

T.C.
ONDOKUZ MAYIS ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ
HESAPLAMALI BİLİMLER ANA BİLİM DALI



SİS BİLİŞİM VE IOT TABANLI SAĞLIK VE TAKTİK
ANALİZ İZLEME MODELİ

Doktora Tezi

Aykut KARAKAYA

Danışman

Doç. Dr. Sedat AKLEYLEK

SAMSUN
2021

TEZ KABUL VE ONAYI

Aykut KARAKAYA tarafından, Doç. Dr. Sedat AKLEYLEK danışmanlığında hazırlanan “Sis Bilişim ve IoT Tabanlı Sağlık ve Taktik Analiz İzleme Modeli ” başlıklı bu çalışma, jürimiz tarafından 10.12.2021 tarihinde yapılan sınav sonucunda oy birliği ile başarılı bulunarak Doktora Tezi olarak kabul edilmiştir.

	Unvanı Adı Soyadı Üniversitesi Ana Bilim/Ana Sanat Dalı	İmza	Sonuç
Başkan	Doç. Dr. Hakan KUTUCU Karabük Üniversitesi Yazılım Mühendisliği Anabilim Dalı		<input checked="" type="checkbox"/>
			Kabul
Üye (Danışman)	Doç. Dr. Sedat AKLEYLEK Ondokuz Mayıs Üniversitesi Bilgisayar Mühendisliği Anabilim Dalı		<input type="checkbox"/>
			Ret
Üye	Prof. Dr. Erdal KILIÇ Ondokuz Mayıs Üniversitesi Bilgisayar Mühendisliği Anabilim Dalı		<input checked="" type="checkbox"/>
			Kabul
Üye	Dr. Öğr. Üyesi Meltem KURT PEHLİVANOĞLU Kocaeli Üniversitesi Bilgisayar Mühendisliği Anabilim Dalı		<input type="checkbox"/>
			Ret
Üye	Dr. Öğr. Üyesi Korhan CENGİZ Trakya Üniversitesi Elektrik Elektronik Mühendisliği Anabilim Dalı		<input checked="" type="checkbox"/>
			Kabul
			<input type="checkbox"/>
			Ret

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

ONAY

... / ... / ...

Prof. Dr. Ali BOLAT
Enstitü Müdürü

BİLİMSEL ETİĞE UYGUNLUK BEYANI

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

Etik Kurul Gerekli mi?

Evet (Gerekli ise ekler kısmına ekleyiniz)

Hayır

İmza

14/12/2021

Aykut KARAKAYA

TEZ ÇALIŞMASI ÖZGÜNLÜK RAPORU BEYANI

Tez Başlığı: Sis Bilişim ve IoT Tabanlı Sağlık ve Taktik Analiz İzleme Modeli

Yukarıda başlığı belirtilen tez çalışması için şahsım tarafından 08/11/2021 tarihinde intihal tespit programından alınmış olan özgünlük raporu sonucunda;

Benzerlik oranı : % 4

Tek kaynak oranı : % 1 çıkmıştır.

İmza

14/12/ 2021

Doç Dr. Sedat AKLEYLEK

ÖZET

SİS BİLİŞİM VE IOT TABANLI SAĞLIK VE TAKTİK ANALİZ İZLEME MODELİ

Aykut KARAKAYA

Ondokuz Mayıs Üniversitesi

Lisansüstü Eğitim Enstitüsü

Hesaplamalı Bilimler Ana Bilim Dalı

Doktora, Aralık/2021

Danışman: Doç. Dr. Sedat AKLEYLEK

Spor müsabakalarında sporcular için maç esnasında heyecan, stres, yorgunluk gibi koşullara bağlı olarak sakatlık veya taktiksel zafiyet gibi olumsuz sonuçlar ortaya çıkabilmektedir. Bu nedenle sağlık durumlarının sürekli kontrol edilmesi ve maç esnasında bir takım stratejik analizlerin yapılması gerekmektedir. Taktiksel analizlerin sonucuna göre teknik ekip maçın gidişatına etki eden hamleler yapmaktadır. Etkiler, bazen olumlu bazen olumsuz sonuçlar doğurabilmektedir. Faydalı hamleler için analizlerin doğruluk oranlarının yüksek olması gerekmektedir. Bu tez kapsamında olumsuz durumların oyuna etkisini azaltmak ve sakatlıklardan korunmak için sis bilişim ve nesnelerin İnterneti (Internet of things-IoT) tabanlı bir mimari ve bu mimari içinde çalışabilen makine öğrenmesi temelli bir taktik analiz modeli önerilmektedir.

Sporcular maç esnasında kan basıncı, vücut ısısı, kalp atış hızı, hassas konum gibi sensörler ile izlenmektedir. Sensörlerden alınan veriler sis bilişim katmanında işlenmekte ve ortaya çıkan bilgiler teknik ekip ve kulüp doktorlarının cihazlarına gönderilmektedir. Önerilen mimaride önemli olayların çok daha kısa sürede işlenebilmesi için sis katmanında öncelik kuyruğu kullanan bir algoritma önerilmektedir. Bu mimari iFogSim kullanılarak test edilmektedir. Benzer çalışmalar ile karşılaştırıldığında önerilen yapının gecikme süresini önemli ölçüde düşürdüğü ve yüksek verim sağladığı görülmektedir. Mimaride kullanılan nesnelerin birleşik hafif kimlik doğrulaması (federated lightweight authentication of things-FLAT) yöntemi ile kimlik doğrulama ve veri gizliliği sağlanmaktadır. Ayrıca, blokzincir teknolojisine dayalı yazılım tanımlı ağ denetleyicisi kullanılarak veri bütünlüğü sağlanmaktadır.

Sis bilişim ve IoT tabanlı altyapıda çalışması için örnek olarak makine öğrenmesi ve birleşik öğrenme tabanlı taktik analiz yöntemi önerilmektedir. Önceki maçlardan elde edilen veriler kullanılarak bir makine öğrenmesi modeli eğitilmekte ve sınıflandırılmaktadır. Model diskriminant analizi, k-en yakın komşuluk (k-nearest neighbor-KNN), naive Bayes, destek vektör makinesi (support vector machine-SVM), karar ağaçları ve topluluk öğrenmesi yöntemlerinin farklı türevleri kullanılarak test edilmektedir. Topluluk öğrenme yöntemlerinden biri olan ve en uygun parametreler için optimize edilen AdaBoost %87,2 ile en yüksek başarıma sahiptir. Önerilen mimaride önerilen model için bu yöntem kullanılabilir.

Anahtar Sözcükler: Nesnelerin İnterneti, sağlık izleme, taktik analiz, sis bilişim, kaynak tahsisi, güvenlik, futbol formasyonu, makine öğrenmesi.

ABSTRACT

FOG COMPUTING AND IOT BASED HEALTH AND TACTICAL ANALYSIS MONITORING MODEL

Aykut KARAKAYA

Ondokuz Mayıs University

Institute of Graduate Studies

Department of Computational Sciences

Phd, December/2021

Supervisor: Assoc. Prof. Dr. Sedat AKLEYLEK

In sports, negative consequences such as injury or tactical weakness may occur for the teams depending on the conditions such as excitement, stress, and fatigue during the match. For this reason, it is necessary to constantly control the health of the player and make some strategic analyzes during the match. According to the results of the tactical analysis, the technical team can make moves that affect the match. The effects can sometimes lead to positive and sometimes negative results. For positive moves, the accuracy of the analyzes must be high. In this dissertation, an architecture based on fog computing and Internet of things (IoT) is proposed in order to reduce the effects of negative situations on the game and to protect against injuries. In addition, a machine learning-based tactical analysis model that can work in this architecture is proposed.

Players are monitored during the match with sensors such as blood pressure, body temperature, heart rate, precise location. The data received from the sensors are processed in the fog computing layer and the resulting information is sent to the devices of the technical team and club doctors. In the proposed architecture, an algorithm that uses a priority queue in the fog layer is proposed so that important events can be processed in a much shorter time. This architecture is tested using iFogSim. When compared with similar studies, it is determined that the proposed structure significantly reduces the delay time and provides high efficiency. Authentication and data privacy are ensured by the federated lightweight authentication of things (FLAT) method used in architecture. In addition, data integrity is ensured by using a software-defined network controller based on blockchain technology.

A machine learning and federated learning based tactical analysis model that can work in fog computing and IoT based infrastructure is proposed. A machine learning model is trained and classified using data from previous matches. The model is tested using different variants of discriminant analysis, k-nearest neighbor (KNN), naive bayes, support vector machine (SVM), decision trees, and ensemble learning methods. AdaBoost, which is one of the ensemble learning methods and optimized for the most suitable parameters, has the highest performance with 87.2%. This method can be used for the proposed model in the proposed architecture.

Keywords: Internet of things, health monitoring, tactical analysis, fog computing, resource allocation, security, football formation, machine learning.

ÖNSÖZ VE TEŞEKKÜR

Bu tezin tamamlanmasında öncelikle; desteğini ve tecrübesini hiçbir zaman esirgemeyen, tez danışmanım, değerli hocam Doç. Dr. Sedat AKLEYLEK'e; tez çalışmama yaptıkları yorumlar ile katkı veren tez izleme komitesindeki değerli hocalarım Prof. Dr. Erdal KILIÇ'a ve Doç. Dr. Hakan KUTUCU'ya teşekkürlerimi bir borç bilirim.

Tez sürecinde; çalışma ortamımı sağlamada hep yardımcı olmaya çabalayan ve bana olan güvenlerini her an hissettiren sevgili aileme, karamsarlığa düştüğüm anlarda bana olan inancıyla güç bulduğum ve motivasyonumu üst düzeyde tutabilmemi sağlayan sevgili nişanlım İrem ŞAHMUTOĞLU'na teşekkür ederim. Ayrıca, tezin görüntü işleme kısmında destek veren değerli arkadaşım Ahmet ULU'ya, veri toplama aşamasında uzun zamanlarını ayırıp yardım eden değerli arkadaşlarım Fatih SEVEN'e ve Ahmet Alp KUŞPINAR'a teşekkür ederim.

Aykut KARAKAYA

İÇİNDEKİLER

TEZ KABUL VE ONAYI	i
BİLİMSEL ETİĞE UYGUNLUK BEYANI	ii
ÖZET	iii
ABSTRACT	iv
ÖNSÖZ VE TEŞEKKÜR	v
İÇİNDEKİLER	vi
SİMGELER VE KISALTMALAR	viii
ŞEKİLLER DİZİNİ	xi
TABLolar DİZİNİ	xiii
1. GİRİŞ	1
1.1. Literatür İncelemesi.....	2
1.1.1. IoT Tabanlı Spor ve Altyapı Çalışmaları.....	3
1.1.2. IoT Sistemlerde Kaynak Yönetim Çalışmaları.....	5
1.1.2.1. Bulut Bilişimde Kaynak Tahsis Yöntemleri	7
1.1.2.2. Sis Bilişimde Kaynak Tahsis Yöntemleri	7
1.1.2.3. Kenar Bilişimde Kaynak Tahsis Yöntemleri	9
1.1.2.4. Bulut, Sis ve Kenar Bilişimde Kaynak Tahsis Yöntemleri.....	11
1.1.3. Makine Öğrenmesi Tabanlı Taktik Analiz Çalışmaları.....	18
1.2. Motivasyon ve Katkı	22
1.2.1. Sis Bilişim ve IoT Tabanlı Mimarinin Katkıları.....	22
1.2.2. Sis Düğümlerinde Görev Yönetim ve Kaynak Tahsis Algoritmasının Katkıları	23
1.2.3. Makine Öğrenmesi Tabanlı Taktik Analiz Modelinin Katkıları	24
1.3. Organizasyon.....	25
2. NESNELERİN İNTERNETİ VE SİS BİLİŞİM	26
2.1. IoT Sistemlerinde Depolama ve Veri İşleme	27
2.2. Sis Bilişim	31
3. SİS BİLİŞİM VE NESNELERİN İNTERNETİ TABANLI SAĞLIK İZLEME VE TAKTİK ANALİZ MİMARİSİ, SİS DÜĞÜMLERİNDE GÖREV YÖNETİM VE KAYNAK TAHSİS ALGORİTMASI	33
3.1. IoT Tabanlı Mimari Detayları	33
3.2. Sis Düğümlerinde Görev Yönetimi ve Önerilen Algoritma.....	37
3.3. Önerilen IoT Tabanlı Mimarinin Güvenliğinde Kullanılan Yöntemler	44
3.3.1. Veri Gizliliği ve Kimlik Doğrulama.....	44
3.3.2. Blokzincir Tabanlı SDN Denetçisi ile Güvenlik Desteği	47

3.4. Önerilen IoT Tabanlı Mimarinin Güvenlik Analizi	49
3.4.1. Düşük Gecikme Süresi	49
3.4.2. Algılanan Verilerin Doğruluğu	49
3.4.3. Kimlik Doğrulama ve Veri Gizliliği	50
3.4.4. Veri Bütünlüğü ve Erişilebilirlik	50
3.4.5. Bant Genişliği Tasarrufu.....	51
3.5. Önerilen Mimarinin ve Algoritmanın Simülasyonu.....	51
3.5.1. Simülasyon Topolojisi	51
3.5.2. Simülasyonda Kullanılan Sis Düğümlerinin Özellikleri	52
3.5.3. Simülasyonun Uygulama Yapısı	53
3.5.4. Simülasyon Uygulamasının Kod Yapısı.....	53
3.5.5. Simülasyon Uygulamasının Sonuçları.....	55
3.5.6. Simülasyon Sonuçlarının Karşılaştırılması.....	58
3.6. Giyilebilir Teknolojilerdeki Zorluklar.....	60
4. MAKİNE ÖĞRENMESİ TABANLI TEKNİK EKİP UYARI SİSTEMİ... 61	
4.1. Önerilen Modelin Uygulanması	63
4.1.1. Veri Toplama	64
4.1.2. Veri Ön İşleme.....	65
4.1.2.1. Görüntü İşleme.....	65
4.1.2.2. Veri Kümesinin Normalizasyonu.....	69
4.1.3. Makine Öğrenmesi Algoritmaları ile Veri İşleme	69
4.1.3.1. Veri Kümesinin Eğitilmesi.....	69
4.1.3.2. Deneysel Sonuçlar.....	70
4.1.3.3. Bir Topluluk Öğrenme Yöntemi: AdaBoost.....	73
5. SONUÇLAR VE GELECEK ÇALIŞMALAR	75
KAYNAKLAR	79
ÖZ GEÇMİŞ.....	86

SİMGELER VE KISALTMALAR

2r-SAE: Related and Regularized Stacked Auto Encoder (İlgili ve Düzenlenmiş Yığılmış Otomatik Kodlayıcı)

ADMM: Alternating Direction Method of Multipliers (Çarpanların Alternatif Yön Yöntemi)

AES: Advanced Encryption Standard (Gelişmiş Şifreleme Standardı)

AHP: Analytical Hierarchical Process (Analitik Hiyerarşik Süreç)

ARP: Address Resolution Protocol (Adres Çözümleme Protokolü)

ASA: Adaptive Simulated Annealing (Uyarlanabilir Simüle Tavlama)

Bi-GISIS: Bilateral Generalization Inhomogeneous Short Integer Solution (İkili Genelleme Homojen Olmayan Kısa Tamsayı Çözümü)

CNN: Convolutional Neural Network (Evrışimli Sinir Ağı)

CPU: Central Process Unit (Merkezi İşlem Birimi)

C-RAN: Cloud Radio Access Network (Bulut Radyo Erişim Ağı)

DDoS: Distributed Denial of Service (Dağıtık Hizmet Reddi)

DFL: Dispersed Federated Learning (Dağınık Birleşik Öğrenme)

DQN: Deep Q Network (Derin Q Ağı)

DRL: Deep Reinforcement Learning (Derin Pekiştirmeli Öğrenme)

ECC: Elliptic-Curve Cryptography (Eliptik Eğri Kriptografisi)

ECQV: Elliptic Curve Qu-Vanstone (Eliptik Eğri Qu-Vanstone)

FIdM: Federated Identity Management (Birleştirilmiş Kimlik Yönetimi)

FIS: Fuzzy Inference System (Bulanık Çıkarım Sistemi)

FLAT: Federated Lightweight Authentication of Things (Nesnelerin Birleştirilmiş Hafif Kimlik Doğrulaması)

Fog-RAN: Fog Radio Access Network (Sis Radyo Erişim Ağı)

GMA: Gateway Placement and Multihop Routing Algorithm (Ağ Geçidi Yerleştirme ve Çok Atlamalı Yönlendirme Algoritması)

HAP: Hybrid Access Point (Hibrit Erişim Noktası)

HSV: Hue, Saturation, Value (Ton, Doygunluk, Değer)

ID: Identification Number (Kimlik Numarası)

IDE: Integrated Development Environment (Tümleşik Geliştirme Ortamı)

IoT: Internet of Things (Nesnelerin İnterneti)

KBPS: Kilobits Per Second (Saniye Başına Kilobit)

KNN: K-Nearest Neighbors (K-En Yakın Komşular)

LB-2PAKA: Lattice Based Two Party Authenticated Key Agreement (Kafes Tabanlı İki Taraflı Kimliği Doğrulanmış Anahtar Anlaşması)

M2M: Machine to Machine (Makineden Makineye)

MAC: Message Authentication Code (Mesaj Doğrulama Kodu)

MB: Megabyte (Megabayt)

MCDM: Multi-Criteria Decision Making (Çok Kriterli Karar Verme)

MCPR: Matching Considered Preferences (Dikkate Değer Tercihleri Eşleştirme)

MCRC: The Matching Design Considers The Resource Cube (Dikkate Değer Kaynak Küplerini Eşleştirme Tasarımı)

MDP: Markov Decision Process (Markov Karar Süreci)

MEC: Multi-Access Edge Computing (Çoklu Erişim Kenar Bilgi İşlem)

MIPS: Million Instructions Per Second (Saniye Başına Milyon İşlem)

MITM: Man-in-The-Middle (Ortakdaki Adam)

MJ: Megajoule (Megajul)

ML: Machine Learning (Makine öğrenmesi)

MQTT: Message Queuing Telemetry Transport (Mesaj Kuyruğu Telemetri Aktarımı)

NB-IoT: Narrow Band IoT (Dar Bant IoT)

NFV: Network Functions Virtualization (Ağ İşlevleri Sanallaştırma)

NOMA: Non Orthogonal Multiple Access (Ortogonal Olmayan Çoklu Erişim)

OFDM: Orthogonal Frequency Division Multiplexing (Ortogonal Frekans Bölmeli Çoğullama)

PER: Prioritized Experience Replay (Öncelikli Deneyim Tekrarı)

QBDT: QoS-based Task Distribution (QoS Tabanlı Görev Dağıtım)

QoE: Quality of Experience (Deneyim Kalitesi)

QoS: Quality of Service (Hizmet Kalitesi)

RGB: Red, Green, Blue (Kırmızı, Yeşil, Mavi)

RJCC: Reinforcement Learning based Joint Communicational-and-Computational Resource Allocation (Pekiřtirmeli Öğrenme Tabanlı Ortak İletişimsel ve Hesaplamalı Kaynak Tahsisi)

RL: Reinforcement Learning (Pekiřtirmeli Öğrenme)

RNN: Deep Recurrent Neural Network (Derin Tekrarlayan Sinir Ađı)

SARSA: State-Action-Reward- State-Action (Durum-Eylem-Ödöl- Durum-Eylem)

SCA: Successive Convex Approximation (Ardışık Dışbükey Yaklaşım)

SDN: Software Defined Network (Yazılımla Tanımlanan Ağ)

SE: Stackelberg Equilibrium (Stackelberg Dengesi)

SIR: Signal Interference Ratio (Sinyal Parazit Oranı)

SLA: Service Level Agreement (Hizmet Seviyesi Anlaşması)

SP1A: Service Placement Algorithm 1 (Hizmet Yerleştirme Algoritması 1)

SP2A: Service Placement Algorithm 2 (Hizmet Yerleştirme Algoritması 2)

SPA: Student Project Allocation (Öğrenci Proje Tahsisi)

SPT: Shortest Path Tree (En Kısa Yol Ağacı)

SPTC: SPT with Centrality (Merkezi SPT)

STN: Satellite-Terrestrial Network (Uydu-Karasal Ağ)

SVM: Support Vector Machine (Destek Vektör Makinesi)

TCA-IPSO: Two-edge-node Cooperative-task Allocation based on Improved Particle Swarm Optimization (İyileştirilmiş Parçacık Sürü Optimizasyonuna Dayalı İki Uçlu Ortak Görev Tahsisi)

UOC: User Oriented Cooperation (Kullanıcı Odaklı İşbirliği)

WBAN: Wireless Body Area Networks (Kablosuz Vücut Alan Ağları)

WSN: Wireless Sensor Networks (Kablosuz Sensör Ağlar)

ŞEKİLLER DİZİNİ

Şekil 1.1. IoT'de kaynak yönetim süreçlerinin sınıflandırılması	6
Şekil 2.1. IoT sistemlerin genel yapısı	27
Şekil 2.2. Bulut bilişim kullanan IoT sistemi	28
Şekil 2.3. Bulut ve sis bilişimi birlikte kullanan IoT sistemi.....	29
Şekil 2.4. Bulut ve kenar bilişimi birlikte kullanan IoT sistemi	30
Şekil 2.5. Sis bilişim katmanı (Akleyek ve Karakaya, 2020).....	32
Şekil 3.1. Önerilen mimarinin veri akışı ve kullanılan yapılar	33
Şekil 3.2. Sistemin aktörleri, olası konumları ve veri iletim yönleri	35
Şekil 3.3. Algılama ve izleme uç cihazlarının veri iletişim şeması	35
Şekil 3.4. Önerilen mimarinin katmanları	36
Şekil 3.5. Bir sis düğümünün yapısı.....	38
Şekil 3.6. Sis ve bulut düğümlerindeki görevler	38
Şekil 3.7. Sis katmanı görev grafi	39
Şekil 3.8. Öncelik atama algoritması.....	41
Şekil 3.9. Önerilen görev yönetim ve kaynak tahsis algoritması	42
Şekil 3.10. FLAT Kimlik doğrulama şemasının önerilen mimari üzerinde işleyişi	45
Şekil 3.11. Blokzincir tabanlı SDN denetleyicisinin yapısı	47
Şekil 3.12. Simülasyonda baz alınan topoloji	52
Şekil 3.13. Modüller arasındaki demet iletim yönü.....	53
Şekil 3.14. İsteğin tek bir sis düğümüne atanabildiği durumda modül yerleşimi.....	54
Şekil 3.15. İsteğin birden fazla sis düğümüne atanabildiği durumda modül yerleşimi	55
Şekil 3.16. İsteğin hiçbir sis düğümüne atanamadığı durumda yürütülen işlem	55
Şekil 3.17. Önerilen algoritmanın simülasyon sonuçları.....	56
Şekil 3.18. Önerilen algoritmanın simülasyon sonuçlarının grafikleri ((a) Yürütme süresi, (b) Toplam ağ kullanımı, (c) Uygulama döngüleri gecikme süresi, (d) Bulutta yürütme maliyeti).....	57
Şekil 4.1. Birleştirilmiş öğrenme modeli.....	62
Şekil 4.2. Teknik ekip uyarı sistemindeki süreçlerin akış şeması	63
Şekil 4.3. FM'den alınan görüntüde sahanın bölgelere ayrılması (temsili çizgiler ile).....	65
Şekil 4.4. FM'den alınan örnek görüntü.....	66

Şekil 4.5. Görüntünün kırılması	67
Şekil 4.6. CHT uygulanmış görüntü.....	67
Şekil 4.7. Maskelenmiş görüntü	68
Şekil 4.8. En iyi performansa sahip ilk 3 yöntemin karmaşıklık matrisleri.....	71
Şekil 4.9. Topluluk öğrenmesinin optimizasyonu sonucundaki MCE grafiği	73
Şekil 4.10. AdaBoost algoritması (Hastie, vd., 2001)	74



TABLolar DİZİNİ

Tablo 1.1. Veri depolama ve işleme yapılarına göre sınıflandırma.....	14
Tablo 1.2. Kullanım alanlarına göre sınıflandırma.....	14
Tablo 1.3. IoT sistemlerinde kaynak tahsis çalışmalarının özeti.....	15
Tablo 3.1. Önerilen algoritmadaki parametreler	40
Tablo 3.2. Simülasyonda kullanılan sis düğümlerinin özellikleri	53
Tablo 3.3. Sis düğüm sayısı 1 iken simülasyonların karşılaştırılması	58
Tablo 3.4. Sis düğüm sayısı 2 iken simülasyonların karşılaştırılması	58
Tablo 3.5. Sis düğüm sayısı 3 simülasyonların karşılaştırılması.....	58
Tablo 3.6. Sis düğüm sayısı 8 iken simülasyonların karşılaştırılması	59
Tablo 3.7. Sis düğüm sayısı 13 iken simülasyonların karşılaştırılması	59
Tablo 4.1. Önerilen model için en iyi performansa sahip yöntemler	70
Tablo 4.2. F1 puanlarına göre yöntemlerin karşılaştırılması	72

1. GİRİŞ

Teknolojik gelişmelerin sağlık ve spor alanlarında kullanılması ile geleneksel yöntemlere göre daha verimli çalışmalar elde edilmektedir. Sporda sporcu sağlığı, seyirci sağlığı, oyun içi taktiksel analiz gibi veriler oldukça önemlidir. Saha içinde ve dışında yer alan kişilerin anlık sağlık durumlarını kontrol etmek, taktik analizde bir takım olayları önceden sezmek rekabeti daha kaliteli hale getirmektedir. Bu nedenle teknik ekibin karar verme süreçlerini daha verimli bir şekilde ele alabilmek için sporda bilişim sistemlerinden destek alınması gerekmektedir. Saha içinden veri almak ve bu verilerin işlenerek anlamlandırılması süreçleri için nesnelerin İnterneti (Internet of Things - IoT) tabanlı uygulamalar oldukça uygundur. Çoğu IoT uygulaması, depolama ve bilgi işlem için bulut hizmetlerini kullanmaktadır. Seyircilerin, sporcuların ve diğer aktörlerin sisteme dâhil edildiği büyük ölçekli bir IoT uygulaması için bulut bilişim yöntemleri ile birlikte sis bilişim kullanmak verimliliği artırmaktadır. Bir spor alanı belirli sınırlar içinde olduğu için sis bilişim ile verilerin uzaktaki bulut sunucu yerine yerel ağda işlenmesi ve kullanıcıya gecikme süresi düşük yanıtların iletilmesi sağlanabilmektedir. Sağlık izlemede veya taktik analizde gecikmenin düşük olması teknik ekip ve kulüp doktorlarının olaya erken müdahale edebilmesini sağlamaktadır (Ikram, vd., 2015).

Sis bilişim, çok büyük ölçekli IoT uygulamalarını desteklemek için depolama, bilgi işlem ve ağ kaynaklarını ağ kenarında genişleterek bulut bilişimde merkezi olmayan mimari sunmaktadır. Bulut ve sis ikisi de hesaplama, depolama, uygulama, altyapı ve veri kaynakları sağlamaktadır (Luan, vd., 2015). Aralarındaki en önemli fark erişilebilirlik ve yakınlıktır. Sis, sistemdeki düğümlere yakındır ve genellikle yerel ağda bulunmaktadır. Bulut ise dünyanın herhangi bir yerinde İnternet üzerinden erişilen sunucu veya veri merkezidir. Sis, çeşitli hizmetler için gerekli olan sanal sensörleri ve ağları oluşturmak için sanallaştırmayı kullanarak bulutu genişletmektedir. Bir başka deyişle, sis düğümleri, uç düğümler ile bulut arasında bir katman gibi çalışmaktadır (Mukherjee, vd., 2017). Veri analizine, veri işlemeye ve filtrelemeye yardımcı olmakta ve hassas veriler için güvenliği artırmaktadır. Bulut daha merkezi iken sis dağıtılmış uygulamalarda daha verimli çalışmaktadır.

Sporda taktik analiz olarak en uygun pas analizlerinin belirlenmesi, oyuncu değişikliklerinde antrenörlere yardım edilmesi, taktik düzen bozulduğunda ortaya çıkabilecek sorunlara karşı önlemler alınması gibi durumlar; sağlık açısından ise

sporcuların ve seyircilerin maç esnasında kan basıncı, vücut ısısı, kalp atış hızı gibi verilere bakarak sakatlanma ve hastalık durumlarının tespit edilmesi oldukça önemlidir.

Yakın zamana kadar sporda taktik analiz çıkarımları insan gözlemleri ile elde edilen verilere dayanmaktaydı (Rein ve Memmert, 2016). Günümüzde ise taktik analizlerin teknolojik gelişmeler ile desteklenmesine önem verilmektedir. Buna rağmen, bilişim sistemlerinin desteği birçok sektörde oldukça yüksek oranda iken spor alanında nispeten hala düşüktür. Çoğu spor dallarında irrasyonel argümanlar rol oynasa bile özellikle karar alma süreçlerinde rasyonel argümanlara ihtiyaç artmaktadır (Boon ve Sierksma, 2003). Optimum takım oluşumu da bunlardan biridir. Takımın dizilişi takım sporlarında bir menajerin karar vermesi gereken en önemli konular arasındadır. Başka bir deyişle, bir futbol menajeri için en önemli taktik kararlardan biri, maçın farklı aşamalarında takımın veya dizilişinin mekânsal konfigürasyonunu belirlemektir (Shaw ve Glickman, 2019). Oyuncu dizilişleri takımların genel oyun tarzını etkilemektedir.

Bu çalışmada futbolda sağlık izleme ve taktik analiz için IoT ve sis bilişim tabanlı giyilebilir teknolojiler kullanan bir altyapı modeli ve bu model üzerinde çalışan makine öğrenmesi tabanlı bir taktik analiz uygulaması önerilmektedir. Ayrıca, IoT sistemlerdeki depolama, işlemci, ağ yapısı gibi farklı kaynakların verimli bir şekilde kullanılması ve görevlere tahsis edilmesi için önerilen çözümler incelenmekte ve öncelik kuyruğu kullanan yeni bir çözüm algoritması önerilmektedir.

1.1. Literatür İncelemesi

Bu bölümde literatürdeki sis bilişim ve IoT tabanlı mimariler, sporda sağlık izleme ve taktik analiz yapıları, sis bilişim kullanan mimariler için görev yönetim ve kaynak tahsis yöntemleri ve futbolda taktik analiz çalışmaları incelenmektedir. Günümüzde spor alanında bilişim sistemlerinin kullanımı hızla artmaktadır. Bu nedenle sporda kaliteyi artırmak ve rekabeti güçlendirmek için yeni yöntemler geliştirilmektedir. Literatürde taktik analiz, sağlık izleme, gerçek zamanlı geri bildirim sistemleri, görev yönetim, kaynak tahsisi, düşük gecikme süresi, yüksek verim gibi konularda çalışmalar bulunmaktadır. Bu bölümde IoT tabanlı spor ve altyapı çalışmaları, bulut, sis ve kenar bilişim sistemlerinde kaynak tahsis yöntemleri ve makine öğrenmesi tabanlı taktik analiz uygulamaları incelenmektedir.

1.1.1. IoT Tabanlı Spor ve Altyapı Çalışmaları

Bu bölümde IoT sistemlerde spor ve altyapı problemleri ve bu problemlerin çözümüne ilişkin literatürde son dönemlerde yapılan çalışmalar incelenmektedir. Sporda geri bildirim sistemi olarak yapılan bir çalışmada, egzersiz uygulamalarını incelemek için veriler genişletilmektedir (Baca ve Kornfeind, 2006). Masa tenisi, biatlon ve kürek çekme sporlarında denemeleri yapılmıştır. Masa tenisinde sistem atış doğruluğunu tespit etmek için topun masa üzerindeki etkilerini tespit etmektedir. Biatlonda bir lazer konumlandırma sistemi ile tüfek namlusunun atıştan önceki ve sonraki durumunu analiz etmektedir. Kürek çekmede ise sistem kürek çekmek için harcanan çaba hesaplanmaktadır. Bu çalışmada sistem spora bağlı olduğu için esnek değildir. Başka bir çalışmada, bir WSN yardımı ile maraton koşucularının sağlık durumları izlenmektedir (Pfisterer, vd., 2006). Ancak, bu çalışmalar gerçek zamanlı değildir.

Gerçek zamanlı yapılan çalışmalar olarak; bisikletçileri izleyen bir WSN'de algılanan veriler önceden tanımlanmış veri kümeleriyle karşılaştırılarak bisikletçiler grubuna geri bildirim sağlamaktadır (Vales-Alonso, vd., 2010). Sistem bisiklete grubunun oluşumunun değişmesini, bölünmelerini veya hızlarını artırmalarını söyleyebilmektedir. Bu çalışmada çevresel veriler dikkate alınmamıştır. Rüzgâr hızı, sıcaklık, nem gibi çevresel faktörler de bu duruma etki edebilmektedir. Çevresel faktörleri değerlendiren dinamik bir çalışmada, karar verme yöntemi olarak spline tabanlı sayısal yaklaşım teknikleri önerilmektedir (Vales-Alonso, vd., 2010). Sporcunun fiziksel koşullarının yanında arazinin koşulları da değerlendirilmekte ve sporcuyu yönlendirilmektedir. SVM gibi bazı yapıların önceden çok fazla veriye ihtiyaç duymasından dolayı spline yaklaşımı kullanılmıştır. Bu sayede çok değişkenli problemler çözülebilmekte ve hesaplama yükü düşük olmaktadır. Ancak, bu çalışmada koşu, bisiklet gibi sporlarda antrenman durumları için önerilmiştir. Sağlık izleme ve sporcunun nereye gideceği konusunda yönlendirmek için kullanılmaktadır. Önerilen model ise futbol sporunu dikkate alarak sporcuyu sağlığının yanı sıra maç esnasında taktik analiz yaparak teknik ekibe yardımcı olmaktadır.

Akıllı şehir planlanmasında stadyumdaki verilerden elde edilen acil durumların sağlık kuruluşlarına iletildiği, güvenli ve saldırıları algılayabilen yapılar geliştirilmiştir. Ayrıca, IoT hizmet paydaşlarının açık bir IoT ekosistemine özgürce katılmalarını katkıda bulunmalarını ve bu yapıdan faydalanılmasını sağlayan bir

çerçeve oluşturulmuştur (Kubler, vd., 2017). Bir başka çalışmada, gelecekte sis hesaplama tabanlı yapıların çokça kullanılacağından ve gerçek zamanlı uygulama mimarilerinin geliştirilmesi gerektiğinden bahsetmiştir. Ayrıca, sis bilişim gibi dağıtık kaynaklarda kaynak tahsisi kritik bir konu olduğu ve sis tabakasında eş zamanlı işlemlerin uygun şekilde planlanması gerektiği açıklanmıştır (Naha, vd., 2018).

Hastanelerde, spor alanlarında veya diğer alanlarda uygulanabilir olan sağlık izleme çalışmaları da mevcuttur. Sporcuların sakatlanması gibi durumları algılayıp antrenöre gönderen sistem önerilmektedir (Ikram, vd., 2015). Bu sistemde M2M standardı kullanılmaktadır. Ayrıca, bir WBAN içinde sensörler barındıran algılama katmanı, ZigBee teknolojisi kullanılan ağ katmanı ve tüm verileri kontrol etmesi için bulut servisleri kullanan bulut katmanı olmak üzere üç katman bulunmaktadır. Sis bilişim kullanan bir başka sağlık izleme sistemi hastaneler için önerilmektedir (Paul, vd., 2018). Bu modelde WSN ve WBAN kullanılarak algılanan veriler sis katmanına iletilmektedir. Sis katmanında veriler öncelik işleminden geçirilmekte ve görevler sis düğümlerine uygun şekilde dağıtılmaktadır. Böylece daha önemli işlemlere öncelik verilerek daha hızlı sonuçlar alınabilmektedir. Sis bilişimin yapısı gereği bazı güvenlik durumları mevcuttur. Ancak, sis ile bulut arasındaki güvenlik işlemleri gizli anahtar kullanılarak sağlanmaktadır. Modelin uygulaması mevcut olmayan yazılımlar ve var olan yazılımların karmaşıklığı gibi engellerden dolayı kısmi olarak gerçekleştiği vurgulanmaktadır (Paul, vd., 2018). WSN kullanan mimariler kablosuz iletişimden dolayı güvenlik sorunları ile karşı karşıya kalabilmektedir. Bu alandaki saldırı çeşitleri, kullanılan protokoller açıkları ve yapılması gereken güvenlik işlemleri değerlendirilmelidir (Karakaya ve Akleylek, 2018).

Sis ağının kurulması, ölçeklendirilebilmesi, veriminin artırılması ve maliyetinin azaltılması için SDN ve NFV tabanlı yapılardan yararlanılmaktadır (Krishnan, vd., 2020). SDN, küresel ağın daha akıllı, soyut ve işlevsel olmasını sağlayan yazılım tabanlı olarak gerçekleştirilen teknolojiler, modeller ve protokolleri kapsamaktadır. NFV ise ağ işlevlerini sanallaştırmak, ağ entegrasyonunu artırmak için geliştirilen yüksek hacimli yapıları kapsamaktadır. Sis ağının görevi, sisin her bileşenini birbirine bağlamaktır. Bu ağları yönetmek ve sürdürülebilir bağlantı elde etmek zordur. Kolay yönetim, ölçeklenebilirlik gibi sistemin verimini artıran işlemler için SDN ve NFV teknikleri kullanılmaktadır (Yi, vd., 2015). Her bir sis düğümünün multi-hop (çok atlamalı – düğümden düğüme) iletişim kurabilmesi gerekmektedir. Bu nedenle her

düğüm sis ağı içinde yönlendirici görevi görmektedir. Sis platformunda verimli yönlendirme optimizasyonunu sağlamak için SDN kullanılmaktadır. IoT platformunda sis düğümlerinin etkin yönlendirme yapabilmesi için hiyerarşik SDN tabanlı sis mimarisi çalışmaları yapılmaktadır (Okay ve Ozdemir, 2018). NFV, sanallaştırma ile yazılımı donanımdan ayırarak ağ işlevlerinin daha hızlı gerçekleşmesini sağlamaktadır. Ağın maliyetini en aza indirmek için NFV tabanlı hibrit bulut ve sis sistemleri geliştirilmektedir (Mouradian, vd., 2019).

