşılaşılabilecek daha geniş tehdit yelpazesini kapsamamaktadır. İkinci olarak, artırılmış öğrenme sürecinde kullanılan veri seti boyutu (5,064 görüntü) ve artırılmış tur sayısı (6 tur) sınırlıdır; uzun vadeli sürekli öğrenme senaryolarında katastrofik unutma probleminin nasıl evrimleşeceği daha fazla araştırma gerektirmektedir. Üçüncü olarak, 1-5. turlar boyunca yalnızca yeni verilerle yapılan artırılmış öğrenmede katastrofik unutma gözlemlenmemiştir; bu durum kısmen veri setinin içsel homojenliğinden kaynaklanabileceği gibi, kullanılan kaliteye dayalı veri seçim stratejisinin ($IoU \geq 0,5$ ve sınıf eşleşme kriterleri) de etkisi olabilir. Bu strateji, her turda modelin yüksek güven skorlu ve doğru lokalizasyona sahip örnekleri seçmesini sağlamış, bu da modelin kademeli olarak benzer görsel özelliklere sahip verilerle beslenmesine yol açmış olabilir. Sonuç olarak, model önceki bilgilerle çelişmeyen, uyumlu örneklerle eğitildiği için katastrofik unutma ortaya çıkmamıştır. Ancak 6. turda tamamen farklı bir domain'den (gerçek dünya kameraları) gelen verilerle karşılaşıldığında, domain shift nedeniyle katastrofik unutma problemi ortaya çıkmıştır. Bu bulgu, artırılmış öğrenme sürecinde veri seçim stratejisinin ve domain benzerliğinin katastrofik unutma üzerinde kritik etkiye sahip olduğunu göstermektedir. Farklı veri seçim stratejileri (rastgele örnekleme, belirsizlik tabanlı seçim) veya daha heterojen veri akışlarında katastrofik unutmanın daha erken turlarda ortaya çıkabileceği göz önünde bulundurulmalıdır. Dördüncü olarak, hibrit güvenlik protokolünün performans değerlendirmesi kontrollü laboratuvar ortamında yapılmıştır; ancak protokolün tüm işlem adımları (ECC imzalama, steganografik gömme, UUID doğrulama) kenar cihazda gerçekleştiği için gerçek dünya ağ koşullarında (değişken gecikme, paket kaybı, bant genişliği dalgalanmaları) elde edilen zamansal ölçümlerde ciddi değişiklikler beklenmemektedir. Bununla birlikte, sistemin farklı ağ topolojilerinde ve yüksek yük senaryolarında kapsamlı olarak test edilmesi, operasyonel güvenilirliğin doğrulanması açısından önemlidir. Beşinci olarak, DDoS koruması

mekanizması uygulama katmanı saldırılarına odaklanmış olup, ağ katmanı ve transport katmanı saldırılarına karşı ek savunma mekanizmalarına ihtiyaç duyulmaktadır. Altıncı olarak, sistem şu an tek bir iMC ile operasyonel olarak çalışmakta olup, çoklu düğüm koordinasyonu, sürü içi iletişim güvenliği ve ölçeklenebilirlik testleri henüz gerçekleştirilmemiştir.

Gelecek çalışmalar için aşağıdaki öneriler sunulmaktadır: Tehdit tespit kapasitesinin genişletilmesi amacıyla, ateşli silahlar, kesici aletler, patlayıcılar, şüpheli davranış kalıpları ve araç anomalileri gibi daha geniş bir tehdit kategorisi spektrumunu kapsayan veri setleri ile model eğitimi gerçekleştirilmelidir. Artırımlı öğrenme stratejisinin uzun vadeli performansını ve katastrofik unutma davranışını daha kapsamlı değerlendirmek için, farklı kaynaklardan toplanmış heterojen veri setleri, 20-30 tur gibi daha uzun öğrenme döngüleri ve binlerce görüntü içeren büyük ölçekli veri akışları ile testler yapılmalıdır. Özellikle, farklı veri seçim stratejilerinin (rastgele örnekleme, belirsizlik tabanlı aktif öğrenme, çeşitlilik tabanlı seçim) katastrofik unutma üzerindeki etkisi sistematik olarak araştırılmalı ve domain shift senaryolarında hibrit eğitim yönteminin etkinliği farklı uygulama alanlarında doğrulanmalıdır. Kuantum bilgisayar tehdidine karşı sistem güvenliğini güçlendirmek amacıyla, NIST tarafından standartlaştırılan kuantum-sonrası kriptografi algoritmalarının (CRYSTALS-Kyber, CRYSTALS-Dilithium) hibrit protokole entegrasyonu araştırılmalıdır. Çoklu iMC düğümleri ile gerçek sürü davranışlarının test edilmesi, düğümler arası güvenli iletişim protokollerinin geliştirilmesi ve sürü koordinasyonu algoritmalarının entegrasyonu, sistemin operasyonel olgunluğunu artıracaktır. Hibrit güvenlik protokolünün gerçek dünya ağ koşullarında (değişken gecikme, paket kaybı, bant genişliği kısıtlamaları, çoklu düğüm trafiği) kapsamlı performans testleri yapılmalı ve protokolün farklı ağ topolojilerindeki davranışı sistematik olarak değerlendirilmelidir. Açıklanabilir yapay zeka bileşeninin geliştirilmesi için, Grad-CAM'in yanı sıra SHAP, LIME ve dikkat mekanizması gibi tamamlayıcı XAI yöntemlerinin entegrasyonu, model kararlarının daha kapsamlı anlaşılmasını sağlayacaktır. Enerji verimliliğini daha da artırmak için, model kuantizasyonu (8-bit veya 4-bit), model budama ve bilgi damıtma gibi model optimizasyon tekniklerinin uygulanması araştırılmalıdır. YOLOv9, YOLOv10, YOLOv11 ve gelecekte çıkacak daha güncel YOLO versiyonları ile karşılaştırmalı performans analizleri yapılması, önerilen artırımlı öğrenme stratejisinin farklı model mimarileri üzerindeki etkinliğini değerlendirmek ve potansiyel performans iyileştirmelerini ortaya koymak açısından

önemli bir araştırma yönü oluşturmaktadır. Son olarak, sistemin farklı çevresel koşullarda (gece görüşü, yağmur, sis, kar) performansını değerlendirmek ve çok-modlu sensör füzyonu (kamera + termal görüntüleme + radar) ile tespit doğruluğunu artırmak, gelecek çalışmaların odak noktalarını oluşturmalıdır.

Sonuç olarak, bu tez çalışması sürü robotik sistemlerde güvenli iletişim ve adaptif tehdit tespitinin entegre edilebileceğini deneysel olarak göstermiş ve kaynak kısıtlı platformlarda uygulanabilir bir çözüm sunmuştur. Önerilen hibrit güvenlik çerçevesi, hem akademik literatüre metodolojik katkı sağlamakta hem de gerçek dünya güvenlik uygulamaları için pratik bir temel oluşturmaktadır. Gelecek çalışmalarla birlikte, bu çerçevenin daha geniş tehdit spektrumlarında, daha büyük sürü boyutlarında ve daha zorlu operasyonel koşullarda test edilmesi, sürü robotik güvenlik alanında önemli ilerlemeler sağlayacaktır.