Bir başka çalışmada sis ağının servis kalitesi için önemli ölçümler yapılmaktadır. Bunlar bağlantı, güvenilirlik, kapasite ve gecikme ölçümleridir. Bağlantı ölçümünde ağın hızlı ve güvenilir olması beklenmektedir. Güvenilirlik ölçümünde, verilerin doğru ve eksiksiz iletilmesi beklenmektedir. Ancak, bunu gerçekleştirmek için veriler bir takım kontrol (veri bütünlüğünün sağlanması) işlemlerinden geçirilmelidir (Madsen, vd., 2013). Bu da sisteme ek gecikme yükü getirmektedir. Kapasite ölçümünde, ağın bant genişliği ve depolama alanlarını verimli kullanması beklenmektedir. Bu yüzden önce veri toplanmakta, sonra hesaplama gerçekleştirilmektedir. Böylece önbellek farklı işler için yeniden tasarlanabilmektedir (Wang, vd., 2014). Bu işlem de gecikmeye neden olmaktadır. Gecikme ölçümünde ise düğümlerin hesaplama ve depolama işlemlerinden hızlı yanıt alması beklenmektedir. Akış madenciliği, karmaşık olay işleme gibi teknikler gelecek işlemleri öngörerek gecikmeyi azaltabilir.

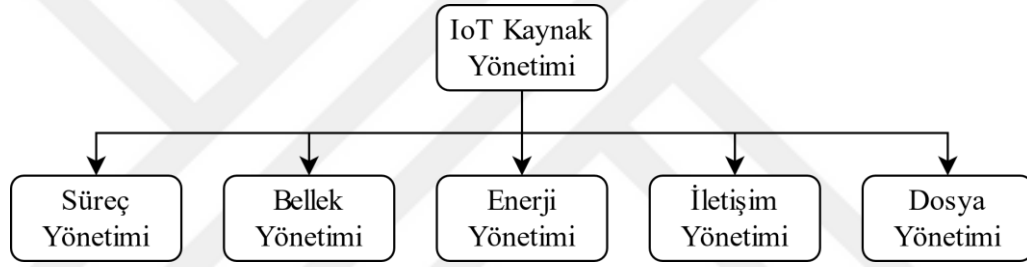
IoT geliştiricilerinin uygulamalarını sis bilişim platformuna taşımalarını kolaylaştırmak için ara yüz ve programlama modellerine ihtiyaç vardır. Ortamdaki bileşenlerin farklı platformlara uyum sağlayabilmesi için dinamik, hiyerarşik kaynakların buna göre optimize edilmesi gerekmektedir (Yi, vd., 2015). IoT uygulamalarında uç düğümler hareketli olabilmektedir. Bant genişliği, depolama, hesaplama, gecikme durumları dinamik olarak değiştiği için kaynak yönetimi durumunda uç düğümün hareketli olması zorluklar oluşturmaktadır. Kaynak yönetimi ve paylaşımı için işlemci, bant genişliği, depolama gibi heterojen kaynakları sis düğümlerine uygun şekilde paylaşır yapılar üzerine çalışmalar yapılmaktadır (Liu, vd., 2014).

1.1.2. IoT Sistemlerde Kaynak Yönetim Çalışmaları

Bu bölümde IoT sistemlerde kaynak yönetimi ve tahsis problemleri ve bu problemlerin çözümüne ilişkin literatürde son dönemlerde yapılan çalışmalar

incelenmektedir. IoT sistemlerde çok sayıda görev ve istek tarafından kullanılan sunucu kaynakları, iletim hattı, kablosuz kaynaklar gibi ortak kullanılan her kaynak paylaşılmak zorundadır. Bu paylaşımın verimli ve etkili yapılması oldukça önemlidir. Kaynak yönetimi üzerine en büyük problemler ve zorluklar arasında yüksek hesaplama karmaşıklığı, yüksek iletim gecikmesi, kapsam darlığı, kısıtlı kaynakların paylaşımı, düşük enerji verimliliği, QoS gereksinimleri, verimsiz kaynak paylaşımının yol açabileceği güvenlik sorunları sayılabilmektedir.

Kaynak yönetiminin alt dalları Şekil 1.1’de gösterilmiştir (Musaddiq, vd., 2018). Son dönemlerde yapılan çalışmalarda, süreç yönetimi için "görev zamanlama" ve "görev yükü boşaltma", bellek yönetimi için "kaynak tahsisi", enerji yönetimi için "enerji verimliliği", iletişim yönetimi için "yük dengeleme", dosya yönetimi için ise "sanallaştırma" ve "depolama" yaklaşımları oldukça popülerdir.



Şekil 1.1. IoT'de kaynak yönetim süreçlerinin sınıflandırılması

Standart IoT, NB-IoT, bilişsel IoT, M2M gibi kaynak kısıtlı sistemler için kaynak yönetiminin ve tahsisinin önemli bir yeri vardır. Bu sistemlerde kaynak tahsisi ile ilgili son dönemlerde yapılan çalışmalarda araştırmacılar, çok sayıda kullanıcıya hizmet veren, kaynakları tam verimle kullanmayı sağlayan, düşük gecikmeli ve yüksek performanslı çerçeveler için makine öğrenmesi yaklaşımları kullanma eğilimindedir. Bu konuda yaşanan her türlü zorluk farklı çalışmalarda ele alınmakta ve zorluğun üstesinden gelmek için etkili yöntemler geliştirilmektedir.

IoT sistemlerinde kaynak tahsisi bulut, sis ve kenar bilişim gibi sunucu kaynakları üzerinde yapılabileceği gibi frekans kanalları, spektrum bandı ve iletim bank genişliği gibi iletim kaynaklarında da yapılabilmektedir. Kaynak tahsis problemini çözmek için ortaya konan çalışmalar çok sayıda parametreye göre incelenmekte ve analiz edilmektedir.

1.1.2.1. Bulut Bilişimde Kaynak Tahsis Yöntemleri

Bulut bilişim sistemlerinde kaynak tahsis problemini gidermek için literatürde çok sayıda çözüm önerilmektedir. (Choi ve Lim, 2016)'da kombinatorial açık artırma (combinatorial auction) yöntemi kullanılarak bulut sistemlerin kaynak tahsisinde hizmet kalitesi standartlarını sağlayan yöntem önerilmiştir. Verimli kaynak tahsisi için acil durumlar dikkate alınarak görevlerin son tarihinin ihlal olasılığı ve beklenen değeri belirlenmiştir. Karşılaştırmalar ile daha yüksek iş tamamlama başarı oranına sahip olduğu sonucuna varılmıştır. (Ramirez, vd., 2019)'da bulut sistemlerinde farklı nesnelerin ortak kaynakları paylaşması için bir algoritma önerilmiştir. Ağdan topladığı bilgiler ile makine öğrenmesi tabanlı bir algoritma kullanan bir yapay zekâ denetleyicisi merkezi yönetimi sağlamıştır. MQTT protokolünün daha düşük bir bant genişliği tüketmesi için ve makineler arası ilişkilerin kurulması için algoritmalar tasarlanmıştır. Geleneksel IoT iletişim sistemlerine göre daha yüksek performans ve kullanılabilirliğe sahip olduğu belirtilmiştir.

(Wang, vd., 2019)'da görev ön planlaması ile WebIDE bulut sisteminin gerçekleştireceği görevlerin tahmini ve buna dayalı olarak mevcut kaynakların tahsisi için bir yöntem önerilmiştir. Görevler üç sınıfta (yürütme durumu, yürütme işlemleri, bulut kaynak gereksinimi) gruplandırılmıştır. Daha sonra, Markov durum geçişi olasılık matrisi ile görevler arası geçiş olasılığı hesaplanmış ve tahmin modeli tasarlanmıştır. Son olarak, karınca kolonisi algoritması ile ön planlama yapılarak bulutta kaynak tahsisi sağlanmıştır. Bu sayede görev yanıt süresi azaltılmış ve kaynak kullanımı verimli hale getirilmiştir. (Sangaiah, vd., 2020)'de bulut sistemlerde görevlerin maliyetini ve çalışma süresini azaltmak ve görevlere en uygun kaynak tahsisini yapmak için balina optimizasyon algoritması tabanlı model önerilmiştir. Modelde, iki balina arasındaki mesafe algoritması, bir balinanın üretilen değere göre avının etrafını sarması işlevi için spiral algoritması, bir balınayı avına doğru hareket ettirmek için küçülme algoritması ve bir balınayı rastgele seçilen bir balınaya doğru hareket ettirmek için arama avı algoritması yeniden tasarlanmıştır. Böylece kaynak tahsis problemleri tespit edilerek en uygun optimizasyon işlemi gerçekleştirilmiştir.

1.1.2.2. Sis Bilişimde Kaynak Tahsis Yöntemleri

Çok sayıda sis bilişim sistemlerine özgü kaynak tahsis çalışmaları yapılmaktadır. Bu alandaki eğilim kaynak tahsisini akıllı ve verimli şekilde gerçekleştiren, sis

kaynaklarını maksimum oranda kullanan ve algoritmalar ve matematiksel modeller ile desteklenen çerçeveler geliştirmek üzerinedir. (Abedin, vd., 2018)'de güvenilir ve düşük gecikmeli iletişim ve gelişmiş mobil geniş bant için artan QoS gereksinimleri dikkate alınarak kullanıcı ilişkilendirmesi ve kaynak tahsisi yöntemi önerilmiştir. Analitik hiyerarşi süreci ile IoT uygulamalarının QoS gereksiniminin öncelikleri belirlenmiştir. Daha sonra, IoT cihazlarında ilişki kurmak için sis ağları koordinatöründe ikili karşılaştırma matrisi oluşturulmuş ve iki taraflı eşleştirme yapılmıştır. Bu eşleştirmede gecikmeden dolayı oluşan dışsallıklar için en uygun kaynak tahsisi stratejisinin uygulanması gerçekleştirilmiştir. Bu yöntem ile faydalı ve verimli kaynak tahsisi ve kullanıcı ilişki kararlılığı sonucuna ulaşılmıştır. (Gu, vd., 2018)'de sistem optimizasyonu ve kullanıcı memnuniyetini artırmak amacıyla radyo veya hesaplamalı kaynak tahsis problemi incelenmiştir. Kaynakların en uygun şekilde tahsis edilmesi için SPA oyun algoritması ve buna ek olarak dış etkilerin oluşturduğu uyuşmazlığı ortadan kaldırmak için UOC yöntemi önerilmiştir. Böylece en uygun kaynak tahsisi stratejisi ile dışsallıkların giderilmesi sağlanmıştır. Ayrıca, benzer çalışmalara göre daha verimli çalıştığı belirtilmiştir.

Fog-RAN kaynak tahsisi için (Nassar ve Yılmaz, 2019)'da MDP kullanılmıştır. Buradaki sorunların giderilmesi için ise SARSA, beklenen SARSA, Q-öğrenme ve Monte Carlo gibi pekiştirmeli öğrenme yöntemleri kullanılarak optimum politikayı keşfetmek için bir yöntem önerilmiştir. Bu yöntem ile görevlerin sis düğümlerindeki kaynaklardan maksimum oranda fayda sağladığı ve düşük gecikme için iyi performans elde edildiği belirtilmiştir. (Banaie, vd., 2020)'de ağ geçitlerinde oluşabilecek tıkanıkları engellemek için MCDM tabanlı yük dengeleme yöntemi önerilmiştir. Bu tıkanıklık güvenlik sorunlarına yol açabilmektedir. Yük dengeleme işlemi ve çok amaçlı optimizasyon sorunu AHP yaklaşımı ile çözülmüştür. Bu sayede tıkanıklıklar önlenerek ve yük dengeleme sağlanarak kullanıcılar için güvenilir ve hızlı sonuçlar elde edilmiştir. (Huang, vd., 2020)'de sis bilgi işlem ek yükünü en aza indirmek ve QoS gereksinimlerini sağlayan kaynak tahsis şeması önerilmiştir. Bu şema üç yöntem içermektedir. QoS parametreleri ve görev önceliklerini belirlemek için AHP tabanlı yöntem geliştirilmiştir. Tahsis edilecek olan kaynak bloklarının öncelik, memnuniyet gibi parametrelere göre belirlenmesi için kaynak bloğu tahsis yöntemi tasarlanmıştır. Kurulan ilişki optimizasyonu için ikili eşleştirme oyunu (biletral matching game) yöntemi oluşturulmuştur. Bu yöntemlerden oluşan kaynak tahsis şeması ile verimli bir

yük dengeleme sağlanabileceği ve kullanıcılara QoS sağlayarak ek yüklerin azaltılabileceği vurgulanmıştır.

1.1.2.3. Kenar Bilişimde Kaynak Tahsis Yöntemleri

Kenar bilişim kaynak kısıtlı yapısı itibari ile verimli kaynak tahsisine en çok gereksinim duyan hesaplama yaklaşımlarından biridir. Uç sistemler uç sunucularla desteklenerek daha yüksek veri işleme yapabilmektedir. Kenar sistemler üzerine yapılan çalışmalar da sis ve bulutta olduğu gibi genellikle verimli kaynak yönetimi, yük dengeleme, öncelikli görevler şeklindedir. (Na, vd., 2018)'de Lagrange ve Karush-Kuhn-Tucker (KKT) yapısına dayalı kenar sunucular ve kenar ağ geçitleri arasında kaynak yönetim şeması önerilmiştir. Bu şema kapsamında kenar sunucular ve ağ geçitleri için en verimli kaynak tahsisini belirleyen bir algoritma ve ağ geçitlerinin iş yükünü dikkate alan ve onların arasında planlama yapan bir sezgisel algoritma sunulmuştur. Şemanın adaletli iş yükü ve verimli kaynak tahsisi sağladığı belirtilmiştir. (Deng, vd., 2020)'de güvenilirlik bakımından SLA'ya uymak için en iyi politikayı bulmayı sağlayan kaynak tahsisi algoritması ve kaynakların hizmetlere uyarlanmasını sağlayan MDP modeli önerilmiştir. Bu yöntemler ile pekiştirmeli öğrenme oluşturularak güvenilirlik bakımından üst düzey kaynak tahsisi sağlanmıştır. Çalışmada yöntemin benzer temel yaklaşımlara kıyasla daha iyi performans gösterdiği belirtilmiştir. (Jiang, vd., 2020)'de iletim gücü, kaynak tahsisi, yük dengeleme ve gecikmeyi azaltmak için DRL tabanlı kaynak planlama şeması önerilmiştir. Şemada yüksek boyutlu verileri sıkıştırmak ve azaltmak için 2r-SAE algoritması kullanılmıştır. Daha sonra ASA özelleştirilmiş yapısı kullanılarak DRL'nin verimli arama yapabilmesi sağlanmıştır. Sonuç olarak, DRL yapısının ağı eğitmesi ve en uygun yük dengesini bulması sağlanmıştır. Bu şemanın optimum performansla yaklaştığı ve işlem süresini azalttığı belirtilmiştir.

(Xu, vd., 2020)'de akıllı şehir uygulamalarındaki MEC sistemlerde enerji ve kaynak kısıtlı işlem gecikmesini optimize etmek için RJCC algoritması önerilmiştir. Hesaplama maliyetini optimize etmek için çoklu segmentlerde Q-öğrenme tabanlı yük boşaltma algoritması ve uç platformlarda ise Lagrange tabanlı göç algoritması geliştirilmiştir. Bu sayede yöntemin düşük gecikme süresini korunduğu, enerji tüketimi ve yük boşaltma sorunlarında daha iyi performans gösterdiği ve QoS garantisi sağladığı belirtilmiştir. (Liu, vd., 2020)'de makine öğrenmesi yaklaşımları kullanılarak kenar sistemlerde dinamik kaynak tahsisi için algoritmalar önerilmiştir.

Kullanıcılar öncelik parametreleri ile k-ortalama algoritmasına göre gruplanmıştır. En yüksek öncelikli grup hesaplama islerini kenar sunucusunda, en düşük öncelikli grup ise yerel olarak gerçekleştirmiştir. Diğerleri ise MDP modeline göre işlenmiştir. Yüksek boyutlu verilerle basa çıkmak ve en uygun politikayı bulmak için DQN kullanılmıştır. Bu algoritmaların benzer yaklaşımlardan daha düşük maliyet ile benzer performanslar gösterdiği belirtilmiştir. (Tang ve Hu, 2020)'de MEC kullanan IoT sistemlerdeki heterojen ağlarda verimli enerji tüketimi, optimum kaynak tahsisi ve yük hesaplama yönetimi için şema önerilmiştir. Bu şema alt problemlere bölünmüştür. Bunlar, optimum CPU frekans çözümü, SCA tabanlı logaritmik yaklaşım kullanan güç yönetim algoritması ve dağıtılmış bir yük hesaplaması ve kaynak tahsisidir. Bu problemleri gidermek için algoritmalar geliştirilmiştir. Bu sayede, düşük gecikme süresi, verimli hesaplama karmaşıklığı ile işlemlerin gerçekleştirildiği belirtilmiştir.

(Wang, vd., 2020)'de iş birlikçi kenar bilişim tabanlı ve ortalama görev bitirme süresini düşüren bir görev tahsis şeması önerilmiştir. Bu şemada genetik algorithmada genetik değişim (crossover) ve mutasyon stratejisini uygulayarak optimizasyon sürecini iyileştirmek için TCA-IPSO algoritması tasarlanmıştır. Bu sayede, ortalama görev bitirme süresini QBTD algoritmasına göre çok daha azalttığı ve büyük IoT sistemlerine uygulanabileceği belirtilmiştir. (Xiong, vd., 2020)'de ortalama görev bitirme süresi ve ortalama kaynağa gereksinimi en aza indirmeyi hedefleyen verimli kaynak tahsisi politikası önerilmiştir. MDP yaklaşımı ile problem formülize edilmiş ve problemi çözmek için DQN tabanlı algoritma tasarlanmıştır. Bu yaklaşımın standart DQN algoritmasına ve benzer çalışmalara göre daha az kaynak kullanımı ve daha düşük görev bitirme süresi ile daha verimli kaynak tahsisi sağladığı belirtilmiştir.

(Zhang, vd., 2020)'de MEC ile birlikte çalışan C-RAN yapılarında ağ enerji verimliliğini optimize etmek amacıyla görev yük boşaltma ve kaynak tahsis yaklaşımı önerilmiştir. Karışık tam sayılı doğrusal olmayan programlama problemi ile incelenen sorun için dört alt problem oluşturulmuştur. Bunlar, Lyapunov optimizasyon teorisi, görev yük boşaltma ve kaynak planlaması için dışbükey ayrıştırma yöntemleri, gerçek problem ve radyo kaynak tahsisi için eşleşme oyunu alt problemleridir. Çalışmada yöntemlerin enerji verimliliği ve gecikme süresi bakımından iyi sonuçlar verdiği vurgulanmıştır.

Endüstrilerde kullanılan IoT sistemlerde veri gizliliği sorunları nedeniyle makine öğrenimi yaklaşımlarının kullanımı kısıtlanmaktadır. (Khan, vd., 2020)'de

kaynak optimizasyonunu sađlayan ve endüstrilerde gizli bilgileri koruyarak özellikleri ortaya çıkaran DFL yapısı önerilmiştir. İlişki ve kaynak tahsisi için dışbükey optimizasyon yapısı ve ilişki ve kaynak tahsis deđişkenlerini elde etmek için yuvarlama tekniđi kullanılmıştır. Tüm kenar bilişim özellikli küçük ve büyük baz istasyonlarının arasında hafif bir kimlik dođrulama yaklaşımı kullanılmıştır. DFL tabanlı yaklaşımın, akıllı endüstriler için rastgele kaynak tahsisi ve ilişkilendirme şemalarına göre daha verimli çalıştığı ve gizliliđi koruduđu belirtilmiştir. (Lee, vd., 2020)'de buluttaki bir koordinatöre iletilen ve MEC sistemlerde çalışan birleşik öğrenme modelinin sonuçlarına göre bir Stackelberg oyun modeli önerilmiştir. IoT sensörlerinden alınan veriler çok sayıda MEC sistemi ile yerel olarak eğitilmiş ve sonuçlar bulut koordinatörüne iletilmiştir. Bulut sistem bu sonuçlar ile küresel bir piyasa davranış modeli oluşturmuştur. Memnuniyet ve maliyet arasında SE noktası oluşturulmuştur. Bu yaklaşımın tüm piyasa katılımcıları için faydayı maksimize ettiği sonucuna ulaşıldığı vurgulanmıştır.

(Pham, 2020)'de bir NFV destekli IoT sisteminde ađ geçitlerinin optimum yerleşimi (deployment), optimum kaynak tahsisini ve optimum yönlendirmeyi sađlayan optimizasyon modelleri önerilmiştir. Bunlar, geçit (gateway) yerleştirme ve çok atlamalı yönlendirme için GMA, servis yerleştirme için SP1A ve SP2A algoritmalarıdır. Bu algoritmalar benzer işlemlerin optimizasyon modelleri (GMO, SP1O, SP2O) baz alınarak geliştirilmiştir. Çalışmada, SP1A ve SP2A arasında hesaplama ve enerji maliyetleri açısından farklıklar olduđu belirtilmiştir. Ayrıca, bu modeller kullanılarak daha az hesaplama süresi ile optimuma yakın sonuçlar elde edildiđi vurgulanmıştır.

1.1.2.4. Bulut, Sis ve Kenar Bilişimde Kaynak Tahsis Yöntemleri

Sis, bulut ve kenar bilişim sistemlerinin hepsinde kullanılabilen kaynak tahsis çalışmaları da yapılmaktadır. (He, vd., 2018)'de yeşil IoT'de QoE sađlamak için kaynak tahsisi ve iletim hızı iyileştirme modelleri önerilmiştir. Bu kapsamda SPT, SPTC ve DQN algoritmaları kaynak tahsisi için kullanılmıştır. İçerik merkezli IoT sistemlerde heterojen kaynaklar için tek bir algoritma yerine üç algoritma birlikte kullanılmıştır. Böylece, çalışmada içerik merkezli IoT sistemlerinde verimli kaynak tahsisi ve düşük iletim hızı ile yüksek QoE performansı elde edildiđi bildirilmiştir. (Zhang, vd., 2018)'de yeşil IoT sistemlerde enerji maliyetlerini optimize etmek için çevrimiçi ve iki zaman ölçekli kaynak tahsisi algoritması geliştirilerek Lyapunov

tabanlı şema önerilmiştir. Bu şema geniş zamanda enerji yönetimi, dar zamanda veri toplama kontrolü ve çapraz entropi tabanlı kanal planlama olmak üzere üç bölümden oluşturulmuştur. Zaman içinde enerji ve spektrum sağlanması amaçlanmıştır. Çalışmada yapılan analizlerin ve algoritmanın verimliliğinin doğrulandığı belirtilmiştir.

(Liu, vd., 2018)'de heterojen IoT sistemler için NOMA tabanlı ve enerji verimli kaynak tahsisi önerilmiştir. Bilişsel radyo ağlarında mobil kullanıcılar ve IoT kullanıcıları için aşamalı bir kaynak tahsisi planlanmış ve bu plan RNN tabanlı bir algoritma ile çözümlenmiştir. Bu algoritmanın hızlı yakınsama ve düşük hesaplama karmaşıklığı ile optimum çözümü ürettiği ve benzer yaklaşımlardan güç ve enerji bakımından daha verimli olduğu belirtilmiştir. NB-IoT sistemlerde hesaplama karmaşıklığını azaltmak ve geniş kapsam için zaman farkı ve tekrarlama işlemleri kullanılmaktadır. (Malik, vd., 2018)'de NB-IoT sistemler için parazitlere duyarlı ve hızı en yükseğe çıkaran kaynak tahsis şeması önerilmiştir. Veri hızının üst sınırı için üretilen çerçevede uydu-yer bağlantı hızının 89.2 Kbps ve yer-uydu bağlantı hızının 92 Kbps olduğu belirtilmiştir. Kaynak tahsisinde optimum çözümü bulmak için işbirlikçi ve yinelemeli algoritma geliştirilmiştir. Bu algoritma ile iyileşme ve enerji tasarrufu sağlandığı bildirilmiştir. (Su, vd., 2018)'de bir HAP ile çok sayıda sensörden oluşan IoT sisteminde Stckelberg oyun yaklaşımı tabanlı kaynak tahsis şeması önerilmiştir. Bu şema, iletim maliyetini ve enerji transfer maliyetini en aza indirmek için tasarlanmıştır. Problemin analizi için Steckelberg oyun modeli ve amaç fonksiyonunu oluşturmak için ortalama alan oyunu yaklaşımı kullanılmıştır. Yöntemin HAP ve sensörler için optimum güç kontrolü sağladığı ve maliyetleri azalttığı belirtilmiştir.

(Dou, vd., 2019)'da IoT ağları için parazit ve girişimleri önleyen otomatik kaynak tahsis modeli önerilmiştir. Modelde düşük güç, bant genişliği sınırı ve parazitleri önlemek için yayılma zamanı (spreading-time) teknolojisi ve OFDM yöntemi kullanılmıştır. Ek olarak, modelde sıkışma (jamming) olmadan kaynak tahsis etmek için ortak özyinelemeli bir algoritma geliştirilmiştir. Çalışmada, modelin etkili olduğu ve kaynak sınırlı IoT sistemlerine uygulanabileceği belirtilmiştir. (Hussain, vd., 2020)'de kümelenmiş IoT ağları için slot planlaması yapan dağıtık ve Q-öğrenme tabanlı algoritma önerilmiştir. Bu algoritma ile çok kanallı sistemlerde parazit önleme ve SIR iyileştirmek amaçlanmıştır. Çalışmada her bir iletişim cihazına, uygun bir SIR

seviyesinde kanal tahsis yapıldığı ve IoT ağının çarpışma olmayan bir iletme yakınlştırıldıđı belirtilmiştir. (Liu, vd., 2019)'da kablosuz IoT ağlarında spektrum ve enerji gibi sınırlı kaynakların verimli tahsis için düşük karmaşıklıđa sahip bir çerçeve önerilmiştir. Çerçevde kullanıcılar kanal bazlı olarak gruplandırılmış ve çoktan bire eşleştirme oyunu olarak ele alınmıştır. Bu kapsamda MDP tabanlı bir güç tahsis algoritması ve bir kanal tahsis algoritması geliştirilmiştir. Çalışmada algoritmaların verimli olduđu, rastgele kanal atama ve merkezi güç tahsis yaklaşımlarından daha üstün olduđu ve düşük karmaşıklık sağladıđı belirtilmiştir.

(He, vd., 2020)'de büyük ölçekli kaynak tahsis problemi için ADMM tabanlı kaynak tahsis yaklaşımı önerilmiştir. Optimum enerji verimliliđi için Dinkelbach algoritması ve denetleyici ağ gecikmesini optimize etmek için Jacobian ve ADMM paralel algoritması kullanılmıştır. Ek olarak, denetleyici ile bölgesel IoT sunucuları arasındaki hedef çelişkisini gidermek için Steckelberg oyun yaklaşımı kullanan iki katmanlı iteratif kaynak tahsis yaklaşımı kullanılmıştır. Çalışmada, bu algoritmaların ölçeklenebilirlik ve optimuma yaklaşma bakımından verimli olduđu belirtilmiştir. (Yang, vd., 2020)'de ortak kanal atama ve iletim gücünü optimize etmek için çok etmenli DRL tabanlı kaynak tahsis şeması önerilmiştir. Ağ performansını ve iletim başarımını artırmak ve dağıtık ortamda ortak çalışmayı sağlamak için PER kullanılmıştır. Bu şemanın enerji tasarrufu ve QoS gereksinimlerini karşılayarak kaynak yönetim performansını artırdığı bildirilmiştir.

(Librino ve Santi, 2020)'de spektrum bandında girişimlerin yüksek olduđu zorlu bir senaryo üzerinde paylaşımlı iletişimde mobil kaynakları tahsis etmek için bir grafik tabanlı algoritma ve metot önerilmiştir. Ultra güvenilir düşük gecikmeli iletişimde ortaya çıkan sorunları gidermek için yöntem akıllı fabrika ortamındaki zorlu koşullarda gerçekleştirilmiştir. Çalışmada, yöntemin spektral verimliliđinin benzer yaklaşımlardan çok daha yüksek olduđu ve yoğun ağlarda hızlı kaynak tahsis sağladıđı belirtilmiştir. (Chen, vd., 2020)'de STN için kaynak yönetimi yaklaşımı önerilmiştir. Bu yaklaşımda, kaynak kullanımını maksimize etmek için bir MCPR algoritması ve toplam sistem gecikmesini minimize etmek için Markov yaklaşım analizini kullanan MCRC algoritması geliştirilmiştir. Kaynakların yönetimi için katmanlı mimari ve kuyruklama yöntemi kullanılmıştır. Bu algoritmaların kaynak kullanımını düşük ölçüde artırdığı, gecikmeyi büyük ölçüde azalttığı belirtilmiştir.

IoT sistemlerinde kaynak tahsisine yönelik son çalışmaların uygulama alanları

ve veri depolama ve işleme yapıları kategorilere ayrılmaktadır. Veri depolama ve işleme yapılarına göre yapılan sınıflandırma Tablo 1.1’de gösterilmektedir.

Tablo 1.1. Veri depolama ve işleme yapılarına göre sınıflandırma

Veri Depolama ve İşleme Yapısı	Çalışmalar
Bulut	(Choi ve Lim, 2016), (Ramirez, vd., 2019), (Wang, vd., 2019), (Sangaiah, vd., 2020)
Sis	(Abedin, vd., 2018), (Banaie, vd., 2020), (Huang, vd., 2020), (Karakaya ve Akleyek, 2021)
Kenar	(Na, vd., 2018), (Deng, vd., 2020), (Jiang, vd., 2020), (Liu, vd., 2020), (Tang ve Hu, 2020), (Wang, vd., 2020), (Xiong, vd., 2020), (Khan, vd., 2020)
Bulut-Sis	(Gu, vd., 2018), (Nassar ve Yilmaz, 2019)
Bulut-Kenar	(Zhang, vd., 2020), (Lee, vd., 2020), (Pham, 2020), (Xu, vd., 2020)
Genel	(He, vd., 2018), (Liu, vd., 2018), (Malik, vd., 2018), (Su, vd., 2018), (Zhang, vd., 2018), (Dou, vd., 2019), (Hussain, vd., 2020), (Liu, vd., 2019), (He, vd., 2020), (Yang, vd., 2020), (Librino ve Santi, 2020), (Chen, vd., 2020)

Tablo 1.1’deki "Genel" terimi tüm kaynak tedarikçi türlerinin ortak kullanıldığı durumları ifade etmektedir. Kaynak tahsis yöntemlerinde kullanım alanlarına göre yapılan sınıflandırma ise Tablo 1.2’de gösterilmektedir.

Tablo 1.2. Kullanım alanlarına göre sınıflandırma

Kullanım Alanları	Çalışmalar
Akıllı Endüstri	(Khan, vd., 2020)
Pazar Analizi	(Lee, vd., 2020)
Akıllı Şehir	(Xu, vd., 2020)
Yeşil IoT	(He, vd., 2018), (Liu, vd., 2018), (Zhang, vd., 2018)
NB-IoT	(Pham, 2020), (Malik, vd., 2018), (Dou, vd., 2019), (Liu, vd., 2019)
Bilişsel IoT	(Yang, vd., 2020), (Chen, vd., 2020)
Çoğu IoT Sistemleri	(Choi ve Lim, 2016), (Ramirez, vd., 2019), (Wang, vd., 2019), (Sangaiah, vd., 2020), (Abedin, vd., 2018), (Banaie, vd., 2020), (Huang, vd., 2020), (Gu, vd., 2018), (Nassar ve Yilmaz, 2019), (Na, vd., 2018), (Deng, vd., 2020), (Jiang, vd., 2020), (Liu, vd., 2020), (Tang ve Hu, 2020), (Wang, vd., 2020), (Xiong, vd., 2020), (Zhang, vd., 2020), (Su, vd., 2018), (Hussain, vd., 2020), (He, vd., 2020), (Librino ve Santi, 2020), (Karakaya ve Akleyek, 2021)

IoT sistemlerdeki kaynak tahsis yöntemleri için incelenen çalışmaların özeti Tablo 1.3’te gösterilmektedir.

Tablo 1.3. IoT sistemlerinde kaynak tahsis çalışmalarının özeti

Çalışmalar	Yöntem Açıklaması	Avantajları	Geliştirilen veya Kullanılan Algoritmalar
(Choi ve Lim, 2016)	Kombinatoryal açık artırma görev yönetimi yaklaşımı	Yüksek görev tamamlama ve QoS desteği	Algoritma yok
(Ramirez, vd., 2019)	IoT cihazlarının kaynak paylaşımı ve kendi kendini yönetme algoritması	Daha yüksek performans ve iletişim kapasitesi	IoT ağ geçidinde kontrol, M2M ilişkileri kurulması
(Wang, vd., 2019)	Karınca kolonisi optimizasyonu ile görev ön planlaması yapan WebIDE bulut sunucusu için kaynak ayırma yaklaşımı	Daha az görev yanıt süresi ve verimli kaynak kullanımı	Algoritma yok
(Sangaiah, vd., 2020)	Balina optimizasyonu tabanlı sezgisel yaklaşımla IoT kaynak tahsisi ve optimizasyonu	Kaynak tahsisinde problem keşfi ve optimum çözüm	Balinaların mesafe alg., spiral alg., küçülen alg. arama avı alg.
(Abedin, vd., 2018)	Yüksek QoS ihtiyaçları olan IoT uygulamaları için kullanıcı ilişkilendirme ve kaynak tahsis şeması	Kullanıcı ilişkileri istikrarı ve verimli kaynak tahsisi	İkili karşılaştırma matrisi oluşturma, her sis cihazında küresel ağırlık vektörü, dışsallıklarla eşleştirme algoritması, en uygun kaynak tahsisi
(Gu, vd., 2018)	SPA ve UOC yaklaşımlarıyla kaynak tahsis şeması	Yüksek performans	SPA modelleme algoritması, UOC stratejisi
(Nassar ve Yilmaz, 2019)	Fog-RAN kaynak tahsisi için pekiştirmeli öğrenme yöntemleri ile çözüm yaklaşımı	Sınırlı kaynaklar için verimli kullanım	Monte Carlo kullanarak optimum politikayı öğrenme, Q-öğrenme, SARSA ve beklenen SARSA kullanarak optimum politikayı öğrenme
(Banaie, vd., 2020)	AHP kullanan MCDM yöntemiyle çoklu ağ geçitleri için yük dengeleme yaklaşımı	Tıkanıklık önleme, güvenilir ve hızlı sonuç	AHP tabanlı yük ilişkilendirme algoritması
(Huang, vd., 2020)	IoT ağlarında ortak görev boşaltma ve QoS destekli kaynak yönetimi	QoS desteği, düşük ek yük, yük dengeleme	AHP tabanlı QoS değerlendirme algoritması, araştırma blok tahsisi algoritması, ikili eşleştirme oyunu algoritması.
(Na, vd., 2018)	MEC'de kaynak tahsisi ve girişim yönetimi	Performans: %42 toplam verimlilik, %59 gecikme, %37 veri erişim oranı, %40 yük dengesi	Kenar sunucu ve kenar geçidi için optimize edilmiş araştırma tahsis şeması, kenar geçitler arasında zamanlama algoritması

Tablo 1.3. (devam)

Çalışmalar	Yöntem Açıklaması	Avantajları	Geliştirilen veya Kullanılan Algoritmalar
(Deng, vd., 2020)	Uç sistemlerde güvenilirlik açısından SLA sağlayan RL tabanlı dinamik kaynak tahsisi	%21,72 daha yüksek performans	DeraDE algoritması
(Jiang, vd., 2020)	DRL tabanlı kaynak zamanlama şeması	Optimuma yakın performans, düşük gecikme süresi, gerçek zamanlı	2r-SAE alg., ASA alg., ASA ve 2r-SAE ile DRL alg.
(Liu, vd., 2020)	ML yaklaşımları ile uç sistemlerde kaynak tahsisi	Daha düşük maliyet, diğer yaklaşımlarla aynı performans	K-ortalama tabanlı öncelik odaklı kümeleme, hesaplama görevi boşaltma için DQN eğitimi, DQN tabanlı hesaplama görevi boşaltma şeması
(Tang ve Hu, 2020)	MEC tabanlı sistemlerde heterojen yapılarda verimli kaynak tahsisi, enerji tüketimi ve yük dengeleme şeması	Düşük gecikme süresi, verimli hesaplama karmaşıklığı	İteratif ortak alt kanal ve güç tahsisi alg., logaritmik ve ardışık dışbükey yaklaşımlara dayalı güç tahsisi alg., ortak hesaplama boşaltma ve kaynak tahsisi alg.
(Wang, vd., 2020)	Genetik algoritmaya dayalı olarak geliştirilen TCA-IPSO algoritması ile optimum görev dağılımı yaklaşımı	İki benzer algoritmadan ortalama görev tamamlama süresinde %53,8 ve %36 azalma	TCA-IPSO algoritması
(Xiong, vd., 2020)	DRL tabanlı verimli kaynak tahsis çerçevesi	Düşük görev tamamlama süresi, daha az kaynak kullanımı	Çoklu tekrar hafızalı DQN
(Zhang, vd., 2020)	C-RAN ve MEC entegre sistemlerinde optimum görev boşaltma ve kaynak tahsisi yaklaşımı	Enerji tüketiminde %59 azalma, ortalama görev gecikmesinde %57 azalma	Eşleşen oyun tabanlı alt kanal tahsisi alg., Dinamik görev boşaltma ve kaynak tahsisi alg.
(Khan, vd., 2020)	Akıllı endüstrilerde DFL tabanlı kaynak optimizasyonu yaklaşımı	Yüksek verimlilik, veri gizliliği koruması	Ayrışma gevşemesi tabanlı algoritma
(Lee, vd., 2020)	Pazar analizi için MEC sistemlerinde çalışan ve bulutta DFL sonuçlarını kullanan Stackelberg oyun modeli	Tüm piyasa katılımcıları için faydayı maksimize etmek	Algoritma yok

Tablo 1.3. (devam)

Çalışmalar	Yöntem Açıklaması	Avantajları	Geliştirilen veya Kullanılan Algoritmalar
(Pham, 2020)	NFV özellikli IoT sistemlerinde ağ geçidi yerleşimi, kaynak tahsisi ve yönlendirme için optimizasyon modeli	Düşük hesaplama süresi ile optimuma yakın sonuçlar	Komşu nesil ağ geçidi yerleşimi ve çok atlamalı yönlendirme, Komşu nesil hizmet yerleşimi
(Xu, vd., 2020)	RL tabanlı ortak iletişim ve hesaplamalı kaynak tahsis mekanizması	Enerji tüketiminde ve işlem süresinde azalma, QoS desteği	RJCC, Q-Yük boşaltma, L-Göç
(He, vd., 2018)	Yeşil IoT'de iletim hızı ve kaynak tahsis modeli	Yüksek QoE performansı	SPT'ye dayalı kaynak tahsisi, SPTC'ye dayalı kaynak tahsisi, DQN'ye dayalı kaynak tahsisi
(Liu, vd., 2018)	Heterojen IoT için NOMA teknolojisini kullanan RNN tabanlı kaynak tahsis modeli	Düşük hesaplama karmaşıklığı ve enerji, yüksek güç	Birincil kullanıcıların alt kanal ataması için kademeli kenar kopma alg., ikincil alt kanal ataması için yeniden ağırlıklı mesaj geçirme alg., bağlı parametre eğitimi için öğrenme alg.
(Malik, vd., 2018)	NB-IoT sistemleri için maksimum hız kaynak tahsis şeması	Yüksek hız, enerji tasarrufu ve genel iyileştirme	İteratif verimli ve güvenilir radyo kaynak yönetimi prosedürü
(Su, vd., 2018)	Birçok sensör ve HAP içeren IoT sistemlerinde Steckelberg tabanlı kaynak tahsis şeması	HAP ve sensörler için yüksek güç, daha az maliyet	HAP ve sensör düğümleri için ortalama alan kontrol algoritması
(Zhang, vd., 2018)	Çevrimiçi ve iki zaman ölçekli kaynak tahsis algoritmasının ve Lyapunov tabanlı şemanın geliştirilmesi	Enerji ve spektrum optimizasyonu, verimli kaynak tahsisi	Kanal tahsisi alg., İki zaman ölçeğinde kaynak tahsisi alg.
(Dou, vd., 2019)	Girişim ve sıkışmaları önleyen OFDM tabanlı kaynak tahsis modeli	Kaynak kısıtlı IoT için uygun ve etkili	IoT ağında çok kullanıcılı OFDM sisteminin alt taşıyıcı ataması, ortak kod uzunluğu ve güç tahsisi
(Hussain, vd., 2020)	Çok kanallı sistemlerde parazit önleme ve uygun kanal tahsisi için Q-öğrenme tabanlı algoritma	Uygun SIR seviyesinde kanal tahsisi, çarpışmasız iletim	SIR tabanlı slot tahsisi
(Liu, vd., 2019)	Kanal ve güç tahsisi için MDP tabanlı kaynak tahsisi yaklaşımı	Daha iyi performans, düşük bilgi işlem ve uzun ağ ömrü	Verimli kanal tahsisi alg., eşleşen başlatma alg., MDP tabanlı güç tahsisi alg.