KAYNAKLAR

- Adebayo, J., Gilmer, J., Muelly, M., Goodfellow, I., Hardt, M., ve Kim, B. (2018). Sanity Checks for Saliency Maps. *Google Brain*.
- Ahn, H., Choi, H., Kang, M., ve Moon, S. (2019). Learning-Based Anomaly Detection and Monitoring for Swarm Drone Flights. *Applied Sciences*.
- Ahsan, K., ve Kundur, D. (2002). Practical Data Hiding in TCP/IP. *University of Toronto*.
- Albani, D., Nardi, D., ve Trianni, V. (2017). Field Coverage and Weed Mapping by UAV Swarms. *Google Brain*.
- Almomani, I., Alkhayer, A., ve El-Shafai, W. (2022). A Crypto-Steganography Approach for Hiding Ransomware within HEVC Streams in Android IoT Devices. *Sensors*.
- Ansari, S. A., & Ali, S. (2025). A systematic review of lightweight cryptographic schemes for security and privacy in IoT. *Discover Computing*, 28(1), 266.
- Apau, R., Asante, M., Twum, F., Ben Hayfron-Acquah, J., & Peasah, K. O. (2024). Image steganography techniques for resisting statistical steganalysis attacks: A systematic literature review. *PLOS ONE*, 19(9), 1-47.
- Bhattacharyya, S., Banerjee, I., & Sanyal, G. (2010). A Novel Approach of Secure Text-Based Steganography Model using Word Mapping Method (WMM). *International Journal of Computer and Information Engineering*, 4(2), 96-103.
- Barker, E. (2016). Recommendation for Key Management: Part 1 – General (NIST Special Publication 800-57 Part 1, Revision 4). *National Institute of Standards and Technology*.
- Barker, E. (2020). Recommendation for Key Management: Part 1 – General (NIST Special Publication 800-57 Part 1, Revision 5). *National Institute of Standards and Technology*.
- Beaulieu, R., Shors, D., Smith, J., Treatman-Clark, S., Weeks, B., ve Wingers, L. (2015). The SIMON and SPECK Lightweight Block Ciphers. *National Security Agency*.
- Becher, M., Freiling, F. C., Hoffmann, J., Holz, T., Uellenbeck, S., ve Wolf, C. (2006). Mobile Security Catching Up? Revealing the Nuts and Bolts of the Security of

- Mobile Devices. *Proceedings of the Third International Conference on Security in Pervasive Computing (SPC 2006)*, Lecture Notes in Computer Science, 3934.
- Bengio, Y., Louradour, J., Collobert, R., ve Weston, J. (2009). Curriculum Learning. *Proceedings of the 26th International Conference on Machine Learning (ICML)*.
- Bernstein, D. J. (2005). The Poly1305-AES Message-Authentication Code. *Fast Software Encryption (FSE 2005)*, Lecture Notes in Computer Science.
- Bernstein, D. J. (2008). ChaCha, a Variant of Salsa20. *University of Illinois at Chicago*.
- Bhuyan, M. H., Bhattacharyya, D. K., ve Kalita, J. K. (2014). Network Anomaly Detection: Methods, Systems and Tools. *IEEE Communications Surveys & Tutorials*, 16(1), 303-336.
- Biggio, B., ve Roli, F. (2018). Wild Patterns: Ten Years After the Rise of Adversarial Machine Learning. *Pattern Recognition*, 84, 317-331.
- Bochkovskiy, A., Wang, C.-Y., ve Liao, H.-Y. M. (2020). YOLOv4: Optimal Speed and Accuracy of Object Detection. *arXiv preprint arXiv:2004.10934*.
- Bogdanov, A., Knudsen, L. R., Leander, G., Paar, C., Poschmann, A., Robshaw, M. J. B., Seurin, Y., ve Vikkelsoe, C. (2007). PRESENT: An Ultra-Lightweight Block Cipher. *Cryptographic Hardware and Embedded Systems (CHES 2007)*, Lecture Notes in Computer Science.
- Bogdanov, A., Knežević, M., Leander, G., Toz, D., Varıcı, K., ve Verbauwhede, I. (2011). A Lightweight Hash Function. *Cryptographic Hardware and Embedded Systems (CHES 2011)*, Lecture Notes in Computer Science, 6917.
- Bonomi, F., Milito, R., Zhu, J., ve Addepalli, S. (2012). Fog Computing and Its Role in the Internet of Things. *Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing*, 13-16.
- Boroumand, M., Chen, M., ve Fridrich, J. (2019). Deep Residual Network for Steganalysis of Digital Images. *IEEE Transactions on Information Forensics and Security*, 14(5), 1181-1193.

- Braga, R., Mota, E., ve Passito, A. (2010). Lightweight DDoS Flooding Attack Detection Using NOX/OpenFlow. *Proceedings of the 35th Annual IEEE Conference on Local Computer Networks (LCN)*.
- Brambilla, M., Ferrante, E., Birattari, M., ve Dorigo, M. (2013). Swarm Robotics: A Review from the Swarm Engineering Perspective. *Swarm Intelligence*, 7(1), 1-41.
- Buczak, A. L., & Guven, E. (2016). A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection. *IEEE Communications Surveys & Tutorials*, 18(2), 1153-1176.
- Cabuk, S., Brodley, C. E., & Shields, C. (2004). IP Covert Timing Channels: Design and Detection. *Proceedings of the 11th ACM Conference on Computer and Communications Security*.
- Campion, M., Ranganathan, P., & Faruque, S. (2018). UAV Swarm Communication and Control Architectures: A Review. *Journal of Unmanned Vehicle Systems*.
- Chan, C. K., & Cheng, L. M. (2004). Hiding Data in Images by Simple LSB Substitution. *Pattern Recognition*, 37, 469-474.
- Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly Detection: A Survey. *ACM Computing Surveys*, 41(3), 1-58.
- Cheddad, A., Condell, J., Curran, K., & Mc Kevitt, P. (2010). Digital Image Steganography: Survey and Analysis of Current Methods. *Signal Processing*, 90(3), 727-752.
- Chung, S. J., Paranjape, A. A., Dames, P., Shen, S., & Kumar, V. (2018). A Survey on Aerial Swarm Robotics. *IEEE Transactions on Robotics*, 34(4), 837-855.
- Clark, A. (2023). Pillow Documentation. *Pillow (PIL Fork) Documentation*.
- Cohen, F., et al. (2005). *Digital Forensic Evidence Examination*.
- Corsaro, A., & Schmidt, D. C. (2012). Distributed Systems and System of Systems. *Encyclopedia of Sustainability Science and Technology*.
- Cox, I. J., Miller, M. L., Bloom, J. A., Fridrich, J., & Kalker, T. (2007). *Digital Watermarking and Steganography* (2nd ed.). Morgan Kaufmann Publishers.