Tablo 1.3. (devam)

Çalışmalar	Yöntem Açıklaması	Avantajları	Geliştirilen veya Kullanılan Algoritmalar
(He, vd., 2020)	Dinkelbach, Jacobian-ADMM algoritmaları ile büyük ölçekli kaynak tahsisi yaklaşımı	Ölçeklenebilir, optimum enerji tüketimi ve gecikme süresi	ADMM ve Dinkelbach tabanlı kaynak tahsisi alg., Jacobian-ADMM tabanlı kaynak tahsisi paralel alg., Oyun ve Jacobian-ADMM tabanlı iki katmanlı iteratif kaynak tahsisi alg.
(Yang, vd., 2020)	DRL tabanlı verimli kaynak tahsis şeması	Enerji tasarrufu, QoS desteği, verimli kaynak tahsisi	Koordineli çok aracıli DRL-PER ile kaynak yönetimi
(Chen, vd., 2020)	Uydu-karasal ağlarda Markov yaklaşımına dayalı kaynak yönetimi şeması	Kaynak kullanımı artar, toplam gecikme azalır	MCPR alg., MCRC alg.
(Librino ve Santi, 2020)	Ultra güvenilir düşük gecikmeli iletişim için grafik tabanlı kaynak tahsis yaklaşımı	Dengeli trafik, spektrum verimliliği, hızlı kaynak tahsisi	Algoritma yok
(Karakaya ve Akleyek, 2021)	Öncelik kuyruğu tabanlı görev yönetimi algoritması ve hafif IoT uygulama şeması	Yüksek performans, düşük gecikme süresi, güvenli iletişim	Öncelik atama algoritması, sis düğümlerinin yönetimi

1.1.3. Makine Öğrenmesi Tabanlı Taktik Analiz Çalışmaları

Bu bölümde başta futbol olmak üzere sporda taktik analiz problemleri ve bu problemlerin çözümüne ilişkin literatürde son dönemlerde yapılan çalışmalar incelenmektedir. Literatürde taktik dizilişlerin incelendiği çok sayıda spor ve bilişim çalışmaları yer almaktadır. (McLean, vd., 2018)'de bir futbol takımı üzerinde ağ analizi yapılarak iki farklı dizilişte oyunun pozisyon geçişleri değerlendirilmiştir. İki sezon için yapılan analizlerde bir sezonda 4-2-2-2 dizilişi diğer sezonda ise 4-2-3-1 dizilişi kullanılmıştır. Oyun dizilişlerine göre bu iki sezon karşılaştırıldığında, dizilişlerin oyuncuların katkılarında farklılıklar ortaya koyduğu ve ileri uç oyuncularının pas trafiğine daha fazla katılmasından dolayı 4-2-3-1 dizilişinin daha çok tercih edildiği belirtilmiştir. (Hirotso ve Wright, 2003)'te sabit iki takım arasındaki eski maçlar üzerinden maç öncesinde ve maç anında en uygun formasyon değişimlerini ve oyuncu değişikliği stratejisini belirlemek için MDP destekli bir maç simülasyonu önerilmiştir. En çok kullanılan 5 diziliş dikkate alınarak Newton-Raphson yöntemi ile formasyon tahmini yapılmıştır. Sonuç olarak iki takım için en iyi

ve en kötü strateji arasında beklenen lig puanı açısından önemli bir fark oluđu belirtilmiştir.

(Horton, vd., 2015)'te Voronoi grafi ve Delaunay üçgenlemesi yöntemleri ile olası pas kombinasyonları ve kaliteli pas durumlarının analizi oluşturulmuştur. (Taki ve Hasegawa, 2000)'de oyuncuların belli olaylara karşı yapması en muhtemel hareketleri tahmin eden gölgelendirme yöntemi, Voronoi grafi kullanılarak bir oyuncunun diđer oyunculardan daha erken gelebileceđi bölgelerin belirlenmesini sağlayan baskın bölgeler yöntemi gibi yapılar önerilmiştir.

(Boon ve Sierksma, 2003)'te bir spor takımı için lineer programlama yöntemi kullanılarak ekip içinde optimum görev dağılımı ve yeni ekip üyelerinin katkılarını hesaplayan model önerilmiştir. Doğrusal optimizasyon formülleri futbol ve voleybol sporları için geliştirilmiştir. Sistemin takımların dizilişı ve yerleşimi dikkate alınarak ekibe sporcu ve personel alımlarındaki kararlar için destek sağladığı belirtilmiştir. (Carling, 2011)'de bir futbol takımının üç sezonda üç rakip takım ile yaptığı maçlar çoklu kamera sistemi ile analiz edilerek üç farklı dizilişteki rakipler savunma, orta alan ve hücum yönlerinden karşılaştırılmıştır. İncelenen maçlar kapsamında, rakibin dizilişlerinin fiziksel performansı önemli derecede etkilemediđi ancak beceri performansında deđişiklikler oluştuduğu belirtilmiştir.

(Narizuka ve Yamazaki, 2019)'da bir futbol takımının oluşumu için Delaunay yöntemine dayalı kümeleme algoritması önerilmiştir. Tüm futbol oyunlarının ısı haritaları dikkate alınarak kümeler belirlenmiştir. Rakip takımın verilerinden yoksun bir şekilde önceki maç verilerinden belirli bir takım için standart formasyon geçişleri modellenmiştir. (Trequattrini, vd., 2016)'da teknolojik taktik-teknik yeniliklerin profesyonel futbol alanında kullanıldığında kulüpler için deđer üretip üretmeyeceđi araştırılmıştır. Eski yıllardan beri çok sayıda menajerin ve takımın tercih ettikleri formasyonlar incelenmiştir. Teknolojik yeniliklerin bir problemi doğrudan çözmek yerine genel kapsamda çözüme destek için kullanılabilmesi ve uzun vadede kulübe ekonomik getiri sağlayabileceđi bildirilmiştir.

Sporun teknolojik gelişmeler ile desteklenmesi son yıllarda popülerliğini oldukça artırmıştır. Özellikle istatistiksel analizlerin bilişim sistemleri tarafından desteklenmesiyle birlikte spordaki taktik savaşlar çok daha güçlenmektedir. Bu kapsamda az da olsa futbolda bilişim destekli olarak taktik dizilişlerin oyuna etkilerini

inceleyen ve yöntemler üreten çalışmalar vardır. (Shaw ve Glickman, 2019)'da Wasserstein metriği kullanılarak oyuncular arasındaki mesafeler ölçülmüş ve takım dizilişleri çıkarılmıştır. Bu dizilişler Bayes modeli kullanılarak sınıflandırılmış ve maçın taktiksel özeti çıkarılmıştır. Menajerlerin önemli olaylara nasıl tepki verdiği ortaya konmuştur. 180 maç analiz edilerek formasyon değişimleri, savunma zafiyetleri ve menajerin alışkanlıkları ile maçın gidişatı arasındaki ilişki ortaya çıkarılmıştır. (Al-Asadi, 2018)'de akıllı karar destek sistemleri kullanılarak futbolda belirli formasyon için oyuncuların mevkilerini ve zaman içindeki gelişimlerini belirleyen bir model önerilmiş ve temel 14 mevki için oyuncular sınıflandırılmıştır.

(Dobreff, vd., 2019)'da oyun teorisi kullanılarak en uygun diziliş seçimi modellenmiştir. Maçların formasyonu, evde veya deplasmanda oynama durumu, maçların skorları gibi bilgiler kullanılarak oyuncu dizilişleri çıkarılmıştır. Modern futbolda en çok kullanılan 4-4-2, 4-3-3, 4-2-3-1 dizilişlerinin uygulanan yöntemde de en iyi sonuçları verdiği belirtilmiştir. (Müller-Budack, vd., 2019)'da oyuncuların pozisyonlarına göre takım formasyonunu oluşturan ve sınıflandıran model önerilmiştir. Farklı dizilişler için önceden tanımlanan durumlar arasında benzerlik hesaplanmış ve formasyon sınıflandırması yapılmıştır. (Frey, vd., 2019)'da giyilebilir cihazlar ile oyuncuların saha içindeki konumlarının tahmini için üç yöntem karşılaştırmalı olarak incelenmiştir. Rastgele orman modeli, gradyan artırma ve CNN yöntemlerinin başarımı yaklaşık aynı olmakla birlikte %80'in üzerinde olduğu belirtilmiştir. Ancak, veriler aynalama ve rastgele veri ekleme yöntemleri ile artırılmıştır. (Andrienko, vd., 2019)'da futbolda takımın hareket kalıplarını çıkarmak için dinamik veri toplama ve bu verilerden hareket kalıplarını keşfetmek için bilgisayarla görme yöntemleri kullanılarak bir model önerilmiştir. Sistem, 2018/2019 Almanya sezonu için test edilmiştir. Elde edilen veriler orijinal hareketler ile karşılaştırılarak ve değerlendirilerek genel hareket modellerinin keşfedilebildiği belirtilmiştir. (Rein ve Memmert, 2016)'da takım sporlarında taktiksel karar verme süreçlerine destek vermek için büyük veri ve modern makine öğrenimi yöntemleri kullanılarak bir teorik model önerilmiştir. İzleme sistemlerinin gelişmesiyle daha etkin bir şekilde elde edilen büyük verilerin yönetimi problemi tartışılmıştır. Büyük veri ve makine öğrenimi yöntemlerinin karmaşık veri analizlerini sağlayabileceği sonucuna ulaşılmıştır. (Tavana, vd., 2013)'te karmaşık ve çok kriterli bir sorun olan oyuncu seçimi ve takım oluşumunu belirlemek için belirsiz sıralama yöntemi ve FIS

kullanılarak bir model önerilmiştir. Yöntemde modelin doğruluğundan (accuracy) ziyade menajere yardımcı olan bir sistem amaçlandığı vurgulanmıştır.

(Asali, vd., 2016)'da Robocup 2 boyutlu futbol simülasyon liginde rakip takımın oluşumunu elde etmek için makine öğrenmesi yaklaşımları kullanılarak bir model önerilmiştir. Sistemde sahada ızgaralar kullanmak yerine yakınlık durumuna göre dizilişlerin oluşturulduğu ve başarılı sonuçlar elde edildiği belirtilmiştir. (Faria, vd., 2010)'da takımların oluşumlarının sınıflandırılması için Robocup 2 boyutlu futbol simülasyonu üzerinde dört makine öğrenmesi yöntemi karşılaştırılmıştır. Aralarından en iyi sonucu destek vektör makinelerinin sağladığı vurgulanmıştır. (Joash Fernandes, vd., 2020)'de ulusal ligler için maçın skorlarını tahmin etmek için makine öğrenmesi tabanlı model önerilmiştir. Birden çok makine öğrenmesi yöntemi karşılaştırılmıştır. Sistemin tahmin doğruluğunun %75,3 olduğu belirtilmiştir. (Moura, vd., 2012)'de takım yayılımını ve organizasyonunu analiz etmek için dış bükey gövde alanı ve Frobenius normu yöntemleri kullanılarak çıkarımlar elde edilmiştir. Takımlar arasındaki hücum ve savunma etkileşiminin analiz edildiği bildirilmiştir.

(Stübinger, vd., 2020)'de makine öğrenmesi yöntemleri kullanılarak Avrupa'nın beş büyük ligi ve bunların 2.ligleri için 12 yıl süreli maçlarını veri kümesi olarak kullanan bir maç skor tahmin yöntemi önerilmiştir. Yöntemin maç başına %1,58 istatistiksel ve ekonomik getiri sağladığı belirtilmiştir. (Tosato ve Wu, 2018)'de Football Manager (FM) 2018 kullanılarak Roma takımı için 47 öznitelik üzerinde projektif uyarlanabilir rezonans teorisi tabanlı bir karar modeli önerilmiştir. Oyuncular performanslarına göre kümelenecek ve finansal kısıtlamalar dâhilinde oyuncu satın alma ve satma kararları elde edilmiştir. (Landers ve Duperrouzel, 2018)'de fantezi futbol dünyası için kullanıcıların fantezi noktalarını tahmin etmek ve güçlü takım oluşturmaya yardımcı olmak için makine öğrenmesi tabanlı bir model önerilmiştir. 11 haftanın 5'inde rastgele seçilen 300000 takım arasından en yüksek puan alan takımı oluşturmayı başardığı ve test edilen 3 sezonun hepsinde birincilik ödülünü aldığı saptanmıştır. (Abidin, 2021)'de, bir futbol akademisinin 13 yaş altı takımındaki en iyi oyuncu kombinasyonunu bulmak için oluşturulan veri kümesine yedi farklı makine öğrenme algoritması uygulanmıştır. Sonuçlar, teknik direktör kadrosu ve 2019-2020 sezonunda oynanan 20 maçın kadrosu ile karşılaştırılmıştır. Veri kümesi iki farklı veri alt kümesinden oluşturulmuştur. Bunlardan biri Hit/It Assistant ile toplanan antrenman verileri, diğeri ise literatürde belirtilen on sekiz kritere göre teknik direktör

değerlendirmeleridir. En iyi performans gösteren algoritmanın %89,36 oranında başarımlı sağladığı ve kadro önerilerinin antrenörün ideal takımına %97,16 oranında benzediği belirtilmiştir.

1.2. Motivasyon ve Katkı

Bu bölümde seçilen çalışma alandaki eksikler ve bu eksikleri gidermek için önerilen mimari, algoritma ve modelin özgün değerleri ön plana çıkarılmaktadır. Bu çalışmada, oyuncuların ve seyircilerin sağlığını izlemek ve maçta taktiksel analiz yapmak için sis bilişim tabanlı güvenli bir IoT altyapı mimarisi ve bu mimarinin sis bilişim katmanında öncelik kuyruğu tabanlı kaynak tahsis algoritması önerilmektedir. Ayrıca, bu altyapıda çalışabilen makine öğrenmesi tabanlı futbolda taktik analiz modeli önerilmektedir.

IoT mimarilerinde veri depolama ve işleme merkezi olarak sis bilişim tabanlı yapılar yerel ağ ortamında çalıştığı için uzaktaki bulut bilişim tabanlı yapılara göre daha düşük gecikme sağlayabilmektedir. Sis bilişim yapılarının bulunduğu sis katmanında birden fazla sis düğümü bulunabilmektedir. Bu düğümler arasında uç cihazlarından gelen isteklerin sis düğümlerine atanması için bir planlama yapılmalıdır. Sağlık ve taktik analiz amacıyla önerilen mimaride bazı isteklerin öncelikli olarak işlenmesi gerekmektedir. Bu nedenle sis düğümleri arasında öncelik tabanlı algoritmalar istek yönetimini etkili ve düşük gecikme süresi ile sağlayabilmektedir. Önerilen makine öğrenmesi modelinde ise verilerin işlenmesi için bir veri merkezine ihtiyaç vardır. Ayrıca bu modelin tahmin sonuçlarının teknik ekibe geri bildirim olarak iletilmesi için modelin IoT tabanlı bir mimaride çalışması gerekmektedir. Bu nedenle önerilen modelin, önerilen IoT ve sis bilişim tabanlı mimaride çalıştığı varsayılmaktadır. Ek olarak, bu sistemin yüksek verimle çalışabilmesi için başarımlı yüksek bir makine öğrenmesi yaklaşımı seçilmelidir. Tez kapsamında literatüre yapılan katkılar takip eden alt başlıklar altında toplanmıştır.

1.2.1. Sis Bilişim ve IoT Tabanlı Mimarinin Katkıları

Sis bilişim ve IoT tabanlı sağlık izleme ve taktik analiz mimarisinin katkıları aşağıdaki gibidir (Karakaya ve Akleylek, 2021).

- Sis bilişim sistemlerinin karşılaştığı güvenlik sorunları incelenmektedir. Önerilen modelin bu sorunlara karşı aldığı önlemler ve güvenlik analizleri açıklanmaktadır.

- Sistem belirli kapsama alanı içinde yer aldığı için yerel ağ ortamında verilerin işlenebilmesi ve gecikme sürelerinin düşürülmesi için mimaride sis bilişim yapısı kullanılmaktadır. Verilerin İnternet ortamına çıkmadan işlenebilmesini sağladığı için sis bilişim bulut sisteme göre daha güvenli kabul edilmektedir.
- Sis bilişimin beraberinde getirdiği güvenlik durumlarına ek olarak mimaride mahremiyet, gizlilik, kimlik doğrulama, saldırıların tespit edilmesi ve bant genişliği tasarrufu gibi sistemin güvenli ve hızlı olmasını sağlayan özellikler de yer almaktadır.
- Sis düğümlerinde öncelik kuyruğu tabanlı kaynak yöntemi algoritması ile önemli işlemler daha hızlı işlenmekte ve gecikme etkin bir şekilde azaltılmaktadır.
- Mimaride kimlik doğrulama ve gizlilik gibi yüksek işlem kapasitesi gerektiren işlemlerde örtük sertifikalar sayesinde bant genişliğinden tasarruf edilmektedir. Blokzincir tabanlı SDN denetleyicisi ile kötü niyetli değişiklikler fark edilmekte, sel baskını gibi saldırı durumlarında veri kayıpları önlenmekte ve veri bütünlüğü sağlanmaktadır.

1.2.2. Sis Düğümlerinde Görev Yönetim ve Kaynak Tahsis Algoritmasının Katkıları

IoT'de veri depolama ve işleme için genellikle bulut, sis ve kenar bilişim sistemleri tek başlarına veya birlikte kullanılmaktadır. Bu sistemlerdeki kaynaklar farklı görevler tarafından paylaşılmaktadır. Kaynakların paylaşılması, dağıtılmış veri işleme ve zaman planlaması gibi işlemler genellikle kaynak tahsisi, kaynak yönetimi, yük dengeleme olarak adlandırılmaktadır (Min, vd., 2019).

Sis düğümlerinde görev yönetimi ve kaynak tahsisi için önerilen algoritmanın katkıları aşağıdaki gibidir (Karakaya ve Akleylek, 2021).

- IoT sistemlerinde sınırlı ve paylaşılan kaynaklar incelenmekte ve kaynakların verimli paylaşımı için çözüm yaklaşımları değerlendirilmektedir (Karakaya ve Akleylek, 2022).
- Önerilen mimarinin sis katmanında çalışan öncelik kuyruğu tabanlı bir kaynak tahsis algoritmasıdır.

- Bu mimari ve algoritma iFogSim kullanılarak test edilmektedir. Sonuçlar benzer çalışmalar ile kıyaslandığında önerilen algoritmanın aynı topoloji üzerinde daha düşük gecikme süresi ile daha yüksek performans sağladığı görülmektedir.
- Simülatör kodlarına <https://github.com/akarakaya/SimApp> adresi üzerinden erişilebilmektedir.

1.2.3. Makine Öğrenmesi Tabanlı Taktik Analiz Modelinin Katkıları

Taktik analiz modern futbolda başarı için ana bileşendir. Şimdiye kadar taktik analiz ile ilgili az sayıda çalışma olmasının temel nedeni ilgili verilerin olmamasından veya temin edilmesinin zor olmasından kaynaklanmaktadır (Rein ve Memmert, 2016). Profesyonel futbol sektörü üzerine yapılan analizden, teknik-taktik yeniliğin futbol kulüpleri için değer yaratmadaki rolünü inceleyen çok az sayıda çalışma olduğu açıktır (Trequattrini, vd., 2016). Karar verme konusunda teknik ekibe destek olan çalışma sayısı da çok azdır (Abidin, 2021). Benzer şekilde, oyuncu seçimi ve takım oluşumu konusunda da çalışma sayısı oldukça azdır (Tavana, vd., 2013). Futbol teknolojisi alanında maç analizlerinin dinamik, etkileşimli ve karmaşık yönlerinin ele alınmadığı çok sayıda çalışma vardır. Ancak, oyunun sürekli ve ardışık yönlerini içeren yöntemlerin geliştirilmesi gerekmektedir (Sarmiento, vd., 2014). Futbol teknolojisi çalışmalarının büyük çoğunluğu maçın skorunun tahmin edilmesi ile ilgili iken taktik analiz için futbol teknolojisinin kullanımı üzerine yapılan çalışmaların büyük çoğunluğu ise futbolcu yerleşimlerinin sınıflandırılması ile ilgilidir.

Önerilen mimaride çalışan makine öğrenmesi tabanlı taktik analiz modelinin katkıları ise şu şekildedir:

- Spor bilimi literatüründeki etkili ve verimli oyuncu seçimi ve takım oluşumu konusundaki boşlukları ele almaktadır.
- Makine öğrenmesi tabanlı yeni bir teknik ekip destek yöntemi sunmaktadır. Veri kümesi birçok makine öğrenme algoritması ile test edilmektedir. Parametreleri optimize edilmiş AdaBoost algoritması, %87,2 doğruluk oranıyla en iyi sonuçları vermektedir. Ayrıca, tüm algoritmalar karşılaştırıldığında, topluluk sınıflandırıcıları, karar ağaçları, SVM ve diskriminant analiz yöntemleri de oldukça iyi performans göstermektedir.
- Gol atma, gol yeme ve golsüz geçilen pozisyonlar için tahminleme

yapılmakta ve sonuçlar yedek kulübesine geri bildirim olarak gönderilmektedir.

- Önerilen altyapıda çalışabilecek şekilde tasarlanan, maç anında formasyon çıkarılarak teknik ekip analizlerine katkı sağlayan bir uygulamadır.
- Sistemin modeli, dağıtık ve güvenliğe yardımcı olan bir federe öğrenme yapısı tarafından desteklenmektedir.

Önerilen modelde en iyi performansı gösteren yöntemlerde ilk sırada %87,2 doğruluk oranı ile AdaBoost algoritması yer almakta iken bunu %85,9 doğruluk oranı ile güçlendirilmiş (boosted) ağaç ve %85,1 ile kaba (coarse) ağaç takip etmektedir.

1.3. Organizasyon

Bu bölümde ilerleyen bölümlerdeki içerik ile ilgili bir özet yer almaktadır. Önceki bölümler sis bilişim ve IoT tabanlı mimari şemaları, bu mimarilerdeki görev yönetim ve kaynak tahsis yaklaşımları ve makine öğrenmesi tabanlı taktik analiz yaklaşımlarını kapsayan detaylı bir literatür taramasını içermektedir.

Bölüm 2’de nesnelerin İnterneti ve sis bilişime genel bakış yer almaktadır. Bölüm 3 önerilen mimarinin ve algoritmanın detaylarını içermektedir. Bu kapsamda, IoT tabanlı mimari detayları, sis düğümlerinde görev yönetimi ve önerilen algoritma, önerilen mimarinin güvenliğinde kullanılan yöntemler ve güvenlik analizleri, önerilen mimarinin ve algoritmanın simülasyonu yer almaktadır. Bölüm 4’te makine öğrenmesi tabanlı teknik ekip uyarı sistemine yer verilmektedir. Bu bölümde verilerin toplanmasını, veri ön işlemeyi ve çeşitli makine öğrenmesi yöntemleri ile modelin test edilmesini kapsayan önerilen modelin uygulanması adımları anlatılmaktadır. Bölüm 5’te önerilen yöntemlerin fiziksel olarak uygulanmasının zor olduğunu kanıtlayan giyilebilir teknolojilerdeki zorluklar yer almaktadır. Son bölümde ise sonuçlar ve gelecek çalışma konuları ele alınmaktadır.

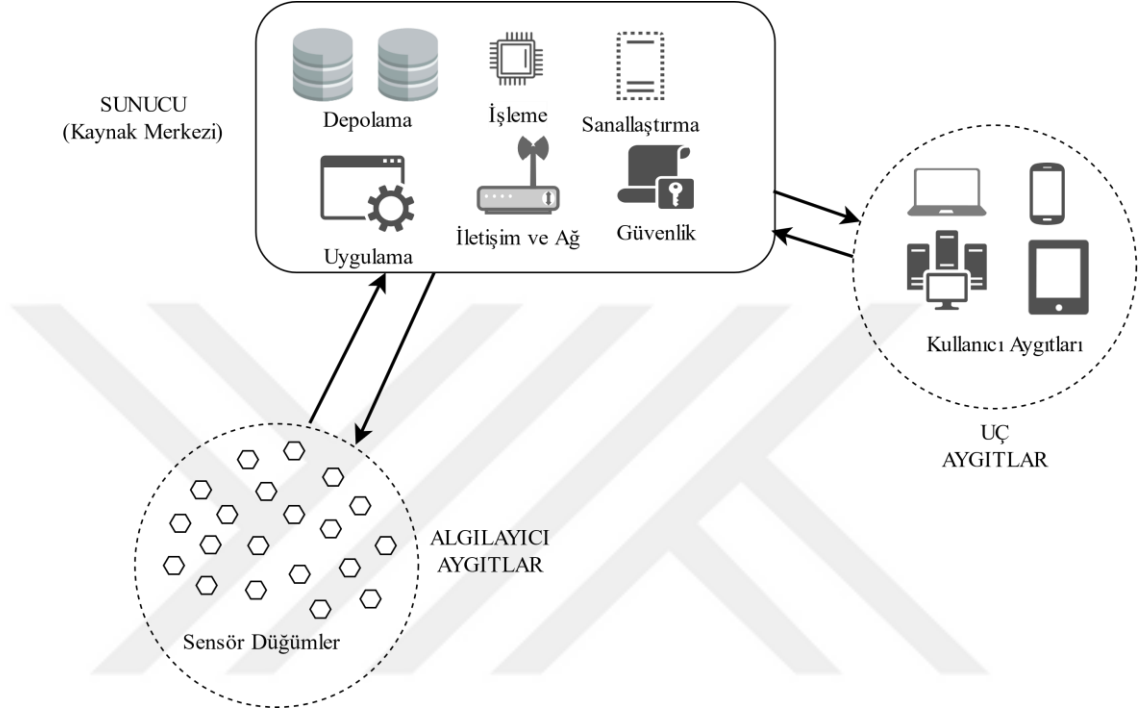
2. NESNELERİN İNTERNETİ VE SİS BİLİŞİM

Bu bölümde nesnelere İnterneti ve sis bilişim konularına genel bir bakış yapılmaktadır. IoT sistemlerindeki veri depolama ve işleme birimleri olan bulut, sis ve kenar bilişim yöntemleri açıklanmakta ve farkları değerlendirilmektedir. Nesnelere İnterneti izleme, tanıma, yönetim gibi işleri yerine getirebilmek için belirlenen protokollere dayalı olarak sensörler gibi algılama donanımları ve sonuçları alan uç donanımları İnternet'e bağlayan ve haberleştiren ağ yapısıdır (Min, vd., 2019). IoT altyapısı ile akıllı şehir, sağlık, endüstri, tarım, askeri vb. gibi çok fazla alanda efektif uygulamalar geliştirilebilmektedir. IoT tabanlı yaklaşımlar küçük giyilebilir cihazlardan büyük endüstri sistemlerine kadar çok geniş bir ölçekte incelenebilmektedir. Genellikle giyilebilir cihazlar ve özelleşmiş sensörler ile insanlar, hayvanlar, yer altı, su altı, orman alanları gibi çeşitli noktalar üzerinde kişisel alan ağları, kablosuz sensör ağları, kablosuz vücut ağları oluşturularak bir takım veriler elde edilmektedir. Algılanan veriler üst katmandaki bir veri işleme ve depolama cihazlarına gönderilmektedir. Bu katmanda üretilen sonuçlar ise izleyen durumda olan uç cihazlara gönderilmektedir. Bu tez kapsamında işlenen konular ile ilgili, IoT sistemlerde görev yönetimi ve kaynak tahsis yaklaşımlarının incelendiği bir uluslararası kitap bölümü (Karakaya ve Akleylek, 2022), sis bilişimi ve uygulamalarında veri güvenliğinin ele alındığı bir ulusal kitap bölümü (Akleylek ve Karakaya, 2020), WSN ve IoT sistemlerde kimlik doğrulama yaklaşımlarının değerlendirildiği bir uluslararası bildiri makalesi (Karakaya ve Akleylek, 2018) yayınlanmıştır.

IoT sistemler ile yüksek kaliteli, hızlı, güvenilir, akıllı ve modern uygulamalar geliştirilebilmektedir. Bu nedenle çok çeşitli alanlarda tercih edilmektedir. Ancak, IoT sistemler bu kadar avantajının yanında birçok zorluğu da beraberinde getirmektedir. Güvenlik ihtiyacı, var olmayan yazılım ve sensör durumları, dış etkenlerden etkilenen (giyilebilir cihazlar gibi) sensörler, düşük güç tüketimi, verimli depolama ve veri iletişimi ihtiyacı bunlardan bazılarıdır.

Genellikle IoT'de depolama ve veri işleme amacıyla bulut, sis ve kenar sistemleri tek başına veya bir arada kullanılmaktadır. Sensörlerden gelen veriler ile oluşturulan isteklerin işlenmesi için bu sistemleri farklı görevler birlikte kullanmak zorundadır. Bu nedenle kaynakların paylaşılması gerekmektedir. Kaynakların paylaşılması, dağıtık işlem yapma ve zaman çizelgesi oluşturma işlevleri için yapılan

çalışmalar genellikle kaynak tahsisi, kaynak yönetimi, yük dengeleme gibi isimler ile adlandırılır. Ayrıca, enerji tasarrufu sağlama, pil ömrünü uzatma, yoğun hesaplamalı işleri azaltma ile kaynak kullanımı, işlem hacmi ve düşük cevap süresi konusunda verim sağlanabilmektedir (Patel ve Patel, 2016). Bu alanda yapılan çalışmalar ise yük boşaltma olarak adlandırılmaktadır.



Şekil 2.1. IoT sistemlerin genel yapısı

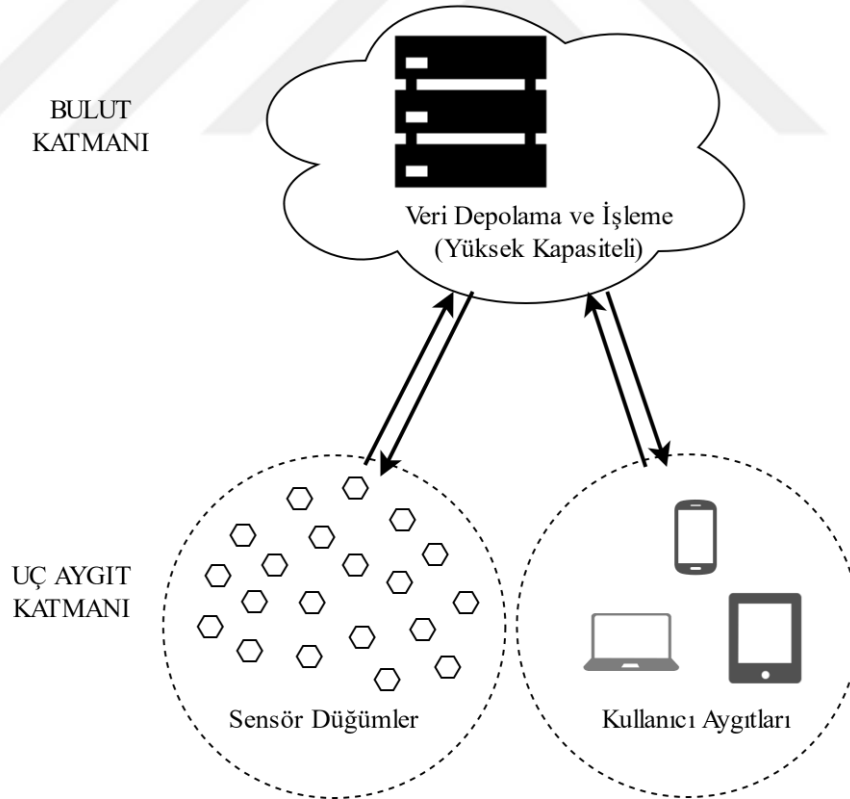
IoT sistemlerin genel yapısı, görevlerin ve isteklerin paylaştığı kaynaklar ve işlemler Şekil 2.1’de gösterilmektedir. Bu işlemler genellikle depolama, proses yürütme, sanallaştırma, uygulama, ağ, güvenlik, dağıtık yapı şeklindedir.

2.1. IoT Sistemlerinde Depolama ve Veri İşleme

IoT sistemlerindeki en önemli ayrıntılardan biri olan ve depolama ve veri işleme işlerini kotaran kaynak merkezleri bu bölümde incelenmektedir. IoT uygulamalarında kaynak merkezleri olarak bulut, sis veya kenar bilişim yapılarından yararlanılabilmektedir. Sis ve kenar bilişim veri depolama ve işleme için yerel ağ kapsamında bir çözüm sunabilmekte ve bulut ile entegre çalışabilmektedir. Sis bilişim, bulut bilişimin sunduğu depolama, ağ oluşturma ve bilgi işlem olanaklarını IoT ağının kenarına doğru genişletir. Bu nedenle sis bilişim, sistemin performansını ve verimliliğini artıran bir faktördür (Bader, vd., 2016). Ancak, kapsama alanı kısıtlaması vardır ve belli alan sınırları içinde gerçekleştirilebilecek uygulamalarda kullanılması

uygundur. Kenar bilişimde hesaplama yükünün uç cihazlara verilmiş olması sis bilişim ile arasındaki en temel farklardandır (Varghese, vd., 2016). Sis bilişimde hesaplama yükü sis katmanındaki sis düğümlerine aittir. Bir başka deyişle, sis uç cihazlar ile bulut sistemler arasında bir katmandır.