- Cvejic, N., & Seppanen, T. (2004). Increasing Robustness of LSB Audio Steganography by Reduced Distortion LSB Coding. *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*.
- Daemen, J., & Rijmen, V. (2020). *The Design of Rijndael: The Advanced Encryption Standard (AES)* (2nd ed.). Springer-Verlag.
- Dallaire-Demers, P.-L., Doyle, W., & Foo, T. (2025). *Brace for impact: ECDLP challenges for quantum cryptanalysis*. *arXiv preprint arXiv:2508.14011*.
- Diro, A. A., & Chilamkurti, N. (2018). Distributed Attack Detection Scheme Using Deep Learning Approach for Internet of Things. *Future Generation Computer Systems*, 82, 761-768.
- Dong, P., & Chen, Q. (2017). *LiDAR Remote Sensing and Applications*. CRC Press.
- Dorigo, M., Birattari, M., & Brambilla, M. (2014). Swarm Robotics. *Scholarpedia*, 9(1), 1463.
- Douceur, J. R. (2002). The Sybil Attack. *Proceedings of the 1st International Workshop on Peer-to-Peer Systems (IPTPS)*.
- Dudek, G., & Jenkin, M. (2010). *Computational Principles of Mobile Robotics* (2nd ed.). Cambridge University Press.
- Dunbabin, M., & Marques, L. (2012). Robots for Environmental Monitoring: Significant Advancements and Applications. *IEEE Robotics & Automation Magazine*, 19(1), 24-39.
- Dumitrescu, S., Wu, X., & Wang, Z. (2003). Detection of LSB Steganography via Sample Pair Analysis. *IEEE Transactions on Signal Processing*, 51(7), 1995-2007.
- Durrant-Whyte, H., & Henderson, T. C. (2008). Multisensor Data Fusion. In B. Siciliano & O. Khatib (Eds.), *Springer Handbook of Robotics* (pp. 585-610). Springer.
- Eastlake, D., Schiller, J., & Crocker, S. (2005). *Randomness Requirements for Security* (RFC 4086). Internet Engineering Task Force.
- Eddy, W. (2007). *TCP SYN Flooding Attacks and Common Mitigations* (RFC 4987). Internet Engineering Task Force.

- Eisenbarth, T., Kasper, T., Moradi, A., Paar, C., Salmasizadeh, M., & Shalmani, M. T. M. (2012). Physical Cryptanalysis of KeeLoq Code Hopping Applications. In A. Mitrokotsa & S. Vaudenay (Eds.), *Progress in Cryptology - AFRICACRYPT 2012* (Lecture Notes in Computer Science, Vol. 7374). Springer.
- eProxima. (2023). *Fast DDS Documentation*. Retrieved from <https://fast-dds.docs.eprosima.com>
- Fagiolini, A., Dini, G., & Bicchi, A. (2014). Distributed Intrusion Detection for the Security of Industrial Cooperative Robotic Systems. *IFAC Proceedings Volumes*, 47(3), 9923-9928.
- Fernandez-Gago, M. C., Roman, R., & Lopez, J. (2007). A Survey on the Applicability of Trust Management Systems for Wireless Sensor Networks. *Proceedings of the Third International Workshop on Security, Privacy and Trust in Pervasive and Ubiquitous Computing*.
- Ferrag, M. A., Maglaras, L., Moschoyiannis, S., & Janicke, H. (2020). Deep Learning for Cyber Security Intrusion Detection: Approaches, Datasets, and Comparative Study. *Journal of Information Security and Applications*, 50, 102419.
- Finn, C., Abbeel, P., & Levine, S. (2017). Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks. *Proceedings of the 34th International Conference on Machine Learning (ICML)*, 1126-1135.
- French, R. M. (1999). Catastrophic Forgetting in Connectionist Networks. *Trends in Cognitive Sciences*, 3(4), 128-135.
- Frew, E. W., & Brown, T. X. (2008). Networking Issues for Small Unmanned Aircraft Systems. *Journal of Intelligent & Robotic Systems*, 54, 21-37.
- Fridrich, J., & Goljan, M. (2002). Practical Steganalysis of Digital Images – State of the Art. *Proceedings of SPIE - The International Society for Optical Engineering*, 4675, 1-13.
- National Institute of Standards and Technology. (2013). *Digital Signature Standard (DSS)* (FIPS PUB 186-4). U.S. Department of Commerce.

- Fridrich, J. (2009). *Steganography in Digital Media: Principles, Algorithms, and Applications*. Cambridge University Press.
- Fridrich, J., Goljan, M., & Du, R. (2001). Reliable Detection of LSB Steganography in Color and Grayscale Images. *Proceedings of the ACM Workshop on Multimedia and Security*, 27-30.
- Fridrich, J., Goljan, M., & Hoge, D. (2001). Attacking the OutGuess. *Proceedings of SPIE - The International Society for Optical Engineering*, 4675, 3-6.
- Gama, J., Žliobaitė, I., Bifet, A., Pechenizkiy, M., & Bouchachia, A. (2014). A Survey on Concept Drift Adaptation. *ACM Computing Surveys*, 46(4), Article 44.
- Ganeriwal, S., & Srivastava, M. B. (2004). Reputation-Based Framework for High Integrity Sensor Networks. *Proceedings of the 2nd ACM Workshop on Security of Ad Hoc and Sensor Networks*, 66-77.
- Ge, Z., Liu, S., Wang, F., Li, Z., & Sun, J. (2021). YOLOX: Exceeding YOLO Series in 2021. *arXiv preprint arXiv:2107.08430*.
- Gerkey, B. P., & Mataric, M. J. (2002). Sold!: Auction Methods for Multirobot Coordination. *IEEE Transactions on Robotics and Automation*, 18(5), 758-768.
- Giffin, J., Greenstadt, R., Litwack, P., & Tibbetts, R. (2002). Covert Messaging through TCP Timestamps. *Proceedings of the 2nd International Workshop on Privacy Enhancing Technologies*, 194-208.
- Gil, S., Kumar, S., Mazumder, M., Katabi, D., & Rus, D. (2017). Guaranteeing Spoof-Resilient Multi-Robot Networks. *Autonomous Robots*, 41(6), 1383-1400.
- Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 580-587.
- Girshick, R. (2015). Fast R-CNN. *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 1440-1448.
- Goodfellow, I. J., Shlens, J., & Szegedy, C. (2015). Explaining and Harnessing Adversarial Examples. *Proceedings of the International Conference on Learning Representations (ICLR)*.

- Goodrich, M. A., & Schultz, A. C. (2007). Human–Robot Interaction: A Survey. *Foundations and Trends in Human–Computer Interaction*, 1(3), 203-275.
- Gu, T., Dolan-Gavitt, B., & Garg, S. (2017). BadNets: Identifying Vulnerabilities in the Machine Learning Model Supply Chain. *Proceedings of the IEEE Access*, 1-12.
- Gubbi, J., Buyya, R., Marusic, S., & Palaniswami, M. (2013). Internet of Things (IoT): A Vision, Architectural Elements, and Future Directions. *Future Generation Computer Systems*, 29(7), 1645-1660.
- Guo, J., Peyrin, T., Poschmann, A., & Robshaw, M. (2011). The LED Block Cipher. In P. Rogaway (Ed.), *Advances in Cryptology – CRYPTO 2011* (Lecture Notes in Computer Science, Vol. 6841, pp. 326-343). Springer.
- Gupta, V., Millard, M., Fung, S., Zhu, Y., Gura, N., Eberle, H., & Shantz, S. C. (2005). Sizzle: A Standards-Based End-to-End Security Architecture for the Embedded Internet. *Pervasive and Mobile Computing*, 1(4), 425-445.
- Gura, N., Patel, A., Wander, A., Eberle, H., & Shantz, S. C. (2004). Comparing Elliptic Curve Cryptography and RSA on 8-bit CPUs. In M. Joye & J.-J. Quisquater (Eds.), *Cryptographic Hardware and Embedded Systems – CHES 2004* (Lecture Notes in Computer Science, Vol. 3156, pp. 119-132). Springer.
- Hall, D. L., & Llinas, J. (2001). *Handbook of Multisensor Data Fusion*. CRC Press.
- Hamann, H. (2018). *Swarm Robotics: A Formal Approach*. Springer International Publishing.
- Hampapur, A., Brown, L., Connell, J., Ekin, A., Haas, N., Lu, M., Merkl, H., & Pankanti, S. (2005). Smart Video Surveillance: Exploring the Concept of Multiscale Spatiotemporal Tracking. *IEEE Signal Processing Magazine*, 22(2), 38-51.
- Hankerson, D., Menezes, A. J., & Vanstone, S. (2004). *Guide to Elliptic Curve Cryptography*. Springer-Verlag.
- He, K., Zhang, X., Ren, S., & Sun, J. (2015). Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(9), 1904-1916.

- He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask R-CNN. *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2961-2969.
- Higgins, F., Tomlinson, A., & Martin, K. M. (2009). Survey on Security Challenges for Swarm Robotics. *Proceedings of the Fifth International Conference on Autonomic and Autonomous Systems*, 307-312.
- Hinton, G., Vinyals, O., & Dean, J. (2015). Distilling the Knowledge in a Neural Network. *arXiv preprint arXiv:1503.02531*.
- Holzinger, A. (2016). Interactive Machine Learning for Health Informatics: When Do We Need the Human-in-the-Loop? *Brain Informatics*, 3(2), 119-131.
- Horé, A., & Ziou, D. (2010). Image Quality Metrics: PSNR vs. SSIM. *Proceedings of the 20th International Conference on Pattern Recognition (ICPR)*, 2366-2369.
- Hou, S., Pan, X., Loy, C. C., Wang, Z., & Lin, D. (2019). Learning a Unified Classifier Incrementally via Rebalancing. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 831-839.
- Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., & Adam, H. (2017). MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. *arXiv preprint arXiv:1704.04861*.
- Hughes, A., & Drury, B. (2019). *Electric Motors and Drives: Fundamentals, Types and Applications* (5th ed.). Newnes/Elsevier.
- Hussain, M., Wahab, A. W. A., Idris, Y. I. B., Ho, A. T. S., & Jung, K.-H. (2018). Image Steganography in Spatial Domain: A Survey. *Signal Processing: Image Communication*, 65, 46-66.
- Huynh-Thu, Q., & Ghanbari, M. (2008). Scope of Validity of PSNR in Image/Video Quality Assessment. *Electronics Letters*, 44(13), 800-801.
- Jang, K., Srivastava, V., Baksi, A., Sarkar, S., & Seo, H. (2025). New quantum cryptanalysis of binary elliptic curves (Extended version). *Cryptology ePrint Archive, Paper 2025/017*.

- Jain, S., & Wallace, B. C. (2019). Attention is not Explanation. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, 3543-3556.
- Jebrane, J., Chhaybi, A., Lazaar, S., & Nitaj, A. (2025). Elliptic Curve Cryptography with Machine Learning. *Cryptography*, 9(1), 3.
- Jocher, G., Chaurasia, A., & Qiu, J. (2023). *Ultralytics YOLOv8* [Computer software]. Retrieved from <https://docs.ultralytics.com>
- Johar, L., & Al-Ali, B. (2025). *Enhancing LSB Steganography using Chaotic Maps* [Master's thesis, Blekinge Institute of Technology].
- Johnson, D., Menezes, A., & Vanstone, S. (2001). The Elliptic Curve Digital Signature Algorithm (ECDSA). *International Journal of Information Security*, 1(1), 36-63.
- Johnston, S. J., & Cox, S. J. (2017). The Raspberry Pi: A Technology Disrupter, and the Enabler of Dreams. *Electronics*, 6(3), 51.
- Kadhim, I. J., Premaratne, P., Vial, P. J., & Halloran, B. (2019). Comprehensive Survey of Image Steganography: Techniques, Evaluations, and Trends in Future Research. *Neurocomputing*, 335, 299-326.
- Kaplan, E. D., & Hegarty, C. J. (Eds.). (2017). *Understanding GPS/GNSS: Principles and Applications* (3rd ed.). Artech House.
- Karlof, C., & Wagner, D. (2003). Secure Routing in Wireless Sensor Networks: Attacks and Countermeasures. *Ad Hoc Networks*, 1(2-3), 293-315.
- Ker, A. D. (2005). Steganalysis of LSB Matching in Grayscale Images. *IEEE Signal Processing Letters*, 12(6), 441-444.
- Kfoury, E. F., Saab, J., Younes, P., & Achkar, R. (2019). A Self-Organizing Map Approach for Intrusion Detection in a UAV Network. *International Journal of Interdisciplinary Telecommunications and Networking*, 11(1), 23-44.
- Khalastchi, E., & Kalech, M. (2019). Fault Detection and Diagnosis in Multi-Robot Systems: A Survey. *Sensors*, 19(18), 4019.

- Khraisat, A., Gondal, I., Vamplew, P., & Kamruzzaman, J. (2019). Survey of Intrusion Detection Systems: Techniques, Datasets and Challenges. *Cybersecurity*, 2(1), 20.
- Kim, S., & Ahn, H. (2025). Integrating Risk Assessment and Deep-Learning-Based Anomaly Detection for Efficient Risk-Informed Decision Making for Swarm Flight Robot Systems. *IEEE Access*, 13, 8847-8862.
- Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A. A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., Hassabis, D., Clopath, C., Kumaran, D., & Hadsell, R. (2017). Overcoming Catastrophic Forgetting in Neural Networks. *Proceedings of the National Academy of Sciences*, 114(13), 3521-3526.
- Koblitz, N., Menezes, A., & Vanstone, S. (2000). The State of Elliptic Curve Cryptography. *Designs, Codes and Cryptography*, 19(2-3), 173-193.
- Kutlu, Z., & Emiroğlu, B. G. (2025). Image-Based Threat Detection and Explainability Investigation Using Incremental Learning and Grad-CAM with YOLOv8. *Computers*, 13(11), 305.
- Kuznetsov, O., Frontoni, E., & Chernov, K. (2024). Beyond traditional steganography: enhancing security and performance with spread spectrum image steganography. *Applied Intelligence*, 54(7), 5253-5277.
- LaValle, S. M. (2006). *Planning Algorithms*. Cambridge University Press. Retrieved from <http://planning.cs.uiuc.edu/>
- Leach, P., Mealling, M., & Salz, R. (2005). *A Universally Unique IDentifier (UUID) URN Namespace* (RFC 4122). Internet Engineering Task Force.
- Li, Z., & Hoiem, D. (2017). Learning without Forgetting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(12), 2935-2947.
- Li, N., Ho, C. P., Xue, J., Lim, L. W., Chen, G., Fu, Y. H., & Lee, L. Y. T. (2022). A Progress Review on Solid-State LiDAR and Nanophotonics-Based LiDAR Sensors. *Laser & Photonics Reviews*, 16(11), 2100511.
- Li, X., Chen, L., Jia, J., Qin, Z., & Fu, Z. (2025). A Lightweight Image Forgery Prevention Scheme for IoT Using GAN-Based Steganography. *IEEE Transactions on Industrial Informatics*, 21(1), 167-178.