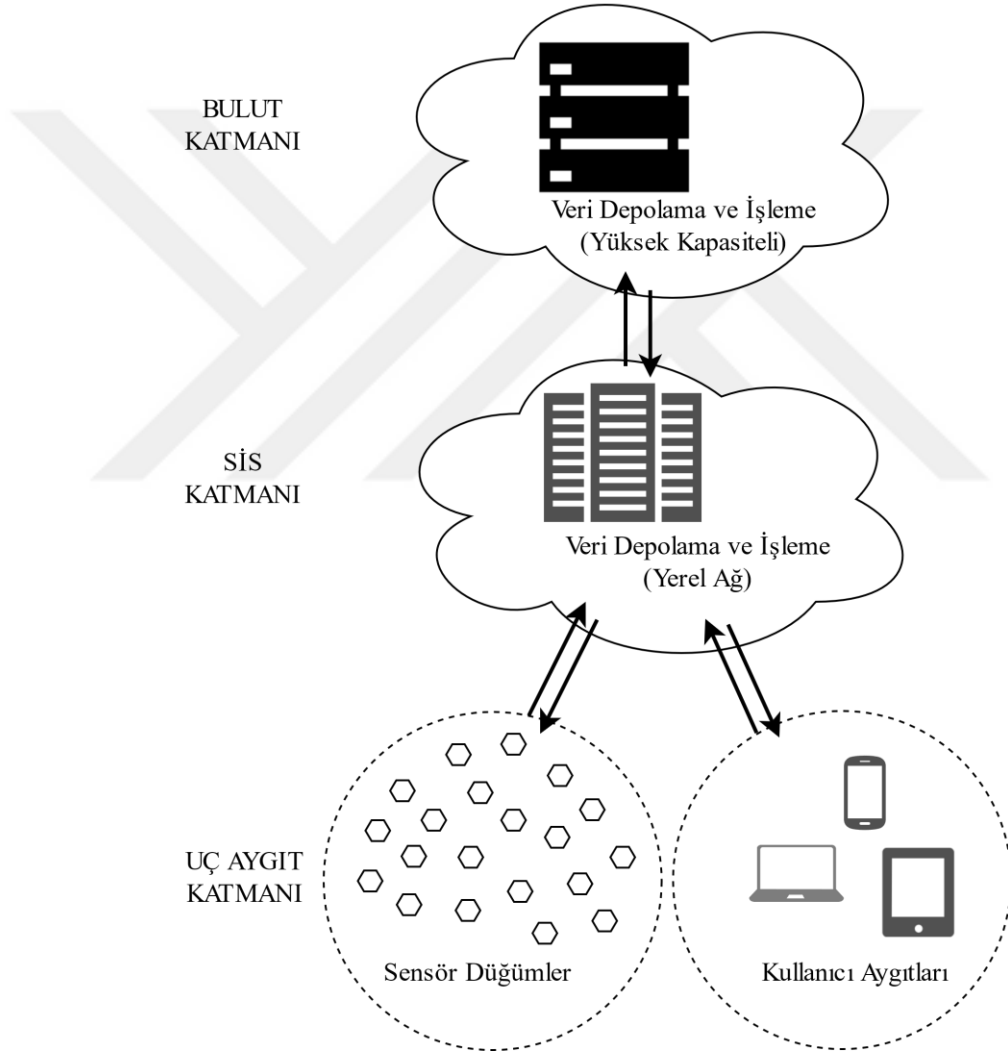
Bulut bilişimde veriler, depolama ve işleme için doğrudan uzaktaki bulut sunucusuna gönderilmektedir. Veriler İnternet üzerinden şifrelenmeden buluta gönderildiğinden veri güvenliği riskleri oluşabilmektedir. Uç cihazlarda veri güvenliği işlemleri yapılabilmekte, ancak uç cihazlar kaynak sınırlı olduğu için maliyet yüksek olmaktadır. Bu nedenle gecikme süresinin önemsenmediği uygulamalarda bulut sistemler tek başına kullanılabilir. Ayrıca, sis ve kenar bilişim kullanan uygulamalarda kalıcı depolama gibi amaçlar için bulut sistemler de birlikte kullanılabilir. Bulut bilişim kullanan IoT sistemi Şekil 2.2’de gösterilmektedir. Uç aygıt katmanı ile bulut katmanı arasındaki iletişim İnternet ağından sağlanmaktadır.



Şekil 2.2. Bulut bilişim kullanan IoT sistemi

Sis bilişimde genellikle üç katmanlı mimari kullanılmaktadır. İlk katman, sensör düğümleri, akıllı cihazlar, IoT özellikli cihazlar gibi uç cihazlardan oluşmaktadır.

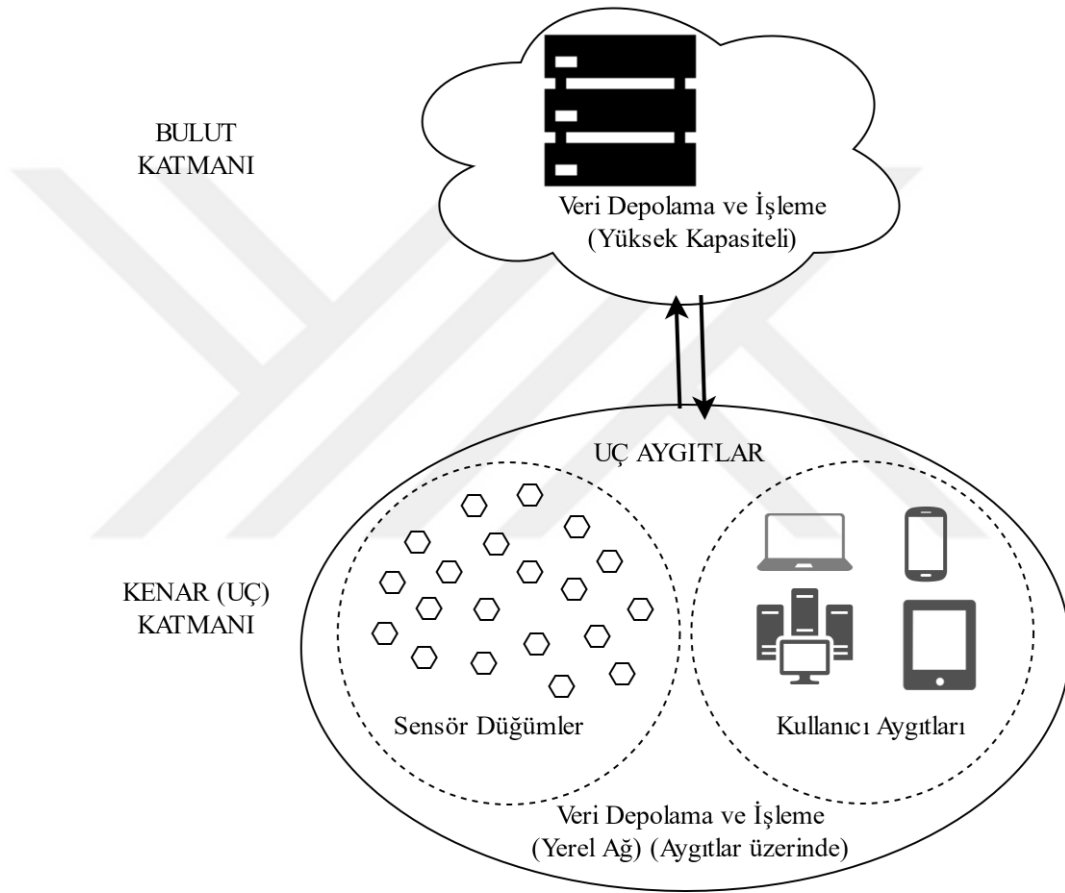
İkinci katman, yönlendiriciler, ağ geçitleri, anahtarlar ve erişim noktaları gibi sis düğümlerinden oluşmakta ve sis düğümleri, depolama ve bilgi işlem faaliyetlerini yürütmektedir. Üçüncü katman, uzak bulut sunucularından meydana gelmektedir. Ayrıca, bu katman, yüksek depolama ve bilgi işlem hizmeti sağlamaktadır (Mukherjee, vd., 2017). Bulut ve sis bilişimi entegre bir şekilde kullanan IoT sisteminin genel yapısı Şekil 2.3'te gösterilmektedir. Bulut katmanı ile sis katmanı arasındaki iletişim yerel ortamdan sağlanırken sis katmanı ile uç aygıt katmanı arasındaki iletişim yerel ağ üzerinden sağlanmaktadır.



Şekil 2.3. Bulut ve sis bilişimi birlikte kullanan IoT sistemi

Kenar bilişim sistemlerinde zamana duyarlı veriler akıllı uç cihazlarda işlenmektedir. Mobil uç cihazlar kısıtlı kaynaklara sahip olduklarından, birçok zorluğu bulunmaktadır. Ancak, avantajları, sis bilişime göre kısıtlı alanlar yerine geniş coğrafi alanlara yayılabilmesidir. Sis bilişim, uç düğümlerin veri merkezine yakın olduğu

ortamlarda (hastane, spor sahası, sanayi şirketleri gibi) kullanılabilir. İken kenar bilişim uzak ve geniş coğrafi ortamlarda (otomobil sistemleri, akıllı şehirler gibi) kullanılabilir. Bu bilgi işlem yapılarının her ikisi de bulutu uç cihazlara kadar genişletmekte ve bulutun optimizasyonunu desteklemektedir. Bulut ve kenar bilişimi birlikte kullanan IoT sisteminin genel yapısı Şekil 2.4'te gösterilmektedir. Bulut katmanı ile kenar veya uç katmanı arasındaki iletişim İnternet ortamından sağlanmaktadır.



Şekil 2.4. Bulut ve kenar bilişimi birlikte kullanan IoT sistemi

IoT sistemlerde donanım, veri işleme ve depolama, görev yönetimi, kaynak tahsisi ve güvenliğin geliştirilmesi ile düşük maliyetli, hızlı ve güvenilir sonuçlar elde edilmektedir. Bu nedenle birçok alanda IoT tabanlı yaklaşımlar tercih edilmekte ve bu yaklaşımlar ile uygulamalar geliştirilmektedir. Bu üç bilgi işlem yapısı ise IoT tabanlı uygulamaların performanslarını artırmak için kullanılabilir.

2.2. Sis Bilişim

Bu bölüm, sis bilişiminin özelliklerini ve güvenlik ihtiyaçlarını içerir. Sis bilişim, uç cihazlar ve bulut veri merkezleri arasında veri işleme ve depolama gibi hizmetler sağlayan yerleştirilmiş ve sanallaştırılmış bir platformdur (Bonomi, vd., 2012). IoT, sis bilişimi ile daha verimli ve daha güvenli hale gelmektedir (Abdulkareem, vd., 2019). IoT uygulamalarında kısıtlı kaynaklara sahip sensörler kullanılmaktadır. Veri işleme, depolama ve şifreleme gibi büyük bilgi işlem gerektiren işlemler genellikle sensörler tarafından yapılamamaktadır. Bu işlemleri genellikle uzak sunucu olarak hizmet veren bulut düğümleri veya yerel olarak hizmet veren sis düğümleri gerçekleştirmektedir.

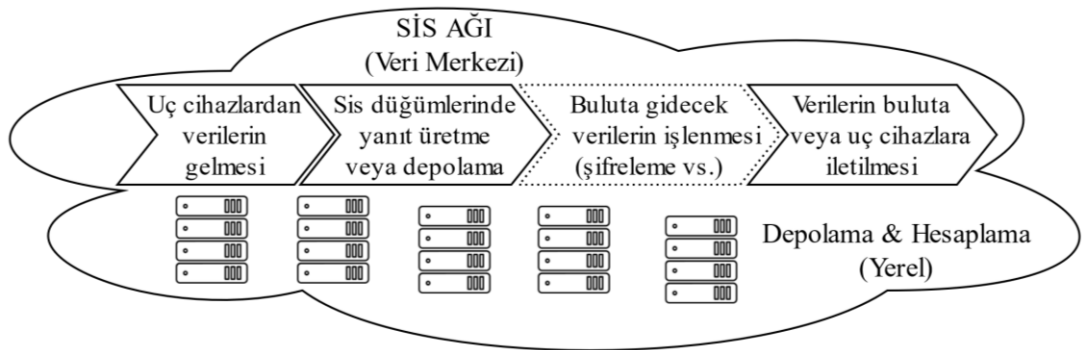
Sis bilişim, sensörlerden gelen verileri işlemek, depolamak ve yönetmek için kullanılmaktadır. Buluta veri gönderirken güvenli olmayan İnternet ağı kullanılmaktadır. IoT cihazları tespit edilen verileri buluta gönderirse, verilerin çalınması ve değiştirilmesi gibi riskler taşımaktadır. Ayrıca, verilerin işlenmesi ve saklanması yerel ağa göre daha yavaştır ve daha çok zaman almaktadır. Sis bilişiminde veriler yerel olarak işlendiğinden dolayı uç cihazlara iletilen yanıtın gecikme süresi bulut sistemlere göre daha kısa olabilmektedir. Sistemin performansını artırmak için IoT uygulamalarında sis bilişimi kullanmak önemlidir. IoT uygulamalarında büyük miktarda veri elde edilmekte ve bu verileri buluta aktarmak için yüksek bant genişliği gerekmektedir. Bu nedenle, sis katmanındaki verilerin analiz edilmesi ve yalnızca kalıcı olarak saklanacak verilerin buluta gönderilmesi bant genişliğini azaltmaktadır (Yi, vd., 2015). Sis bilişimin genel özellikleri aşağıdaki gibi listelenebilir (Akleylek ve Karakaya, 2020):

- **Düşük gecikme süresi:** Veriler uzaktaki bulut sunucusu yerine yerel ağdaki sis düğümlerinde işlenebilmekte ve depolanabilmektedir. Yerel ağda bant genişliğinin tamamı kullanılabilen İnternet ortamında paylaşımlı bağlantıdan dolayı kısıtlı bant genişliğinin kullanılmaktadır. Bu nedenle sis bilişim sistemleri iç ağda kullanılan donanımsal teknolojiye de bağlı olarak bulut sistemlere göre daha düşük veri işleme, iletim ve depolama gecikme süresine sahip olabilmektedir.
- **Geniş coğrafi alan:** Sis bilişim IoT uygulamalarının performansını önemli ölçüde artırmaktadır. Bu nedenle sis bilişim sistemleri akıllı şebeke, akıllı şehir gibi IoT uygulamalarında farklı yerel ortamlara konuşlandırılarak geniş

bir coğrafi alanda IoT hizmetlerinin etkisini artırabilmektedir.

- **Konum bilgisi:** Sis düğümleri ağın kenarında bulunan servisleriyle uç düğümlerin konumunu saptayabilmektedir. Hareketli uç düğümlerin ağa dâhil olabilmelerini sağlamaktadır.
- **Çok sayıda düğüme hizmet:** Özellikle çok sayıda uç düğüme hizmet veren geniş coğrafi alanda çalışan IoT uygulamalarında sis bilişim kullanılabilir. kullanılabilmektedir.
- **Gerçek zamanlı uygulamalar:** Sis bilişim oldukça düşük gecikme süresine sahip olduğu için güçlü veri iletimi sağlamaktadır. Bu yüzden gerçek zamanlı IoT sistemlerde sis bilişim kullanmak performansı artırmaktadır.
- **Heterojenlik:** Sis bilişim sistemleri farklı aygıtların veya farklı servislerin birlikte çalışabilmesini sağlamaktadır. Geniş coğrafi alandaki uygulamalarda mobil uç düğümler ile iletişim kurabilen sis düğümleri birçok farklı yapıdaki aygıt ve servislere hizmet verebilmektedir.
- **Güvenlik:** Kaynak kısıtlı uç düğümler tarafından gerçekleştirilmesi zor olan şifreleme gibi hesaplama yükleri sis düğümleri tarafından yapılabilmektedir. Böylece, veriler bulut sistemde kalıcı olarak depolanmak üzere iletilirken güvensiz olan internet ortamından geçmeden önce sis düğümleri tarafından yerel ağda güvenliği sağlanabilmektedir.

Bu verimli özelliklerinden dolayı, önerilen IoT tabanlı taktik analiz ve sağlık izleme mimarisinde veri işleme ve depolama amacıyla sis bilişim kullanılmaktadır. Sis bilişim katmanının yapısı Şekil 2.5'te gösterilmektedir.



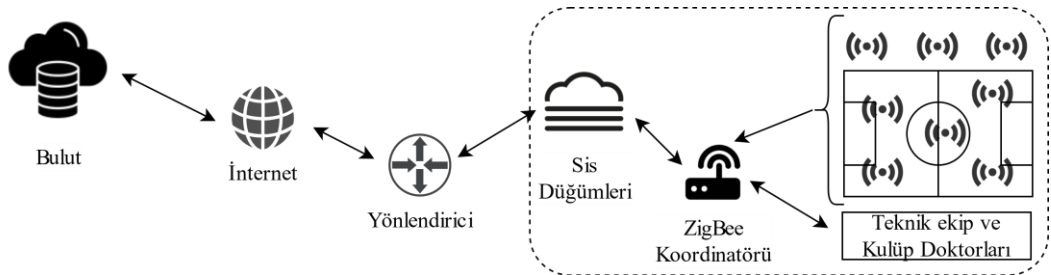
Şekil 2.5. Sis bilişim katmanı (Akleyek ve Karakaya, 2020)

3. SİS BİLİŞİM VE NESNELERİN İNTERNETİ TABANLI SAĞLIK İZLEME VE TAKTİK ANALİZ MİMARİSİ, SİS DÜĞÜMLERİNDE GÖREV YÖNETİM VE KAYNAK TAHSİS ALGORİTMASI

Bu bölümde spor alanında önerilen sağlık ve taktik analiz yapısı için IoT tabanlı mimari detayları ve bu mimaride kullanılan sis düğümleri üzerinde etkili kaynak yönetimini sağlayan ve iFogSim simülatöründe test edilen öncelik kuyruğu tabanlı görev yönetim algoritmasının detayları yer almaktadır. Önerilen IoT ve sis bilişim tabanlı mimari ve görev yönetim ve kaynak tahsis algoritması (Karakaya ve Akleylek, 2021) çalışmasında yayınlanmıştır.

3.1. IoT Tabanlı Mimari Detayları

Sporda sağlık izleme ve taktik analiz işlemlerinin daha etkili bir şekilde yapılabilmesi için önerilen IoT tabanlı mimari ve güvenlik detayları incelenmektedir. Mimaride oyuncular ve seyircilere uygun giyilebilir teknolojiler yerleştirilerek sensörler aracılığı ile sağlık durumlarının izlenmesi ve takım bazlı taktik analizlerin oluşturulması amaçlanmaktadır. Ayrıca, uygulama bölgesine yerleştirilen sensörler (sıcaklık, nem, topun konumu gibi) aracılığıyla çevresel veriler de toplanmaktadır. Sensörlerden elde edilen veriler ZigBee ağındaki sis düğümlerine iletilmektedir. ZigBee protokolü düşük güç uygulamalarında ve dış ortamda 300 metreye kadar olan mesafelerde çalışabilmektedir (Li, vd., 2010). Bu nedenle spor alanı için uygun bir protokoldür. Veriler sis düğümlerinde işlenmekte ve sonuçlar kulüp doktorlarına ve teknik ekibe gönderilmektedir. Sis düğümleri yerel ağda hizmet verebildiğinden dolayı veriler bulut sistemlere göre daha güvenli bir ortamdan iletilmekte ve gecikme süresi daha düşük olabilmektedir. Önerilen mimarinin veri akışı ve mimaride kullanılan yapılar Şekil 3.1’de gösterilmektedir.



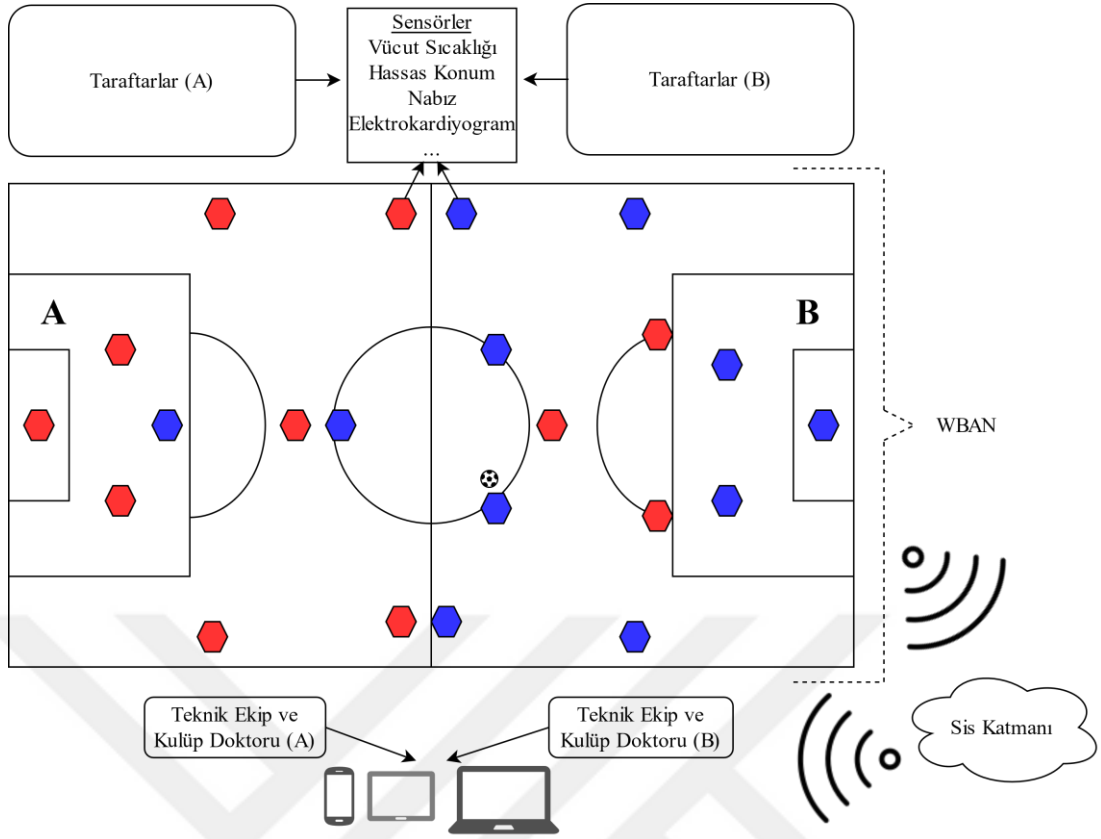
Şekil 3.1. Önerilen mimarinin veri akışı ve kullanılan yapılar

Sensörlerden alınan veriler çok büyük olduğundan ve özellikle sağlık izleme noktasında düşük gecikmeli geri bildirim gerektiğinden dolayı verilerin sis düğümlerinde depolanması, görevler halinde işlenmesi ve oluşan yanıtın iletilmesi oldukça düşük gecikme sürelerinde gerçekleşmelidir. Sis bilişim sistemleri verilerin İnternet ortamına geçmeden yerel ağ ortamında depolanabilmesine ve işlenebilmesine olanak sağladığı için sistemin mimarisinde sis bilişim yararlanılmaktadır. Ayrıca, geri bildirimde veya geri bildirimden sonra kullanılmayan veriler kalıcı depolama amacıyla bulut sistemlerinde depolanmaktadır. Verilerin uzaktaki bulut sunucularına gitmeden önce sis düğümlerinde değerlendirilmesi aynı zamanda verilerin güvensiz olan İnternet ortamına çıkmadan önce şifrelenebilmesini ve korunabilmesini de sağlamaktadır.

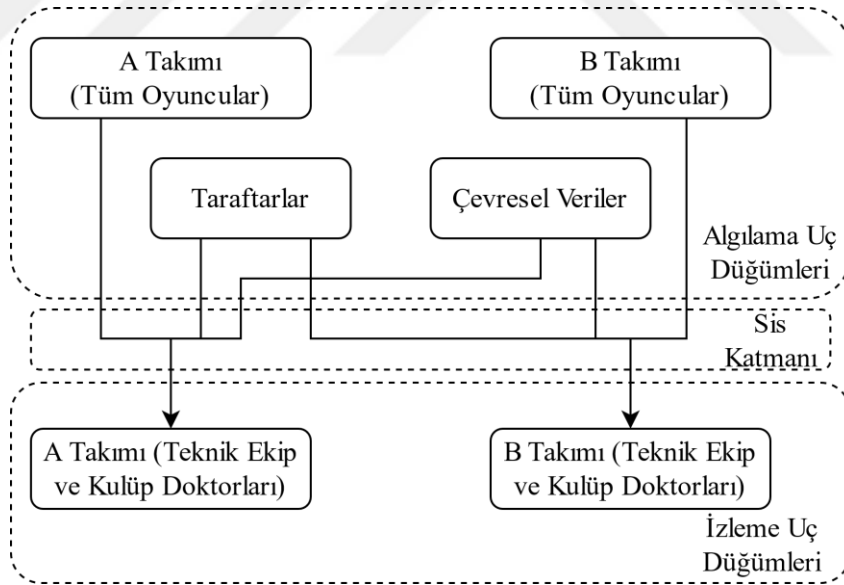
Oyunculara vücut ısı, nabız, mesafe ve hareket, hassas konum, elektrokardiyografi gibi sensörler giyilebilir teknolojiler olarak yerleştirilmektedir. Bu sensörler oyuncunun kıyafetinde veya bileklik veya vücut bantları gibi doğrudan vücudunda bulunarak WBAN oluşturmaktadır. Ancak, giyilebilir teknolojiler hala gelişmekte olduğundan bu ürünlerin temin edilmesi veya temin edilse bile verimli çalışması zordur. Bu nedenle giyilebilir teknolojiler bu önerilen mimari detayları kapsamına alınmamaktadır. Önerilen mimari spor alanında sağlık ve taktik analiz için düşük gecikme sağlayan güvenli bir altyapının oluşmasına odaklanmaktadır.

WBAN ile her bir oyuncudan alınan veriler sis düğümlerine iletilmekte ve sis düğümlerinde depolanarak görevler halinde işlenmektedir. Elde edilen sonuçlar, tahminler veya çıkarımlar teknik ekibe ve kulüp doktorlarına iletilmektedir. Bir futbol ortamı düşünüldüğünde sistemin aktörleri, olası konumları ve veri iletim yönleri Şekil 3.2’de gösterilmektedir.

Sistemin yerel ağda çalışabilmeyi desteklemesi verilerin İnternet ortamına çıkıp uzaktaki sunucuda işlenmesi durumuna kıyasla iletimdeki gecikmeleri azaltabilmektedir. Sağlık izleme veya taktik analiz işlemlerinde oluşabilecek olumsuzluklara karşı daha hızlı müdahale edilebilmesi için iletim ve veri işleme gecikmelerinin oldukça düşük olması çok önemlidir. Bu durum sakatlanma veya fiziksel belirtisi olmayan hastalıkların nüksetmesi durumunda daha erken tedaviye geçilmesine yardımcı olmaktadır. Ayrıca, teknik ekibin maç içi analizlerden anlık olarak haberdar olmasına ve taktiksel değişiklik kararları almasına da yardımcı olmaktadır. Algılama ve izleme işlemlerini gerçekleştiren uç düğüm cihazlarının veri iletişim şeması Şekil 3.3’te gösterilmektedir.



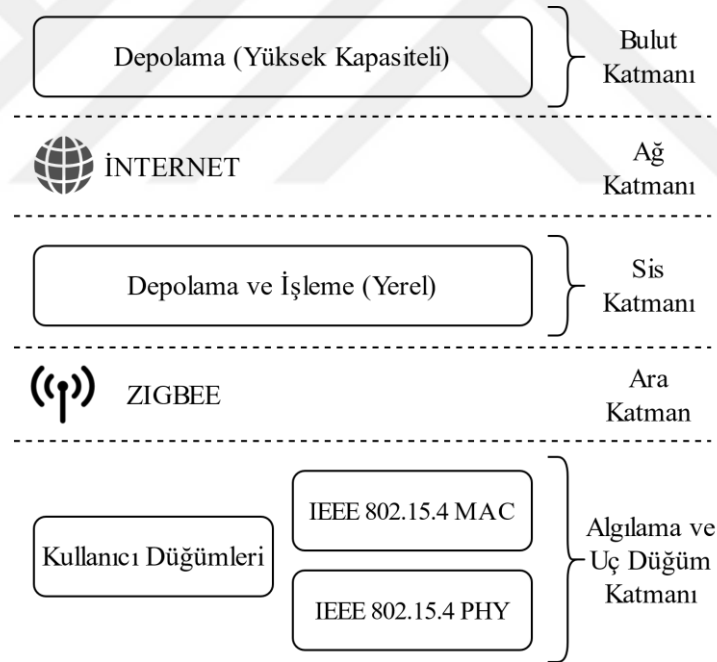
Şekil 3.2. Sistemin aktörleri, olası konumları ve veri iletim yönleri



Şekil 3.3. Algılama ve izleme uç cihazlarının veri iletişim şeması

İki takım arasındaki taktik verilerinin korunması için her takımın düğümleri kimlik doğrulaması yapmaktadır. Bir başka deyişle, iki takım taktiksel açıdan birbirinde izole olmaktadır. Kaynak kısıtlı sensör cihazlarda kimlik doğrulama yapabilmek için önerilen mimaride FLAT yöntemi kullanılmaktadır. FLAT, örtük sertifikalar kullanarak bant genişliğini azaltmaktadır. Ayrıca, önerilen mimaride

işlemlerin kaydedilmesi ve veri bütünlüğünün kontrol edilmesi için var olan bir blokzincir tabanlı SDN kontrol mekanizması kullanılmaktadır. Bulut tabanlı IoT sistemlerinde veriler İnternet ortamına çıktıklarında uzaktaki bulut sunucularına ya korumasız bir şekilde iletilmekte ya da kaynak kısıtlı uç cihazlarda şifrelenerek iletilmektedir. Bu durum ise verilerin ele geçirilerek taşınan bilgilerin çalınmasına veya uç cihazların hali hazırda kısıtlı olan kaynaklarının verimsiz kullanılmasına yol açmaktadır. Önerilen mimaride ise algılanan veriler önce yüksek kaynak kapasitesine sahip sis düğümlerine gelerek burada şifrelendikten sonra uzaktaki bulut sunucusuna kalıcı depolama için gönderilmektedir. Ayrıca, sis düğümlerinin yüksek miktarda iş yüklediği durumlarda kapasite yetmeme ihtimaline karşı önceliği düşük olan görevler de bulut sunucularında işlenebilmektedir. Güvenlik aşamasında yüksek bilgi işlem gücü gerektiren tüm işlemler sis düğümlerinde gerçekleşmektedir. Bu nedenle önerilen mimari hafif ve verimlidir. Her katmandaki sistem kaynaklarının verimli kullanımı sağlanmaktadır.



Şekil 3.4. Önerilen mimarinin katmanları

Önerilen mimarinin katmanlı yapısı Şekil 3.4'te gösterilmektedir. Katmanlarda gerçekleşen işler şu şekilde listelenebilmektedir:

- Algılama ve uç düğüm katmanı, veri toplamak için kullanılan sensörler ve sonuçları alan kullanıcı uç cihazlarından oluşmaktadır.
- Ara katmanda, verileri sis katmanına iletmek için bir ağ gerekmektedir. Bu

ağ yereldir ve çok sayıda düğümün iletişim kurmasını sağlayan ZigBee protokolünü kullanmaktadır.

- Sis katmanında, sis düğümleri tarafından veri işleme ve depolama gerçekleştirilmektedir. Veriler işlendikten sonra geri bildirim olarak algılama ve uç düğüm katmanına veya kalıcı depolanmak üzere bulut katmanına gönderilmektedir. Bulut katmanına gönderilen veriler, güvenli olmayan İnternet nedeniyle sis katmanında şifrelenmektedir.
- Ağ katmanı, sis ve bulut katmanı arasındaki ağ bağlantısını içermektedir. Bu katman, bulut ve sis katmanları arasında veri alışverişi yapmaktadır.
- Bulut katmanı, verilerin kalıcı olarak depolanması ve işlenmesi için sis katmanına destek sağlamaktadır.

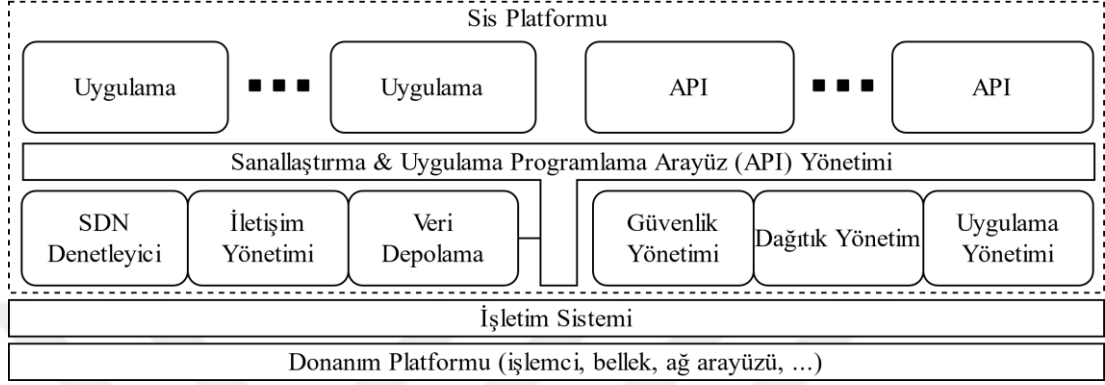
Algılama ve uç düğüm katmanından elde edilen veriler, ara katmandaki ZigBee ağları aracılığıyla sis katmanındaki sis düğümlerine gönderilmektedir. Veriler görev yönetim süreçlerinden geçtikten sonra sis düğümlerinde işlenmektedir. Sis katmanında ortaya çıkan sonuçlar için iki durum vardır. İlk durum, sonuçların kalıcı depolama için ağ katmanı üzerinden bulut katmanına gönderilmesidir. İkinci durum ise sonuçların sistem aktörlerini bilgilendirmek için ara katmandaki ZigBee protokolü aracılığıyla algılama ve uç düğüm katmanındaki kullanıcı cihazlarına geri bildirim sağlanmasıdır. Algılama ve uç düğüm katmanındaki cihazlar, aynı alanda buldukları için aynı katmanda tanımlanmaktadır. Sensörler sporculardan veri toplarken, uç cihazlar sonucu teknik ekibe bildirmektedir.

3.2. Sis Düğümlerinde Görev Yönetimi ve Önerilen Algoritma

Bu bölümde önerilen mimarinin sis katmanında görev yönetimi ve kaynak tahsisi için öncelik kuyruğu tabanlı bir algoritma önerilmektedir. Bu algoritma iFogSim ortamında test edilmekte ve benzer algoritmalar ile kıyaslanmaktadır.

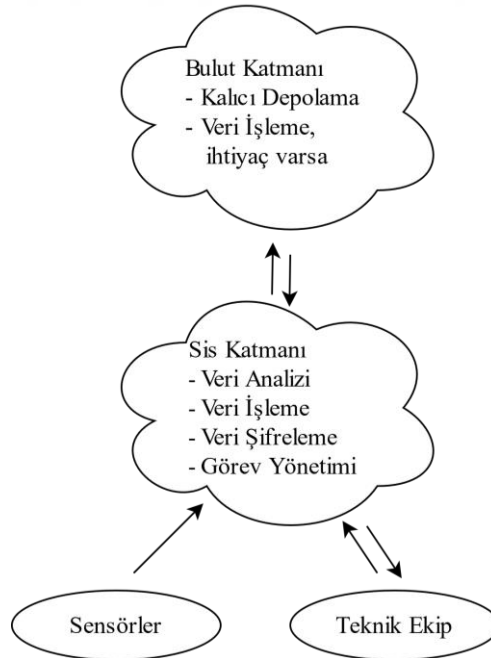
Sis düğümleri yerel ağ ortamında veri işleme ve depolama sağlamaktadır. Bir sis düğümünün yapısı Şekil 3.5'te gösterilmektedir. IoT sistemleri sık ve periyodik olarak veri üretebilmektedir. Bu veriler veri deposunda sıkıştırılmış bir biçimde saklanmaktadır. Verilerin hangi IoT cihazından ve ne zaman geldiğini belirlemek için bir zaman damgası kullanılmaktadır. İletişim yönetimi, veri dağıtım servisinden aldığı sonuçları veya geri bildirimleri CoAP üzerinden IoT uç cihazlarına dağıtmaktadır (Yoon, vd., 2019). Güvenlik yönetimi hem sis güvenlik durumlarını hem de buluta

gönderilecek verilerin şifrenmesi işlerini yürütmektedir. Dağıtık yönetim, veriler üzerinde hesaplama işlerinin yapılmasını sağlamaktadır. Uygulama yönetimi ise sis düğümlerinde kullanılan uygulamaların yönetimini sağlamaktadır. SDN denetçisi kontrol ve veri bütünlüğüne yardımcı olan dağıtılmış blokzincir tabanlı yapının denetleme işlemlerini gerçekleştirmektedir.



Şekil 3.5. Bir sis düğümünün yapısı

Önerilen sağlık ve taktik analiz izleme mimarisinde uç düğümler, sis ve bulut arasında veri alışverişi yapılmaktadır. Sis ve bulut düğümlerinde gerçekleşen görevler Şekil 3.6'da gösterilmektedir.

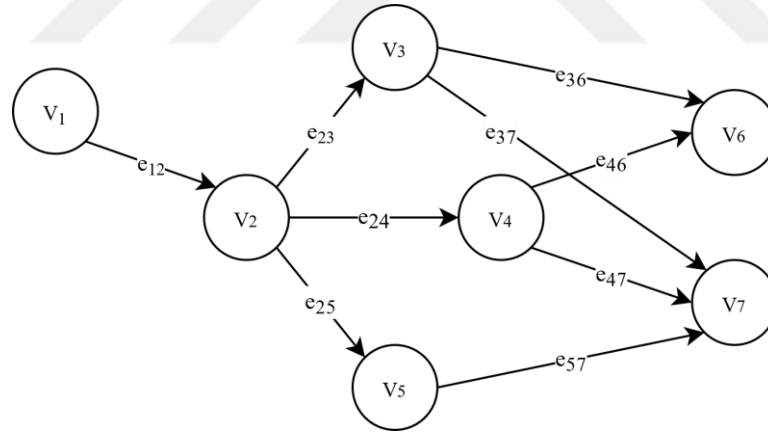


Şekil 3.6. Sis ve bulut düğümlerindeki görevler

Şekil 3.6'da sis ve bulut içindeki temel görevler ve ham veri ile sonuç verisinin yolu gösterilmektedir. Sensörler verileri toplayan aygıtlardır. Toplanan bu veriler sis

katmanına gönderilmektedir. Sis katmanında işlemlerden geçirilerek sonuçlar ya teknik ekibin ve kulüp doktorlarının cihazına ya da kalıcı depolanmak üzere şifrelenerek buluta iletilmektedir.

Şekil 3.7’deki grafta, sis düğümlerinde; veri ön işleme ve doğrulama (v_1), veri analizi ve sınıflandırma (v_2), sağlık verilerinin hesaplanması (v_3), Voronoi grafi ve Delaunay üçgenlemesi gibi yöntemlerle olası pas kombinasyonları ve kaliteli pas durumları, konumlara göre saha içi diziliş gibi taktik analiz verilerinin hesaplanması (v_4), buluttan gelen ve buluta giden veriler için şifreleme ve şifre çözme işlemleri (v_5), uç düğümlere giden verilerin hazırlanması (v_6), buluta giden verilerin hazırlanması (v_7), gibi görevler yer almaktadır. Her bir e_{ij} değeri ise i ve j görevleri arasındaki ilişkileri ve veri akışını göstermektedir. Graf, modeldeki verilerin algılandıktan sonra sınıflandırılması, analiz edilmesi ve sonuçlarının uç cihazlara veya buluta gönderilmesi süreçlerini kapsamaktadır. Şekil 3.5’teki sis düğümü yapısında bu işlemleri gerçekleştiren yönetim ve depolama alanları bulunmaktadır. Bunlar dışında kimlik doğrulama işlemleri ve kimlik sağlayıcı olarak görev yapan sis düğümleri de bulunmaktadır.



Şekil 3.7. Sis katmanı görev grafi

IoT cihazları düşük gecikme toleransı ile gerçek zamanda işlenmesi için çok sayıda veri üretmektedir. Bu nedenle sis düğümlerinde görev yönetim algoritmaları kullanılmalıdır. Önerilen mimaride Şekil 3.7’deki işlemler için öncelik kuyruğu tabanlı bir görev yönetim algoritması kullanılmaktadır. Uç cihazlardan gelen istekler en yakın sis düğümüne gelmektedir. Sis düğümü içinde yüksek (Q_H), orta (Q_M) ve düşük (Q_L) olmak üzere üç adet kuyruk bulunmaktadır. Her istek (i), isteğin orijinal öncelik düzeyi (SB_{CAT}) ile isteğin işleneceği gecikmenin (veya toplam hizmet

süresinin) ($delay_i^T$) iki eşik (T_1 ve T_2) arasında olup olmaması kriterine göre bu kuyruklara atanmaktadır. Bu süreç öncelik atama algoritması ile gerçekleşmektedir. Sis düğümlerinde, bir kullanıcıdan gelen isteğin işleneceği gecikme süresi ($delay_i^T$), istek tarafından verilen son teslim (deadline) süresi ve sis düğümüne geliş zamanının farkı alınarak bulunmaktadır. Sis katmanında bir istek için harcanan toplam süre (W_i) ise isteğin kuyruқта beklediği süre ile isteğin işlem süresinin toplamı kadardır. Algoritmada kullanılan SLA gereği isteğin (i) QoS gereksinimini karşılayabilmesi için $W_i < delay_i^T$ koşulu sağlanmalıdır.

Şekil 3.8'deki öncelik atama algoritmasına göre (Choudhari, vd., 2018); bir istek (i), kullanıcıdan gelen talebin toplam hizmet süresi ($delay_i^T$), (i)'nin tahmin edilen hizmet süresinden (ST_{est}^i) küçük veya eşitse en yüksek önceliğe sahip olan (Q_H) kuyruğa atanmaktadır. Diğer durumda, talebin toplam hizmet süresinin T_1 ve T_2 eşik değerlerinin arasında olup olmadığına bakılmaktadır. Eğer bu aralıkta ise isteğin orijinal öncelik düzeyine (SB_{CAT}) göre Q_H , Q_M ve Q_L kuyruklarından hangisine atanacağı belirlenmektedir. Son olarak, $delay_i^T$ değeri T_2 'den büyük ise istekler orijinal önceliklerine uygun olan kuyruğa atanmaya zorlanmaktadır. Kuyrukların kapasitesi doluyorsa istek bir alt düzeydeki kuyruğa atanmaktadır. Böylece, daha yüksek öncelikli ve daha düşük gecikmeye sahip olan işlerin yürütülmesine öncelik verilmektedir.

Görev yönetim ve kaynak tahsisi için önerilen algoritmanın parametreleri ve tanımları Tablo 3.1'de verilmektedir.

Tablo 3.1. Önerilen algoritmadaki parametreler

Parametreler	Açıklamalar
Q_H, Q_M, Q_L	Öncelik kuyrukları (yüksek-orta-düşük)
req_i	İstek
$delay_i^T$	i isteği için gecikme maliyeti
ST_{est}^T	Kuyruktaki isteklerin tahmin edilen toplam kapasitesi
ST_{est}^k	k düğümünün tahmin edilen hizmet kapasitesi
$node_{dt}$	Dağıtıcı düğümü
$node_k$	k numaralı düğüm
L_{node}	Düğüm kapasitelerinin sıralı listesi
L_{node}^k	Düğüm kapasite listesinin k .elemanı

```

Priority(delayiT, STestk)
if (delayiT ≤ STestk) then
    return(H)
else if (T1 < delayiT ≤ T2)
then
    if (SBCAT = 1) then
        return(H)
    else if (SBCAT = 2) then
        return(M)
    else if (SBCAT = 3) then
        return(L)
else if (delayiT > T2) then
    if (SBCAT = 1) then
        if (QH is not full) then
            return(H)
        else
            return(M)
    else if (SBCAT = 2) then
        if (QM is not full) then
            return(M)
        else
            return(L)
    else if (SBCAT = 3) then
        return(L)

```

Şekil 3.8. Öncelik atama algoritması

Sis düğümlerinde görev yönetim için önerilen algoritma Şekil 3.9’da gösterilmektedir. Şekil 3.8’deki algoritma görevlerin öncelik kuyruğuna atanması için kullanılmakta iken Şekil 3.9’daki algoritma kuyruklardaki isteklerin sis düğümlerine atanıp işlenmesi ve kaynakların verimli değerlendirilmesi için kullanılmaktadır. Şekil 3.9’daki görev yönetim ve kaynak tahsis algoritması iki adımdan oluşmaktadır. Adım 1’de, istek en yakın sis düğümüne ($node_{dt}$) gönderilmektedir. $node_{dt}$, diğer düğümlere görev dağıtımını yapmaktadır. Her düğüm boşta olan kaynak miktarı değerinden yapabileceği iş süresini tahmin etmektedir. Her bir düğümün tahmin edilen toplam hizmet süresi (k.düğüm için ST_{est}^k), $node_{dt}$ ’ye bildirilmektedir. $node_{dt}$ ’de her düğümün alabileceği istek bilgisi bir listede (L_{node}) artan sıralı halinde tutulmakta ve bu liste sürekli güncellenmektedir. Gelen istekler (req_i) Şekil 3.8’deki öncelik atama algoritmasına göre kuyruklara yerleştirilmektedir. Adım 2’de, tüm kuyruklardaki istekler Q_H , Q_M , Q_L sırasına göre işlenmektedir. $node_{dt}$ ’deki L_{node} listesinde düğümün verebileceği toplam hizmet süresi (ST_{est}^k), isteğin toplam gecikme süresine ($delay_i^T$) eşit veya büyük olduğunda istek (req_i) o düğüme atanmakta ve döngüden

çıkılarak sonraki istek değerlendirilmektedir. Eğer istek tek bir düğüme atanamamışsa, L_{node} listesi bu kez sondan başa doğru (büyükten küçüğe doğru) dolaşmakta ve her adımda sis düğümlerinin tahmin edilen toplam işlem süreleri toplanmaktadır.

Adım 1

Her sis düğümü ST_{est}^k (k. düğüm için) değerini $node_{dt}$ 'ye gönderir. $node_{dt}$ bu değerleri L_{node} listesine atar.

foreach req_i **do**

req_i , $node_{dt}$ 'ye gönderilir.

$(Pr_i) = Priority(delay_i^T, ST_{est}^T)$

if $(Pr_i = H)$ **then**

$Q_H(last) \leftarrow req_i$

else if $(Pr_i = M)$ **then**

$Q_M(last) \leftarrow req_i$

else if $(Pr_i = L)$ **then**

$Q_L(last) \leftarrow req_i$

Adım 2

/* ST_{est}^k ve $delay_i^T$ 'ye göre sis düğümlerine görev dağılımı */

foreach req_i **in** Q_H, Q_M, Q_L **do**

for L_{node}^k **do**

if $(ST_{est}^k \geq delay_i^T)$ **then**

$node_k \leftarrow req_i$

break

if $(req_i$ herhangi bir sis düğümüne atanmamış ise) **then**

$reversedList \leftarrow reverse(L_{node}^k)$

$sum \leftarrow L_{node}^{first}$

for $reversedList$ **do**

if $(sum \geq delay_i^T)$ **then**

$L_{node}^{first}, L_{node}^{second}, \dots, L_{node}^k \leftarrow req_i$

break

else

$sum \leftarrow sum + L_{node}^{next}$

if $(req_i$ düğümler grubuna da atanmamış ise) **then**

if $(req_i$ in $Q_H)$ or $(req_i$ in $Q_M)$ **then**

$reject(req_i)$

else if $(req_i$ in $Q_L)$ **then**

$cloud \leftarrow req_i$

Şekil 3.9. Önerilen görev yönetim ve kaynak tahsis algoritması

Döngünün her adımında ortaya çıkan toplam süre (sum), $delay_i^T$ 'ye eşit ve

büyükse isteği karşılayan düğümler grubuna atanmakta ve döngüden çıkılarak sonraki istek değerlendirilmektedir. Bu aşamada listenin ilk elemanı (L_{node}^{first}) listenin ters çevrilmeden önceki son düğümün tahmin edilen hizmet süresini (ST_{est}^{last}) tutmaktadır. Eğer istek düğümler grubuna da atanamamış ise bu istek sis katmanında boştaki kaynaklar tarafından karşılanamamaktadır. Bu durumda isteğin bulunduğu kuyruğa göre işlem yapılmaktadır. Eğer istek Q_H veya Q_M kuyruklarında ise reddedilmekte, Q_L kuyruğunda ise öncelik düzeyi düşük olduğundan dolayı buluta gönderilmektedir.

Eşitlik 3.1, req_i talebini karşılayan düğümleri ve bu düğümlerin sınırlarını belirlemek için kullanılmaktadır.

$$L_{node}^i \leq \sum_{k=1}^n L_{node}^k \quad n \leq size(L_{node}), L_{node}^i \geq delay_i^T \quad (3.1)$$

Her istek için kaynakların toplam kapasitesi Eşitlik 3.2'ye göre belirlenmektedir. Bu eşitlikte L_{node}^U güncellenmiş kapasiteyi gösterirken L_{node}^C mevcut kapasiteyi göstermektedir.

$$L_{node}^U = L_{node}^C - L_{node}^i \quad (3.2)$$

Her istek tamamlandıktan sonra kaynakların toplam kapasitesi Eşitlik 3.3'e göre güncellenmektedir.

$$L_{node}^U = L_{node}^C + L_{node}^i \quad (3.3)$$

Şekil 3.8'deki öncelik atama algoritmasının Big-O karmaşıklık analizinde $delay_i^T$ ve ST_{est}^i değerlerinin hesaplandığı varsayıldığında her bir istek $O(1)$ karmaşıklığında gerçekleşmektedir. n adet istek için toplam karmaşıklık $O(n)$ olmaktadır. Şekil 3.9'daki önerilen görev yönetim ve kaynak tahsis algoritmasının Big-O karmaşıklık analizinde Adım 1, Şekil 3.8'deki algoritmayı kullandığı için n adet istek için $O(n)$ karmaşıklığına sahiptir. Adım 2'de her bir kuyrukta işlenmek zorunda olan n adet istek ve bu istekleri karşılamaya çalışan m adet sis düğümü olduğu varsayılmaktadır. Adım 2'nin en kötü durumu, isteğin herhangi bir sis düğümüne atanmamış olmasıdır. Bu nedenle Adım 2 için en kötü durum karmaşıklığı $n \times m + m + 6$ şeklindedir. Adım 1 ve Adım 2 birlikte düşünüldüğünde en kötü durum karmaşıklığı $n + n \times m + m + 6$ olarak bulunmaktadır. Böylece, Şekil 3.9'da önerilen algoritmanın Big-O notasyonuna göre karmaşıklık değeri $O(n \times m)$ olmaktadır.

3.3. Önerilen IoT Tabanlı Mimarinin Güvenliğinde Kullanılan Yöntemler

Bu bölümde önerilen mimarinin güvenliğinin sağlanması için kullanılan yöntemler anlatılmaktadır. Sensör düğümler ile algılanan verilerin sis düğümlerine iletilmesi, sis düğümlerinde işlendikten sonra ortaya çıkan sonuçların uç düğümlere geri bildirilmesi veya İnternet ortamı üzerinden buluta iletilerek depolanması işlemlerinde çok sayıda düğüm arasında iletişim meydana gelmektedir. Önerilen mimaride sporda sağlık izleme ve taktik analiz için olumsuzluklar en aza indirilmeye çalışılırken elde edilen verilerin manipülasyonu, başka kişilerin sisteme sızması bu olumsuzlukları daha da kötü hale getirebileceği için sistemin güvenliğinin sağlanması çok önemlidir. Hem takımlar arasında hem de dış ortamdaki kişilerin sisteme erişimleri belirli yetkiler ile kısıtlanmalıdır. Ayrıca, kablosuz ortamların doğası gereği dış tehditlere karşı savunmasız olduğundan dolayı da sistemin güvenliğinin sağlanması gerekmektedir. Önerilen mimarinin ara katmanında tanımlanan yakın mesafe ZigBee ağlarında yenilik sayaçları, AES şifreleme yöntemi, şifrelerin kısa periyotlarda yenilenmesi gibi güvenlik mekanizmaları bulunmaktadır (Li, vd., 2010). Ancak, bunlara ek olarak düğümlerin birbirini tanınması, kimlik doğrulama yapılması ve İnternet ortamında verilerin şifreli iletilmesi gerekmektedir. Bu yüzden önerilen modelde kimlik doğrulama ve veri gizliliği için FLAT yöntemi kullanılmaktadır (Santos, vd., 2020). Ayrıca, blokzincir tabanlı SDN denetçisi veri bütünlüğünün korunmasına destek vermekte ve özellikle su baskını saldırıları tespit edilebilmektedir. Sis ve bulut katmanı arasındaki iletişim açık anahtar şifreleme sistemleri kullanılarak sağlanabilmektedir.

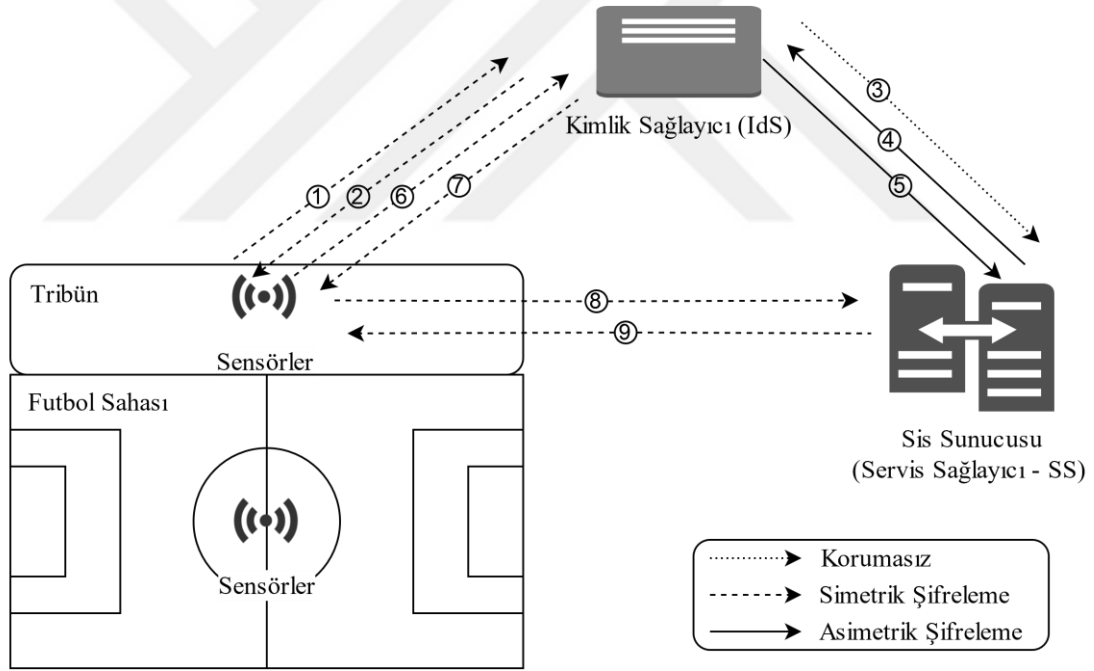
Önerilen mimaride, kaynak kısıtlı IoT sensörleri arasında ve bu sensörler ile sis düğümleri arasında kimlik doğrulama için FLAT yöntemi kullanılmaktadır. Sis düğümlerine yalnızca yasal sensörlerin verileri alınmaktadır. Saldırgan bir düğüm herhangi bir şekilde FLAT kimlik doğrulama aşamasını başarıyla geçerse, veriler SDN denetleyicisinden de geçmekte ve oluşabilecek su baskını gibi saldırılar denetlenebilmektedir. Bu sayede güvenilmez durumlara karşı ağın tıkanması ihtimaline karşı uyarı alınabilmektedir. Önerilen mimarinin güvenliği ZigBee ağının sağladığı güvenlik yöntemlerinin yanı sıra bu iki aşama ile desteklenmektedir.

3.3.1. Veri Gizliliği ve Kimlik Doğrulama

Veriler iletilirken kimlik doğrulama yapılması ve mahremiyetin korunması gerekmektedir. Bunun için önerilen mimaride FLAT protokolü kullanılmaktadır.

FLAT, bulut kimlik doğrulama sistemi olan birleşik kimlik yönetimi yapan FIDM'nin sis bilişime uyarlanmış halidir (Santos, vd., 2020). IoT teknolojisi kaynak kısıtlı cihazlardan oluştuğu için FIDM'nin avantajlarından tam olarak yararlanamamaktadır. Çünkü FIDM kimlik doğrulama ve şifreleme işlemlerini yüksek maliyetli olarak gerçekleştirmektedir. FIDM, bulut ve sis arasındaki güvenlik işlemlerini sağlamak için kullanılabilir. Ancak, kaynak kısıtlı cihazlar üzerinde kimlik doğrulama ve şifreleme işlemleri FLAT ile yapılabilir.

FLAT, hesaplama gücü düşük olan istemci, hesaplama gücü yüksek olan servis sağlayıcı (SS) ve kimlik sağlayıcı (IdS) cihazlar ile çalışmaktadır. İstemciler önerilen mimarideki algılama ve uç düğüm katmanında yer alan aygıtlardan oluşmaktadır. Bunlar arasında kaynak kısıtlı sensör cihazlar da yer almaktadır. SS, bir sis sunucusu iken, IdS ise kesintisiz çalışması gereken ve kimlik onaylarını sağlayan sis ağı içindeki bir sunucudur.



Şekil 3.10. FLAT Kimlik doğrulama şemasının önerilen mimari üzerinde işleyişi

FLAT protokolü için kaynak kısıtlı IoT cihazlarına ortama konuşlanmadan önce simetrik anahtar atanmaktadır. Bu anahtarlar fiziksel saldırılara karşı dirençli olduğu gerekçesi ile fiziksel kopyalanamayan işlevler (Physical Unclonable Functions - PUF) ile oluşturulmaktadır. PUF çıktısı üretildikten sonra biçimlendirilerek saklanmaktadır. Kaynak kısıtlı IoT cihazları ile IdS arasında güven sağlanması için biçimlendirilmiş PUF çıktıları kullanılmaktadır. IdS ve SS arasındaki güven ise ortak bir sertifika

yetkilisi tarafından verilen dijital sertifikalar ile sağlanmaktadır.

FLAT protokolünde süreç kayıt ve kimlik doğrulama olmak üzere iki aşamadan oluşmaktadır. Kayıt aşamasında, güvenilen taraflar, sertifikalarını farklı bir güvenilir sertifika yetkilisinden almaktadır. Kimlik doğrulama aşamasında ise, iki düğüm karşılıklı kimlik doğrulamak ve güvenilir iletişim için gizli bir anahtar oluşturmak amacıyla sertifikalarını değiştirebilmektedir. Kayıt aşaması, protokol çalışmaya başlamadan önce yürütülmelidir. Ancak, FLAT yöntemi, protokolün çalışmasından önceki işlemleri ele almamıştır (Santos, vd., 2020). Bu nedenle düğümlere verilen sertifikaların süresi ile ilgili bir detay içermemektedir.

Spor alanındaki sensörler, IdS ve SS FLAT sistemi üzerindeki Şekil 3.10'daki gibi iletişim sağlamaktadır. FLAT protokolünün işleyişine göre;

- Adım 1: kaynak kısıtlı istemci cihaz servise erişmek istediğinde IdS'den bir oturum anahtarı istemektedir.
- Adım 2: IdS, oturum anahtarını istemciye iletmektedir.
- Adım 3, 4 ve 5: IdS ve SS birbirlerine sertifikalarını göndermektedir. Güvenli bir iletişim kanalı oluşturduktan sonra IdS, SS'ye istemci anahtarını göndermektedir.
- Adım 6 ve 7: FIdM sisteminde olduğu gibi istemci IdS'den bir onay talebinde bulunmakta ve IdS'den onayı almaktadır.
- Adım 8: İstemci, IdS'den aldığı onayı SS'ye sunarak hizmet talebinde bulunmaktadır.
- Adım 9: SS, onayın doğru olması durumunda hizmet sağlamaktadır.

Oyunculara ve spor sahasına yerleştirilen her sensör, sis düğümlerine veri göndermeden önce sis düğümleriyle güvenli bir oturum kurmalıdır. Bu nedenle mimaride FLAT hafif kimlik doğrulama protokolü kullanılmaktadır. Bu işlemten sonra görev yönetimi süreci başlar.

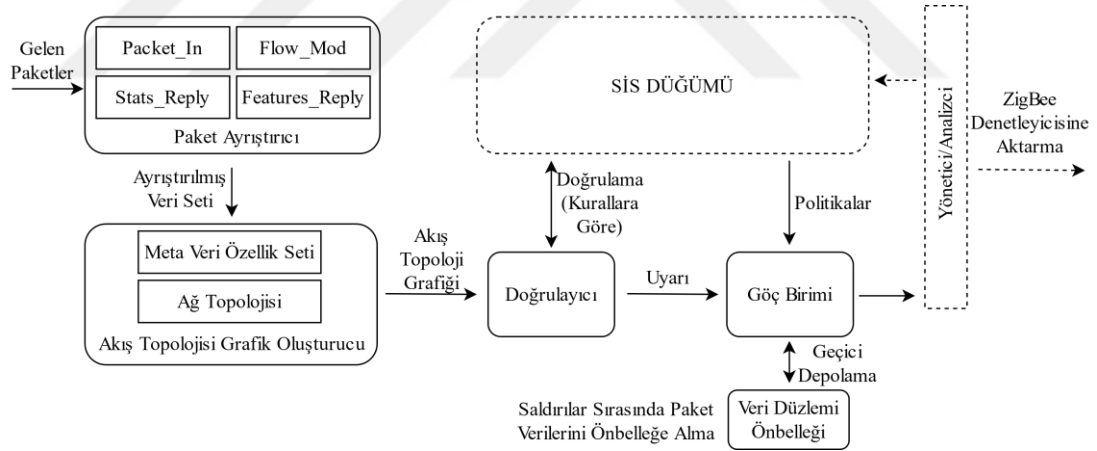
FLAT sisteminde hem SS hem de istemci, IdS'ye güvenmek zorundadır. IdS ile SS arasındaki iletişim asimetrik şifreleme yöntemleri ile, IdS ile istemci ve SS ile istemci arasındaki iletişimler ise simetrik şifreleme yöntemleri ile korunmaktadır. Böylece kaynak kısıtlı istemci cihazları maliyeti yüksek olan asimetrik şifreleme yapısını kullanmak zorunda değildir ve sis bilişim kullanan IoT sistemleri için FLAT

verimli bir kimlik doğrulama yöntemi sunmaktadır.

FLAT protokolü bilgi güvenliği ilkelerinden, kimlik doğrulama için MAC'leri ve dijital imzayı, veri gizliliği için simetrik ve asimetrik şifrelemeyi, veri bütünlüğü için MAC'leri, dijital imza ve PUF'leri, kullanılabilirlik için iki tarafın kimlik doğrulamasını ve mahremiyet için ise kimliğe erişimi sınırlayan FIDM modelini kullanmaktadır (Santos, vd., 2020).

3.3.2. Blokzincir Tabanlı SDN Denetçisi ile Güvenlik Desteği

Blokzincir bir düğümde yapılan işlemlerin her düğüm tarafından takip edilebilmesini sağlayan, kayıt defteri görevi gören bir depolama teknolojisidir (Nofer, vd., 2017). Bir blokzincir, veri paketlerinden oluşmakta ve eklenen yeni bloklar ile genişletilmektedir. Bu nedenle eksiksiz bir işlem geçmişi defterini temsil etmektedir. Sporda sağlık ve taktik analiz izleme mimarisinin sis katmanında SDN denetleyicisine sahip sis düğümleri kullanılmaktadır. Blokzincir tabanlı SDN denetleyicisinin yapısı Şekil 3.11'de gösterilmektedir (Sharma, vd., 2017). Bu yapı, Şekil 3.5'te gösterilen sis düğümündeki SDN denetleyicisi bölümünü kapsamaktadır.



Şekil 3.11. Blokzincir tabanlı SDN denetleyicisinin yapısı

Algılama ve uç düğüm katmanındaki aygıtlar ile sis düğümleri arasındaki iletişim ZigBee denetleyicisi ile sağlanmaktadır. ZigBee denetleyicisi sis düğümündeki SDN denetleyicisi için bir ağ geçidi veya yönlendirme anahtarı olarak kabul edilmektedir. Şekil 3.11'deki SDN denetleyicisinin yapısı üç aşamadan oluşmaktadır (Sharma, vd., 2017).

- Adım 1: Paket ayrıştırıcı, gelen paketlerden temel mesajları tanımlamak için izleme ve ayrıştırma işlevlerini yürütmektedir. Anormal davranışları

tanımlayabilmek için gelen her paket yakalanmalı ve ayrıştırılmalıdır. Paket ayrıştırma aşaması Features_Reply, Stats_Reply, Flow_Mod ve Packet_In şeklinde dört mesajdan oluşur. Meta verileri ayıklamak için paket ayrıştırıcı gelen paketleri dinamik olarak izlemektedir. Bir saldırgan SDN denetleyicisinin ağ yapısını değiştirmek isterse bu mesajların bir alt kümesini ele geçirmek zorundadır.

- Adım 2: Akış topolojisi grafik oluşturucu, paket ayrıştırma aşamasından gelen veri kümelerini analiz etmek ve ağ akış topolojisinin bir grafiğini oluşturmak için meta veri özellikleri kümesini ve ağ topolojisini ayrıştırma işlevleri yürütülmektedir. Her akış ile ilgili meta verilerin mantıksal ve fiziksel topoloji bilgileri saklanmaktadır. Sistem, oluşturulan akış topolojisinin grafiğine ve her akış için aktarılan bayt/paket istatistiklerine bakarak yapılan değişiklikleri, kötü niyetli güncellemeleri ve güvenlik stratejisi ihlallerini ayırt edebilmektedir.
- Doğrulayıcı, bir kümenin meta veri akışını ve yönetici tarafından belirlenen yönetim stratejileri boyunca izin verilen meta veri değerlerini doğrulamaktadır. Yönetim stratejileri doğrultusunda tanımlanan saldırıları işaretlemektedir. Her akış için değil sadece güvenilir olmayan bir durumu tanıdığında uyarı vermektedir. Göç birimi, alınan uyarılar ile özellikle su baskını gibi saldırıları tanımakta ve kararlar vermektedir. Bu kararlar ayrıştırıcının reaktif kurallarına eklenmektedir. Ayrıca, su baskını saldırısı durumunda denetleyiciyi taşımamak ve aşırı yüklememek için paketleri geçici depolama alanı olan veri düzlemi önbelleğine göndermektedir. Akış kuralı güncellendikten sonra önbellekteki paketler işlenmektedir.

Sis katmanında blokzincir tabanlı SDN denetleyici sayesinde veri değişiklikleri ve su baskını saldırıları algılanmaktadır (Sharma, vd., 2017). Böylece veri bütünlüğü korunmaktadır. Sensörlerden sis düğümlerine veri aktarımı sırasında veri bütünlüğünü sağlamak için SDN denetleyici yöntemi önerilen mimariye entegre edilebilmektedir. Şekil 3.11'deki gelen paketler önerilen mimaride sensörlerden gelen ham veriler olarak kabul edilebilmektedir. Bu yöntem kullanıldığında sis düğümlerinde görev yönetimi başlamadan önce topoloji üzerindeki veri akışı incelenebilmekte ve verilerin sis düğümlerine güvenli bir şekilde ulaşmasına destek sağlanabilmektedir.

3.4. Önerilen IoT Tabanlı Mimarinin Güvenlik Analizi

Bu bölümde önerilen mimari, düşük gecikme süresi, veri doğruluğu, kimlik doğrulama ve gizlilik, veri bütünlüğü ve erişilebilirliği, bant genişliği tasarrufu açısından analiz edilmektedir.

3.4.1. Düşük Gecikme Süresi

Sporla sis bilişim ve IoT tabanlı sağlık ve taktik analiz izleme için önerilen mimaride sis bilişimin yerel ağda çalışmasından dolayı yanıt gecikme süresi düşüktür. Böylece sağlık konusunda büyük öneme sahip olan hızlı müdahale için olumlu bir yaklaşım sunmaktadır. Aynı zamanda taktiksel açıdan maç anında yapılacak değişikliklere karar verilmesi gibi teknik konular için de hızlı sonuç alınması önemlidir. STATS analitik şirketindeki yapay zekâ uzmanlarının “Rakibin dağılmaya eğilimli olduğu özel bir senaryo tespit ederseniz, bu size rakibin organize olamadığı 30 saniyelik bir pencere verir.” şeklindeki araştırma verisi de taktik analiz olarak hızlı sonuçlar alınmasının önemli olduğunu göstermektedir (Burn-Murdoch, 2018). Veriler İnternet ortamına çıkmadan sis sunucularında işlenerek yanıt üretildiği için gecikme süresi bulut sistemlere daha düşük olabilmektedir.

Önerilen öncelik kuyruğu tabanlı görev yönetim ve kaynak tahsis algoritması ile sonuçların işlenmesi ve iletilmesi işlemleri daha düşük gecikmeler daha az maliyet ile yapılabilmektedir. Böylece öncelik değeri yüksek olan görevlere öncelik verilmekte ve işlem süresi kısaltılmaktadır. Bu durum mimari simülasyonunda detaylı bir şekilde ele alınmaktadır. Algılanan verilerden yanıt gecikme süresinin önemsenmediği veya kalıcı olarak depolanması gereken veriler ise sis katmanında şifrelenerek uzaktaki bulut sunucuda depolanmaktadır.

3.4.2. Algılanan Verilerin Doğruluğu

Pozisyon analizlerinin yapılabilmesi için sporculardan ve toptan doğru verilerin elde edilmesi gerekmektedir. Bunun için lineer konum sensörleri kullanılmaktadır. Bu sensörlerden alınan veriler ile her bir düğümün anlık konumları belirlenerek Voronoi grafi, Delaunay üçgenlemesi, baskın bölgeler ve gölgelendirme gibi taktiksel yapılar kontrol edilebilmektedir. Bu yöntemler taktik analizinin çıkarılmasına yardımcı olmaktadır. ZigBee ağı kullanılarak kaynak kısıtlı cihazlar birbirleri ile iletişime geçebilmektedir. ZigBee ağının kapsamının saha sınırları içinde olması ve içerisinde AES simetrik şifreleme yapısını kullanması verilerin manipülasyon riskini azaltarak

dođru bir Őekilde elde edilmesine katkı sađlamaktadır. Sis dđđümleri uĉ cihazlara yakın yerlerde konuŐlandırıldıklarından dolayı uzaklıktan kaynaklı veri bozulma riski oldukĉa azalmaktadır.

3.4.3. Kimlik Dođrulama ve Veri Gizliliđi

Önerilen mimarinin güvenliđinde kimliklerin dođrulanması ve verilerin gizliliđi için FLAT protokolü kullanılmaktadır. FLAT kaynak kısıtlı cihazlar barındıran ve sis biliŐim kullanan IoT uygulamaları için verimlidir (Santos, vd., 2020). Bulut biliŐim kullanan FIdM'de istemci önce SS ile iletiŐime geĉtikten sonra IdS'ye yönlendirilmektedir. FLAT protokolünde bu yükü azaltmak için istemci dođrudan IdS ile iletiŐime geĉmektedir. SS ile IdS arasındaki sertifika deđiŐimi asimetrik Őifreleme ile yapılmaktadır. Her iki sađlayıcı da yüksek hesaplama gücüne sahip olduđu için asimetrik Őifrelemenin yükü önemsenmemektedir. Kaynak kısıtlı istemcilere sistem kurulmadan önce PUF biçimlendirilmiş ĉıktısı olarak simetrik anahtarlar verilmektedir. Bu yüzden istemciler ile SS ve IdS sunucuları arasındaki iletiŐim daha hafif olan simetrik Őifreleme ile güvence altına alınmaktadır. Önerilen mimaride, kimlik dođrulama ve veri gizliliđi için FLAT yöntemi, asimetrik ve simetrik Őifrelemeler kullanılmaktadır.

FLAT protokolüne göre taŐınan veriler Őifrelendiđi için MITM saldırılarında veriler ele geĉirilse bile içeriklere eriŐilememektedir. Ayrıca, MITM'nin önlenmesi için, IdS ve SS iletiŐiminde paylaŐılan anahtar SS'ye gönderilmeden önce sertifika deđiŐimi ile dijital imzalar dođrulanmaktadır. İstemcinin kimliđi de simetrik anahtarlar kullanılarak IdS tarafından dođrulanmaktadır. Saldırgan, meŐru bir sertifika yetkilisi tarafından verilmeyen bir örtük sertifikayı ve sis dđđümünün özel anahtarını elde edememektedir. Bu nedenle sis dđđümleri sertifika ve dijital imza kullandıkları için kimliđe bürünme saldırılarının da önüne geĉilebilmektedir.

3.4.4. Veri Bütünlüđu ve EriŐilebilirlik

Önerilen mimaride sensör cihazlardan alınan veriler kablosuz ortamlardan sis dđđümlerine iletildiđi için saldırılara karşı zafiyetleri olabilmektedir. Bu nedenle ZigBee ađı tarafından bazı güvenlik önlemleri (AES Őifreleme gibi) alınmakta ve kimlik dođrulama için hafif ve güvenli FLAT yöntemi kullanılmaktadır. Ayrıca, FLAT dijital imza yöntemi ile sis katmanı tarafında veri bütünlüđünün korunmasında da destek olmaktadır. Kaynak kısıtlı uĉ dđđümlerden gelen verilerin bütünlüđünün

korunmasında ise sis katmanında blokzincir tabanlı SDN denetleyicisi yapısı kullanılmaktadır. Gelen paketler ayrıştırılarak bir akış topoloji grafiği oluşturulmakta ve tanımlanmış kurallar doğrultusunda doğrulayıcı alanında veriler doğrulanmaktadır. Özellikle su baskını saldırılarına karşı etkili bir şekilde çalışmaktadır. SDN denetleyicisinin taşmaması ve aşırı yüklenmemesi için gelen paketler veri düzlemi önbelleğinde depolamaktadır. Böylece bilgi kaybı ve taşma olmadan sistem çalışmasına devam etmektedir. Daha sonra önbelleğe alınan paketler sırası geldikçe işlenmeye devam edecektir. Bu sayede SDN denetleyicisi sahte topoloji, ARP zehirlenmesi ve DDos gibi saldırıları ayırt edip uyarı verebilmektedir (Sharma, vd., 2017). Blokzincirin dağıtık veri yönetimi ile sistemin maliyeti düşürülerek verilerin bütünlüğü ve güvenliği sağlanmaktadır (Yang, vd., 2018).