- Liao, G., Yang, J., Shao, W., & Huang, Y. (2025). A framework for designing provably secure steganography. In *Proceedings of the 34th USENIX Security Symposium*. USENIX Association.
- Lin, T.-Y., Dollár, P., Girshick, R., He, K., Hariharan, B., & Belongie, S. (2017). Feature Pyramid Networks for Object Detection. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2117-2125.
- Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., & Zitnick, C. L. (2014). Microsoft COCO: Common Objects in Context. *Proceedings of the European Conference on Computer Vision (ECCV)*, 740-755.
- Liu, A., & Ning, P. (2008). TinyECC: A Configurable Library for Elliptic Curve Cryptography in Wireless Sensor Networks. *Proceedings of the 7th International Conference on Information Processing in Sensor Networks (IPSN)*, 245-256.
- Liu, S., Qi, L., Qin, H., Shi, J., & Jia, J. (2018). Path Aggregation Network for Instance Segmentation. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 8759-8768.
- Loshchilov, I., & Hutter, F. (2019). Decoupled Weight Decay Regularization. *Proceedings of the International Conference on Learning Representations (ICLR)*.
- Lundberg, S. M., & Lee, S.-I. (2017). A Unified Approach to Interpreting Model Predictions. *Advances in Neural Information Processing Systems (NeurIPS)*, 30, 4765-4774.
- Luo, J., & Oracevic, A. (2024). Adaptive Machine Learning for Efficient Anomaly Detection in Autonomous UAVs Swarm Operations. *Proceedings of the 2024 IEEE International Conference on Robotics and Automation (ICRA)*, 1-7.
- Luo, W., Huang, F., & Huang, J. (2010). Edge Adaptive Image Steganography Based on LSB Matching Revisited. *IEEE Transactions on Information Forensics and Security*, 5(2), 201-214.
- Lyu, L., Yu, H., & Yang, Q. (2020). Threats to Federated Learning: A Survey. *arXiv preprint arXiv:2003.02133*.

- Madry, A., Makelov, A., Schmidt, L., Tsipras, D., & Vladu, A. (2019). Towards Deep Learning Models Resistant to Adversarial Attacks. *Proceedings of the International Conference on Learning Representations (ICLR)*.
- Malan, D. J., Welsh, M., & Smith, M. D. (2004). A Public-Key Infrastructure for Key Distribution in TinyOS Based on Elliptic Curve Cryptography. *Proceedings of the 1st IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks (SECON)*, 71-80.
- Mallya, A., & Lazebnik, S. (2018). PackNet: Adding Multiple Tasks to a Single Network by Iterative Pruning. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 7765-7773.
- Mandal, P. C., Mukherjee, I., Paul, G., & Chatterji, B. N. (2022). Digital image steganography: A literature survey. *Information Sciences*, 609, 1451-1488.
- Mankins, J. C. (2009). Technology Readiness Assessments: A Retrospective. *Acta Astronautica*, 65(9-10), 1216-1223.
- McKay, K. A., Bassham, L., Sönmez Turan, M., & Mouha, N. (2017). *Report on Lightweight Cryptography* (NISTIR 8114). National Institute of Standards and Technology. <https://doi.org/10.6028/NIST.IR.8114>
- McMahan, H. B., Moore, E., Ramage, D., Hampson, S., & Agüera y Arcas, B. (2017). Communication-Efficient Learning of Deep Networks from Decentralized Data. *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 1273-1282.
- Mei, Y., Lu, Y.-H., Hu, Y. C., & Lee, C. S. G. (2004). Energy-Efficient Motion Planning for Mobile Robots. *Proceedings of the 2004 IEEE International Conference on Robotics and Automation (ICRA)*, 4344-4349.
- Microsoft. (2023). *ASP.NET Core Documentation*. Microsoft Learn. Retrieved from <https://learn.microsoft.com/tr-tr/aspnet/core>
- Microsoft Corporation. (2023). *SQL Server Documentation*. Microsoft Docs. Retrieved from <https://docs.microsoft.com/en-us/sql/sql-server>

- Miele, A. (2015). *On the Analysis of Public-Key Cryptologic Algorithms* (Doctoral dissertation, École Polytechnique Fédérale de Lausanne). <https://doi.org/10.5075/epfl-thesis-6603>
- Mielikainen, J. (2006). LSB Matching Revisited. *IEEE Signal Processing Letters*, 13(5), 285-287.
- Mirkovic, J., & Reiher, P. (2004). A Taxonomy of DDoS Attack and DDoS Defense Mechanisms. *ACM SIGCOMM Computer Communication Review*, 34(2), 39-53.
- Mishra, B., & Kertesz, A. (2020). The Use of MQTT in M2M and IoT Systems: A Survey. *IEEE Access*, 8, 201071-201086.
- Murdoch, S. J., & Lewis, S. (2005). Embedding Covert Channels into TCP/IP. In J. Fridrich (Ed.), *Information Hiding: 7th International Workshop* (pp. 247-261). Springer.
- Murphy, R. R., Tadokoro, S., Nardi, D., Jacoff, A., Fiorini, P., Choset, H., & Erkmen, A. M. (2008). Search and Rescue Robotics. In B. Siciliano & O. Khatib (Eds.), *Springer Handbook of Robotics* (pp. 1151-1173). Springer.
- Murphy, R. R. (2000). *Introduction to AI Robotics*. MIT Press.
- NATO. (2012). *STANAG 4586 – Standard Interfaces of UAV Control System (UCS) for NATO UAV Interoperability* (Edition 3). North Atlantic Treaty Organization.
- Nguyen, P. Q., & Shparlinski, I. E. (2003). The Insecurity of the Elliptic Curve Digital Signature Algorithm with Partially Known Nonces. *Designs, Codes and Cryptography*, 30(2), 201-217.
- NIST. (2015). *Secure Hash Standard (SHS)* (FIPS PUB 180-4). National Institute of Standards and Technology. <https://doi.org/10.6028/NIST.FIPS.180-4>
- NIST. (2020). *Recommendation for Key Management: Part 1 – General* (NIST Special Publication 800-57 Part 1, Revision 5). National Institute of Standards and Technology. <https://doi.org/10.6028/NIST.SP.800-57pt1r5>
- NIST. (2022). *Status Report on the Third Round of the NIST Post-Quantum Cryptography Standardization Process* (NIST IR 8413-upd1). National Institute of Standards and Technology. <https://doi.org/10.6028/NIST.IR.8413-upd1>