3.4.5. Bant Genişliği Tasarrufu

Sertifika değişimlerinde örtük sertifika yapısı kullanılmaktadır. ECQV olarak da adlandırılan örtük sertifika bir ortak anahtar sertifikasıdır (Santos, vd., 2020). Örtük sertifikalar ID, açık anahtar ve sertifika yetkilisinin dijital imzasını içermektedir. Hem açık anahtar hem dijital imza ayrı ayrı değil bir ortak anahtar sertifikası olarak gönderildiği için sertifika boyutları azalmaktadır. Böylece bant genişliğinden önemli ölçüde tasarruf edilmektedir.

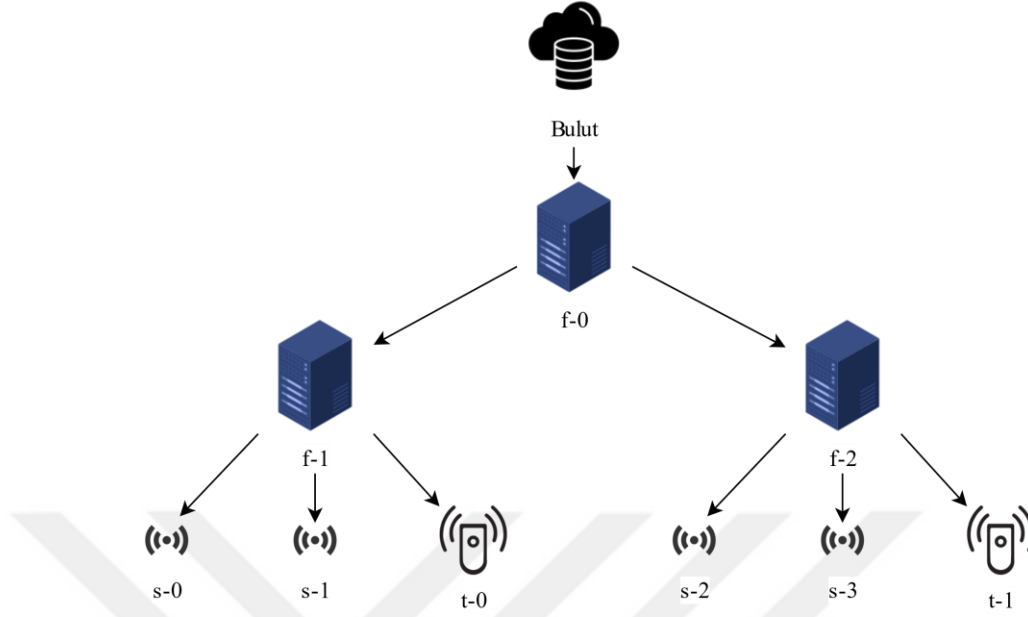
3.5. Önerilen Mimarinin ve Algoritmanın Simülasyonu

Bu bölüm, önerilen mimarinin sis katmanında çalışan önerilen algoritmanın iFogSim programı kullanılarak detaylı simülasyonunu içermektedir. Simülasyonda sensörlerden elde edilen verilerin güvenlik adımlarını başarıyla geçtiği ve gelen verilerin tamamen yasal veriler olduğu varsayılmaktadır. Simülasyon, mimariyi temsilen için bir ağ topolojisi oluşturulmasını ve sis düğümlerinde önerilen görev yönetimi ve kaynak tahsisi algoritmasını içermektedir. Simülasyon kodlarına <https://github.com/akarakaya/SimApp> adresinden erişilebilmektedir.

3.5.1. Simülasyon Topolojisi

Önerilen mimarinin sis katmanındaki düğüm sayısı ve buna bağlı olarak topoloji yapısı değişebilmektedir. Önerilen algoritmanın simülasyonunda, sis katmanında iki seviyeden oluşan üç sis düğümü kullanılmaktadır. Şekil 3.12'de belirtilen topolojide, üç düğümden biri yalnızca bulutla iletişimden sorumludur ve depolama görevlerini yerine getirmektedir. Diğer iki düğüm uç düğümlerin isteklerini gerçekleştirmekte ve

sonuçları sistem aktörlerine iletmektedir.



Şekil 3.12. Simülasyonda baz alınan topoloji

Uç düğümlerin isteklerini işlemekle görevli f-1 ve f-2 düğümlerini simülasyon kodunda “numOfEndDevPerGateway” değişkeni temsil etmekte iken, bulut ile iletişimden sorumlu f-0 düğümünü simülasyonda “numofGateways” değişkeni temsil etmektedir. s-0, s-1, s-2 ve s-3 düğümleri, oyuncularadaki ve spor alanındaki sensörleri belirtmektedir. t-0 ve t-1 düğümleri, teknik ekip ve kulüp doktorları gibi sistem aktörlerini tanımlamaktadır.

3.5.2. Simülasyonda Kullanılan Sis Düğümlerinin Özellikleri

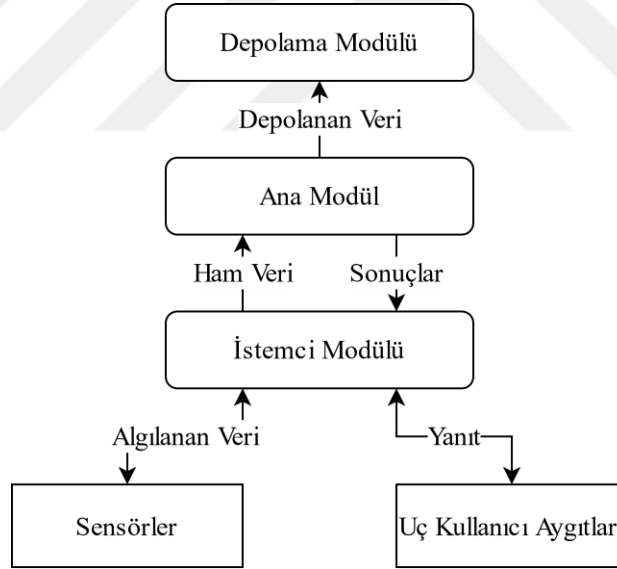
Simülasyonda işletim sistemi Linux, mimarisi x86 ve sanal makine yöneticisi olarak ise Xen kullanan sis düğümleri oluşturulmaktadır. Bunların yanı sıra yeni bir sis düğümü oluşturmak için kullanılan metodun tanımı “private static FogDevice createFogDevice(String nodeName, long mips, int ram, long upBw, long downBw, int level, double ratePerMips, double busyPower, double idlePower)” şeklindedir. Bu metod ile tanımlanan sis düğümlerinin parametre değerleri Tablo 3.2’de gösterilmektedir. Algoritmanın simülasyon karşılaştırmasında (Mahmud ve Buyya, 2019) çalışması kullanıldığı için önerilen mimari simülasyonunda da aynı topoloji ve parametreler kullanılmaktadır.

Tablo 3.2. Simülasyonda kullanılan sis düğümlerinin özellikleri

Parametreler	1.seviye sis düğümleri	2.seviye sis düğümleri
Mips (Saniye başına milyon işlem)	2800 MIPS	3200 MIPS
Ram (Ana bellek)	4000 MB	1000 MB
upBw (Yukarı bant genişliği)	10000 KBPS	10000 KBPS
downBw (Aşağı bant genişliği)	10000 KBPS	270 KBPS
Level (Seviye)	1	2
ratePerMips (Mips oranı)	0	0
busyPower (Meşgul durum gücü)	107.339 MJ	87.53 MJ
idlePower (Boş andaki güç)	83.4333 MJ	82.44 MJ

3.5.3. Simülasyonun Uygulama Yapısı

Uygulamada “clientModule”, “mainModule”, “storageModule” olmak üzere üç modül kullanılmaktadır. “clientModule” uç sis cihazlarına, “storageModule” bulut cihazına yerleştirildiği varsayılmaktadır. “mainModule” ise başlama modülüdür. Modüller arasında veriler demet veri yapısı şeklinde iletilmektedir. Simülasyonda modüller arasındaki demetlerin iletim yönü Şekil 3.13’te gösterilmektedir.



Şekil 3.13. Modüller arasındaki demet iletim yönü

3.5.4. Simülasyon Uygulamasının Kod Yapısı

iFogSim çatısı altında simülasyon uygulamasının ana bloğunun bulunduğu “TestApplication.java” dosyasında sis düğüm sayısını “numOfGateways” ve “numOfEndDevPerGateway” değişkenleri belirlemektedir. Şekil.3.12’deki topolojide “numOfGateways” değeri 1, “numOfEndDevPerGateway” değeri 2 olarak ayarlanmaktadır. Bu değerler önerilen algoritmanın tüm durumlarını gösterebilmek

için düşük tutulmaktadır. Simülasyon uygulamasının ana bloğunda modül yerleştirme için önerilen algoritmanın uygulandığı “MyModulePlacement.java” sınıfının nesnesi çağrılmaktadır. Sensör cihazla ile düzenli bir veri akışı sağlanamadığı için simülasyon ortamında istekler (mips ve süre değerleri) rastgele belirlenmektedir. Bu istekler önceliklerine göre sıralandığı varsayılarak tek tek sis düğümlerine önerilen algoritmaya göre atanmaktadır. Her işlemten sonra sis düğümlerinin tutulduğu liste sis düğümlerinin boştaki kaynak durumuna göre artan sırada tutulmaktadır. Bir döngüde eğer sis düğümlerinden biri isteği tek başına karşılayabilir ise istek o sis düğümlerine yerleştirilmekte ve döngü kırılmaktadır. İsteğin tek bir sis düğümlerine atanabildiği durumda yürütülen modül yerleştirme işlemi Şekil 3.14’teki koda göre yapılmaktadır.

```

166     if (requestMipsInfo.get(i)<=currentMips) {
167         currentMips = currentMips - requestMipsInfo.get(i);
168         sortedChildDeadline.put(key, currentMips);
169         System.out.println("The request "+requestMipsInfo.get(i)+" is assigned to node with ID "+
170             key+", for "+requestDeadlineInfo.get(i)+" units of time.");
171
172         sortedChildDeadline = sortedChildDeadline.entrySet()
173             .stream()
174             .sorted(Map.Entry.comparingByValue())
175             .collect(Collectors.toMap(
176                 Map.Entry::getKey,
177                 Map.Entry::getValue,
178                 (oldValue, newValue) -> oldValue, LinkedHashMap::new));
179
180         System.out.println("Remaining sorted fog device capacity: "+sortedChildDeadline+"\n");
181
182         if(!getDeviceToModuleMap().containsKey(key)){
183             List<AppModule>placedModules = new ArrayList<AppModule>();
184             placedModules.add(appModule);
185             getDeviceToModuleMap().put(key, placedModules);
186         }
187         else{
188             List<AppModule>placedModules = getDeviceToModuleMap().get(key);
189             placedModules.add(appModule);
190             getDeviceToModuleMap().put(key, placedModules);
191         }
192
193         finishedRequestMipsInfo.add(requestMipsInfo.get(i));
194         finishedRequestDeadlineInfo.add(requestDeadlineInfo.get(i));
195         requestMipsInfo.remove(i);
196         requestDeadlineInfo.remove(i); i=0;
197         break;

```

Şekil 3.14. İsteğin tek bir sis düğümlerine atanabildiği durumda modül yerleşimi

İstek tek bir sis düğümlerine atanamaz ise, sis düğümlerinin listesi geriye doğru taranmakta ve düğümlerin boştaki kapasiteleri toplanmaktadır. Uygun kapasiteye ulaşırsa, istek belirlenen sis düğümlerine dağıtılmaktadır. Bu durumda modül yerleştirme işlemi Şekil 3.15’teki koda göre gerçekleştirilmektedir.

```

220 for(int k=sortedValues.size()-2; k>=0;k--){
221     AppModule appModule = getApplication().getModuleByName(moduleToPlace);
222     sum = sum + sortedValues.get(k);
223     usedKeys.add(sortedKeys.get(k));
224     usedValues.add(sortedValues.get(k));
225     if (request<=sum) {
226         int remain = request;
227         for (int l=0; l<usedKeys.size();l++) {
228             System.out.println("Part of the remaining request ("+"remain+") is assigned to the node with ID "+
229                 usedKeys.get(l)+" for "+requestDeadlineInfo.get(l)+" units of time.");
230             if(remain-usedValues.get(l)>=0) {
231                 remain=remain-usedValues.get(l);
232                 usedValues.set(l,0);
233             } else {
234                 int x = usedValues.get(l)-remain;
235                 usedValues.set(l,x);
236             }
237             sortedChildDeadline.put(usedKeys.get(l), usedValues.get(l));
238             System.out.println("Remaining sorted fog device capacity: "+sortedChildDeadline+"\n");
239         }
240     }

```

Şekil 3.15. İsteğin birden fazla sis düğümüne atanabildiği durumda modül yerleşimi

Eğer sis düğümlerinden hiçbiri tek başına veya dağıtık bir şekilde isteği karşılayamaz ise sis katmanında yeterli alan olmadığı için istek iptal edilmektedir. Bu durumda yürütülen işlem Şekil 3.16'daki koda göre yapılmaktadır.

```

274     if(request>sum) {
275         AppModule appModule = getApplication().getModuleByName(moduleToPlace);
276         List<AppModule>placedModules = getDeviceToModuleMap().get(deviceParent);
277         placedModules.add(appModule);
278         getDeviceToModuleMap().put(deviceParent, placedModules);
279         System.out.println(" The request "+request+"is not assigned to any fog node.");
280         System.out.println("getDeviceToModuleMap"+getDeviceToModuleMap());
281     }
282 }
283

```

Şekil 3.16. İsteğin hiçbir sis düğümüne atanamadığı durumda yürütülen işlem

3.5.5. Simülasyon Uygulamasının Sonuçları

Rastgele oluşturulan [1079, 749, 1192, 1047, 596, 944] (mips) istek listesine göre simülasyon sonuçları Şekil 3.17'de gösterilmektedir. Aynı zamanda, her istek için rastgele oluşturulmuş bir son teslim tarihi belirlenmektedir.

Şekil 3.17'de sis düğüm sayılarını belirleyen “numofGateways” ve “numOfEndDevPerGateway” değişkenlerinin değerleri sırasıyla 1 ve 2 olarak atanmaktadır. “Tuple CPU execution delay” başlığında, modüller arasında iletilen demet tipindeki verilerin türleri ve gecikme süreleri belirtilmektedir. Şekil 3.18'deki grafiklerde sis düğümlerinin sayısı için “numOfEndDevPerGateway” değeri artırılarak program yeniden çalıştırılmaktadır.

```

Problems @ Javadoc Search Console
<terminated> TestApplication (1) [Java Application] C:\Program Files\Java\jre1.8.0_181\bin\javaw.exe (29.Eki.2021 16:3
Starting Health and Tactical Analysis System...
Modulscld and storageModule
Modulse-0-1 and clientModule
Modulse-0-0 and clientModule
Children: [5, 8]
Children deadline: {5=3200, 8=3200}
Requests: [1079, 749, 1192, 1047, 596, 944]

Sorted fog device capacity{5=3200, 8=3200}

The request 1079, is assigned to node with ID 5, for 5.725144892420063 units of time.
Remaining sorted fog device capacity: {5=2121, 8=3200}

The request 1192, is assigned to node with ID 5, for 3.3767522100000544 units of time.
Remaining sorted fog device capacity: {5=929, 8=3200}

The request 1047, is assigned to node with ID 8, for 5.67524669905512 units of time.
Remaining sorted fog device capacity: {5=929, 8=2153}

The request 596, is assigned to node with ID 5, for 2.1159033401981793 units of time.
Remaining sorted fog device capacity: {5=333, 8=2153}

The request 944, is assigned to node with ID 8, for 2.142471036154175 units of time.
Remaining sorted fog device capacity: {5=333, 8=1209}

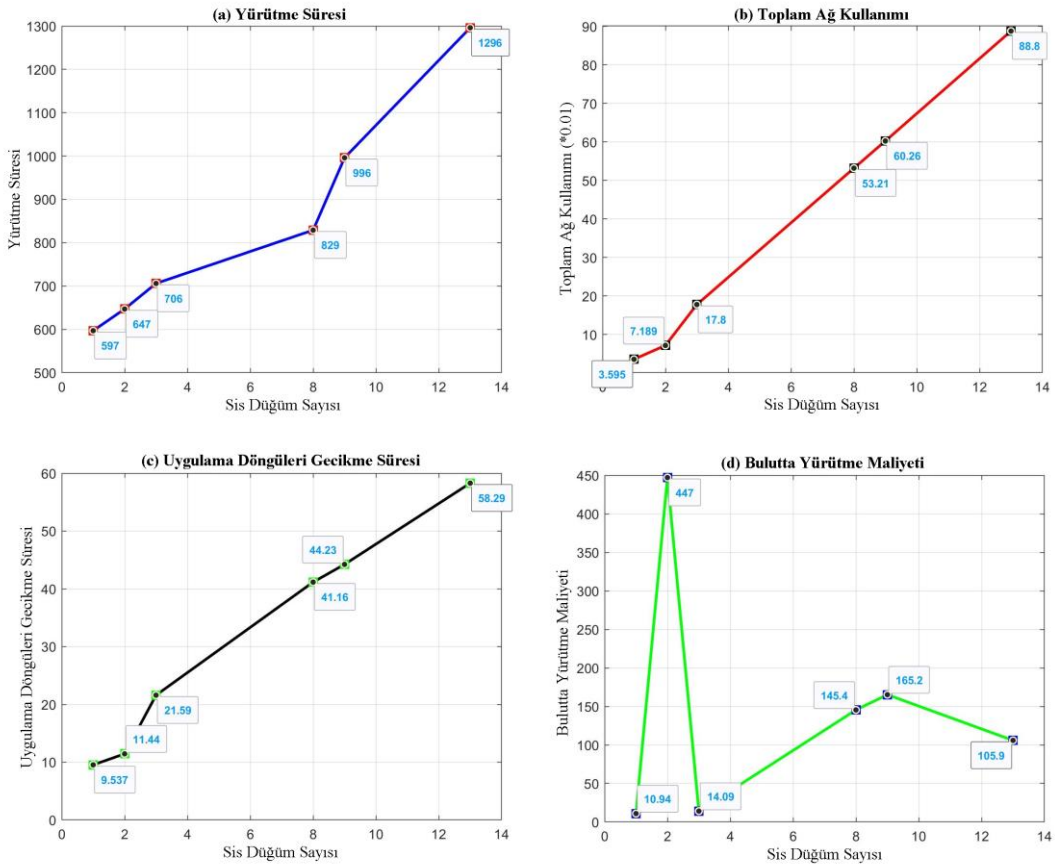
=====
===== RESULTS =====
=====
EXECUTION TIME : 142
=====
APPLICATION LOOP DELAYS
=====
[IoTSensor, clientModule, mainModule, clientModule, IoTActuator] ---> 11.443749999999975
=====
TUPLE CPU EXECUTION DELAY
=====
RawData ---> 3.8500000000000227
ResultData ---> 0.16250000000002274
IoTSensor ---> 0.13125000000002274
StoreData ---> 0.07154761904761622
=====
cloud : Energy Consumed = 401772.0913337049
g-0 : Energy Consumed = 25029.989999999998
e-0-0 : Energy Consumed = 25971.57406249996
e-0-1 : Energy Consumed = 26192.830000000002
Cost of execution in cloud = 3079.4206250000343
Total network usage = 684.0

```

Şekil 3.17. Önerilen algoritmanın simülasyon sonuçları

Sis platformları ile ilgili gelişmiş simülatör eksikliği göz önüne alındığında iFogSim’de sensörler üzerinden veri alınamadığından her durumu görebilmek için isteklerin düğümlere geliş süresinin sis düğümlerinde işlenme süresinden çok daha kısa olduğu varsayılmaktadır. Bu nedenle düğüm sayıları da istek listesinin uzunluğundan daha az (“numOfEndDevPerGateway” = 2) olarak seçilmektedir. Daha fazla sis düğümü ile oluşan sonuçlar için ana modüldeki “numOfGateways” ve “numOfEndDevPerGateway” değerleri değiştirilerek yeniden çalıştırılabilir. Şekil 3.17’de iFogSim simülatörü tarafından standart olarak üretilen “RESULTS” bölümü sistemin çalışması için harcanan zamanı, enerjiyi ve genel maliyeti göstermektedir.

Sis düğüm sayısı değiştirilerek elde edilen simülasyon sonuçlarından Yürütme süresi (a), toplam ağ kullanımı (b), uygulama döngüleri gecikme süresi (c), bulutta yürütme maliyeti (d) değerleri için grafikler Şekil 3.18'de gösterilmektedir. Düğüm sayısı arttıkça, yürütme süresi ve uygulama döngü gecikmesi artmaktadır. Düğüm sayısı 4 olduğunda, “RawData” demet verisinde yüksek gecikme meydana gelmektedir. Bu nedenle, yürütme süresi ve uygulama döngü gecikmesi çok daha yüksektir. Toplam ağ kullanımı düğüm sayısı ile artarken, 4 düğüm olduğunda azalmaktadır. “RawData” demet verisinin yüksek gecikmesi ağ aktivitelerini etkilememektedir. Ek olarak, toplam ağ kullanımı istek maliyetlerine göre değişebilmektedir. Bulut kullanım maliyeti ise, düğüm sayısı 2 olduğunda en yüksek seviyededir. Bu durum rastgele belirlenen isteklerin maliyet değerlerine göre değişiklik göstermektedir.



Şekil 3.18. Önerilen algoritmanın simülasyon sonuçlarının grafikleri ((a) Yürütme süresi, (b) Toplam ağ kullanımı, (c) Uygulama döngüleri gecikme süresi, (d) Bulutta yürütme maliyeti)

3.5.6. Simülasyon Sonuçlarının Karşılaştırılması

Önerilen mimari ve algoritma simülasyonu, simülasyon kodlarını açıkça paylaşan (Mahmud ve Buyya, 2019) ile karşılaştırılmaktadır. Her iki çalışma da “numOfGateways” ve “numOfEndDevPerGateway” parametrelerinin değerleri sırayla 1 ve 2 olarak ayarlanmaktadır.

Tablo 3.3. Sis düğüm sayısı 1 iken simülasyonların karşılaştırılması

Özellikler	(Mahmud ve Buyya, 2019)	Önerilen Mimari ve Algoritma
Yürütme süresi	278	79
Toplam ağ kullanımı	479,0	344,0
Uygulama döngü gecikme süresi	13,870357142857134	9,5375
Bulutta yürütme maliyeti	479,0	3133,283125000039
Tuple yürütme gecikmesi (RawData)	2,242857142857133	1,975000000000227
Tuple yürütme gecikmesi (ResultData)	0,1312500000002268	0,1312500000002268
Tuple yürütme gecikmesi (IoTSensor)	0,1312500000002268	0,1312500000002268
Tuple yürütme gecikmesi (StoreData)	0,12232142857141076	0,12232142857141076

Tablo 3.4. Sis düğüm sayısı 2 iken simülasyonların karşılaştırılması

Özellikler	(Mahmud ve Buyya, 2019)	Önerilen Mimari ve Algoritma
Yürütme süresi	125	104
Toplam ağ kullanımı	809,3333333333334	684,0
Uygulama döngü gecikme süresi	17,71321428571426	11,44374999999975
Bulutta yürütme maliyeti	3407,7379464284923	3079,4206250000343
Tuple yürütme gecikmesi (RawData)	4,2317142857143	3,850000000000227
Tuple yürütme gecikmesi (ResultData)	0,1907500000002545	0,1625000000002274
Tuple yürütme gecikmesi (IoTSensor)	0,1312500000002274	0,1312500000002274
Tuple yürütme gecikmesi (StoreData)	0,12704761904762257	0,07154761904761622

Tablo 3.5. Sis düğüm sayısı 3 simülasyonların karşılaştırılması

Özellikler	(Mahmud ve Buyya, 2019)	Önerilen Mimari ve Algoritma
Yürütme süresi	167	109
Toplam ağ kullanımı	1111,0	1710,0
Uygulama döngü gecikme süresi	25,747454545454435	22,047697368421073
Bulutta yürütme maliyeti	2498,1106249999866	7417,84000000009
Tuple yürütme gecikmesi (RawData)	4,746627594627586	14,799983310707375
Tuple yürütme gecikmesi (ResultData)	0,8733928519536377	0,09375
Tuple yürütme gecikmesi (IoTSensor)	0,1312500000002274	0,11155821917810493
Tuple yürütme gecikmesi (StoreData)	0,3004395604395589	0,048469387755110976

Tablo 3.6. Sis düğüm sayısı 8 iken simülasyonların karşılaştırılması

Özellikler	(Mahmud ve Buyya, 2019)	Önerilen Mimari ve Algoritma
Yürütme süresi	13013	156
Toplam ağ kullanımı	2522,6666666666665	5088,0
Uygulama döngü gecikme süresi	91,423197544643	41,59062500000002
Bulutta yürütme maliyeti	64840,653125	42967,400000000074
Tuple yürütme gecikmesi (RawData)	48,853586865236174	34,200000000000315
Tuple yürütme gecikmesi (ResultData)	1,6688360533211914	0,600000000000218
Tuple yürütme gecikmesi (IoTSensor)	0,2562500000002274	0,351361846923851
Tuple yürütme gecikmesi (StoreData)	0,4156389508928515	0,06875420168064506

Tablo 3.7. Sis düğüm sayısı 13 iken simülasyonların karşılaştırılması

Özellikler	(Mahmud ve Buyya, 2019)	Önerilen Mimari ve Algoritma
Yürütme süresi	1152	180
Toplam ağ kullanımı	2410,0	8502,0
Uygulama döngü gecikme süresi	234,71987499999986	60,11250000000007
Bulutta yürütme maliyeti	17908,864374999986	44802,940000000046
Tuple yürütme gecikmesi (RawData)	268,54760881423954	48,888110277065806
Tuple yürütme gecikmesi (ResultData)	4,436298875855698	0,912500000000227
Tuple yürütme gecikmesi (IoTSensor)	2,6289916992187727	0,506250000000227
Tuple yürütme gecikmesi (StoreData)	0,501258979524859	4,896754800565221

Her iki yapıda da Tablo 3.2'deki özelliklere sahip sis düğümleri kullanılmaktadır. Karşılaştırma sonuçları Tablo 3.3, Tablo 3.4, Tablo 3.5, Tablo 3.6 ve Tablo 3.7'de gösterilmektedir. Bu sonuçlara göre, önerilen algorithmada modüller arasındaki veri iletim gecikme süreleri ve ağ kullanımı daha düşüktür. Bulut düğümü maliyetleri, sis düğümü maliyetlerinden daha yüksektir. Bir isteğin uzaktaki bulut sunucusu ile iletişim kurabilmesi için İnternet bağlantısı gerekmekte iken sis düğümleri ile yerel ağda iletişim kurabilmektedir. Bu, sis bilişiminin bulut bilişimden daha düşük gecikmeye neden olabildiğini göstermektedir. Bu nedenle önerilen mimaride sis bilişim sistemleri kullanılmaktadır. Karşılaştırma sonuçlarında da bu durum görülebilmektedir. Önerilen algorithmada simülasyondaki her istek sis katmanı tarafından karşılanabilirse (Şekil 3.16'daki durum oluşmazsa), önerilen modelin gecikme süreleri ve bulut maliyetleri çok daha düşük çıkmaktadır. Ancak, bu durum isteklerin boyutuna ve sis düğümlerinin kapasitesine de bağlı olarak değişmektedir.

3.6. Giyilebilir Teknolojilerdeki Zorluklar

Bu bölüm, giyilebilir teknolojilerdeki zorlukları ve gelecekteki bazı çalışma önerilerini içermektedir. Günümüzde giyilebilir cihaz teknolojisi, sis bilişim ile birleştirildiğinde büyük verilerin hızlı işlenmesi ve iş kararlarına daha hızlı ulaşma ya da düşük gecikme isteyen uygulamalara çözüm olabilecek seviyeye gelebilmektedir (Fiandrino, vd., 2019). Ancak, giyilebilir teknolojilerle ilgili çok fazla zorluk bulunmaktadır. Bu yüzden önerilen modelimizde giyilebilir teknolojiler çalışma kapsamı dışında tutulmaktadır. Giyilebilir teknoloji ile ilgili bazı zorluklar şu şekilde listelenebilmektedir.

- Her aygıt her kullanıcının vücuduna uygun olmayabilir. Bu nedenle yaş gruplarına göre en uygun ve ergonomik ürün geliştirilmeye çalışılmaktadır (Hänsel, vd., 2015).
- Yeni sağlık aygıtı tasarımları için psikologlar ve mühendisler arasında yapılan disiplinler arası çalışma ile ürün hem teknolojik hem de davranış değişikliği yönünden ele alınmaktadır. Giyilebilir sağlık cihazlarının tıbbi bilgi olmadan kendi kendine teşhis yapması kullanıcıyı yanlış yönlendirebileceği için bu durumun da geliştirilen giyilebilir ürünlerde ele alınması gerekmektedir. Giyilebilir uygulamaların geleceği, sadece sağlık verileri için yeni analiz algoritmalarının geliştirilmesinde değil, aynı zamanda kullanıcının verileri anlamlandırması ve davranış değişikliği için geri bildirim almasını desteklenmesinde yatmaktadır (Hänsel, vd., 2015).
- Sağlık algılama konusunda kimyasal sensörler kullanılmaktadır. Mevcut kimyasal sensörlerin boyutu, sertliği ve çalışma gereksinimleri, giyilebilir teknolojiyle uyumlu olmadığı için uygulamalarda kullanılması zordur (Bandodkar, vd., 2016). Mevcut giyilebilir cihazların güç kaynakları, düşük enerji yoğunlukları ve yavaş şarj olmaları nedeniyle gereklilikleri karşılayamamaktadır. Ayrıca, giyilebilir sensörlerin ürettiği büyük verilerin işlenmesi ve güvenliğinin sağlanmasıyla ilgili de büyük zorluklar vardır. Veri güvenliği ve kullanıcı gizliliğini sağlamak için yeni nesil kriptolojik algoritmaların geliştirilmesi gerekmektedir (Bandodkar, vd., 2016).

4. MAKİNE ÖĞRENMESİ TABANLI TEKNİK EKİP UYARI SİSTEMİ

Bu bölümde bir futbol maçındaki taktik diziliş ve başka çevresel veriler yardımı ile gol oluşumu hakkında teknik ekibi uyaran yöntemin detayları yer almaktadır. Bu yöntemde önerilen IoT mimarisi üzerinde çalışan makine öğrenmesi tabanlı bir taktik analiz modeli önerilmektedir. Bu model, bir makale haline getirilerek uluslararası bir dergi de yayınlanmak üzere gönderilmiş olup hakem değerlendirme sürecindedir.

Sporculardan alınan veriler ile makine öğrenmesi tabanlı taktik analiz modelinin sonuçlarından üretilen geri bildirim teknik ekibe iletilebilmesi için modelin bir IoT sistemi üzerinde çalışması gerekmektedir. Bu nedenle, bu modelin önerilen IoT ve sis bilişim tabanlı sağlık ve taktik analiz mimarisinde çalıştığı ve verilerin sis katmanındaki veri merkezlerinde işlendiği varsayılmaktadır.

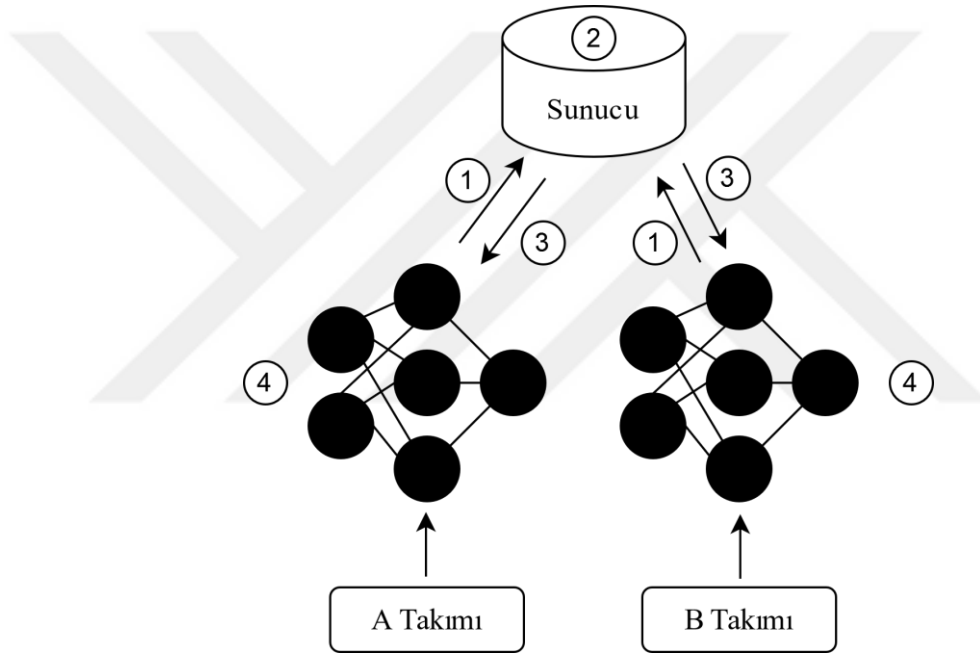
Futbolculara ve topa yerleştirilen uygun sensörler (hassas konum gibi) ile veriler elde edilerek ve saha parçalara bölünerek bir takımın formasyonu ortaya çıkarılabilmektedir. Önerilen yöntemin ana katkıları şu şekildedir:

- Literatürde oldukça az bulunan maç esnasında anlık formasyonların çıkarılmasına olanak sağlamaktadır.
- Gol atma, gol yeme gibi olaylardan önceki belirli zaman aralıklarında elde edilen sıralı formasyon bilgileri maç analizlerinde kullanılabilir.
- Önemli olaylardan önceki zaman dilimlerinde elde edilen formasyon verilerine çevresel veriler de eklenerek makine öğrenmesi yöntemleri ile gelecekteki maçlarda gol atma, gol yeme tahminleri yapılabilir.
- Teknik ekibe yardımcı bir taktik analiz uyarı yöntemi sağlanabilir.

Futbolculara ve topa yerleştirilen giyilebilir teknoloji sensörleri ile veriler toplanarak sis düğümlerine iletilmektedir. Veriler sis düğümlerinde işlendikten sonra sonuçlar teknik ekibe iletilmektedir. IoT tabanlı bu sistem maç esnasında formasyon oluşturulmasına ve bu sayede teknik ve taktik analizlere yardımcı olabilmektedir.

Her iki ekibin verileri, IoT destekli giyilebilir uç cihazlar aracılığıyla veri sporculardan elde edilmektedir. Bu veriler, sağlık ve taktik analiz için önemli bilgiler içermektedir. Bu nedenle, her takımın verileri farklı sis düğümlerine iletilmekte ve orada yerel olarak eğitilmektedir. Eğitilen verilerin parametre ağırlık değerleri, bulutta

veya yerelde farklı bir sunucuya iletilmekte ve yerel olarak eğitilmiş tüm modeller bu sunucuda birleştirilmektedir. Veriler tek bir merkezde toplanmadan birleştirilmiş (federated) öğrenme modeli ile dağıtık bir şekilde işlenmektedir. Böylece hem iş yükleri dengelenmekte hem de sistem güvenliği artırılmaktadır. Google tarafından önerilen birleştirilmiş öğrenme yöntemi, veri sızıntılarını önleyen bir makine öğrenme modeli oluşturmaktadır (Yang, vd., 2019). Birleştirilmiş öğrenme modeli Şekil 4.1'de gösterilmektedir. Veriler, sis düğümlerinde yerel olarak eğitildikten sonra, Adım 1'de model sonuçları sunucuya gönderilmektedir. Adım 2'de veriler birleştirilmektedir. Adım 3'te güncellenen model yerel sistemlere geri gönderilmekte ve Adım 4'te yerel modeller güncellenmektedir.

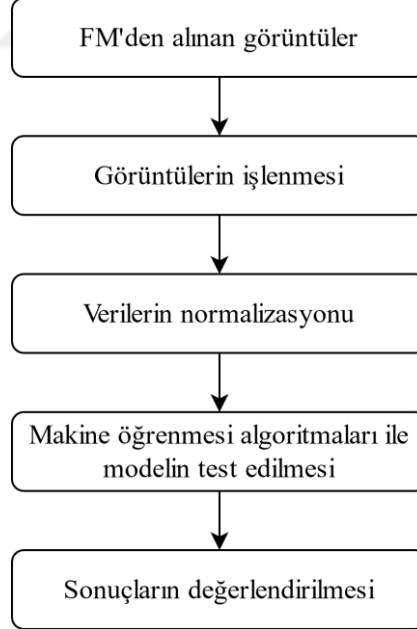


Şekil 4.1. Birleştirilmiş öğrenme modeli

Taktik analizlerde belirli zaman aralıklarında alınan anlık takımın diziliş ve topun konum verilerine ek olarak iki takımada açık olan takımların başlangıç formasyonları, önceki maçlardaki oyuncu derecelerinin ortalaması, maçın oynandığı stadyumun zemin durumu, pozisyonun zaman aralığı, maçın önemi, maçın yeri gibi bilgiler alınmaktadır. Bu bilgiler çeşitli makine öğrenmesi teknikleri kullanılarak sisteme öğretilmekte ve maç anında alınan yeni veri sınıflandırılarak teknik ekibe uyarı gönderilmektedir. Önerilen sis bilişim tabanlı güvenli IoT altyapısında bir takımın maç esnasında elde ettiği verilere diğer takımların erişimine izin verilmemektedir. Bir başka deyişle, iki takımdan maç anında toplanan bilgiler birbirinden izole edilmekte

ve ma esnasında gol atma ve gol yeme tahminlerinin yapılmasında rakip takımın gizli verileri kullanılamamaktadır. Bu nedenle veri kümesi yalnızca ilgili takımın ma anındaki bilgileri toplanarak oluşturulmaktadır. Ancak, her takımın verileri farklı sis düğümünde öğrenildikten sonra sonuç modelleri birleştirilmektedir. Rakip takımın sadece herkese açık olan (genel itibar derecesi, başlangı formasyonu gibi) verileri veri kümesine eklenmektedir. Veri kümesi normalize edildikten sonra birçok makine öğrenmesi yöntemiyle test edilmektedir. Parametreleri optimize edilmiş AdaBoost algoritması en iyi sonuçları vermektedir. Bu nedenle AdaBoost algoritması, önerilen sis bilişime dayalı güvenli IoT altyapısında ma sırasında taktiksel analiz için kullanılabilir.

Günümüzde giyilebilir teknolojilerin istenilen seviyelere gelememiş olmasından ve hiç bir takımda ma esnasında hali hazırda önerilen mimarideki gibi bir yapı kullanılmıyor olmasından dolayı veri edinme konusunda sorunlar yaşanmaktadır. Bu nedenle veriler gereğe çok yakın olan Football Manager (FM) 2020 oyunundan alınmaktadır (Hocquet, 2016) (Bin Mohd Rafeek ve McLean, 2015). Süreçlerin akış şeması Şekil 4.2'de gösterilmektedir.



Şekil 4.2. Teknik ekip uyarı sistemindeki süreçlerin akış şeması

4.1. Önerilen Modelin Uygulanması

Bu bölümde önerilen makine öğrenmesi modelinin test edilmesi için uygulanması ve detayları verilmektedir. Ma anında geri bildirim sağlayarak teknik

ekibe yardımcı olan bu sistemde veriler FM oyunu üzerinden alınmaktadır. FM'den elde edilen iki boyutlu maç görüntülerinde takımın anlık diziliş bilgileri ve topun pozisyonu görüntü işleme teknikleri ile sayısal verilere dönüştürülmektedir. Pozisyonun dakika aralığı ve diğer sabit veriler eklenerek veri kümesi oluşturulmaktadır. Çoğu modern öğrenme algoritması, sonraki olaylar hakkında bir tahmin yapmak için bu veri kümesi ile çalıştırılmaktadır. En iyi sonucu veren öğrenme yöntemi önerilen IoT tabanlı altyapıda kullanılmak üzere seçilebilmektedir.

4.1.1. Veri Toplama

Az gelişmiş giyilebilir teknolojiler nedeniyle, IoT tabanlı mimariyi fiziksel ortamda oluşturmak ve veri elde etmek oldukça zordur. Bu nedenle, veriler FM oyunundan alınmaktadır. FM, oyuncunun sadece menajerlik yaptığı, maçları kendisinin oynatmadığı bir oyun tarzıdır. Maçlar FM oyun motorları tarafından otomatik bir şekilde oynatılmaktadır. Veri toplanırken FM'deki maç esnasında belirli aralıklarla ekran görüntüleri alınmaktadır. 1 saniyelik aralıklarla alınan 50 görüntü oyundaki maç süresi olarak 5'er dakikalık dilimlere denk gelmektedir. Bu nedenle peş peşe alınan 50 görüntünün formasyonları oluşturulmaktadır. Görüntülerden formasyon çıkarma işlemi tamamen görüntü işleme teknikleri kullanılarak sayısal verilere dönüştürülmektedir. d-min-maks normalizasyon yöntemi ile veriler normalize edildikten sonra çok sayıda makine öğrenmesi algoritması ile sınıflandırılarak sonuçlar karşılaştırılmaktadır.

FM oyunun da Türkiye Süper Lig takımlarından biri seçilerek istenilen sayıda veriye ulaşıncaya kadar 470 maç oynatılmıştır. Tüm maçlar seçilen takımın 2020 yılındaki standart formasyonu olan 4-2-3-1 formasyonu ile başlatılmıştır. Veri kümesinde 663'er tane gol atma ve gol yeme pozisyonları ile 670 tane golsüz geçilen pozisyonlar olmak üzere toplam 1996 adet veri bulunmaktadır. Veri kümesinde, her bir pozisyon için alınan 50 görüntüden elde edilen takımın anlık formasyon verileri ve topun anlık konum verileri ile birlikte her iki takımın maç başındaki formasyon verileri ve önceki maçlardaki maçta oynayan oyuncuların aldıkları puanların ortalamaları, her iki takımın genel itibar derecesi (1-5 yıldız arasında buçuklu değerler de dahil toplam 10 skaladan oluşan değerler), maçın yeri (ev sahibi, deplasman), maçın önemi (resmi maç, hazırlık maçı), zemin durumu (normal zemin, karlı zemin), pozisyonun dakikası (her pozisyon 5 dakikalık dilimi kapsadığı için maçın 90 dakikası 5 dakikalık dilimlere bölünerek numaralandırılmaktadır.) olmak üzere toplam 110 girdi yer almaktadır. Tek

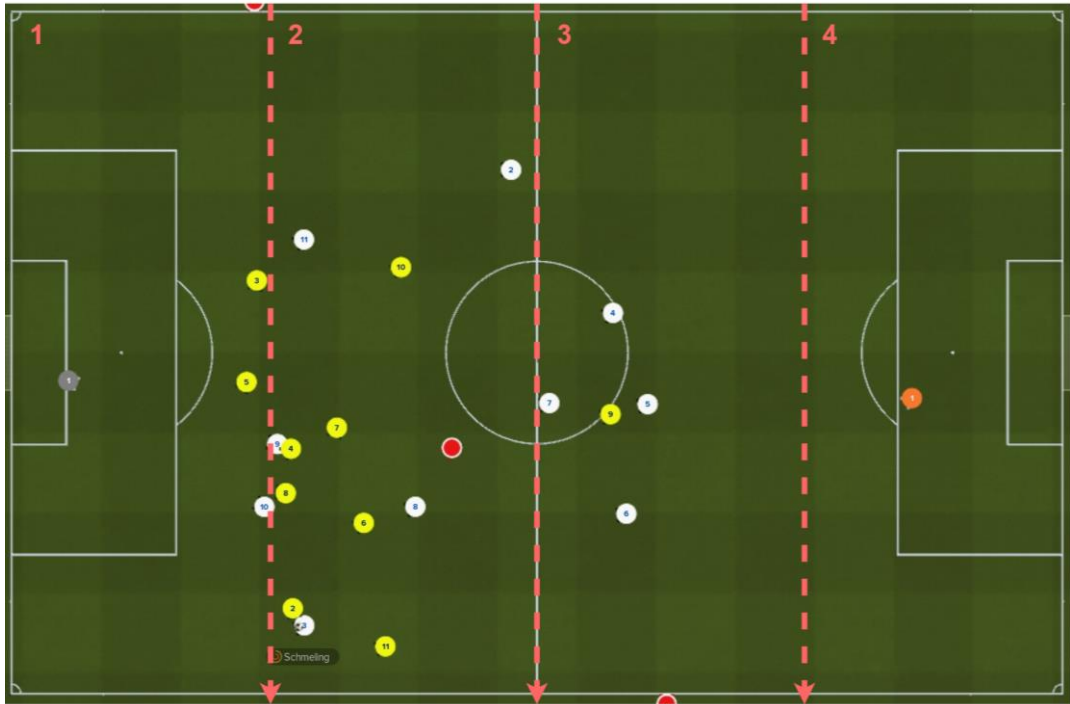
bir çıktı ise gol atma (scoring a goal - SG), gol yeme (conceding a goal - CG) ve golsüz geçilme (no goal - NG) pozisyonları olmak üzere üç sınıftan oluşmaktadır.

4.1.2. Veri Ön İşleme

Modern futbolda bilişim sistemleri çokça kullanılmasına rağmen giyilebilir teknolojilerin istenilen seviyelere gelememesinden dolayı kulüpler tarafından futbolculara sensörler yerleştirilerek maç anında veri toplanması işlemi yapılmamaktadır. Bu nedenle veri toplarken görüntü işleme tekniklerinden yararlanılmaktadır. Tam açılı gerçek maç görüntülerine ulaşmanın güçlüğünden dolayı önerilen modelde gerçeğe en yakın futbol simülasyon oyun olan FM'den görüntüler elde edilmektedir.

4.1.2.1. Görüntü İşleme

FM'deki görüntülerden formasyonlar ile topun konumları görüntü işleme teknikleri kullanılarak sayısal veri haline dönüştürülmektedir. Şekil 4.3'te sahanın formasyonu oluşturabilmek için bölgelere ayrılması gösterilmektedir. Saha 4 parçaya bölünmekte ve takımın kaleleri solda olduğu için soldan itibaren numaralandırılmaktadır. Kaleler değiştiğinde ise tam tersi numaralama yapılmaktadır. Takımın kalecisinin olduğu bölge her zaman 1 numaralı bölgedir.



Şekil 4.3. FM'den alınan görüntüde sahanın bölgelere ayrılması (temsili çizgiler ile)

FM görüntüsündeki formasyonlar ve topun konumları Hough dönüşümü (Hough transform - HT) yöntemi kullanılarak tespit edilmekte ve sayısal verilere dönüştürülmektedir. HT, Paul Hough tarafından dijital görüntülerdeki geometrik özellikleri tespit etmek için önerilmiştir (Hough, 1962). Bu teknik, şekil sınırındaki boşluklara ve görüntü gürültüsüne toleranslıdır. HT, ilk olarak sadece çizgi tespiti için tasarlanmış, daha sonra dairesel şekilleri tanımlamak için genişletilmiş ve dairesel Hough dönüşümü (Circular HT - CHT) olarak adlandırılmıştır (Pedersen, 2007).

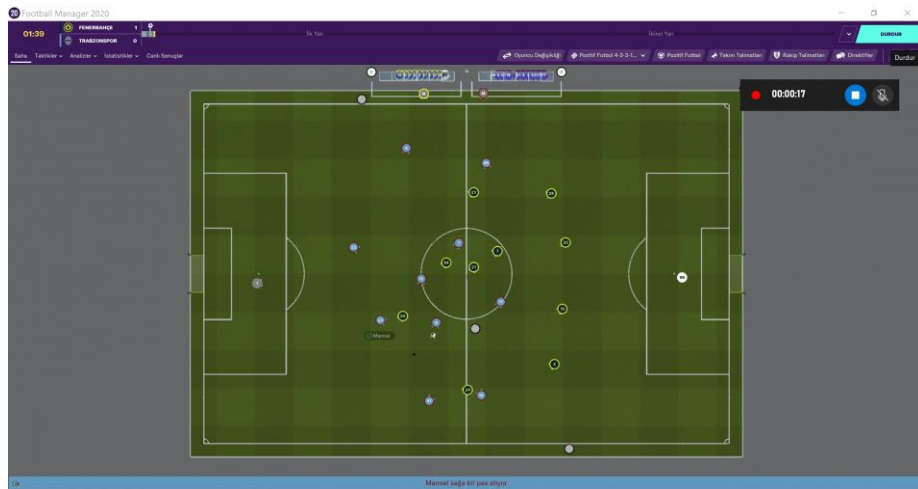
2 boyutlu uzayda, merkez noktası ve yarıçapı sırasıyla (a, b) ve r olan bir daire Eşitlik 4.1 kullanılarak temsil edilebilmektedir.

$$(x - a)^2 + (y - b)^2 = r^2 \quad (4.1)$$

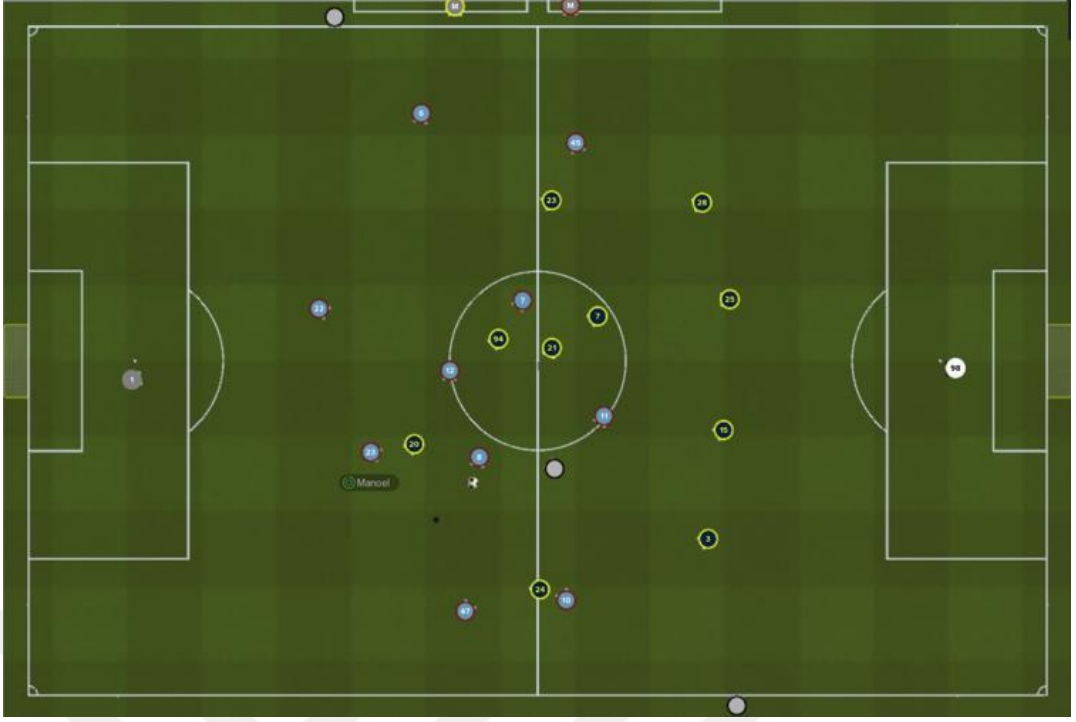
CHT, dairenin tanımladığı geometrik koordinatları (x, y) alır ve bunları Hough parametre uzayına dönüştürür. Buna göre bir daire Eşitlik 4.2 ile tanımlanmaktadır.

$$\begin{aligned} x &= a + r \times \cos(\theta) \\ y &= b + r \times \sin(\theta) \end{aligned} \quad (4.2)$$

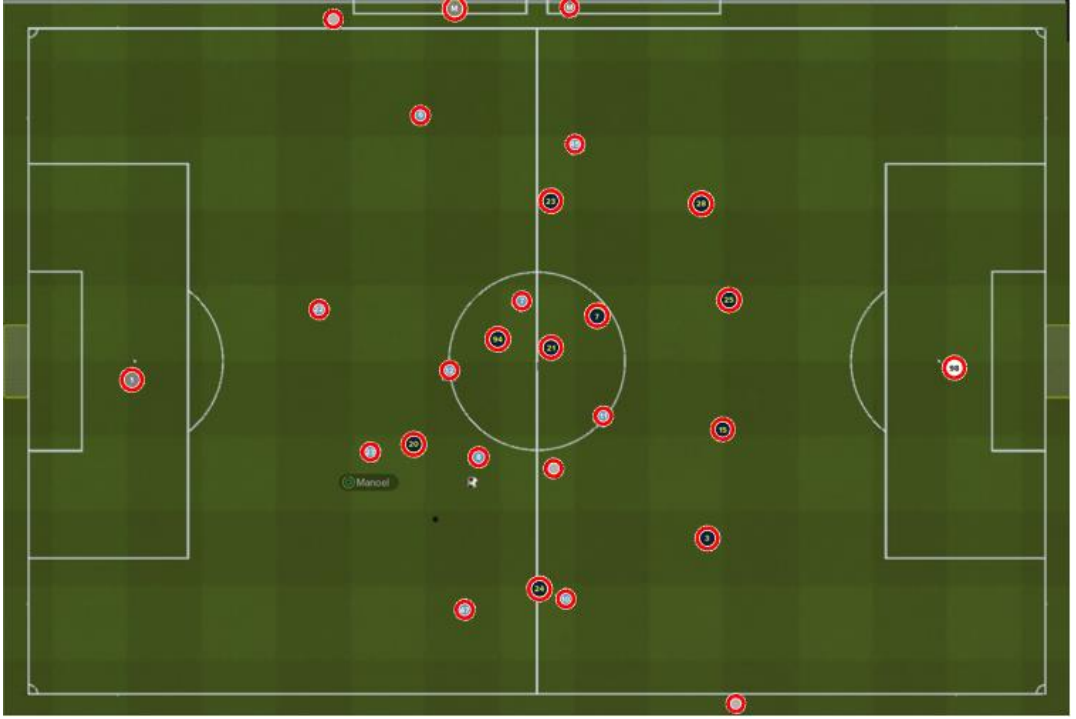
θ , x ekseninden yukarı doğru olan açıdır. CHT algoritması (x, y) noktalarını tanımlayan (a, b, r) parametrelerini aramaktadır. Hough parametre uzayında, geometrik koordinatlardaki çember üzerinde her nokta için bir nokta oylanmaktadır. Hough uzayındaki oylara bağlı olarak birikimli bir matris oluşturulmaktadır. En çok oy alan noktalara göre geometrik koordinatlardaki daire adayları seçilmektedir. CHT, özellikle gürültü toleransı nedeniyle görüntü işleme ve yapay görme uygulamalarında kullanılan bir daire algılama algoritmasıdır (Rizon, vd., 2005) (Cherabit, vd., 2012). Bu nedenle futbolcuların çemberinin tespiti için CHT yöntemi kullanılmaktadır.



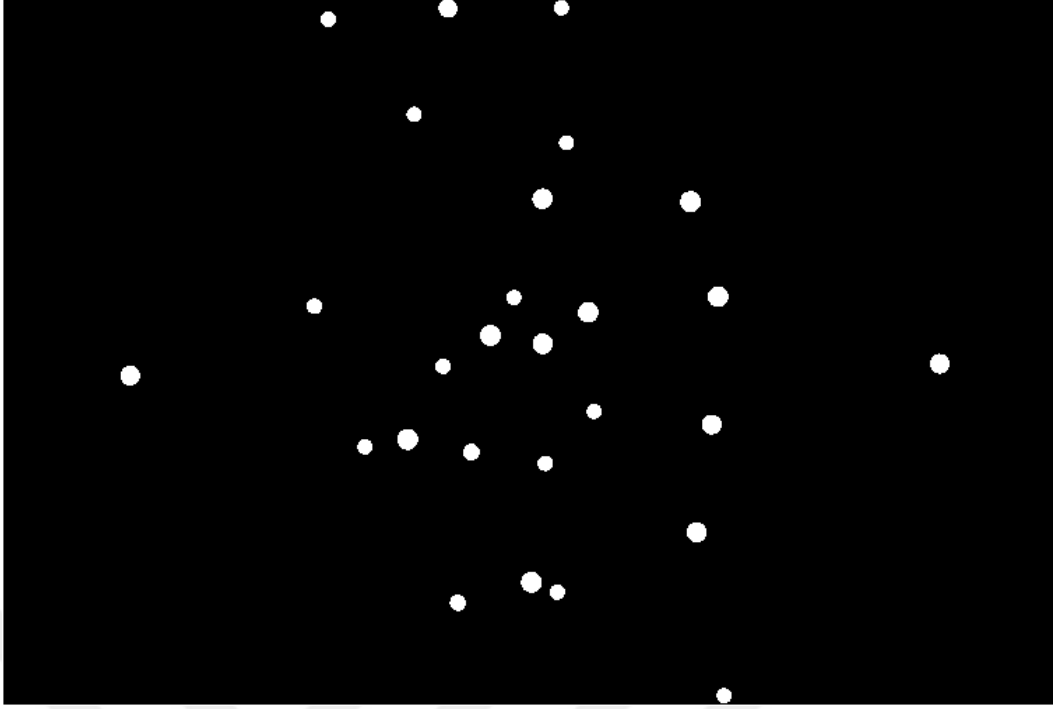
Şekil 4.4. FM'den alınan örnek görüntü



Şekil 4.5. Görüntünün kırılması



Şekil 4.6. CHT uygulanmış görüntü



Şekil 4.7. Maskelenmiş görüntü

İlk olarak, Şekil 4.4 ve Şekil 4.5'te gösterildiği gibi yedek kulübesinde oturan oyuncularını çıkarmak için görüntü üzerinde görüntü kırpma işlemi yapılmaktadır. Daha sonra Şekil 4.6'daki gibi CHT yöntemi kullanılarak olası çemberlerin merkez noktaları ve yarıçap bilgileri bulunmaktadır. Oyuncuların hangi takıma ait olduğuna karar verebilmek için Şekil 4.7'deki gibi merkez noktalarda ilgili yarıçapta bir maske daire oluşturularak daire içindeki renk bilgisi elde edilmektedir. HSV, renk bilgisi almak için daha uygun ve ışık değişimlerine daha dayanıklı olduğu için dairelerin renk bilgisi RGB renk uzayından HSV renk uzayına dönüştürülmektedir. İki olası daire için, yakınlık bilgisi, HSV uzayındaki ortalama renk tonu bilgisine bağlı olarak Eşitlik 4.3 ve Eşitlik 4.4'te belirtildiği gibi L1 norm mesafe yöntemi kullanılarak ölçülmektedir.

$$f(x) = \left| \frac{1}{m} \sum_{i=1}^m h_i - \frac{1}{n} \sum_{j=1}^n h_j \right| \quad (4.3)$$

$$g(x) = \begin{cases} \text{aynı takım} & , f(x) < t \\ \text{farklı takım} & , f(x) \geq t \end{cases} \quad (4.4)$$

Burada; h_i ve h_j farklı çemberlere ait renk tonu değeri, m ve n toplam renk tonu değerlerinin sayıları, t deneysel ve önceden tanımlanmış bir eşik değeridir. Birbirine yeterince yakın olan çemberler aynı takıma dâhil edilirken birbirine yakın olmayan çemberler (hakem, kaleci vb.) takım formasyon listesinden çıkarılmaktadır.

4.1.2.2. Veri Kümesinin Normalizasyonu

Veri kümesindeki her pozisyon için FM'den alınan 50 görüntüden takımın formasyonu ve topun konumu bilgileri görüntü işleme teknikleri ile sayısal verilere dönüştürülmektedir. Diğer 10 girdi verisi ise FM oyunu menülerinden alınmaktadır. Elde edilen veriler uygun sayısal formlara dönüştürüldükten sonra d-min-maks normalizasyonundan geçirilmektedir. d-min-maks normalizasyonu Eşitlik 4.5'e göre hesaplanmaktadır.

$$x'_i = 0,8 \frac{x_i - x_{min}}{x_{maks} - x_{min}} + 0,1 \quad (4.5)$$

Burada; x'_i verinin normalize edilmiş hali, x_i gerçek veridir. x_{min} , veri kümesinde ilgili alanın en küçük değerini gösterirken, x_{maks} , en büyük değerini göstermektedir.

Veri kümesindeki girdiler sayısal değerlere dönüştürülmektedir. Örneğin; maçın yeri girdisinde ev sahibi ve deplasman durumu 1 ve 2 değerlerini, maçın önemi girdisindeki hazırlık ve resmi maç durumu 1 ve 2 değerlerini, her diziliş girdisi tüm diziliş kombinasyonlarına göre bir sayısal değer almaktadır. Diğer tüm girdiler benzer şekilde düzenlenmektedir. Daha sonra, girdiler Eşitlik 4.5'e göre [0,1-0,9] aralığında uygun değerlere dönüştürülmektedir. Veri kümesindeki girdiler veri ön işleme adımlarından geçtikten sonra çeşitli makine öğrenmesi algoritmalarına göre öğretilmekte ve doğruluklar hesaplanmaktadır.

4.1.3. Makine Öğrenmesi Algoritmaları ile Veri İşleme

Veriler makine öğrenmesi yöntemleri için normalleştirildikten sonra veri işleme aşamasına geçilmektedir. Veri kümesi ile farklı modeller öğretilmekte ve performansları değerlendirilmektedir. Böylece sis bilişime dayalı güvenli IoT altyapısında maç anında taktiksel analiz için en uygun model seçilebilmektedir.

4.1.3.1. Veri Kümesinin Eğitilmesi

FM'den elde edilen veriler sayısal hale getirildikten sonra makine öğrenmesi algoritmaları ile modelin başarısı test edilmektedir. Veri kümesinde 110 girdi 1 çıktı bulunmaktadır. Çıktı gol atma, gol yeme ve golsüz durum için sırası ile SG, CG ve NG etiketleri ile belirtilen 3 sınıftan oluşmaktadır. Veri kümesi 1996 veriyi barındırmaktadır. Sistemi aşırı uyuma (overfitting) karşı korumak için "fold" değeri 10'a ayarlanarak çapraz doğrulama (cross validation) işlemi uygulanmaktadır. Bu

şartlar altında Matlab'ın "Classifier Learner" aracı kullanılarak çeşitli makine öğrenmesi algoritmalarından sonuçlar alınmaktadır. Model test edilirken diskriminant analizi, KNN, naive Bayes, SVM, karar ağaçları ve topluluk öğrenmesi yöntemlerinin farklı türevleri kullanılmaktadır.

4.1.3.2. Deneysel Sonuçlar

Önerilen model birçok makine öğrenmesi yöntemiyle test edilmektedir. En iyi performansı veren 10 yöntemin isimleri, doğruluk değerleri ve eğri altında kalan alan (AUC) değerleri Tablo 4.1'de gösterilmektedir.

Tablo 4.1. Önerilen model için en iyi performansa sahip yöntemler

Sıra	Öğrenme Yöntemleri	Doğruluk (%)	AUC
1	Parametreleri optimize edilmiş topluluk öğrenme (Adaboost)	87,2	0,98
2	Güçlendirilmiş ağaç (Boosted tree)	85,9	0,98
3	Kaba ağaç (Coarse tree)	85,1	0,94
4	Rastgele örneklem ile güçlendirilmiş ağaç (RUSBoosted tree)	84,8	0,97
5	Orta ağaç (Medium tree)	84,3	0,96
6	Doğrusal SVM (Linear SVM)	84,1	0,97
7	Karesel SVM (Quadratic SVM)	83,6	0,97
8	Paketlenmiş ağaç (Bagged tree)	83,4	0,95
9	Doğrusal diskriminant (Linear Discriminant)	83,3	0,96
10	Orta Gauss SVM (Medium Gauss SVM)	83,2	0,97

En iyi başarıyı sağlayan yöntemler topluluk öğrenme, karar ağaçları, SVM ve diskriminant analizi şeklindedir. Bu sonuçlardan en yüksek performansa sahip 3 yöntemin karmaşıklık matrisleri Şekil 4.8'de gösterilmektedir. Tahmin edilen ve doğru bulunan sayılar değerlendirildiğinde CG ve SG için yanlış tahmin edilen durumların büyük bir kısmının NG olarak tahmin edildiği görülmektedir. NG, pozisyonun golsüz geçildiğini ve teknik ekibe herhangi bir uyarı gerektirmediğini belirten etikettir. Bu nedenle yanlış tahmin edilen ve oldukça az olan bu durumların uyarısız geçileceği ancak en azından ortaya çıkan sonucun teknik ekibi yanlış yönlendirmediği görülmektedir. NG durumlarının biraz daha yüksek oranda SG ve CG olarak tahmin edilmesinin çıkarımı ise oluşan atağın gole yatkın olduğu ancak bireysel kabiliyet ve bitiricilik eksikliği nedeniyle pozisyonun golsüz geçtiği şeklinde yorumlanabilmektedir. Başarısı en yüksek olan yöntem AdaBoost olsa da karmaşıklık matrislerindeki değer incelendiğinde CG ve SG durumlarının yüksek doğrulukta

tahmin edildiği kaba ağaç yönteminin de oldukça başarılı olduğu söylenebilmektedir.



Şekil 4.8. En iyi performansa sahip ilk 3 yöntemin karmaşıklık matrisleri

Sınıflandırıcıların doğruluk değeri gibi performansını gösteren önemli parametrelerden biri de F1 puan değerleridir. Kesinlik (precision) ve geri çağırma (recall) değerlerinin harmonik ortalamasıdır ve Eşitlik 4.5'teki gibi hesaplanmaktadır. Her sınıf için F1 puanı hesaplandıktan sonra ağırlıklandırılmış F1 puan değerleri elde edilmektedir. Veri sayılarını da bir parametre olarak değerlendiren ağırlıklandırılmış F1 puanı Eşitlik 4.6'ya göre hesaplanmaktadır.

$$F1^{class} = \frac{2}{recall^{-1} + precision^{-1}} = \frac{tp}{tp + \frac{1}{2}(fp + fn)} \quad (4.5)$$

Burada; tp , doğru pozitif; fp , yanlış pozitif; fn , yanlış negatif değerleri gösterir.

$$\text{Ağırlıklandırılmış } F1 = \frac{F1^{CG} \times VS^{CG} + F1^{SG} \times VS^{SG} + F1^{NG} \times VS^{NG}}{VS^{CG} + VS^{SG} + VS^{NG}} \quad (4.6)$$

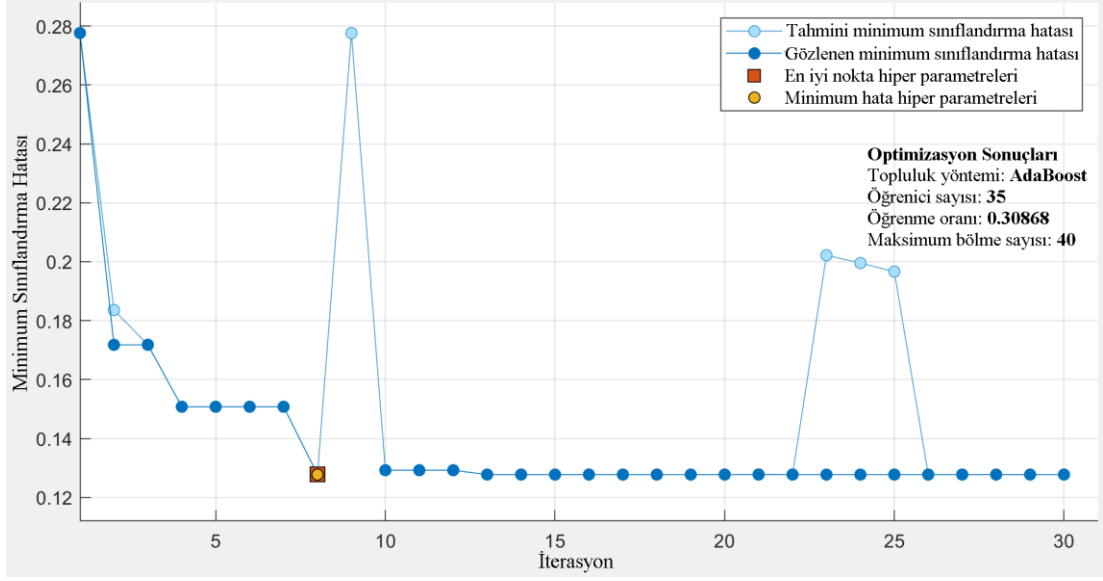
Burada; VS , ilgili sınıfın veri sayısını göstermektedir.

Sınıflandırıcıların ağırlıklı F1 puan değerleri Tablo 4.2'de gösterilmektedir. Doğruluk değerinde olduğu gibi F1 puanı da optimize edilmiş Adaboost yönteminin en yüksek performansa sahip olduğunu göstermektedir.

Tablo 4.2. F1 puanlarına göre yöntemlerin karşılaştırılması

Sıra	Öğrenme Yöntemleri	Ağırlıklandırılmış F1 Puanı (%)
1	Parametreleri optimize edilmiş topluluk öğrenme (Adaboost)	86,78
2	Güçlendirilmiş ağaç (Boosted tree)	85,31
3	Kaba ağaç (Coarse tree)	83,72

Her karar sisteminin sıfırdan farklı bir doğruluk değeri vardır. Bir sınıflandırıcının sınıflandırma hatasının kontrol edilebileceği iki bileşen bulunmaktadır: önyargı ve varyans. Önyargı sınıflandırıcının doğruluğunu, varyans ise farklı eğitim kümeleri ile eğitildiğinde sınıflandırıcının kesinliği belirtmektedir (Polikar, 2012). Düşük önyargılı sınıflandırıcılar yüksek varyansa veya tam tersine sahiptir. Topluluk öğrenmede daha doğru kararlar verebilmek için birden fazla benzer önyargıya sahip sınıflandırıcıların ortalaması alınmaktadır. Böylece birçok sınıflandırıcının verdiği doğru kararlar ortalama da büyük oranda doğru kabul edilmektedir. Bu tür sistemler başarıyı düşük bir sınıflandırıcı seçme ihtimalini azaltmaktadır. Ayrıca, en uygun parametrelerin belirlenip minimum sınıflandırma hatası (minimum classification error - MCE) değerinin optimize edildiği AdaBoost kullanan optimize edilebilir topluluk öğrenme yöntemi veri kümesi üzerindeki en iyi sonucu vermektedir. Yöntemin 30 iterasyondan oluşan optimizasyon sonucundaki MCE değerini gösteren grafik Şekil 4.9'da belirtilmektedir.



Şekil 4.9. Topluluk öğrenmesinin optimizasyonu sonucundaki MCE grafiği

4.1.3.3. Bir Topluluk Öğrenme Yöntemi: AdaBoost

Topluluk öğrenme yöntemi (ensemble learning), paketleme (bagging), güçlendirme (boosting) ve yığılma (stacking) olmak üzere üç yapıdan oluşmaktadır. Paketlemede öncelikle veri kümesi eğitim ve test olarak (genellikle 70/30 oranı ile) ayrılmaktadır. Eğitim verilerinden rastgele ve tekrarlı olarak alınan örnekler belli sayıdaki paketlere yerleştirilmektedir. Her bir örneklem paketi kurulan modellerde eğitilmektedir. Çıktıların ortalaması veya oylaması karar verme için kullanılmaktadır. Güçlendirmede, paketlemeye benzer şekilde veriler bölünmekte ve rastgele örneklenmektedir. Farkları ise paketlemede her bir örneklem kendi içinde eğitilip çıktı üretmekte yani her model eşit şansa sahip olmakta iken, güçlendirmede bir modelde yanlış sınıflandırılan veriler diğer model için önceliklendirilmektedir. Güçlendirmede bir seferde üç sınıflandırıcı kümesi oluşmaktadır. Birinci ve ikinci eğitici paketlemedeki gibi veri kümesinin rastgele farklı bölümlerinde eğitilmektedir. Üçüncü sınıflandırıcı birinci ve ikinci sınıflandırıcının uyuşmadığı veriler üzerinde eğitilmektedir. Daha sonra bu üç sınıflandırıcı çoğunluk oyu yöntemi ile birleştirilmektedir. Yığılamada ise sınıflandırıcıların öznelik uzayının hangi bölümünde başarılı olduğunun çıkarılarak bunlara göre karar verilmesini sağlanmaktadır. Sınıflandırıcıların çıktıları yine ayrı bir sınıflandırıcı ile birleştirilerek karar verilmektedir (Polikar, 2012).

Başlangıç. Gözlem ağırlığı

$$w_i = 1/N, \quad i = 1, 2, \dots, N$$

for $m = 1$ to M **do**

w_i ağırlıklı eğitim kümesine bir $G_m(x)$ sınıflandırıcı uygulanır.

$$err_m = \frac{\sum_{i=1}^N w_i \times I(y_i \neq G_m(x_i))}{\sum_{i=1}^N w_i} \text{ değeri hesaplanır.}$$

$\alpha_m = \log((1 - err_m)/err_m)$ değeri hesaplanır.

$w_i \leftarrow w_i \times \exp[\alpha_m \times I(y_i \neq G_m(x_i))]$, $i = 1, 2, \dots, N$ ağırlık ayarlanır.

Çıktı. $G(x) = \text{sign}[\sum_{m=1}^M \alpha_m \times G_m(x)]$

Şekil 4.10. AdaBoost algoritması (Hastie, vd., 2001)

Adaptif güçlendirme (adaptive boosting - AdaBoost), güçlendirme yapılarının en temel algoritmalarındandır. Her modelin çıktısındaki yanlış tahminler ağırlık değerleri yükseltılarak diğer modeldeki veri kümesinde eğitilmektedir. Böylece tekrarlı olarak örneklemelere ayrılan yapılarıdaki gibi maksimum derinlikteki karar ağaçları yerine iki yaprak düğümden oluşan 1 derinlikli karar ağaçları kullanılmaktadır. Birleştirme işlemi ise ağırlıklı çoğunluk oylaması ile gerçekleştirilmektedir (Polikar, 2012). AdaBoost algoritması Şekil 4.10'da gösterilmektedir. Başlangıçta tüm ağırlıklar toplam örneklem boyutu (N) dikkate alınarak $w_i = 1/N$ olarak atanmaktadır. Her ardışık yinelemede (m) gözlem ağırlıkları ayrı ayrı değiştirilerek sınıflandırma algoritması bu verilere tekrar uygulanmaktadır. Her tekrar işleminde önceki adımdaki sınıflandırıcı (G_{m-1}) tarafından doğru sınıflandırılanlar için ağırlıklar azaltılırken yanlış sınıflandırılanlar için ağırlıklar artırılmaktadır. Daha sonra ağırlıklı hata oranı (err_m) hesaplanmaktadır. Son sınıflandırıcı tarafından $G(x)$ üretilirken $G_m(x)$ 'e verilen α_m ağırlığı hesaplanmaktadır. Ağırlıklar yeni değerler ile bir sonraki döngü için güncellenirken yanlış sınıflandırmaların etkisini artırmak için $\exp(\alpha_m)$ ile ölçeklendirilmektedir (Hastie, vd., 2001). Çıktıya tüm sınıflandırıcıların ağırlıklı sonuçlarına göre karar verilmektedir. AdaBoost algoritması, önerilen sis bilişime dayalı güvenli IoT altyapısında maç sırasında taktiksel analiz modeli için kullanılabilir. Sonuç olarak, hazırlanan veri kümesi kullanılarak parametreleri en uygun değerlere getirilmiş AdaBoost %87,2 doğruluk değeri ve %86,78 ağırlıklandırılmış F1 puanı ile önerilen modelde kullanılabilir en iyi yöntemdir.