- NSA. (2016). *Report on Post-Quantum Cryptography* (NISTIR 8105). National Security Agency/National Institute of Standards and Technology.
- NVIDIA. (2020). *NVIDIA Jetson Platform*. Retrieved from <https://www.nvidia.com/en-us/>
- Oladipupo, E. T., Abikoye, O. C., & Awotunde, J. B. (2024). A Lightweight Image Cryptosystem for Cloud-Assisted Internet of Things. *Applied Sciences*, 14(7), 2808. <https://doi.org/10.3390/app14072808>
- Ollero, A., & Maza, I. (Eds.). (2007). *Multiple Heterogeneous Unmanned Aerial Vehicles* (Springer Tracts in Advanced Robotics, Vol. 37). Springer.
- Paar, C., & Pelzl, J. (2010). *Understanding Cryptography: A Textbook for Students and Practitioners*. Springer.
- Pardo-Castellote, G. (2003). OMG Data-Distribution Service: Architectural Overview. In *23rd International Conference on Distributed Computing Systems Workshops* (pp. 200-206). IEEE.
- Parisi, G. I., Kemker, R., Part, J. L., Kanan, C., & Wermter, S. (2019). Continual Lifelong Learning with Neural Networks: A Review. *Neural Networks*, 113, 54-71.
- Park, D., Park, J., & Hong, S. (2020). Fast Implementation of NIST P-256 Elliptic Curve Cryptography on 8-Bit AVR Processor. *Applied Sciences*, 10(14), 4858.
- Patcha, A., & Park, J.-M. (2007). An Overview of Anomaly Detection Techniques: Existing Solutions and Latest Technological Trends. *Computer Networks*, 51(12), 3448-3470.
- Peng, T., Leckie, C., & Ramamohanarao, K. (2007). Survey of Network-Based Defense Mechanisms Countering the DoS and DDoS Problems. *ACM Computing Surveys*, 39(1), Article 3.
- Pevný, T., Bas, P., & Fridrich, J. (2010). Steganalysis by Subtractive Pixel Adjacency Matrix. *IEEE Transactions on Information Forensics and Security*, 5(2), 215-224.
- Provos, N., & Honeyman, P. (2003). Hide and Seek: An Introduction to Steganography. *IEEE Security & Privacy*, 1(3), 32-44.

- PyTorch Mobile. (2022). *PyTorch Mobile Documentation*. Retrieved from <https://docs.pytorch.org/executorch/>
- Python Cryptographic Authority. (2023). *Cryptography Package*. PyPI. Retrieved from <https://pypi.org/project/cryptography/>
- Python Software Foundation. (2021). *Python 3.9 Documentation*. Retrieved from <https://docs.python.org/3.9/>
- Python Software Foundation. (2023). *Cryptography Library Documentation*. Retrieved from <https://cryptography.io/en/latest/>
- Queralta, J. P., Taipalmaa, J., Pullinen, B. C., Sarker, V. K., Gia, T. N., Tenhunen, H., Gabbouj, M., Raitoharju, J., & Westerlund, T. (2020). Collaborative Multi-Robot Search and Rescue: Planning, Coordination, Perception, and Active Vision. *IEEE Access*, 8, 191617-191643. <https://doi.org/10.1109/ACCESS.2020.3030190>
- Ranjan, S., Swaminathan, R., Uysal, M., Nucci, A., & Knightly, E. (2009). DDoS-Shield: DDoS-Resilient Scheduling to Counter Application Layer Attacks. *IEEE/ACM Transactions on Networking*, 17(1), 26-39.
- Raspberry Pi Foundation. (2019). *Raspberry Pi 4 Model B*. Retrieved from <https://www.raspberrypi.com/products/raspberry-pi-4-model-b/>
- Rebuffi, S.-A., Kolesnikov, A., Sperl, G., & Lampert, C. H. (2017). iCaRL: Incremental Classifier and Representation Learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 2001-2010).
- Redmon, J., & Farhadi, A. (2017). YOLO9000: Better, Faster, Stronger. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 7263-7271).
- Redmon, J., & Farhadi, A. (2018). YOLOv3: An Incremental Improvement. *arXiv preprint arXiv:1804.02767*.
- Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You Only Look Once: Unified, Real-Time Object Detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 779-788).

- Refis, F. C., Mahammedi, N. A., Kerrache, C. A., & Dhelim, S. (2025). From network sensors to intelligent systems: A decade-long review of swarm robotics technologies. *Sensors*, 25(19), 6115.
- Reina, G., Milella, A., & Galati, R. (2017). Terrain Assessment for Precision Agriculture Using Vehicle Dynamic Modelling. *Biosystems Engineering*, 162, 124-139.
- Ren, S., He, K., Girshick, R., & Sun, J. (2017). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6), 1137-1149.
- Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). "Why Should I Trust You?" Explaining the Predictions of Any Classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 1135-1144).
- Riverbank Computing. (2023). *PyQt5 Reference Guide*. Retrieved from <https://www.riverbankcomputing.com/static/Docs/PyQt5/>
- Rizk, Y., Awad, M., & Tunstel, E. W. (2019). Cooperative Heterogeneous Multi-Robot Systems: A Survey. *ACM Computing Surveys*, 52(2), Article 29.
- Roboflow Universe. (2023). *Pascal VOC Dataset*. Retrieved from <https://universe.roboflow.com/pascal-voc>
- Rodola, G. (2023). *psutil Documentation*. Retrieved from <https://psutil.readthedocs.io/en/latest/>
- Roesch, M. (1999). Snort—Lightweight Intrusion Detection for Networks. In *Proceedings of the 13th USENIX Conference on System Administration* (pp. 229-238).
- Roetteler, M., Naehrig, M., Svore, K. M., & Lauter, K. (2017). Quantum Resource Estimates for Computing Elliptic Curve Discrete Logarithms. In *International Conference on the Theory and Application of Cryptology and Information Security* (pp. 241-270). Springer.
- Rossow, C. (2014). Amplification Hell: Revisiting Network Protocols for DDoS Abuse. In *Proceedings of the Network and Distributed System Security Symposium (NDSS)*.