5. SONUÇLAR VE GELECEK ÇALIŞMALAR

Bu tez kapsamında literatürde yer alan IoT ve sis tabanlı güvenli mimari, sis düğümlerinde görev önceliklerini dikkate alan ve düşük gecikme süresi sağlayan kaynak tahsis algoritması ve makine öğrenmesi tabanlı taktik analiz problemlerine çözüm sunulmuştur.

Sporda sağlık ve taktiksel verilerinin analiz edilmesiyle elde edilen sonucun teknik ekibe ve kulüp doktorlarına bildirildiği, hafif ve güvenli sis tabanlı bir IoT mimarisi önerilmektedir. Bu mimaride, uç düğümlere verilen yanıtların düşük gecikme süresi ile iletilmesi, verilerin yerel ağda işlenmesi ve saklanması sağlanmaktadır. Ayrıca, sis düğümlerinde gecikme süresini düşüren bir görev yönetimi ve kaynak tahsis algoritması önerilmektedir. Sis düğümlerinde öncelik kuyruğu yöntemi ile acil işlemlere yüksek öncelik verilmektedir. Önerilen mimari yapı ve algoritma, iFogSim simülatörü kullanılarak test edilmiş ve simülasyon sonuçları benzer bir çalışma ile karşılaştırılmıştır. Bu sonuçlara göre, önerilen mimari ve algoritmanın sis düğümlerinde kaynak tahsisi ve veri işlemenin oldukça düşük gecikme süresi ve düşük maliyet ile gerçekleştiği ortaya çıkmaktadır. Spordaki diğer benzer çalışmalarda, IoT uygulamaları genellikle sadece bulut bilişim kullanılarak tasarlanmış ve yalnızca sağlık izleme sistemleri ele alınmıştır. Ancak, önerilen mimaride veri işleme ve depolama için sis bilişim kullanılması hem güvenlik hem de düşük gecikme süresi açısından daha yüksek performans elde edilmesini sağlamaktadır. Önerilen mimarinin güvenliği kapsamında, kimlik doğrulama için, kaynak kısıtlamalı cihazlarda hafif bir kimlik doğrulama sağlayan FLAT protokolü kullanılmaktadır. Kimlik doğrulamaya ek olarak, bu protokol veri gizliliğine ve sis katmanındaki veri bütünlüğüne de destek vermektedir. Ayrıca, blokzinciri tabanlı SDN denetleyicisi ile özellikle su baskını gibi bazı saldırılar tespit edilmekte ve veri bütünlüğü sağlanmaktadır. Bunlara ek olarak, sis bilişim yapısı yerel ağda çalıştığından veriler güvensiz olan İnternet ortamına çıkmadan işlenebilmesi sisteme güvenlik ve düşük gecikme süresi sağlamaktadır. Ayrıca, önerilen mimari sadece sağlık izleme değil, aynı zamanda maç anındaki taktik analizlere de katkı sağlayan bir alt yapı barındırmaktadır. Bu kapsamda, IoT sistemlerde kullanılan depolama ve veri işleme birimlerinin incelenmesi ve sis bilişimde güvenlik yapıları (Akleyek ve Karakaya, 2020) çalışmasında, IoT sistemlerde kaynak tahsis yaklaşımlarının detaylı incelenmesi (Karakaya ve Akleyek, 2022) çalışmasında, önerilen sis bilişim ve IoT tabanlı mimari ve sis düğümlerinde

kaynak tahsis algoritması da (Karakaya ve Akleylek, 2021) çalışmasında yayınlanmıştır.

Bilişim sistemlerine dayalı spor çalışmaları kapsamında, önerilen sis bilişim ve IoT tabanlı altyapı üzerinde çalışabilen ve maç anında taktik analiz yapabilen makine öğrenmesi tabanlı bir model önerilmektedir. Modelin başarısını test etmek için veriler gerçeğe çok yakın bir simülasyon oyunu olan FM'den alınmaktadır. FM'de Türkiye Süper Ligi'nden bir takım seçilerek 470 maçta toplam 1996 gol atma, gol yeme ve golsüz durum verisi toplanmıştır. Her pozisyon için alınan 50 görüntüde saha 4 parçaya bölünerek, belirli zaman aralıklarında takımın anlık diziliş bilgileri ve top konum verileri oluşturulmaktadır. Bunlara ek olarak, bazı maç bazlı sabit veriler de eklenerek 110 girdiden oluşan bir veri kümesi elde edilmiştir. Görüntü işleme teknikleri ile sayısallaştırılan veriler normalize edildikten sonra çeşitli makine öğrenmesi algoritmalarında test edilmektedir. Bu modelde, önemli pozisyonlar (gol atma, gol yeme gibi) öncesindeki belli zaman aralıklarındaki girdiler değerlendirilerek maç gidişatında benzer örüntünün yakalanması durumunda teknik ekibe uyarı gönderilmektedir. Model, diskriminant analizi, KNN, naive Bayes, SVM, karar ağaçları ve topluluk öğrenme yöntemlerinin farklı varyasyonları kullanılarak test edilmiştir. En iyi performansı sağlayan yöntemler topluluk sınıflandırıcıları, karar ağaçları, SVM ve diskriminant analizidir. Bir topluluk öğrenme yöntemi olan ve en uygun parametreler için optimize edilen AdaBoost, %87,2 doğruluk ve %86,78 ağırlıklandırılmış F1 puanı ile en yüksek performansa sahiptir. Başka bir topluluk öğrenme yöntemi olan güçlendirilmiş ağaç %85,9 doğruluk, %85,31 ağırlıklandırılmış F1 puanı ile ikinci en yüksek değere sahip iken bir karar ağacı yöntemi olan kaba ağaç ise %85,1 doğruluk ve %83,72 ağırlıklandırılmış F1 puanı ile üçüncü en yüksek değere sahiptir. Bu nedenle AdaBoost algoritması, önerilen sis bilişime dayalı güvenli IoT mimarisinde maç sırasında taktiksel analiz modeli için kullanılabilir.

Tez kapsamında elde edilen sonuçların iyileştirilmesi ve farklı bakış açılarını sunabilmek amacıyla yapılması hedeflenen çalışmalar aşağıda listelenmiştir:

- Farklı veri kümeleri üretilerek futbolda taktik analiz yaklaşımının farklı açılardan ele alan yöntemler geliştirilebilir.
- Benzer veri kümeleri kullanılarak farklı spor dalları için taktik analizi destekleyen makine öğrenmesi tabanlı geri bildirim yaklaşımları

geliştirilebilir.

- Futbol sahasının daha fazla parçalara bölünerek daha hassas bir formasyon ve analiz ortaya koyan bir model oluşturulabilir.
- Önerilen IoT mimarisini kullanan makine öğrenmesi tabanlı sağlık izleme modeli oluşturulabilir.

Tez kapsamında kullanılan yöntemler ile ilişkili bazı çalışma önerileri ise şu şekilde listelenebilmektedir:

- Giyilebilir teknolojiler boyut, kapasite, enerji tüketimi, su geçirmezlik, yüksek veri işleme ve güvenlik açısından henüz istenilen seviyelere ulaşamamıştır. Bu nedenle spor uygulamaları için oyunculara ve seyircilere uygun, doğru veri sağlayan ve ter gibi dış etkenlerden etkilenmeyen giyilebilir teknolojiler geliştirilebilir.
- Blokzinciri kullanan IoT uygulamaları için kaynak kısıtlı cihazlarda verimli çalışan iş birliği algoritmaları üretilebilir.
- Önerilen sporlarda sağlık izleme ve taktik analiz mimarisine benzer farklı alanlar için hafif ve güvenli IoT mimarileri kurulabilir.
- IoT düğümlerinin, ağ geçitlerinin, sis düğümlerinin, bulut düğümlerinin iletişimi güvenli olmalıdır. Tüm IoT sistemlerinin gizlilik, kimlik doğrulama, bütünlük gibi ihtiyaçları nedeniyle kuantum sonrası güvenlikten etkilenmektedir (Fernández-Caramés, 2019). Bu nedenle, IoT ve bulut sistemleri için kuantum dirençli şemalar kurulabilir.
- Kuantum sonrası kriptografiye dayalı kimlik doğrulama protokolü geliştirilmiştir. LB-2PAKA olarak adlandırılan bu protokol, kanıtlanabilir güvenliği ölçmek ve ihlal zamanını tahmin etmek için analiz edilmiştir (Islam, 2020). IoT uygulamaları için post-kuantum kriptografiye dayalı hafif ve güçlü şifreleme ve kimlik doğrulama protokolleri geliştirilebilir.
- IoT uygulamaları için çok değişkenli ikinci dereceden ve kübik polinom sistemlerinin çeşitli kutupsal biçimleri kullanan bir kuantum sonrası tanımlama şeması önerilmiştir (Akleyek, vd., 2019). Ayrıca, Bi-GISIS sorunu göre doğrulanmış ve kuantum sonrası güvenli bir anahtar değişimi yaklaşımı önerilmiştir (Akleyek ve Seyhan, vd., 2020). Bu yapılara benzer

şekilde, IoT uygulamaları için kuantum sonrası kriptografiye dayalı tanımlama ve anahtar deęişim yöntemleri geliştirilebilir.



KAYNAKLAR

- Abdulkareem, K. H., Mohammed, M. A., Gunasekaran, S. S., Al-Mhiqani, M. N., Mutlag, A. A., Mostafa, S. A., Ali, N. S., & Ibrahim, D. A. (2019). A review of fog computing and machine learning: concepts, applications, challenges, and open issues. *IEEE Access*, 7, 153123-153140.
- Abedin, S. F., Alam, M. G. R., Kazmi, S. A., Tran, N. H., Niyato, D., & Hong, C. S. (2018). Resource allocation for ultra-reliable and enhanced mobile broadband IoT applications in fog network. *IEEE Transactions on Communications*, 67(1), 489-502.
- Abidin, D. (2021). A case study on player selection and team formation in football with machine learning. *Turkish Journal of Electrical Engineering & Computer Sciences*, 29(3), 1672-1691.
- Akleyek, S., & Karakaya, A. (2020). Sis Bilişimi ve Uygulamalarında Veri Güvenliği. Sağiroğlu, Ş., & Akleyek, S. (Ed.), *Siber Güvenlik ve Savunma: Biyometrik ve Kriptografik Uygulamalar* (pp. 1-28), Nobel Akademik Yayıncılık Eğitim Danışmanlık Tic. Ltd. Şti.
- Akleyek, S., & Seyhan, K. (2020). A Probably Secure Bi-GISIS Based Modified Ake Scheme With Reusable Keys. *IEEE Access*, 8, 26210-26222.
- Akleyek, S., Soysaldı, M., Boubiche, D. E., & Toral-Cruz, H. (2019). A Novel Method for Polar Form of Any Degree of Multivariate Polynomials with Applications in IoT. *Sensors*, 19(4), 903.
- Al-Asadi, M. A. M. (2018). Decision support system for a football team management by using machine learning techniques. *Xinyang Teachers College*, 10(2), 1-15.
- Andrienko, G., Andrienko, N., Anzer, G., Bauer, P., Budziak, G., Fuchs, G., Hecker, D., Weber, H., & Wrobel, S. (2019). Constructing spaces and times for tactical analysis in football. *IEEE transactions on visualization and computer graphics*. vol. 27, no. 4, pp. 2280–2297.
- Asali, E., Valipour, M., Zare, N., Afshar, A., Katebzadeh, M., & Dastghaibfard, G. H. (2016, April). Using Machine Learning approaches to detect opponent formation. In *2016 Artificial Intelligence and Robotics (IRANOPEN)* (pp. 140-144). IEEE.
- Baca, A., & Kornfeind, P. (2006). Rapid feedback systems for elite sports training. *IEEE Pervasive Computing*, 5(4), 70-76.
- Bader, A., Ghazzai, H., Kadri, A., & Alouini, M. S. (2016). Front-end intelligence for large-scale application-oriented internet-of-things. *IEEE Access*, 4, 3257-3272.
- Banaie, F., Yaghmaee, M. H., Hosseini, S. A., & Tashtarian, F. (2020). Load-balancing algorithm for multiple gateways in Fog-based Internet of Things. *IEEE Internet of Things Journal*, 7(8), 7043-7053.
- Bandodkar, A. J., Jeerapan, I., & Wang, J. (2016). Wearable chemical sensors: Present challenges and future prospects. *Acs Sensors*, 1(5), 464-482.
- Bin Mohd Rafeek, M. A., & McLean, G. N. (2015). Evaluation of Organization Development Interventions in a Video Simulation Game: A Case Study of Football Manager 2014™. *Organization Development Journal*, 33(4).
- Bonomi, F., Milito, R., Zhu, J., & Addepalli, S. (2012, August). Fog computing and its role in the internet of things. In *Proceedings of the first edition of the MCC workshop on Mobile cloud computing* (pp. 13-16).
- Boon, B. H., & Sierksma, G. (2003). Team formation: Matching quality supply and quality demand. *European Journal of Operational Research*, 148(2), 277-292.

- Burn-Murdoch J. (2018). How data analysis helps football clubs make better signings. Erişim adresi, <https://www.ft.com/content/84aa8b5e-c1a9-11e8-84cd-9e601db069b8>, (12.12.2021).
- Carling, C. (2011). Influence of opposition team formation on physical and skill-related performance in a professional soccer team. *European Journal of Sport Science*, 11(3), 155-164.
- Chen, D., Yang, C., Gong, P., Chang, L., Shao, J., Ni, Q., & Guizani, M. (2020). Resource cube: Multi-virtual resource management for integrated satellite-terrestrial industrial IoT networks. *IEEE Transactions on Vehicular Technology*, 69(10), 11963-11974.
- Cherabit, N., Chelali, F. Z., & Djeradi, A. (2012). Circular hough transform for iris localization. *Science and Technology*, 2(5), 114-121.
- Choi, Y., & Lim, Y. (2016). Optimization approach for resource allocation on cloud computing for iot. *International Journal of Distributed Sensor Networks*, 12(3), 3479247.
- Choudhari, T., Moh, M., & Moh, T. S. (2018, March). Prioritized task scheduling in fog computing. In *Proceedings of the ACMSE 2018 Conference* (pp. 1-8).
- Deng, S., Xiang, Z., Zhao, P., Taheri, J., Gao, H., Yin, J., & Zomaya, A. Y. (2020). Dynamical resource allocation in edge for trustable Internet-of-Things systems: A reinforcement learning method. *IEEE Transactions on Industrial Informatics*, 16(9), 6103-6113.
- Dobreff, G., Pašić, A., Sonkoly, B., & Toka, L. (2019, September). The formation game in football. In *6th Workshop on Sports Analytics: Machine Learning and Data Mining for Sports Analytics (MLSA)*, Wurzburg, Germany.
- Dou, Z., Si, G., Lin, Y., & Wang, M. (2019). An adaptive resource allocation model with anti-jamming in IoT network. *IEEE Access*, 7, 93250-93258.
- Faria, B. M., Reis, L. P., Lau, N., & Castillo, G. (2010, June). Machine Learning algorithms applied to the classification of robotic soccer formations and opponent teams. In *2010 IEEE Conference on Cybernetics and Intelligent Systems* (pp. 344-349). IEEE.
- Fernández-Caramés, T. M. (2019). From pre-quantum to post-quantum IoT security: A survey on quantum-resistant cryptosystems for the Internet of Things. *IEEE Internet of Things Journal*, 7(7), 6457-6480.
- Fiandrino, C., Allio, N., Kliazovich, D., Giaccone, P., & Bouvry, P. (2019). Profiling Performance of Application Partitioning for Wearable Devices in Mobile Cloud and Fog Computing. *IEEE Access*, 7, 12156-12166.
- Frey, M., Murina, E., Rohrbach, J., Walser, M., Haas, P., & Dettling, M. (2019, June). Machine learning for position detection in football. In *2019 6th Swiss Conference on Data Science (SDS)* (pp. 111-112). IEEE.
- Gu, Y., Chang, Z., Pan, M., Song, L., & Han, Z. (2018). Joint radio and computational resource allocation in IoT fog computing. *IEEE Transactions on Vehicular Technology*, 67(8), 7475-7484.
- Hänsel, K., Wilde, N., Haddadi, H., & Alomainy, A. (2015, December). Challenges with current wearable technology in monitoring health data and providing positive behavioural support. In *Proceedings of the 5th EAI International Conference on Wireless Mobile Communication and Healthcare* (pp. 158-161).
- Hastie, T., Tibshirani, R., & Friedman, J. (2001). The elements of statistical learning. *Springer series in statistics* New York, vol. 1, no. 10.

- He, X., Wang, K., Huang, H., Miyazaki, T., Wang, Y., & Guo, S. (2018). Green resource allocation based on deep reinforcement learning in content-centric IoT. *IEEE Transactions on Emerging Topics in Computing*, 8(3), 781-796.
- He, Y., Zhang, S., Tang, L., & Ren, Y. (2020). Large scale resource allocation for the internet of things network based on ADMM. *IEEE Access*, 8, 57192-57203.
- Hirotsu, N., & Wright, M. (2003). Determining the best strategy for changing the configuration of a football team. *Journal of the Operational Research Society*, 54(8), 878-887.
- Hocquet, A. (2016). Football Manager: Mutual Shaping between Game, Sport, and Community. *Journal of media studies and popular culture= Revue d'études des médias et de culture populaire*, 6(Special Issue).
- Horton, M., Gudmundsson, J., Chawla, S., & Estephan, J. (2015, May). Automated classification of passing in football. In *Pacific-Asia conference on knowledge discovery and data mining* (pp. 319-330). Springer, Cham.
- Hough, P. V. C. (1962). Method and means for recognizing complex pattern. Dec. 18 1962, uS Patent 3,069,654.
- Huang, X., Cui, Y., Chen, Q., & Zhang, J. (2020). Joint task offloading and QoS-aware resource allocation in fog-enabled Internet-of-Things networks. *IEEE Internet of Things Journal*, 7(8), 7194-7206.
- Hussain, F., Hussain, R., Anpalagan, A., & Benslimane, A. (2020). A new block-based reinforcement learning approach for distributed resource allocation in clustered iot networks. *IEEE Transactions on Vehicular Technology*, 69(3), 2891-2904.
- Ikram, M. A., Alshehri, M. D., & Hussain, F. K. (2015, December). Architecture of an IoT-based system for football supervision (IoT Football). In *2015 IEEE 2nd World Forum on Internet of Things (WF-IoT)* (pp. 69-74). IEEE.
- Islam, S. H. (2020). Provably secure two-party authenticated key agreement protocol for post-quantum environments. *Journal of Information Security and Applications*, 52, 102468.
- Islam, S. H., & Biswas, G. P. (2013). Design of improved password authentication and update scheme based on elliptic curve cryptography. *Mathematical and Computer Modelling*, 57(11-12), 2703-2717.
- Jiang, F., Wang, K., Dong, L., Pan, C., & Yang, K. (2020). Stacked autoencoder-based deep reinforcement learning for online resource scheduling in large-scale MEC networks. *IEEE Internet of Things Journal*, 7(10), 9278-9290.
- Joash Fernandes, C., Yakubov, R., Li, Y., Prasad, A. K., & Chan, T. C. (2020). Predicting plays in the National Football League. *Journal of Sports Analytics*, 6(1), 35-43.
- Karakaya, A., & Akleyek, S. (2018, March). A survey on security threats and authentication approaches in wireless sensor networks. In *2018 6th international symposium on digital forensic and security (ISDFS)* (pp. 1-4). IEEE.
- Karakaya, A., & Akleyek, S. (2021). A novel IoT-based health and tactical analysis model with fog computing. *PeerJ Computer Science*, vol. 7, p. :e342. doi:10.7717/peerj-cs.342.
- Karakaya, A., & Akleyek, S. (2022). A Review of Resource Allocation and Management Methods in IoT. *A Fusion of Artificial Intelligence and Internet of Things for Emerging Cyber Systems*, 409-429.
- Khan, L. U., Alsenwi, M., Yaqoob, I., Imran, M., Han, Z., & Hong, C. S. (2020). Resource optimized federated learning-enabled cognitive internet of things for smart industries. *IEEE Access*, 8, 168854-168864.

- Krishnan, P., Duttagupta, S., & Achuthan, K. (2020). SDN/NFV security framework for fog-to-things computing infrastructure. *Software: Practice and Experience*, 50(5), 757-800.
- Kubler, S., Robert, J., Hefnawy, A., Främling, K., Cherifi, C., & Bouras, A. (2017). Open IoT ecosystem for sporting event management. *IEEE Access*, 5, 7064-7079.
- Landers, J. R., & Duperrouzel, B. (2018). Machine learning approaches to competing in fantasy leagues for the NFL. *IEEE Transactions on Games*, 11(2), 159-172.
- Lee, J., Kim, D., & Niyato, D. (2020). Market analysis of distributed learning resource management for internet of things: a game-theoretic approach. *IEEE Internet of Things Journal*, 7(9), 8430-8439.
- Li, H., Jia, Z., & Xue, X. (2010, April). Application and analysis of ZigBee security services specification. In *2010 Second International Conference on Networks Security, Wireless Communications and Trusted Computing* (Vol. 2, pp. 494-497). IEEE.
- Librino, F., & Santi, P. (2020). Resource allocation and sharing in URLLC for IoT applications using shareability graphs. *IEEE Internet of Things Journal*, 7(10), 10511-10526.
- Liu, M., Song, T., & Gui, G. (2018). Deep cognitive perspective: Resource allocation for NOMA-based heterogeneous IoT with imperfect SIC. *IEEE Internet of Things Journal*, 6(2), 2885-2894.
- Liu, W., Nishio, T., Shinkuma, R., & Takahashi, T. (2014). Adaptive resource discovery in mobile cloud computing. *Computer Communications*, 50, 119-129.
- Liu, X., Qin, Z., Gao, Y., & McCann, J. A. (2019). Resource allocation in wireless powered IoT networks. *IEEE Internet of Things Journal*, 6(3), 4935-4945.
- Liu, X., Yu, J., Wang, J., & Gao, Y. (2020). Resource allocation with edge computing in IoT networks via machine learning. *IEEE Internet of Things Journal*, 7(4), 3415-3426.
- Luan, T. H., Gao, L., Li, Z., Xiang, Y., Wei, G., & Sun, L. (2015). Fog computing: Focusing on mobile users at the edge. arXiv 2015. *arXiv preprint arXiv:1502.01815*.
- Madsen, H., Burtschy, B., Albeanu, G., & Popentiu-Vladicescu, F. L. (2013, July). Reliability in the utility computing era: Towards reliable fog computing. In *2013 20th International Conference on Systems, Signals and Image Processing (IWSSIP)* (pp. 43-46). IEEE.
- Mahmud, R., & Buyya, R. (2019). Modelling and simulation of fog and edge computing environments using iFogSim toolkit. *Fog and edge computing: Principles and paradigms*, 1-35.
- Malik, H., Pervaiz, H., Alam, M. M., Le Moullec, Y., Kuusik, A., & Imran, M. A. (2018). Radio resource management scheme in NB-IoT systems. *IEEE Access*, 6, 15051-15064.
- McLean, S., Salmon, P. M., Gorman, A. D., Wickham, J., Berber, E., & Solomon, C. (2018). The effect of playing formation on the passing network characteristics of a professional football team. *Human Movement*, 19(5), 14-22.
- Min, M., Xiao, L., Chen, Y., Cheng, P., Wu, D., & Zhuang, W. (2019). Learning-based computation offloading for IoT devices with energy harvesting. *IEEE Transactions on Vehicular Technology*, 68(2), 1930-1941.
- Moura, F. A., Martins, L. E. B., Anido, R. D. O., De Barros, R. M. L., & Cunha, S. A. (2012). Quantitative analysis of Brazilian football players' organisation on the pitch. *Sports biomechanics*, 11(1), 85-96.
- Mouradian, C., Kianpisheh, S., Abu-Lebdeh, M., Ebrahimnezhad, F., Jahromi, N. T., & Glitho, R. H. (2019). Application component placement in NFV-based hybrid cloud/fog systems with mobile fog nodes. *IEEE Journal on Selected Areas in Communications*, 37(5), 1130-1143.

- Mukherjee, M., Matam, R., Shu, L., Maglaras, L., Ferrag, M. A., Choudhury, N., & Kumar, V. (2017). Security and privacy in fog computing: Challenges. *IEEE Access*, 5, 19293-19304.
- Musaddiq, A., Zikria, Y. B., Hahm, O., Yu, H., Bashir, A. K., & Kim, S. W. (2018). A survey on resource management in IoT operating systems. *IEEE Access*, 6, 8459-8482.
- Müller-Budack, E., Theiner, J., Rein, R., & Ewerth, R. (2019, October). " Does 4-4-2 exist?" - An Analytics Approach to Understand and Classify Football Team Formations in Single Match Situations. In *Proceedings Proceedings of the 2nd International Workshop on Multimedia Content Analysis in Sports* (pp. 25-33).
- Na, W., Jang, S., Lee, Y., Park, L., Dao, N. N., & Cho, S. (2018). Frequency resource allocation and interference management in mobile edge computing for an Internet of Things system. *IEEE Internet of Things Journal*, 6(3), 4910-4920.
- Naha, R. K., Garg, S., Georgakopoulos, D., Jayaraman, P. P., Gao, L., Xiang, Y., & Ranjan, R. (2018). Fog computing: Survey of trends, architectures, requirements, and research directions. *IEEE access*, 6, 47980-48009.
- Narizuka, T., & Yamazaki, Y. (2019). Clustering algorithm for formations in football games. *Scientific reports*, 9(1), 1-8.
- Nassar, A., & Yilmaz, Y. (2019). Reinforcement learning for adaptive resource allocation in fog RAN for IoT with heterogeneous latency requirements. *IEEE Access*, 7, 128014-128025.
- Nofer, M., Gomber, P., Hinz, O., & Schiereck, D. (2017). Blockchain. *Business & Information Systems Engineering*, 59(3), 183-187.
- Okay, F. Y., & Ozdemir, S. (2018). Routing in fog-enabled IoT platforms: A survey and an SDN-based solution. *IEEE Internet of Things Journal*, 5(6), 4871-4889.
- Patel, K. K., & Patel, S. M. (2016). Internet of things-IOT: definition, characteristics, architecture, enabling technologies, application & future challenges. *International journal of engineering science and computing*, 6(5).
- Paul, A., Pinjari, H., Hong, W. H., Seo, H. C., & Rho, S. (2018). Fog computing-based IoT for health monitoring system. *Journal of Sensors*, 2018.
- Pedersen, S. J. K. (2007). Circular hough transform. *Aalborg University, Vision, Graphics, and Interactive Systems*, 123(6).
- Pfisterer, D., Lipphardt, M., Buschmann, C., Hellbrueck, H., Fischer, S., & Sauselin, J. H. (2006, May). Marathonnet: adding value to large scale sport events-a connectivity analysis. In *Proceedings of the first international conference on Integrated internet ad hoc and sensor networks* (pp. 12-es).
- Pham, T. M. (2020). Optimization of Resource Management for NFV-Enabled IoT Systems in Edge Cloud Computing. *IEEE Access*, 8, 178217-178229.
- Polikar, R. (2012). Ensemble learning. In *Ensemble machine learning* (pp. 1-34). Springer, Boston, MA.
- Ramírez, P. L. G., Taha, M., Lloret, J., & Tomás, J. (2019). An intelligent algorithm for resource sharing and self-management of wireless-IoT-gateway. *IEEE Access*, 8, 3159-3170.
- Rein, R., & Memmert, D. (2016). Big data and tactical analysis in elite soccer: future challenges and opportunities for sports science. SpringerPlus, 5(1), 1-13.
- Rizon, M., Haniza, Y., Puteh, S., Yeon, A., Shakaff, M., Abdul Rahman, S., Sugisaka, M., Sazali, Y., Rozailan, M., M., & Karthigayan, M. (2005). Object detection using circular Hough transform. *American Journal of Applied Sciences*, 2.

- Sangaiah, A. K., Hosseinabadi, A. A. R., Shareh, M. B., Bozorgi Rad, S. Y., Zolfagharian, A., & Chilamkurti, N. (2020). IoT resource allocation and optimization based on heuristic algorithm. *Sensors*, 20(2), 539.
- Santos, M. L., Carneiro, J. C., Franco, A. M., Teixeira, F. A., Henriques, M. A., & Oliveira, L. B. (2020). FLAT: Federated Lightweight Authentication for The Internet of Things. *Ad Hoc Networks*, 107, 102253.
- Sarmiento, H., Marcelino, R., Anguera, M. T., Campaniço, J., Matos, N., & Leitão, J. C. (2014). Match analysis in football: a systematic review. *Journal of sports sciences*, 32(20), 1831-1843.
- Sharma, P. K., Chen, M. Y., & Park, J. H. (2017). A software defined fog node based distributed blockchain cloud architecture for IoT. *Ieee Access*, 6, 115-124.
- Shaw, L., & Glickman, M. (2019). Dynamic analysis of team strategy in professional football. *Barça sports analytics summit*, 1-13.
- Stübinger, J., Mangold, B., & Knoll, J. (2020). Machine learning in football betting: Prediction of match results based on player characteristics. *Applied Sciences*, 10(1), 46.
- Su, J., Xu, H., Xin, N., Cao, G., & Zhou, X. (2018). Resource allocation in wireless powered IoT system: a mean field Stackelberg game-based approach. *Sensors*, 18(10), 3173.
- Taki, T., & Hasegawa, J. I. (2000, June). Visualization of dominant region in team games and its application to teamwork analysis. In *Proceedings computer graphics international 2000* (pp. 227-235). IEEE.
- Tang, L., & Hu, H. (2020). Computation offloading and resource allocation for the Internet of Things in energy-constrained MEC-enabled HetNets. *IEEE Access*, 8, 47509-47521.
- Tavana, M., Azizi, F., Azizi, F., & Behzadian, M. (2013). A fuzzy inference system with application to player selection and team formation in multi-player sports. *Sport Management Review*, 16(1), 97-110.
- Tosato, M., & Wu, J. (2018). An application of PART to the Football Manager data for players clusters analyses to inform club team formation. *Big Data & Information Analytics*.
- Trequatrini, R., Del Giudice, M., Cuzzo, B., & Palmaccio, M. (2016). Does sport innovation create value? The case of professional football clubs. *Technology, Innovation and Education*, 2(1), 1-15.
- Vales-Alonso, J., López-Matencio, P., Gonzalez-Castaño, F. J., Navarro-Hellín, H., Baños-Guirao, P. J., Pérez-Martínez, F. J., ... & Duro-Fernández, R. (2010). Ambient intelligence systems for personalized sport training. *Sensors*, 10(3), 2359-2385.
- Varghese, B., Wang, N., Barbhuiya, S., Kilpatrick, P., & Nikolopoulos, D. S. (2016, November). Challenges and opportunities in edge computing. In *2016 IEEE International Conference on Smart Cloud (SmartCloud)* (pp. 20-26). IEEE.
- Wang, H., Li, J., Tian, J., & Wang, K. (2019, November). WebIDE Cloud Server Resource Allocation With Task Pre-Scheduling in IoT Application Development. In *2019 IEEE International Conference on Industrial Internet (ICII)* (pp. 278-286). IEEE.
- Wang, Q., Shao, S., Guo, S., Qiu, X., & Wang, Z. (2020). Task allocation mechanism of power Internet of things based on cooperative edge computing. *IEEE Access*, 8, 158488-158501.
- Wang, X., Chen, M., Taleb, T., Ksentini, A., & Leung, V. C. (2014). Cache in the air: Exploiting content caching and delivery techniques for 5G systems. *IEEE Communications Magazine*, 52(2), 131-139.

- Xiong, X., Zheng, K., Lei, L., & Hou, L. (2020). Resource allocation based on deep reinforcement learning in IoT edge computing. *IEEE Journal on Selected Areas in Communications*, 38(6), 1133-1146.
- Xu, S., Liu, Q., Gong, B., Qi, F., Guo, S., Qiu, X., & Yang, C. (2020). RJCC: Reinforcement-learning-based joint communicational-and-computational resource allocation mechanism for smart city IoT. *IEEE Internet of Things Journal*, 7(9), 8059-8076.
- Yang, H. K., Cha, H. J., & Song, Y. J. (2018). Secure identifier management based on Blockchain technology in NDN environment. *IEEE Access*, 7, 6262-6268.
- Yang, H., Zhong, W. D., Chen, C., Alphones, A., & Xie, X. (2020). Deep-reinforcement-learning-based energy-efficient resource management for social and cognitive Internet of Things. *IEEE Internet of Things Journal*, 7(6), 5677-5689.
- Yang, Q., Liu, Y., Chen, T., & Tong, Y. (2019). Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 10(2), 1-19.
- Yi, S., Li, C., & Li, Q. (2015, June). A survey of fog computing: concepts, applications and issues. In *Proceedings of the 2015 workshop on mobile big data* (pp. 37-42).
- Yoon, G., Choi, D., Lee, J., & Choi, H. (2019). Management of IoT Sensor Data Using a Fog Computing Node. *Journal of Sensors*, 2019, 5107457.
- Zhang, D., Qiao, Y., She, L., Shen, R., Ren, J., & Zhang, Y. (2018). Two time-scale resource management for green Internet of Things networks. *IEEE Internet of Things Journal*, 6(1), 545-556.
- Zhang, Q., Gui, L., Hou, F., Chen, J., Zhu, S., & Tian, F. (2020). Dynamic task offloading and resource allocation for mobile-edge computing in dense cloud RAN. *IEEE Internet of Things Journal*, 7(4), 3282-3299.

ÖZ GEÇMİŞ

Aykut KARAKAYA, Kdz. Ereğli Atatürk Anadolu Lisesi'ni bitirdikten sonra Ondokuz Mayıs Üniversitesi Mühendislik Fakültesi'nden 2013 yılında mezun oldu. 2016 yılında OMÜ LEE Bilgisayar Mühendisliği Yüksek Lisans programını bitirdi. Mezuniyetinden bu yana 2014-2017 yılları arasında Ondokuz Mayıs Üniversitesi Bilgisayar Mühendisliği Bölümü'nde araştırma görevlisi ve 2017'den beri Zonguldak Bülent Ecevit Üniversitesi'nde öğretim görevlisi olarak görev yapan Aykut KARAKAYA iyi derecede İngilizce bilmektedir. Temel çalışma alanları, nesnelerin İnterneti, bilgisayar ağları, bilgisayar yazılımı, bilgi güvenliği ve makine öğrenmesidir (14.12.2021).

İletişim Bilgileri

ORCID ID: <https://orcid.org/0000-0001-6970-3239>

Yayınlanmış Çalışmalar:

1. Karakaya, A., Akleylek, S., & Ulu, A. (2022). GOALALERT: A Novel Real-Time Technical Team Alert Approach Using Machine Learning on an IoT-Based System in Sports. (Hakem değerlendirme sürecinde).
2. Karakaya, A., & Akleylek, S. (2022). A Review of Resource Allocation and Management Methods in IoT. *A Fusion of Artificial Intelligence and Internet of Things for Emerging Cyber Systems*, 409-429.
3. Karacan, E., Akleylek, S., & Karakaya, A. (2021, June). PQ-FLAT: A New Quantum-Resistant And Lightweight Authentication Approach for M2M Devices. In *2021 9th International Symposium on Digital Forensics and Security (ISDFS)* (pp. 1-5). IEEE.
4. Karakaya, A., & Akleylek, S. (2021). A Novel IoT-Based Health and Tactical Analysis Model with Fog Computing. *PeerJ Computer Science*, 7, e342.
5. Akleylek, S., & Karakaya, A. (2020). Sis Bilişimi ve Uygulamalarında Veri Güvenliği. Sağiroğlu, Ş., & Akleylek, S. (Ed.), *Siber Güvenlik ve Savunma: Biyometrik ve Kriptografik Uygulamalar* (pp. 1-28), *Nobel Akademik Yayıncılık Eğitim Danışmanlık Tic. Ltd. Şti.*
6. Karakaya, A., & Akleylek, S. (2018, March). A Survey on Security Threats and Authentication Approaches in Wireless Sensor Networks. In *2018 6th international symposium on digital forensic and security (ISDFS)* (pp. 1-4). IEEE.
7. Karakaya, A., Akleylek, S., Erzurumlu, K., & Kılıç, E. (2018). A Novel Model for E-Book Borrowing Management System. *Academic Platform Journal of Engineering and Science*, 6(3), 49-57.
8. Kiliç, E., Ateş, N., Karakaya, A., & Şahin, D. Ö. (2015, May). Two new feature extraction methods for text classification: TESDF and SADF. In *2015 23rd Signal*

Processing and Communications Applications Conference (SIU) (pp. 475-478). IEEE.

9. Karhan, Z., Karakaya, A., Şenyer, N., & Kayhan, G. (2015, May). Bird's-eye view images taken plant material and counting. In *2015 23rd Signal Processing and Communications Applications Conference (SIU)* (pp. 1541-1544). IEEE.
10. Karakaya, A., & Erzurumlu, K. (2015). A New Approach to E Book. Presented at the A New Approach to E-Book [Abs]. *2nd International Conference on Computational and Experimental Science and Engineering*, Antalya, Turkey.
11. Karakaya, A., & Erzurumlu, K. (2014). Bir Başarı Hikayesi: Açık Kaynak Kodlu Çözümler ile Optik Sınav Kağıtlarının TC Kimlik No Tabanlı Alarak Arşivlenmesi [Öz]. *Presented at the Özgür Yazılım ve Linux Günleri*.