- Rusu, A. A., Rabinowitz, N. C., Desjardins, G., Soyer, H., Kirkpatrick, J., Kavukcuoglu, K., Pascanu, R., & Hadsell, R. (2016). Progressive Neural Networks. *arXiv preprint arXiv:1606.04671*.
- Ruzai, W. N. A., Ying, Y., Muhammad, K. N., Asbullah, M. A., & Ariffin, M. R. K. (2024). Concurrent factorization of RSA moduli via weak key equations. *AIMS Mathematics*, 9(10), 28211-28231.
- Sadek, M. M., Khalifa, A. S., & Mostafa, M. G. M. (2015). Video Steganography: A Comprehensive Review. *Multimedia Tools and Applications*, 74(17), 7063-7094.
- Saied, A., Overill, R. E., & Radzik, T. (2016). Detection of Known and Unknown DDoS Attacks Using Artificial Neural Networks. *Neurocomputing*, 172, 385-393.
- Sargeant, I., & Tomlinson, A. (2019). Intrusion Detection in Robotic Swarms. In *Intelligent Systems and Applications: Proceedings of the 2019 Intelligent Systems Conference (IntelliSys), Volume 2* (pp. 561-576). Springer.
- Satyanarayanan, M. (2017). The Emergence of Edge Computing. *Computer*, 50(1), 30-39.
- Scharre, P. (2014). *Robotics on the Battlefield Part II: The Coming Swarm*. Center for a New American Security.
- SEC. (2009). *SEC 1: Elliptic Curve Cryptography, Standards for Efficient Cryptography* (Version 2.0). Certicom Research.
- Sedjelmaci, H., Senouci, S. M., & Ansari, N. (2018). A Hierarchical Detection and Response System to Enhance Security Against Lethal Cyber-Attacks in UAV Networks. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 48(9), 1594-1606.
- Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., & Batra, D. (2020). Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization. *International Journal of Computer Vision*, 128(2), 336-359.
- Sethi, K., Kumar, R., Prajapati, N., & Bera, P. (2025). Lightweight DWT Steganography With ECC-ChaCha20 for Secure Medical IoT Systems. *IEEE Access*, 13, 115309-115325.

- Setiadi, D. R. I. M. (2021). PSNR vs SSIM: Imperceptibility Quality Assessment for Image Steganography. *Multimedia Tools and Applications*, 80(6), 8423-8444.
- Settles, B. (2009). *Active Learning Literature Survey* (Technical Report #1648). University of Wisconsin-Madison, Computer Sciences Department.
- Sharafaldin, I., Lashkari, A. H., & Ghorbani, A. A. (2018). Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization. In *Proceedings of the 4th International Conference on Information Systems Security and Privacy (ICISSP)* (pp. 108-116).
- Shejwalkar, V., & Houmansadr, A. (2021). Manipulating the Byzantine: Optimizing Model Poisoning Attacks and Defenses for Federated Learning. In *Proceedings of the Network and Distributed System Security Symposium (NDSS)*.
- Shi, W., Cao, J., Zhang, Q., Li, Y., & Xu, L. (2016). Edge Computing: Vision and Challenges. *IEEE Internet of Things Journal*, 3(5), 637-646.
- Shi, W., & Dustdar, S. (2016). The Promise of Edge Computing. *Computer*, 49(5), 78-81.
- Shirai, T., Shibutani, K., Akishita, T., Moriai, S., & Iwata, T. (2007). The 128-Bit Blockcipher CLEFIA (Extended Abstract). In A. Biryukov (Ed.), *Fast Software Encryption: 14th International Workshop, FSE 2007, Luxembourg, Luxembourg, March 26-28, 2007, Revised Selected Papers* (Lecture Notes in Computer Science, Vol. 4593, pp. 181-195). Springer.
- Siegwart, R., & Nourbakhsh, I. R. (2004). *Introduction to Autonomous Mobile Robots*. MIT Press.
- Silverman, J. H. (2009). *The Arithmetic of Elliptic Curves* (2nd ed.). Springer.
- Simonyan, K., Vedaldi, A., & Zisserman, A. (2014). Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps. In *Proceedings of the International Conference on Learning Representations (ICLR) Workshop*.
- Skulimowski, S., Rybka, S., Tatara, B., & Welman, M. (2025). Systematic drift characterization in differential wheeled robot using external VR tracking: Effects of route complexity and motion dynamics. *Applied Computer Science*, 21, 117-136.

- Specht, D. F., & Lee, H. (2004). *Neural Networks in Business: Techniques and Applications*. IGI Global.
- Stallings, W. (2017). *Cryptography and Network Security: Principles and Practice* (7th ed.). Pearson.
- Stallings, W. (2022). *Cryptography and Network Security: Principles and Practice* (8th ed.). Pearson Education.
- Stentz, A., Bares, J., Singh, S., & Rowe, P. (1998). A Robotic Excavator for Autonomous Truck Loading. In *Proceedings of the 1998 IEEE/RSJ International Conference on Intelligent Robots and Systems* (Vol. 3, pp. 1885-1893). IEEE.
- Szynkiewicz, W., Niewiadomska-Szynkiewicz, E., & Lis, K. (2023). Deep Learning of Sensor Data in Cybersecurity of Robotic Systems: Overview and Challenges. *Sensors*, 23(5), 2476.
- Şahin, E. (2005). Swarm Robotics: From Sources of Inspiration to Domains of Application. In E. Şahin & W. M. Spears (Eds.), *Swarm Robotics: SAB 2004 International Workshop, Santa Monica, CA, USA, July 17, 2004, Revised Selected Papers* (Lecture Notes in Computer Science, Vol. 3342, pp. 10-20). Springer.
- Taha, M. B., Talaat, F. M., & Elhoseny, M. (2019). Combining Steganography and Cryptography for Secure Data Transmission. In *IOP Conference Series: Materials Science and Engineering* (Vol. 518, Article 042025). IOP Publishing.
- Tanenbaum, A. S., & Van Steen, M. (2017). *Distributed Systems: Principles and Paradigms* (3rd ed.). Pearson.
- Tang, W., Li, B., Tan, S., Barni, M., & Huang, J. (2019). CNN-Based Adversarial Embedding for Image Steganography. *IEEE Transactions on Information Forensics and Security*, 14(8), 2074-2087.
- Tao, H., Li, C., Zain, J. M., & Abdalla, A. N. (2014). Robust Image Watermarking Theories and Techniques: A Review. *Journal of Applied Research and Technology*, 12(1), 122-138.

- Terven, J. R., & Córdova-Esparza, D. M. (2023). A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS. *Machine Learning and Knowledge Extraction*, 5(4), 1680-1716.
- Thrun, S., Burgard, W., & Fox, D. (2005). *Probabilistic Robotics*. MIT Press.
- Tian, Z., Shen, C., Chen, H., & He, T. (2019). FCOS: Fully Convolutional One-Stage Object Detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)* (pp. 9627-9636).
- Tolpegin, V., Truex, S., Gursoy, M. E., & Liu, L. (2020). Data Poisoning Attacks Against Federated Learning Systems. In *Proceedings of the 25th European Symposium on Research in Computer Security (ESORICS)* (pp. 480-501). Springer.
- Tran, D. N., Zepernick, H. J., & Chu, T. M. C. (2022). LSB Data Hiding in Digital Media: A Survey. *EAI Endorsed Transactions on Industrial Networks and Intelligent Systems*, 9(30), e3.
- Ullah, S., Zheng, J., Din, N., Hussain, M. T., Ullah, F., & Yousaf, M. (2023). Elliptic Curve Cryptography; Applications, challenges, recent advances, and future trends: A comprehensive survey. *Computer Science Review*, 47, 100530.
- Ultralytics. (2023). *YOLOv8 Documentation*. Retrieved from <https://docs.ultralytics.com/>
- Upton, E., & Halfacree, G. (2020). *The Official Raspberry Pi Beginner's Guide: How to Use Your New Computer* (4th ed.). Raspberry Pi Press.
- Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., ... & SciPy 1.0 Contributors. (2020). SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17(3), 261-272.
- Vujović, V., & Maksimović, M. (2015). Raspberry Pi as a Sensor Web Node for Home Automation. *Computers & Electrical Engineering*, 44, 153-171.
- Wachter, S., Mittelstadt, B., & Russell, C. (2017). Counterfactual Explanations Without Opening the Black Box: Automated Decisions and the GDPR. *Harvard Journal of Law & Technology*, 31(2), 841-887.

- Wang, Z., Bovik, A. C., Sheikh, H. R., & Simoncelli, E. P. (2004). Image Quality Assessment: From Error Visibility to Structural Similarity. *IEEE Transactions on Image Processing*, 13(4), 600-612.
- Wang, W., Zhu, M., Wang, J., Zeng, X., & Yang, Z. (2018). Malware Traffic Classification Using Convolutional Neural Network for Representation Learning. In *Proceedings of the 2017 International Conference on Information Networking (ICOIN)* (pp. 712-717). IEEE.
- Wang, C. Y., Liao, H. Y. M., Wu, Y. H., Chen, P. Y., Hsieh, J. W., & Yeh, I. H. (2019). CSPNet: A New Backbone That Can Enhance Learning Capability of CNN. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*.
- Washington, L. C. (2008). *Elliptic Curves: Number Theory and Cryptography* (2nd ed.). Chapman and Hall/CRC.
- Watro, R., Kong, D., Cuti, S., Gardiner, C., Lynn, C., & Kruus, P. (2004). TinyPK: Securing Sensor Networks with Public Key Technology. In *Proceedings of the 2nd ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN)* (pp. 59-64).
- Wendzel, S., Zander, S., Fechner, B., & Herdin, C. (2015). Pattern-Based Survey and Categorization of Network Covert Channel Techniques. *ACM Computing Surveys*, 47(3), Article 50.
- Winkler, S., & Mohandas, P. (2008). The Evolution of Video Quality Measurement: From PSNR to Hybrid Metrics. *IEEE Transactions on Broadcasting*, 54(3), 660-668.
- Wurman, P. R., D'Andrea, R., & Mountz, M. (2008). Coordinating Hundreds of Cooperative, Autonomous Vehicles in Warehouses. *AI Magazine*, 29(1), 9-19.
- Xie, Y., & Yu, S. Z. (2009). Monitoring the Application-Layer DDoS Attacks for Popular Websites. *IEEE/ACM Transactions on Networking*, 17(1), 15-28.
- Xu, W., Trappe, W., Zhang, Y., & Wood, T. (2005). The Feasibility of Launching and Detecting Jamming Attacks in Wireless Networks. In *Proceedings of the 6th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)* (pp. 46-57).

- Yin, C., Zhu, Y., Fei, J., & He, X. (2017). A Deep Learning Approach for Intrusion Detection Using Recurrent Neural Networks. *IEEE Access*, 5, 21954-21961.
- Yuan, X., Li, C., & Li, X. (2017). DeepDefense: Identifying DDoS Attack via Deep Learning. In *Proceedings of the 2017 IEEE International Conference on Smart Computing (SMARTCOMP)* (pp. 1-8).
- Yudistira, R. (2024). AES (Advanced Encryption Standard) and RSA (Rivest Shamir Adleman) encryption on digital signature document: A literature review. *International Journal of Information Technology and Business*, 7, 29-31.
- Zargar, S. T., Joshi, J., & Tipper, D. (2013). A Survey of Defense Mechanisms Against Distributed Denial of Service (DDoS) Flooding Attacks. *IEEE Communications Surveys & Tutorials*, 15(4), 2046-2069.
- Zheng, Z., Wang, P., Liu, W., Li, J., Ye, R., & Ren, D. (2020). Distance-IoU Loss: Faster and Better Learning for Bounding Box Regression. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(7), 12993-13000.
- Zhou, D.-W., Wang, Q.-W., Qi, Z.-H., Ye, H.-J., Zhan, D.-C., & Liu, Z. (2024). Class-incremental learning: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(12), 9851-9873.

ÖZGEÇMİŞ

Kişisel Bilgiler

Soyadı, adı :

Uyruğu :

Doğum tarihi ve yeri:

Telefon :

e-mail :



Eğitim

Derece	Eğitim Birimi	Mezuniyet
Doktora	SBTÜ/ Savunma Teknolojileri	Devam ediyor
Yüksek lisans	Kırıkkale Üniv/Bilgisayar Müh.	2019
Lisans	Kırıkkale Üniv/Blg. Öğrt.	2016
Ön Lisans	Cumhuriyet Üniversitesi Gürün MYO Blg.Prg	2006

İş Deneyimi

Yıl	Yer	Görev
2021-Halen	Sivas Cumhuriyet Üniv	Öğr.Gör.
2019-2021	Karadeniz Teknik Üniv.	Öğr.Gör
2006-2019	Arçelik A.Ş. Sivas	Bilgi Sistemleri Sorumlusu.

Yabancı Dil

İngilizce

Yayınlar

Kutlu, Z., & Emiroğlu, B. G. (2025). Image-Based Threat Detection and Explainability Investigation Using Incremental Learning and Grad-CAM with YOLOv8. *Computers*, 13(11), 305.

Kutlu, Z., & Ağca, R. K. (2021). Yabancı Dil Öğretiminde Oyun Tabanlı 3 Boyutlu Sanal Gerçeklik Uygulaması. In 8. *Uluslararası Öğretim Teknolojileri ve*

Öğretmen Eğitimi Sempozyumu (ITTES 2021), Trabzon, Türkiye, 17-19 Kasım 2021 (ss. 84-88).

Kutlu, Z., & Ağca, R. K. (2021). Metaverse: Web 3.0 ile Gerçek Hayattan Avatarlar Dünyasına Hızlı Geçiş. In 8. *Uluslararası Öğretim Teknolojileri ve Öğretmen Eğitimi Sempozyumu (ITTES 2021)*, Trabzon, Türkiye, 17-19 Kasım 2021 (ss. 61-63).

Kutlu, Z., & Ağca, R. K. (2019). MOMMS-Mobile Optical Measurement and Rating System. In 6th *International Instructional Technologies & Teacher Education Symposium (ITTES 2019)*, Antalya, Türkiye, 29 Ekim - 01 Kasım 2019 (ss. 45-49).

Yükseltürk, E., & Kutlu, Z. (2017). Bilişim Teknolojileri Öğretmen Adaylarının Okul Deneyimi Dersine Yönelik Tutumları ve Hazırlanan Web Tabanlı Destek Sistemi ile İlgili Görüşleri. *Journal of Science Teacher Education*, 1(2), 58-69.

Hobiler

Yüzme, Amatör Elektronik Çalışmalar, Doğa Gezileri

